

# GL5115 Design Specification

**Latest Version: V2.05**

**Date: 2013-09-13**

# Declaration

Circuit diagrams and other information relating to products of Actions Semiconductor Company, Ltd. ("Actions") are included as a means or illustrating typical applications. Consequently, complete information sufficient for construction purposes is not necessarily given. Although the information has been examined and is believed to be accurate, Actions makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and disclaims any responsibility for inaccuracies. Information in this document is provided solely to enable use of Actions' products. The information presented in this document does not form part of any quotation or contract for sale. Actions assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of Actions' products, except as may expressly be provided in Actions' Terms and Conditions of Sale for such products. All sales of any Actions products are expressly conditional on your agreement to the terms and conditions of the most recently dated version of Actions' Terms and Conditions of Sale Agreement dated before the date of your order.

The provision of this information does not convey to the purchaser of the described semiconductor devices any licenses under any patent rights, copyright, trademark rights, rights in trade secrets and/or know how, or any other intellectual property rights of Actions or others, however denominated, whether by express or implied representation, by estoppels, or otherwise.

Information contained herein relates solely to the Actions products described herein and abrogates and supersedes, as of the release date of this publication, all previously published data and specifications relating to such products provided by Actions or by any other person purporting to distribute such information. Actions reserve the right to make changes to specifications and product descriptions at any time without notice. Contact your Actions sales representative to obtain the latest specifications before placing your product order. Actions product may contain design defects or errors known as anomalies or errata which may cause the products functions to deviate from published specifications. Anomaly or "errata" sheets relating to currently characterized anomalies or errata are available upon request. Designers must not rely on the absence or characteristics of any features or instructions of Actions' products marked "reserved" or "undefined." Actions reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Actions' products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of Actions and further testing and/or modification will be fully at the risk of the customer.

Copies of this document and/or other Actions product literature, as well as the Terms and Conditions of Sale Agreement, may be obtained by visiting Actions' website at <http://www.actions-semi.com> or from an authorized Actions representative. The word "ACTIONS," the Actions' LOGO, whether used separately and/or in combination, are trademarks of Actions Semiconductor Company, Ltd.. Names and brands of other companies and their products that may from time to time descriptively appear in this product data sheet are the trademarks of their respective holders: no affiliation, authorization, or endorsement by such persons is claimed or implied except as may be expressly stated therein.

**ACTIONS DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND AGAINST INFRINGEMENT AND THE LIKE, AND ANY AND ALL WARRANTIES ARISING FROM ANY COURSE OF DEALING OR USAGE OF TRADE.**

**IN NO EVENT SHALL ACTIONS BE RELIABLE FOR ANY DIRECT, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES; OR FOR LOST DATA, PROFITS, SAVINGS OR REVENUES OF ANY KIND; REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF ACTIONS OR OTHERS; STRICT LIABILITY; BREACH OF WARRANTY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY OF BUYER IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER ACTIONS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR NOT.**

#### **ADDITIONAL SUPPORT**

Additional product and company information can be obtained by going to the Actions website at: [www.actions-semi.com](http://www.actions-semi.com).

# Contents

|  |           |
|--|-----------|
| GL5115 Design Specification.....                   | 1         |
| Declaration .....                                  | 2         |
| Contents.....                                      | 4         |
| Revision History.....                              | 33        |
| Acronyms, and Abbreviations .....                  | 52        |
| <b>1 Introduction.....</b>                         | <b>53</b> |
| 1.1 Features.....                                  | 53        |
| 1.2 Application Diagram.....                       | 55        |
| 1.3 Block Diagram.....                             | 55        |
| <b>2 Register List .....</b>                       | <b>56</b> |
| <b>3 Processor (黃少彬、黃俏) .....</b>                  | <b>75</b> |
| <b>4 Debug Unit (黃少彬、黃俏) .....</b>                 | <b>75</b> |
| 4.1 Features.....                                  | 75        |
| 4.2 Function Description.....                      | 75        |
| 4.3 Module Description .....                       | 76        |
| 4.3.1 Block Diagram.....                           | 76        |
| 4.3.2 TAP state machine .....                      | 76        |
| 4.3.3 JTAG Instruction.....                        | 77        |
| 4.4 BIRD Register Group .....                      | 79        |
| 4.5 BIRD Register Description.....                 | 79        |
| 4.5.1 Memory Access Register Description.....      | 79        |
| 4.5.2 MCU Communication Register Description.....  | 80        |
| 4.5.3 Event Manage Register Description .....      | 80        |
| 4.5.4 Breakpoint Manage Register Description ..... | 82        |
| 4.5.5 MCU Manage Register Description.....         | 82        |
| 4.6 SFR accessed BIRD Register .....               | 83        |
| 4.7 MCU accessed Register Description .....        | 83        |
| <b>5 Memory Controller(彭洪、黃俏) .....</b>            | <b>85</b> |
| 5.1 Features.....                                  | 85        |
| 5.2 Function Description.....                      | 85        |
| 5.2.1 Memory controller overview .....             | 85        |
| 5.2.2 Memory access rule .....                     | 86        |
| 5.2.2.1 MCU/DMA access different memory .....      | 86        |
| 5.2.2.2 MCU/DMA read access same memory.....       | 87        |
| 5.2.2.3 Memory access rules conclusion .....       | 87        |
| 5.2.3 Program Address Redirection .....            | 89        |
| 5.2.4 Page Miss Control Mechanism .....            | 91        |
| 5.2.5 Hardware Code Replace Mechanism.....         | 93        |
| 5.2.6 Content protection mechanism .....           | 96        |
| 5.2.6.1 bit jtag_disable.....                      | 97        |
| 5.2.7 BUS extension .....                          | 97        |

---

|         |   |     |
|---------|---|-----|
| 5.2.8   | Signal List .....                                     | 98  |
| 5.3     | Module Description .....                              | 101 |
| 5.3.1   | Memory Organization .....                             | 101 |
| 5.3.2   | Memory Allocation .....                               | 102 |
| 5.3.2.1 | Default memory mapping .....                          | 102 |
| 5.3.2.2 | MP3 decode Memory mapping.....                        | 104 |
| 5.3.2.3 | WMA/WAV decode Memory mapping.....                    | 104 |
| 5.3.2.4 | WAV/MP3 Encode Memory mapping .....                   | 105 |
| 5.3.2.5 | USB speaker & WAV/MP3 encode memory map.....          | 107 |
| 5.3.2.6 | UDISK memory mapping .....                            | 107 |
| 5.3.2.7 | Soundbar memory map .....                             | 109 |
| 5.3.3   | Memory blocks & peripherals access channel list ..... | 110 |
| 5.3.4   | Memory management .....                               | 110 |
| 5.4     | Memory_Controller register list .....                 | 111 |
| 5.5     | Register Description .....                            | 113 |
| 5.5.1   | MemBankCtl.....                                       | 113 |
| 5.5.2   | PageAddr0H .....                                      | 114 |
| 5.5.3   | PageAddr0L.....                                       | 114 |
| 5.5.4   | PageAddr1H .....                                      | 115 |
| 5.5.5   | PageAddr1L.....                                       | 115 |
| 5.5.6   | PageAddr2H .....                                      | 115 |
| 5.5.7   | PageAddr2L.....                                       | 115 |
| 5.5.8   | PageAddr3H .....                                      | 115 |
| 5.5.9   | PageAddr3L.....                                       | 116 |
| 5.5.10  | PageAddr4H .....                                      | 116 |
| 5.5.11  | PageAddr4L.....                                       | 116 |
| 5.5.12  | PageAddr5H .....                                      | 116 |
| 5.5.13  | PageAddr5L.....                                       | 116 |
| 5.5.14  | PageAddr6H .....                                      | 117 |
| 5.5.15  | PageAddr6L.....                                       | 117 |
| 5.5.16  | PageAddr7H .....                                      | 117 |
| 5.5.17  | PageAddr7L.....                                       | 117 |
| 5.5.18  | PageAddr8H .....                                      | 117 |
| 5.5.19  | PageAddr8L.....                                       | 118 |
| 5.5.20  | PageAddr9H .....                                      | 118 |
| 5.5.21  | PageAddr9L.....                                       | 118 |
| 5.5.22  | PageAddrMask0.....                                    | 118 |
| 5.5.23  | PageAddrMask1.....                                    | 119 |
| 5.5.24  | PageAddrMask2.....                                    | 119 |
| 5.5.25  | PageAddrMask3.....                                    | 119 |
| 5.5.26  | PageAddrMask4.....                                    | 120 |
| 5.5.27  | PageAddrMask5.....                                    | 120 |
| 5.5.28  | PageAddrMask6.....                                    | 121 |
| 5.5.29  | PageAddrMask7.....                                    | 121 |
| 5.5.30  | PageAddrMask8.....                                    | 121 |

---

---

|          |                                     |            |
|----------|-------------------------------------|------------|
| 5.5.31   | PageAddrMask9.....                  | 122        |
| 5.5.32   | RedirectAddr0.....                  | 122        |
| 5.5.33   | RedirectAddr1.....                  | 122        |
| 5.5.34   | RedirectAddr2.....                  | 123        |
| 5.5.35   | RedirectAddr3.....                  | 123        |
| 5.5.36   | RedirectAddr4.....                  | 123        |
| 5.5.37   | RedirectAddr5.....                  | 123        |
| 5.5.38   | RedirectAddr6.....                  | 124        |
| 5.5.39   | RedirectAddr7.....                  | 124        |
| 5.5.40   | RedirectAddr8.....                  | 124        |
| 5.5.41   | RedirectAddr9.....                  | 124        |
| 5.5.42   | FixAddr0H.....                      | 124        |
| 5.5.43   | FixAddr0M .....                     | 125        |
| 5.5.44   | FixAddr0L .....                     | 125        |
| 5.5.45   | FixAddr1H.....                      | 125        |
| 5.5.46   | FixAddr1M .....                     | 125        |
| 5.5.47   | FixAddr1L .....                     | 125        |
| 5.5.48   | FixAddr2H.....                      | 126        |
| 5.5.49   | FixAddr2M .....                     | 126        |
| 5.5.50   | FixAddr2L .....                     | 126        |
| 5.5.51   | FixAddr3H.....                      | 126        |
| 5.5.52   | FixAddr3M .....                     | 126        |
| 5.5.53   | FixAddr3L .....                     | 127        |
| 5.5.54   | PageMissEntryEx.....                | 127        |
| 5.5.55   | PageMissEntryH .....                | 127        |
| 5.5.56   | PageMissEntryL.....                 | 127        |
| 5.5.57   | CodeReplaceEntryEx .....            | 127        |
| 5.5.58   | CodeReplaceEntryH .....             | 128        |
| 5.5.59   | CodeReplaceEntryL.....              | 128        |
| 5.5.60   | TestCounterByte0 .....              | 128        |
| 5.5.61   | TestCounterByte1 .....              | 128        |
| 5.5.62   | TestCounterByte2 .....              | 128        |
| 5.5.63   | TestCounterByte3 .....              | 128        |
| 5.5.64   | TestCounterByte4 .....              | 129        |
| 5.5.65   | bist_en1 .....                      | 129        |
| 5.5.66   | bist_en2 .....                      | 129        |
| 5.5.67   | bist_fin1 .....                     | 130        |
| 5.5.68   | bist_fin2 .....                     | 131        |
| 5.5.69   | bist_info1 .....                    | 131        |
| 5.5.70   | bist_info2 .....                    | 132        |
| <b>6</b> | <b>PMU 模块 (冯崧祥、肖丽荣、彭祥、江力) .....</b> | <b>133</b> |
| 6.1      | Features.....                       | 134        |
| 6.2      | Function Description.....           | 134        |
| 6.3      | Module Description .....            | 135        |
| 6.3.1    | Charge Pump.....                    | 135        |

---

|               |   |     |
|---------------|---|-----|
| 6.3.2         | Linear Regulators.....                              | 135 |
| 6. 3. 2. 1    | Regulators Architectonics.....                      | 135 |
| 6. 3. 2. 2    | Regulators Output Voltage Set.....                  | 136 |
| 6. 3. 2. 3    | Regulators Accurate and Maximum Output Current..... | 136 |
| 6. 3. 2. 4    | Regulators Power Down.....                          | 136 |
| 6.3.3         | Reference Voltage .....                             | 136 |
| 6. 3. 3. 1    | EFUSE 与 BANDGAP 的关系及操作过程.....                       | 137 |
| 6.3.4         | A/D Converters .....                                | 138 |
| 6.3.5         | SYSTEM ON/OFF .....                                 | 140 |
| 6. 3. 5. 1    | System ON/OFF 模块框图.....                             | 140 |
| 6. 3. 5. 2    | System ON/OFF 电路框图.....                             | 141 |
| 6. 3. 5. 3    | ON/OFF 多功能键描述.....                                  | 141 |
| 6. 3. 5. 4    | System ON/OFF 状态转换描述.....                           | 142 |
| 6. 3. 5. 5    | System ON/OFF 时序图.....                              | 143 |
| 6. 3. 5. 5. 1 | RTCVDD 上电时序.....                                    | 143 |
| 6. 3. 5. 5. 2 | 正常上电 VCC/VDD 情况.....                                | 143 |
| 6. 3. 5. 5. 3 | 外灌 VCC 或多节干电池.....                                  | 145 |
| 6. 3. 5. 6    | Reset 复位 PMU 说明描述.....                              | 145 |
| 6.4           | Operation Manual .....                              | 147 |
| 6.4.1         | 软件进 standby (S2) 流程.....                            | 147 |
| 6.4.2         | 软件关机 (S3) 流程 .....                                  | 147 |
| 6.4.3         | 软件 SDK 初始化流程 .....                                  | 148 |
| 6.4.4         | 按键与硬开关检测 .....                                      | 148 |
| 6.4.5         | 各电源的开启方法 .....                                      | 149 |
| 6.5           | PMU Register List .....                             | 150 |
| 6.6           | Register Description .....                          | 150 |
| 6.6.1         | VOUT_CTL.....                                       | 150 |
| 6.6.2         | LDOPD_CTL .....                                     | 151 |
| 6.6.3         | BDG_CTL .....                                       | 152 |
| 6.6.4         | BDG_VOL.....  | 153 |
| 6.6.5         | MULTI_USED.....                                     | 153 |
| 6.6.6         | PMUADC_CTL.....                                     | 154 |
| 6.6.7         | BATADC_DATA .....                                   | 155 |
| 6.6.8         | LRADC1_DATA.....                                    | 155 |
| 6.6.9         | LRADC3_DATA.....                                    | 155 |
| 6.6.10        | LRADC4_DATA.....                                    | 156 |
| 6.6.11        | LRADC5_DATA.....                                    | 156 |
| 6.6.12        | CP_CTL0.....  | 156 |
| 6.6.13        | CP_CTL1 .....                                       | 157 |
| 6.6.14        | VDD_reserved .....                                  | 158 |
| 6.6.15        | TEST_CTL .....                                      | 158 |
|               | /*RTCVDD 域寄存器****/ .....                            | 158 |
| 6.6.16        | SYSTEM_CTL .....                                    | 158 |
| 6.6.17        | SYSTEM_ONOFF.....                                   | 159 |
| 6.6.18        | RTCVDD_reserved.....                                | 160 |

---

|  |     |
|--|-----|
| /***EFUSE Control 寄存器***/.....             | 161 |
| 6.6.19 FS_CTL .....                        | 161 |
| 6.6.20 EFUSE_CTL.....                      | 162 |
| 6.6.21 EFUSE0 .....                        | 162 |
| 6.6.22 EFUSE1 .....                        | 162 |
| 6.6.23 EFUSE2 .....                        | 163 |
| 6.6.24 EFUSE3 .....                        | 163 |
| 6.7 TEST MODE.....                         | 163 |
| 6.7.1 VOUT 电压调节 .....                      | 163 |
| 6.7.2 Power ON .....                       | 163 |
| 6.7.3 Reset for TEST MODE.....             | 164 |
| 6.7.4 状态转换测试 .....                         | 164 |
| 6.7.4.1 S4 到 S1 .....                      | 164 |
| 6.7.4.2 S1 到 S3 及唤醒 .....                  | 164 |
| 6.7.5 正常工作模式下的后门信号 .....                   | 164 |
| 7 System Control.....                      | 165 |
| 7.1 ChipVersion (黄少彬、蔡瑞仁) .....            | 165 |
| 7.1.1 ChipVersion Register group.....      | 165 |
| 7.1.2 Register Description .....           | 165 |
| 7.2 RMU Digital (彭洪、刘惠民、蔡瑞仁) .....         | 166 |
| 7.2.1 Features.....                        | 166 |
| 7.2.2 Function Description.....            | 166 |
| 7.2.3 Module Description .....             | 166 |
| 7.2.3.1 Block Diagram.....                 | 166 |
| 7.2.1 RMU_Digital_Part Register List ..... | 167 |
| 7.2.2 Register Description .....           | 168 |
| 7.2.2.1 MRCR1.....                         | 168 |
| 7.2.2.1 PCON.....                          | 168 |
| 7.2.2.2 MRCR2.....                         | 169 |
| 7.2.2.3 MRCR3.....                         | 170 |
| 7.2.2.4 ExWait .....                       | 170 |
| 7.3 CMU Analog (刘惠民、邵要华、陈文杰) .....         | 170 |
| 7.3.1 Features.....                        | 172 |
| 7.3.2 Function Description.....            | 172 |
| 7.3.3 Module Description .....             | 173 |
| 7.3.3.1 Block Diagram .....                | 173 |
| 7.3.4 Operation Manual .....               | 175 |
| 7.3.4.1 高校低电路说明 .....                      | 175 |
| 7.3.4.2 BROM 引导过程说明.....                   | 176 |
| 7.3.5 CMU_analog Register List .....       | 177 |
| 7.3.1 Register Description .....           | 178 |
| 7.3.1.1 MCU_PLL_CTL .....                  | 178 |
| 7.3.1.2 HOSC_CTL .....                     | 178 |
| 7.3.1.3 LOSC_CTL .....                     | 179 |
| 7.3.1.4 HCL_CLK_CTL .....                  | 180 |

---

|          |  |     |
|----------|--|-----|
| 7.3.1.5  | HCL_INTERVAL_CTL.....                  | 180 |
| 7.3.1.6  | AUDIO_PLL_CTL .....                    | 181 |
| 7.3.1.7  | MCU_PLL_SSC_CTL.....                   | 181 |
| 7.3.1.8  | MCU_PLL_SSC_FSTEP.....                 | 182 |
| 7.3.1.9  | MCU_PLL_DEBUG_CTL.....                 | 182 |
| 7.3.1    | TESTMODE .....                         | 183 |
| 7.3.1.1  | BLOCK DIAGRAM .....                    | 183 |
| 7.3.1.2  | SIGNAL DESCRIPTION.....                | 183 |
| 7.3.1.3  | TEST METHOD .....                      | 184 |
| 7.3.1.1  | Debug Signal .....                     | 184 |
| 7.4      | CMU Digital (彭洪、刘惠民、蔡瑞仁) .....         | 184 |
| 7.4.1    | Features.....                          | 187 |
| 7.4.2    | Function Description.....              | 187 |
| 7.4.3    | Module Description .....               | 187 |
| 7.4.3.1  | Block Diagram.....                     | 187 |
| 7.4.3.2  | Clock Requirements.....                | 188 |
| 7.4.4    | Signal List.....                       | 189 |
| 7.4.4.1  | CPU clock .....                        | 189 |
| 7.4.4.2  | DMA01234clock .....                    | 190 |
| 7.4.4.3  | AUIP clock.....                        | 191 |
| 7.4.4.4  | SD Card controller clock .....         | 192 |
| 7.4.4.5  | TK controller clock.....               | 193 |
| 7.4.4.6  | UART controller clock.....             | 194 |
| 7.4.4.7  | SPI controller clock .....             | 195 |
| 7.4.4.8  | USB controller clock .....             | 197 |
| 7.4.4.9  | RTC controller clock.....              | 198 |
| 7.4.4.10 | DAC/IIS-TX controller clock .....      | 198 |
| 7.4.4.11 | ADC and SPDIF rx controller clock..... | 201 |
| 7.4.4.12 | IIS-RX controller clock.....           | 203 |
| 7.4.4.13 | IR controller clock .....              | 204 |
| 7.4.4.14 | ACC controller clock .....             | 204 |
| 7.4.4.15 | FM clock .....                         | 205 |
| 7.4.4.16 | PWM controller clock.....              | 206 |
| 7.4.4.17 | LED & SEG LCD clock .....              | 207 |
| 7.4.4.18 | ASRC clock .....                       | 207 |
| 7.4.4.19 | ROM0/1 clock.....                      | 208 |
| 7.4.4.20 | DRAM251/PRAM251 clock .....            | 209 |
| 7.4.4.21 | URAM clock.....                        | 209 |
| 7.4.4.22 | FIR_CS_RAM clock.....                  | 210 |
| 7.4.4.23 | FIR_AA_RAM clock.....                  | 211 |
| 7.4.4.24 | FIR_MPX_RAM clock.....                 | 212 |
| 7.4.4.25 | FIR_RDS_RAM clock.....                 | 212 |
| 7.4.4.26 | PCMRAM clock .....                     | 213 |
| 7.4.4.27 | MURAM1 clock .....                     | 214 |
| 7.4.5    | CMU_Digital_Part Register List .....   | 215 |

---

|          |                            |     |
|----------|----------------------------|-----|
| 7.4.6    | Register Description ..... | 215 |
| 7.4.6.1  | CLKENCTL0.....             | 215 |
| 7.4.6.2  | CLKENCTL1.....             | 216 |
| 7.4.6.3  | CLKENCTL2.....             | 217 |
| 7.4.6.4  | SDCLKCTL.....              | 217 |
| 7.4.6.5  | CPUCLKCTL .....            | 218 |
| 7.4.6.6  | PWMCLKCTL .....            | 218 |
| 7.4.6.7  | PWMDUTY.....               | 219 |
| 7.4.6.8  | AECLKCTL.....              | 219 |
| 7.4.6.9  | Reserved .....             | 220 |
| 7.4.6.10 | FMCLKCTL .....             | 220 |
| 7.4.6.11 | LED_SEGLCDCLKCTL.....      | 221 |
| 7.4.6.12 | SPICLKCTL .....            | 221 |
| 7.4.6.13 | ADC_DAC_CLK_CTL .....      | 221 |
| 7.4.6.14 | MEMCLKENCTL0.....          | 222 |
| 7.4.6.15 | MEMCLKSELCTL0 .....        | 223 |
| 7.4.6.16 | MEMCLKSELCTL1 .....        | 224 |
| 7.5      | RTC (冯松祥、黄俏) .....         | 224 |
| 7.5.1    | Features.....              | 225 |
| 7.5.2    | Function Description.....  | 225 |
| 7.5.3    | Module Description .....   | 226 |
| 7.5.3.1  | Block Diagram .....        | 226 |
| 7.5.3.2  | Signal List .....          | 227 |
| 7.5.4    | Operation Manual .....     | 228 |
| 7.5.4.1  | 更新 RTCVDD 寄存器函数 .....      | 228 |
| 7.5.4.2  | 万年历时间设置 .....              | 229 |
| 7.5.4.3  | ALARM 设置 .....             | 230 |
| 7.5.4.4  | Watch dog 设置 .....         | 231 |
| 7.5.4.5  | Timer & CTC 设置 .....       | 232 |
| 7.5.4.6  | RTC 日历时钟说明 .....           | 232 |
| 7.5.5    | RTC Register List .....    | 233 |
| 7.5.6    | Register Description ..... | 234 |
| 7.5.6.1  | RTC_CTL0 .....             | 234 |
| 7.5.6.2  | RTC_CTL1 .....             | 235 |
| 7.5.6.3  | RTCTimeS .....             | 236 |
| 7.5.6.4  | RTCTimeMin .....           | 236 |
| 7.5.6.5  | RTCTimeH .....             | 236 |
| 7.5.6.6  | RTCTimeD .....             | 236 |
| 7.5.6.7  | RTCTimeMon .....           | 237 |
| 7.5.6.8  | RTCTimeY .....             | 237 |
| 7.5.6.9  | RTCALMS .....              | 237 |
| 7.5.6.10 | RTCALMM .....              | 237 |
| 7.5.6.11 | RTCALMH .....              | 237 |
| 7.5.6.12 | RTCRUPD .....              | 238 |
| 7.5.6.13 | TimerLB .....              | 238 |

---

|             |  |     |
|-------------|--|-----|
| 7. 5. 6. 14 | TimerMB .....                            | 239 |
| 7. 5. 6. 15 | TimerHB .....                            | 239 |
| 7. 5. 6. 16 | WDCTL .....                              | 239 |
| 7. 5. 6. 17 | CTCCTL .....                             | 240 |
| 7. 5. 6. 18 | CTCCNTL .....                            | 240 |
| 7. 5. 6. 19 | CTCCNTH .....                            | 241 |
| 7. 5. 6. 20 | RTCRDM0 .....                            | 241 |
| 7. 5. 6. 21 | RTCRDM1 .....                            | 241 |
| 7. 5. 6. 22 | RTCRDM2 .....                            | 241 |
| 7. 5. 6. 23 | RTCRDM3 .....                            | 242 |
| 7. 5. 6. 24 | RTCRDM4 .....                            | 242 |
| 7. 5. 6. 25 | RTCRDM5 .....                            | 242 |
| 7. 5. 6. 26 | RTCRDM6 .....                            | 242 |
| 7. 5. 6. 27 | RTCRDM7 .....                            | 242 |
| 7. 5. 6. 28 | RTCRDM8 .....                            | 243 |
| 7. 5. 6. 29 | RTCRDM9 .....                            | 243 |
| 7. 5. 6. 30 | RTCRDM10 .....                           | 243 |
| 7. 5. 6. 31 | RTCRDM11 .....                           | 243 |
| 7. 5. 6. 32 | RTCRDM12 .....                           | 243 |
| 7. 5. 6. 33 | RTCRDM13 .....                           | 244 |
| 7. 5. 6. 34 | RTCRDM14 .....                           | 244 |
| 7. 5. 6. 35 | RTCRDM15 .....                           | 244 |
| 7. 5. 6. 36 | RTCRDM16 .....                           | 244 |
| 7. 5. 6. 37 | RTCRDM17 .....                           | 244 |
| 7. 5. 6. 38 | RTCRDM18 .....                           | 245 |
| 7. 5. 6. 39 | RTCRDM19 .....                           | 245 |
| 7. 5. 6. 40 | RTCRDM20 .....                           | 245 |
| 7. 5. 6. 41 | RTCRDM21 .....                           | 245 |
| 7. 5. 6. 42 | RTCRDM22 .....                           | 245 |
| 7. 5. 6. 43 | CTCCTL2 .....                            | 246 |
| 7. 5. 6. 44 | CTCCNTL2 .....                           | 246 |
| 7. 5. 6. 45 | CTCCNTH2 .....                           | 246 |
| 7.5.7       | DEBUG SIGNAL .....                       | 247 |
| 7.6         | Interrupt Controller (黃少彬、黃俏) .....      | 247 |
| 7.6.1       | Features .....                           | 247 |
| 7.6.2       | Function Description .....               | 247 |
| 7.6.3       | Module Description .....                 | 248 |
| 7.6.3.1     | Block Diagram .....                      | 248 |
| 7.6.3.2     | Signal List .....                        | 249 |
| 7.6.3.3     | Intel 80251 to Flip80251 Migration ..... | 250 |
| 7.6.3.4     | Interrupt Vector .....                   | 251 |
| 7.6.3.5     | Interrupt Priorities .....               | 251 |
| 7.6.3.6     | Interrupt processing .....               | 252 |
| 7.6.3.7     | Interrupt Controller Clock .....         | 255 |
| 7.6.4       | Interrupt Control Register List .....    | 256 |

---

|   |            |
|---|------------|
| 7.6.5 Register Description .....                          | 257        |
| 7.6.5.1 IE0.....  | 257        |
| 7.6.5.2 AIE.....  | 257        |
| 7.6.5.3 IPH0.....   | 259        |
| 7.6.5.4 IPL0 .....  | 259        |
| 7.6.5.5 AIPH .....  | 260        |
| 7.6.5.6 AIPL .....  | 260        |
| 7.6.5.7 IF0.....  | 261        |
| 7.6.5.8 AIF.....  | 261        |
| 7.6.5.9 EXTINT .....                                      | 262        |
| 7.6.5.10 IFDebug0 .....                                   | 263        |
| 7.6.5.11 IFDebug1 .....                                   | 263        |
| <b>8 Co-Processor .....</b>                               | <b>264</b> |
| 8.1 DMA controller (黃少彬、黃俏) .....                         | 264        |
| 8.1.1 Features.....                                       | 264        |
| 8.1.2 Memory and Peripheral Access Description.....       | 265        |
| 8.1.2.1 Access memory.....                                | 265        |
| 8.1.2.2 Access Peripheral FIFO .....                      | 267        |
| 8.1.3 DMA examples .....                                  | 267        |
| 8.1.3.1 Audio applications .....                          | 267        |
| 8.1.3.2 UDISK applications.....                           | 272        |
| 8.1.4 Priorities of DMA0/1/2/3/4.....                     | 272        |
| 8.1.4.1 Priority between DMA0/1/2/3/4 and CPU.....        | 272        |
| 8.1.5 Symbols and abbreviations .....                     | 273        |
| 8.1.6 DMA0/1/2/3/4 state machine.....                     | 273        |
| 8.1.7 DMA transfer type .....                             | 275        |
| 8.1.7.1 Memory to Memory.....                             | 275        |
| 8.1.7.2 Memory to Peripheral or Peripheral to Memory..... | 275        |
| 8.1.8 DMA0/1/2/3/4 access channel list .....              | 276        |
| 8.1.9 Operation Manual .....                              | 277        |
| 8.1.9.1 SRC and DST of DMA .....                          | 277        |
| 8.1.9.2 Length of DMA .....                               | 277        |
| 8.1.9.3 Reload .....                                      | 277        |
| 8.1.9.4 Abort .....                                       | 277        |
| 8.1.9.5 Width.....  | 277        |
| 8.1.9.6 Priorities.....                                   | 277        |
| 8.1.10 DMA Register List.....                             | 278        |
| 8.1.11 DMA0/1/2/3/4 Register Description.....             | 279        |
| 8.1.11.1 DMA0IP.....                                      | 279        |
| 8.1.11.2 DMA0IE .....                                     | 280        |
| 8.1.11.3 DMA01234DBG.....                                 | 280        |
| 8.1.11.4 DMA0CTL0.....                                    | 280        |
| 8.1.11.5 DMA0CTL1.....                                    | 281        |
| 8.1.11.6 DMA0SADDR0H.....                                 | 282        |
| 8.1.11.7 DMA0SADDR0M .....                                | 282        |

---

---

|           |                     |     |
|-----------|---------------------|-----|
| 8.1.11.8  | DMA0SADDR0L .....   | 282 |
| 8.1.11.9  | DMA0DADDR0H .....   | 282 |
| 8.1.11.10 | DMA0DADDR0M .....   | 283 |
| 8.1.11.11 | DMA0DADDR0L.....    | 283 |
| 8.1.11.12 | DMA0FrameLenH.....  | 283 |
| 8.1.11.13 | DMA0FrameLenL ..... | 283 |
| 8.1.11.14 | DMA1IP.....         | 284 |
| 8.1.11.15 | DMA1IE .....        | 284 |
| 8.1.11.16 | DMA1CTL0.....       | 285 |
| 8.1.11.17 | DMA1CTL1.....       | 285 |
| 8.1.11.18 | DMA1SADDR0H.....    | 286 |
| 8.1.11.19 | DMA1SADDR0M .....   | 286 |
| 8.1.11.20 | DMA1SADDR0L .....   | 287 |
| 8.1.11.21 | DMA1DADDR0H .....   | 287 |
| 8.1.11.22 | DMA1DADDR0M .....   | 287 |
| 8.1.11.23 | DMA1DADDR0L.....    | 287 |
| 8.1.11.24 | DMA1FrameLenH.....  | 287 |
| 8.1.11.25 | DMA1FrameLenL ..... | 288 |
| 8.1.11.26 | DMA2IP.....         | 288 |
| 8.1.11.27 | DMA2IE .....        | 289 |
| 8.1.11.28 | DMA2CTL0.....       | 289 |
| 8.1.11.29 | DMA2CTL1.....       | 290 |
| 8.1.11.30 | DMA2SADDR0H.....    | 291 |
| 8.1.11.31 | DMA2SADDR0M .....   | 291 |
| 8.1.11.32 | DMA2SADDR0L .....   | 291 |
| 8.1.11.33 | DMA2DADDR0H .....   | 291 |
| 8.1.11.34 | DMA2DADDR0M .....   | 291 |
| 8.1.11.35 | DMA2DADDR0L.....    | 292 |
| 8.1.11.36 | DMA2FrameLenH.....  | 292 |
| 8.1.11.37 | DMA2FrameLenL ..... | 292 |
| 8.1.11.38 | DMA3IP.....         | 293 |
| 8.1.11.39 | DMA3IE .....        | 293 |
| 8.1.11.40 | DMA3CTL0.....       | 293 |
| 8.1.11.41 | DMA3CTL1.....       | 294 |
| 8.1.11.42 | DMA3SADDR0H.....    | 295 |
| 8.1.11.43 | DMA3SADDR0M .....   | 295 |
| 8.1.11.44 | DMA3SADDR0L .....   | 295 |
| 8.1.11.45 | DMA3DADDR0H .....   | 296 |
| 8.1.11.46 | DMA3DADDR0M .....   | 296 |
| 8.1.11.47 | DMA3DADDR0L.....    | 296 |
| 8.1.11.48 | DMA3FrameLenH.....  | 296 |
| 8.1.11.49 | DMA3FrameLenL ..... | 297 |
| 8.1.11.50 | DMA4IP.....         | 297 |
| 8.1.11.51 | DMA4IE .....        | 298 |
| 8.1.11.52 | DMA4CTL0.....       | 298 |

---

|           |   |     |
|-----------|---|-----|
| 8.1.11.53 | DMA4CTL1.....                                 | 299 |
| 8.1.11.54 | DMA4SADDR0H.....                              | 299 |
| 8.1.11.55 | DMA4SADDR0M .....                             | 300 |
| 8.1.11.56 | DMA4SADDR0L .....                             | 300 |
| 8.1.11.57 | DMA4DADDR0H .....                             | 300 |
| 8.1.11.58 | DMA4DADDR0M .....                             | 300 |
| 8.1.11.59 | DMA4DADDR0L.....                              | 300 |
| 8.1.11.60 | DMA4FrameLenH.....                            | 301 |
| 8.1.11.61 | DMA4FrameLenL .....                           | 301 |
| 8.2       | Audio IP (彭洪、鲍超、郑思、唐振中) .....                 | 302 |
| 8.2.1     | Features.....                                 | 303 |
| 8.2.2     | Function Description.....                     | 304 |
| 8.2.3     | Signal List.....                              | 304 |
| 8.2.4     | Module Description .....                      | 307 |
| 8.2.4.1   | Block Diagram.....                            | 307 |
| 8.2.4.2   | Memory Request.....                           | 307 |
| 8.2.4.3   | Memory Access Table.....                      | 308 |
| 8.2.4.4   | Eufse Option Description.....                 | 308 |
| 8.2.5     | Operation Manual .....                        | 308 |
| 8.2.5.1   | Decoding Control Flow Chart.....              | 308 |
| 8.2.5.2   | Encoding Control Flow Chart.....              | 311 |
| 8.2.5.3   | MP3 Break-Point Play Control Flow Chart ..... | 312 |
| 8.2.5.4   | WMA Break-Point Play Control Flow Chart ..... | 315 |
| 8.2.5.5   | WAV Break-Point Play Control Flow Chart ..... | 317 |
| 8.2.5.6   | Fast Forward Control Flow Chart .....         | 319 |
| 8.2.5.7   | MP3 Fast Reverse Control Flow Chart .....     | 319 |
| 8.2.5.8   | WMA Fast Reverse Control Flow Chart .....     | 320 |
| 8.2.6     | AUIP Register List.....                       | 321 |
| 8.2.7     | Codec Top Register Description .....          | 323 |
| 8.2.7.1   | AuCodecCtl .....                              | 323 |
| 8.2.7.2   | AuDebugLength.....                            | 325 |
| 8.2.7.3   | AuCodecDebug.....                             | 325 |
| 8.2.7.4   | AuCodecFIFOCtl.....                           | 326 |
| 8.2.7.5   | AuCodecFIFOData .....                         | 326 |
| 8.2.7.6   | AuCodecDebug2.....                            | 326 |
| 8.2.7.7   | AuCodecDecStateH .....                        | 327 |
| 8.2.7.8   | AuCodecDecStateL .....                        | 328 |
| 8.2.8     | Mp3 Decoder Register Description .....        | 328 |
| 8.2.8.1   | MP3IE .....                                   | 328 |
| 8.2.8.2   | MP3IP .....                                   | 329 |
| 8.2.8.3   | MP3Ctl.....                                   | 330 |
| 8.2.8.4   | MP3HeaderInfo .....                           | 331 |
| 8.2.8.5   | MP3BitRateH.....                              | 331 |
| 8.2.8.6   | MP3BitRateL .....                             | 332 |
| 8.2.8.7   | MP3TtimeH .....                               | 332 |

---

|          |  |     |
|----------|--|-----|
| 8.2.8.8  | MP3TtimeM.....                         | 332 |
| 8.2.8.9  | MP3TtimeS.....                         | 332 |
| 8.2.8.10 | MP3CtimeH.....                         | 332 |
| 8.2.8.11 | MP3CtimeM.....                         | 333 |
| 8.2.8.12 | MP3CtimeS.....                         | 333 |
| 8.2.8.13 | MP3FrameNumH.....                      | 333 |
| 8.2.8.14 | MP3FrameNumM.....                      | 333 |
| 8.2.8.15 | MP3FrameNumL.....                      | 334 |
| 8.2.8.16 | MP3FileLen3 .....                      | 334 |
| 8.2.8.17 | MP3FileLen2 .....                      | 334 |
| 8.2.8.18 | MP3FileLen1 .....                      | 334 |
| 8.2.8.19 | MP3HeaderSynCtl .....                  | 334 |
| 8.2.8.20 | SynHeaderData1 .....                   | 335 |
| 8.2.8.21 | SynHeaderData2 .....                   | 336 |
| 8.2.9    | WMA Decoder Register Description ..... | 337 |
| 8.2.9.1  | WMAIE .....                            | 337 |
| 8.2.9.2  | WMAIP.....                             | 338 |
| 8.2.9.3  | WMACtl .....                           | 339 |
| 8.2.9.4  | WMAHeaderInfo .....                    | 339 |
| 8.2.9.5  | WMABitRateH .....                      | 340 |
| 8.2.9.6  | WMABitRateL.....                       | 340 |
| 8.2.9.7  | WMATTimeH .....                        | 340 |
| 8.2.9.8  | WMATTimeM .....                        | 341 |
| 8.2.9.9  | WMATTimeS.....                         | 341 |
| 8.2.9.10 | WMACTimeH .....                        | 341 |
| 8.2.9.11 | WMACTimeM.....                         | 341 |
| 8.2.9.12 | WMACTimeS .....                        | 342 |
| 8.2.9.13 | WMAPackNumH.....                       | 342 |
| 8.2.9.14 | WMAPackNumM .....                      | 342 |
| 8.2.9.15 | WMAPackNumL .....                      | 342 |
| 8.2.9.16 | WMAPackSizeH.....                      | 342 |
| 8.2.9.17 | WMAPackSizeM .....                     | 343 |
| 8.2.9.18 | WMAPackSizeL .....                     | 343 |
| 8.2.9.19 | VirtualAddrH .....                     | 343 |
| 8.2.9.20 | VirtualAddrL.....                      | 343 |
| 8.2.9.21 | PhysicalAddrH.....                     | 343 |
| 8.2.9.22 | PhysicalAddrL .....                    | 344 |
| 8.2.9.23 | DiscardLen.....                        | 344 |
| 8.2.10   | WAV Decoder Register Description ..... | 344 |
| 8.2.10.1 | WAVIE .....                            | 344 |
| 8.2.10.2 | WAVIP .....                            | 345 |
| 8.2.10.3 | WAVCtl.....                            | 346 |
| 8.2.10.4 | WAVHeaderInfo1 .....                   | 347 |
| 8.2.10.5 | WAVHeaderInfo2 .....                   | 348 |
| 8.2.10.6 | WAVTTTimeH .....                       | 348 |

---

|           |  |     |
|-----------|--|-----|
| 8.2.10.7  | WAVTTimeM.....                             | 349 |
| 8.2.10.8  | WAVTTimeS.....                             | 349 |
| 8.2.10.9  | WAVCTimeH.....                             | 349 |
| 8.2.10.10 | WAVCTimeM.....                             | 349 |
| 8.2.10.11 | WAVCTimeS.....                             | 350 |
| 8.2.10.12 | WAVBlockNumH .....                         | 350 |
| 8.2.10.13 | WAVBlockNumM .....                         | 350 |
| 8.2.10.14 | WAVBlockNumL .....                         | 350 |
| 8.2.11    | WAV Encoder Register Description.....      | 350 |
| 8.2.11.1  | WAVEncCtl.....                             | 350 |
| 8.2.11.2  | WAVEncInfo .....                           | 351 |
| 8.2.12    | MP3 Encoder Register Description .....     | 352 |
| 8.2.12.1  | MP3EncCtl.....                             | 352 |
| 8.2.12.2  | MP3EncInfo .....                           | 353 |
| 8.2.13    | Software Decoder Register Description..... | 354 |
| 8.2.13.1  | SoftIE.....                                | 354 |
| 8.2.13.2  | SoftIP .....                               | 355 |
| 8.2.13.3  | SoftCtl.....                               | 356 |
| 8.2.13.4  | FrameLen1 .....                            | 357 |
| 8.2.13.5  | FrameLen0 .....                            | 357 |
| 8.2.14    | Post Processor Register Description .....  | 358 |
| 8.2.14.1  | BEPCtl1 .....                              | 358 |
| 8.2.14.2  | BEPCtl2 .....                              | 359 |
| 8.2.14.3  | BEPCtl3 .....                              | 360 |
| 8.2.14.4  | BEPCtl4 .....                              | 360 |
| 8.2.14.5  | BEPCtl5 .....                              | 361 |
| 8.2.14.6  | GlobalGainH.....                           | 362 |
| 8.2.14.7  | GlobalGainM .....                          | 362 |
| 8.2.14.8  | GlobalGainL .....                          | 362 |
| 8.2.14.9  | CurrentEnergy.....                         | 362 |
| 8.2.1     | Karaoke Register Description .....         | 363 |
| 8.2.1.1   | KaraokeCtl.....                            | 363 |
| 8.2.1.2   | LCHGain.....                               | 363 |
| 8.2.1.3   | RCHGain .....                              | 363 |
| 8.2.1.4   | LPFGain.....                               | 364 |
| 8.2.1.5   | HPFGain .....                              | 364 |
| 8.2.2     | EQ Coefficients Description .....          | 364 |
| 8.2.3     | SRS Processor Register Description .....   | 366 |
| 8.2.3.1   | SRSCtl0 .....                              | 366 |
| 8.2.3.2   | SRSCtl1 .....                              | 367 |
| 8.2.3.3   | SRSCtl2 .....                              | 368 |
| 8.2.3.4   | SRS control variable .....                 | 368 |
| 8.2.3.5   | SRS filter temporary variables.....        | 369 |
| 8.2.3.6   | SRS constant table .....                   | 370 |
| 8.2.3.7   | SRS ram allocation .....                   | 372 |

---

|          |  |     |
|----------|--|-----|
| 8.3      | 文件系统加速器模块 (黄少彬、蔡瑞仁) .....  | 373 |
| 8.3.1    | 描述 .....   | 373 |
| 8.3.1.1  | 文件系统加速器框图 .....  | 373 |
| 8.3.1.2  | 加速器时钟 .....  | 373 |
| 8.3.2    | ACC Register List .....  | 374 |
| 8.3.3    | Registers Description .....  | 374 |
| 8.3.3.1  | ACC_CTRL .....   | 374 |
| 8.3.3.2  | ACC_BLKLEN_LENHI .....   | 375 |
| 8.3.3.3  | ACC_BLKNUM_LENLO .....   | 375 |
| 8.3.3.4  | ACC_CMPDATA0_SUMHIE .....  | 375 |
| 8.3.3.5  | ACC_CMPDATA1_SUMHI .....   | 376 |
| 8.3.3.6  | ACC_CMPDATA2_SUMLO .....   | 376 |
| 8.3.3.7  | ACC_CMPDATA3_SUMLOE .....  | 377 |
| 8.3.3.8  | ACC_MATCNTHI .....   | 377 |
| 8.3.3.9  | ACC_MATCNTLO .....   | 377 |
| 8.3.3.10 | ACC_BITMAPHI .....   | 377 |
| 8.3.3.11 | ACC_BITMAPPO .....   | 377 |
| 8.3.3.12 | ACC_FIFO .....   | 378 |
| 8.3.4    | 加速器软件调用 .....  | 378 |
| 8.3.4.1  | 无符号累加的软件调用 .....   | 378 |
| 8.3.4.2  | 数据搜索的软件调用 .....  | 378 |
| 9        | Touch Key Controller (刘惠民、陈国安、黄俏) .....  | 379 |
| 9.1      | Features .....   | 380 |
| 9.2      | Function Description .....   | 380 |
| 9.2.1    | Auto-calibration .....   | 380 |
| 9.2.2    | Baseline Correction .....  | 381 |
| 9.2.3    | Debounce prevention .....  | 381 |
| 9.2.4    | Operation mode .....   | 381 |
| 9.3      | Module Description .....   | 382 |
| 9.3.1    | Block Diagram .....  | 382 |
| 9.3.2    | Signal list .....  | 382 |
| 9.4      | Operation Manual .....   | 383 |
| 9.4.1    | 软件控制流程 .....   | 383 |
| 9.5      | Register List .....  | 384 |
| 9.6      | Register Description .....   | 387 |
| 9.6.1    | TK_EN_CTL .....  | 387 |
| 9.6.2    | TK_FUN_CTL .....   | 387 |
| 9.6.3    | TK_IRQ_PD .....  | 388 |
| 9.6.4    | TK_PRESS_STA .....   | 389 |
| 9.6.5    | TK_DB_CTL .....  | 389 |
| 9.6.6    | TK_PWM_CNTH .....  | 390 |
| 9.6.7    | TK_PWM_CNTL .....  | 390 |
| 9.6.8    | TK_PRESS_TH .....  | 390 |
| 9.6.9    | TK_BL_CAL_TH .....   | 391 |
|          | Note: for details, please refer to Baseline Correction of function Description ..... | 392 |

---

---

|        |                       |     |
|--------|-----------------------|-----|
| 9.6.10 | TK_NOISE_TH .....     | 392 |
| 9.6.11 | TK_BUCKET_TH.....     | 393 |
| 9.6.12 | TK_AUTO_LOWTH_H.....  | 394 |
| 9.6.13 | TK_AUTO_LOWTH_L.....  | 394 |
| 9.6.14 | TK_AUTO_HIGHTH_H..... | 394 |
| 9.6.15 | TK_AUTO_HIGHTH_L..... | 395 |
| 9.6.16 | TK_KEY0_IDA_CTL ..... | 395 |
| 9.6.17 | TK_KEY1_IDA_CTL ..... | 395 |
| 9.6.18 | TK_KEY2_IDA_CTL ..... | 396 |
| 9.6.19 | TK_KEY3_IDA_CTL ..... | 397 |
| 9.6.20 | TK_KEY4_IDA_CTL ..... | 398 |
| 9.6.21 | TK_KEY5_IDA_CTL ..... | 399 |
| 9.6.22 | TK_UPDAT_PD .....     | 400 |
| 9.6.23 | TK_KEY0_DATH .....    | 401 |
| 9.6.24 | TK_KEY0_DATL.....     | 401 |
| 9.6.25 | TK_KEY1_DATH .....    | 401 |
| 9.6.26 | TK_KEY1_DATL.....     | 401 |
| 9.6.27 | TK_KEY2_DATH .....    | 402 |
| 9.6.28 | TK_KEY2_DATL.....     | 402 |
| 9.6.29 | TK_KEY3_DATH .....    | 402 |
| 9.6.30 | TK_KEY3_DATL.....     | 402 |
| 9.6.31 | TK_KEY4_DATH .....    | 402 |
| 9.6.32 | TK_KEY4_DATL.....     | 403 |
| 9.6.33 | TK_KEY5_DATH .....    | 403 |
| 9.6.34 | TK_KEY5_DATL.....     | 403 |
| 9.6.35 | TK_RAW_PD .....       | 403 |
| 9.6.36 | TK_KEY0_RAWH .....    | 404 |
| 9.6.37 | TK_KEY0_RAWL .....    | 404 |
| 9.6.38 | TK_KEY1_RAWH .....    | 404 |
| 9.6.39 | TK_KEY1_RAWL .....    | 404 |
| 9.6.40 | TK_KEY2_RAWH .....    | 405 |
| 9.6.41 | TK_KEY2_RAWL .....    | 405 |
| 9.6.42 | TK_KEY3_RAWH .....    | 405 |
| 9.6.43 | TK_KEY3_RAWL .....    | 405 |
| 9.6.44 | TK_KEY4_RAWH .....    | 405 |
| 9.6.45 | TK_KEY4_RAWL .....    | 406 |
| 9.6.46 | TK_KEY5_RAWH .....    | 406 |
| 9.6.47 | TK_KEY5_RAWL .....    | 406 |
| 9.6.48 | TK_KEY0_BL_H .....    | 406 |
| 9.6.49 | TK_KEY0_BL_L .....    | 406 |
| 9.6.50 | TK_KEY1_BL_H .....    | 407 |
| 9.6.51 | TK_KEY1_BL_L .....    | 407 |
| 9.6.52 | TK_KEY2_BL_H .....    | 407 |
| 9.6.53 | TK_KEY2_BL_L .....    | 407 |
| 9.6.54 | TK_KEY3_BL_H .....    | 407 |

---

|           |                                 |            |
|-----------|---------------------------------|------------|
| 9.6.55    | TK_KEY3_BL_L .....              | 408        |
| 9.6.56    | TK_KEY4_BL_H.....               | 408        |
| 9.6.57    | TK_KEY4_BL_L .....              | 408        |
| 9.6.58    | TK_KEY5_BL_H.....               | 408        |
| 9.6.59    | TK_KEY5_BL_L .....              | 408        |
| 9.6.60    | TK_DEBUG .....                  | 409        |
| 9.7       | TESTMODE .....                  | 409        |
| 9.7.1     | BLOCK DIAGRAM.....              | 409        |
| 9.7.2     | SIGNAL DESCRIPTION.....         | 410        |
| 9.7.3     | TEST METHOD.....                | 410        |
| 9.7.4     | DEBUG METHOD.....               | 410        |
| <b>10</b> | <b>Storage .....</b>            | <b>411</b> |
| 10.1      | SD/MMC 模块 (赵天亮、郑思) .....        | 411        |
| 10.1.1    | Features.....                   | 411        |
| 10.1.2    | Function Description.....       | 411        |
| 10.1.2.1  | Transfer Mode Description.....  | 411        |
| 10.1.2.2  | Hardware WatchDog.....          | 413        |
| 10.1.2.3  | Delay Chain Regulated Unit..... | 414        |
| 10.1.3    | Module Description .....        | 416        |
| 10.1.3.1  | Block Diagram .....             | 416        |
| 10.1.3.2  | Signal List .....               | 416        |
| 10.1.3.3  | Clock Description.....          | 417        |
| 10.1.4    | AC Parameter.....               | 417        |
| 10.1.5    | Operation Manual .....          | 420        |
| 10.1.5.1  | SDC 寄存器使用的一些注意事项.....           | 420        |
| 10.1.5.2  | SDC 常见应用情景软件控制流程.....           | 421        |
| 10.1.5.3  | MFP 的注意事项 .....                 | 421        |
| 10.1.5.4  | 卡初始化注意事项.....                   | 422        |
| 10.1.5.5  | 卡数据传输注意事项.....                  | 422        |
| 10.1.6    | SDC Register List .....         | 422        |
| 10.1.7    | Register Description .....      | 423        |
| 10.1.7.1  | SD_CMD .....                    | 423        |
| 10.1.7.2  | SD_CMD_ARGRSP .....             | 423        |
| 10.1.7.3  | SD_RSP_POIN .....               | 423        |
| 10.1.7.4  | SD_TF_CTL .....                 | 424        |
| 10.1.7.5  | SD_STATE .....                  | 425        |
| 10.1.7.6  | SD_BLK_SIZE_H .....             | 426        |
| 10.1.7.7  | SD_BLK_SIZE_L .....             | 426        |
| 10.1.7.8  | SD_BLK_NUM .....                | 426        |
| 10.1.7.9  | SD_CLK_CTL .....                | 427        |
| 10.1.7.10 | SD_PAD_CTL .....                | 427        |
| 10.1.7.11 | SD_INT_CTL .....                | 428        |
| 10.1.7.12 | SD_DATA_FIFO .....              | 429        |
| 10.1.7.13 | SD_TIMEOUT_CTL .....            | 429        |
| 10.1.7.14 | SD_TIMING_CTL .....             | 429        |

---

|                |   |            |
|----------------|---|------------|
| 10. 1. 7. 15   | SD_DBG_CTL .....                                | 430        |
| 10.2           | External Memory Interface (赵天亮、蔡瑞仁) .....       | 431        |
| 10.2.1         | Features .....                                  | 432        |
| 10.2.2         | Function Description .....                      | 432        |
| 10.2.3         | Module Description .....                        | 432        |
| 10.2.3.1       | Block Diagram .....                             | 432        |
| 10.2.3.2       | Timing .....                                    | 433        |
| 10.2.1         | Operation Manual .....                          | 435        |
| 10.2.1.1       | Operation Flow for Software .....               | 435        |
| 10.2.1.2       | 模块的输出使能信号 .....                                 | 438        |
| 10.2.2         | EM Register List .....                          | 438        |
| 10.2.3         | Register Description .....                      | 438        |
| 10.2.3.1       | EXTMEM_CTL .....                                | 438        |
| 10.2.3.2       | EXTMEM_WT .....                                 | 439        |
| 10.2.3.3       | EXTMEM_DH .....                                 | 439        |
| 10.2.3.4       | EXTMEM_DL .....                                 | 439        |
| <b>11</b>      | <b>Transfer and Communication.....</b>          | <b>440</b> |
| 11.1           | USB (黄少彬、黄俏) .....                              | 440        |
| 11.1.1         | Features .....                                  | 440        |
| 11.1.2         | Function Description .....                      | 441        |
| 11. 1. 2. 1    | OTG state machine .....                         | 442        |
| 11. 1. 2. 2    | Endpoint and memory configuration .....         | 444        |
| 11.1.2.3       | Device Controller .....                         | 447        |
| 11. 1. 2. 3. 1 | Control endpoint0 .....                         | 447        |
| 11. 1. 2. 3. 2 | Non-Control endpoints .....                     | 449        |
| 11. 1. 2. 3. 3 | Data Packet Synchronization .....               | 452        |
| 11.1.2.4       | Host Controller .....                           | 452        |
| 11. 1. 2. 4. 1 | Frame Generator and Transaction Scheduler ..... | 452        |
| 11. 1. 2. 4. 2 | Control Transfer in the Host Mode .....         | 453        |
| 11. 1. 2. 4. 3 | Data Transfer .....                             | 453        |
| 11. 1. 2. 5    | Interrupt .....                                 | 454        |
| 11. 1. 2. 5. 1 | Wakeup Interrupt .....                          | 454        |
| 11. 1. 2. 5. 2 | USB Interrupt .....                             | 454        |
| 11. 1. 2. 6    | Speed Detection .....                           | 456        |
| 11. 1. 2. 7    | USB RESET .....                                 | 456        |
| 11. 1. 2. 8    | USB SUSPEND .....                               | 456        |
| 11. 1. 2. 9    | USB RESUME .....                                | 457        |
| 11. 1. 2. 10   | Efuse .....                                     | 457        |
| 11. 1. 2. 11   | Efuse Control .....                             | 458        |
| 11. 1. 2. 12   | PHY .....                                       | 458        |
| 11.1.3         | Module Description .....                        | 460        |
| 11. 1. 3. 1    | Block Diagram .....                             | 460        |
| 11. 1. 3. 2    | Signal List .....                               | 461        |
| 11. 1. 3. 3    | Clock Description .....                         | 465        |
| 11.1.4         | Operation Manual .....                          | 466        |

---

|              |                                      |     |
|--------------|--------------------------------------|-----|
| 11.1.4.1     | Flow Of Software.....                | 466 |
| 11.1.4.2     | URES AND EFUSE.....                  | 467 |
| 11.1.4.3     | Configuration of EP and Memory ..... | 467 |
| 11.1.4.4     | Interrupt .....                      | 468 |
| 11.1.4.5     | DMA .....                            | 468 |
| 11.1.4.6     | Plug in detect .....                 | 468 |
| 11.1.4.7     | Initialization .....                 | 469 |
| 11.1.4.8     | AUTO mode.....                       | 469 |
| 11.1.4.9     | Exit.....                            | 470 |
| 11.1.4.10    | Error Detection and Recovery.....    | 470 |
| 11.1.5       | USB Register List .....              | 470 |
| 11.1.6       | Register Description .....           | 473 |
| 11. 1. 6. 1  | UDMAM .....                          | 473 |
| 11. 1. 6. 2  | AUTOINTIMER .....                    | 474 |
| 11. 1. 6. 3  | EP1DMALENL .....                     | 474 |
| 11. 1. 6. 4  | EP1DMALENH .....                     | 474 |
| 11. 1. 6. 5  | EP2DMALENL .....                     | 475 |
| 11. 1. 6. 6  | EP2DMALENH .....                     | 475 |
| 11. 1. 6. 7  | OUTPCKCNTL .....                     | 475 |
| 11. 1. 6. 8  | OUTPCKCNTH .....                     | 475 |
| 11. 1. 6. 9  | IDVBUCTRL .....                      | 476 |
| 11. 1. 6. 10 | USBSTATUS .....                      | 476 |
| 11. 1. 6. 11 | DPDMCTRL .....                       | 476 |
| 11. 1. 6. 12 | USB_PHYCTRL .....                    | 477 |
| 11. 1. 6. 13 | Otgirq .....                         | 477 |
| 11. 1. 6. 14 | Otgstate .....                       | 478 |
| 11. 1. 6. 15 | Otgctrl .....                        | 479 |
| 11. 1. 6. 16 | Otgstatus .....                      | 479 |
| 11. 1. 6. 17 | Otgien .....                         | 480 |
| 11. 1. 6. 18 | Hcep0ctrl .....                      | 480 |
| 11. 1. 6. 19 | Hcout1ctrl .....                     | 480 |
| 11. 1. 6. 20 | Hcin2ctrl .....                      | 481 |
| 11. 1. 6. 21 | Hcout3ctrl .....                     | 481 |
| 11. 1. 6. 22 | Hcout0err .....                      | 482 |
| 11. 1. 6. 23 | Hcout1err .....                      | 483 |
| 11. 1. 6. 24 | Hcout3err .....                      | 485 |
| 11. 1. 6. 25 | Hcin0err .....                       | 486 |
| 11. 1. 6. 26 | Hcin2err .....                       | 487 |
| 11. 1. 6. 27 | Hcportctrl .....                     | 489 |
| 11. 1. 6. 28 | Hcfrrmn1 .....                       | 489 |
| 11. 1. 6. 29 | Hcfrrmn1h .....                      | 490 |
| 11. 1. 6. 30 | Hcfrrmremain1 .....                  | 490 |
| 11. 1. 6. 31 | Hcfrrmremainh .....                  | 490 |
| 11. 1. 6. 32 | Hcep03errirrq .....                  | 491 |
| 11. 1. 6. 33 | Hcep03errien .....                   | 491 |

---

|              |                          |     |
|--------------|--------------------------|-----|
| 11. 1. 6. 34 | Hcin0maxpck .....        | 492 |
| 11. 1. 6. 35 | Hcout1maxpckl .....      | 492 |
| 11. 1. 6. 36 | Hcin2maxpckl .....       | 493 |
| 11. 1. 6. 37 | Hcout3maxpck .....       | 493 |
| 11. 1. 6. 38 | Hcout1maxpckh .....      | 493 |
| 11. 1. 6. 39 | Hcin2maxpckh .....       | 493 |
| 11. 1. 6. 40 | Out0bc_hcin0bc .....     | 493 |
| 11. 1. 6. 41 | In0bc_hcout0bc .....     | 494 |
| 11. 1. 6. 42 | Ep0cs_hcep0cs .....      | 495 |
| 11. 1. 6. 43 | In1bcl_hcout1bcl .....   | 498 |
| 11. 1. 6. 44 | Out2bcl_hcin2bcl .....   | 498 |
| 11. 1. 6. 45 | In3bc_hcout3bc .....     | 498 |
| 11. 1. 6. 46 | In1bch_hcout1bch .....   | 498 |
| 11. 1. 6. 47 | Out2bch_hcin2bch .....   | 498 |
| 11. 1. 6. 48 | In1cs_hcout1cs .....     | 499 |
| 11. 1. 6. 49 | Out2cs_hcin2cs .....     | 501 |
| 11. 1. 6. 50 | In3cs_hcout3cs .....     | 503 |
| 11. 1. 6. 51 | In1ctrl_hcout1ctrl ..... | 504 |
| 11. 1. 6. 52 | Out2ctrl_hcin2ctrl ..... | 506 |
| 11. 1. 6. 53 | In3ctrl_hcout3ctrl ..... | 508 |
| 11. 1. 6. 54 | Setupdat0 .....          | 509 |
| 11. 1. 6. 55 | Setupdat1 .....          | 509 |
| 11. 1. 6. 56 | Setupdat2 .....          | 509 |
| 11. 1. 6. 57 | Setupdat3 .....          | 510 |
| 11. 1. 6. 58 | Setupdat4 .....          | 510 |
| 11. 1. 6. 59 | Setupdat5 .....          | 510 |
| 11. 1. 6. 60 | Setupdat6 .....          | 510 |
| 11. 1. 6. 61 | Setupdat7 .....          | 511 |
| 11. 1. 6. 62 | Fif01dat .....           | 511 |
| 11. 1. 6. 63 | Fif02dat .....           | 511 |
| 11. 1. 6. 64 | Fif03dat .....           | 511 |
| 11. 1. 6. 65 | Ep0indata .....          | 511 |
| 11. 1. 6. 66 | Ep0outdata .....         | 512 |
| 11. 1. 6. 67 | Ep03irq .....            | 512 |
| 11. 1. 6. 68 | Ep03ien .....            | 513 |
| 11. 1. 6. 69 | Usbirq_hcusbirq .....    | 514 |
| 11. 1. 6. 70 | Usbien_hcusbien .....    | 516 |
| 11. 1. 6. 71 | Ep03tokirq .....         | 517 |
| 11. 1. 6. 72 | Ep03tokien .....         | 519 |
| 11. 1. 6. 73 | IVECT .....              | 520 |
| 11. 1. 6. 74 | EPRST .....              | 520 |
| 11. 1. 6. 75 | UsbCTRL_STUS .....       | 521 |
| 11. 1. 6. 76 | Fnaddr .....             | 522 |
| 11. 1. 6. 77 | Clkgate .....            | 523 |
| 11. 1. 6. 78 | Fifoctrl .....           | 523 |

---

|              |                                 |     |
|--------------|---------------------------------|-----|
| 11. 1. 6. 79 | FrmCNTL .....                   | 523 |
| 11. 1. 6. 80 | FrmCNTH .....                   | 524 |
| 11. 1. 6. 81 | USBEIRQ .....                   | 524 |
| 11. 1. 6. 82 | AUTONAKCTRL .....               | 524 |
| 11. 1. 6. 83 | HCINCTRL .....                  | 525 |
| 11. 1. 6. 84 | SHORTPCKIRQ .....               | 525 |
| 11. 1. 6. 85 | DBGMODE .....                   | 526 |
| 11. 1. 6. 86 | VDCTRL .....                    | 527 |
| 11. 1. 6. 87 | VDSTAT .....                    | 527 |
| 11. 1. 6. 88 | BKDOOR .....                    | 528 |
| 11. 1. 6. 89 | OTGTRIEN .....                  | 528 |
| 11. 1. 6. 90 | OTGTRIRQ .....                  | 529 |
| 11. 1. 6. 91 | USB_Efuse_Ref .....             | 529 |
| 11. 1. 6. 92 | FSMPRESTATE .....               | 530 |
| 11. 1. 6. 93 | HCIN2CNTL .....                 | 530 |
| 11. 1. 6. 94 | HCIN2CNTH .....                 | 530 |
| 11. 1. 6. 95 | EP1STADDRL .....                | 530 |
| 11. 1. 6. 96 | EP1STADDRH .....                | 530 |
| 11. 1. 6. 97 | EP2STADDRL .....                | 531 |
| 11. 1. 6. 98 | EP2STADDRH .....                | 531 |
| 11.1.7       | PHY Register Description.....   | 531 |
| 11. 1. 7. 1  | APHY Register Description.....  | 531 |
| 11. 1. 7. 2  | DPHY Register Description.....  | 536 |
| 11. 1. 7. 3  | Debug Register Description..... | 538 |
| 11.2         | SPI 子模块 (胡世飞、蔡瑞仁) .....         | 541 |
| 11.2.1       | Features.....                   | 541 |
| 11.2.2       | Function Description.....       | 542 |
| 11.2.3       | Module Description .....        | 543 |
| 11. 2. 3. 1  | Block Diagram .....             | 543 |
| 11.2.4       | Operation Manual .....          | 543 |
| 11. 2. 4. 1  | 软件控制流程 .....                    | 543 |
| 11.2.5       | SPI Register List .....         | 547 |
| 11.2.6       | Register Description .....      | 548 |
| 11. 2. 6. 1  | SPI_CTL .....                   | 548 |
| 11. 2. 6. 2  | SPI_DRQ .....                   | 549 |
| 11. 2. 6. 3  | SPI_STA .....                   | 549 |
| 11. 2. 6. 4  | SPI_CLKDIV .....                | 550 |
| 11. 2. 6. 5  | SPI_TXDAT .....                 | 551 |
| 11. 2. 6. 6  | SPI_RXDAT .....                 | 551 |
| 11. 2. 6. 7  | SPI_BCL .....                   | 551 |
| 11. 2. 6. 8  | SPI_BCH .....                   | 551 |
| 11. 2. 6. 9  | SPI_DEBUG .....                 | 552 |
| 11. 2. 6. 10 | SPI_RAND .....                  | 553 |
| 11. 2. 6. 11 | SPI_SEED0 .....                 | 553 |
| 11. 2. 6. 12 | SPI_SEED1 .....                 | 553 |

---

|             |                                       |     |
|-------------|---------------------------------------|-----|
| 11.2.7      | TESTMODE .....                        | 554 |
| 11. 2. 7. 1 | BLOCK DIAGRAM .....                   | 554 |
| 11. 2. 7. 2 | SIGNAL DESCRIPTION.....               | 554 |
| 11. 2. 7. 3 | TEST METHOD .....                     | 554 |
| 11.2.8      | Electronics parameter .....           | 554 |
| 11.3        | UART 子模块（胡世飞、孙睿） .....                | 555 |
| 11.3.1      | Features.....                         | 555 |
| 11.3.2      | Function Description.....             | 555 |
| 11.3.3      | Module Description .....              | 556 |
| 11.3.4      | Operation Manual .....                | 557 |
| 11. 3. 4. 1 | 软件控制流程 .....                          | 557 |
| 11.3.5      | UART Register List .....              | 558 |
| 11.3.1      | Register Description .....            | 558 |
| 11. 3. 1. 1 | UART_BR .....                         | 558 |
| 11. 3. 1. 2 | UART_MODE .....                       | 558 |
| 11. 3. 1. 3 | UART_CTL .....                        | 559 |
| 11. 3. 1. 4 | UART_DRQ .....                        | 560 |
| 11. 3. 1. 5 | UART_STA .....                        | 561 |
| 11. 3. 1. 6 | UART_TXDAT .....                      | 562 |
| 11. 3. 1. 7 | UART_RXDAT .....                      | 562 |
| 11. 3. 1. 8 | UART_DEBUG .....                      | 562 |
| 11.3.2      | TESTMODE .....                        | 563 |
| 11. 3. 2. 1 | BLOCK DIAGRAM .....                   | 563 |
| 11. 3. 2. 2 | SIGNAL DESCRIPTION.....               | 563 |
| 11. 3. 2. 3 | TEST METHOD .....                     | 563 |
| 11.3.3      | Timing fault-tolerant parameter ..... | 563 |
| 11.4        | IR 子模块（胡世飞、蔡瑞仁） .....                 | 565 |
| 11.4.1      | Features.....                         | 565 |
| 11.4.2      | Function Description.....             | 565 |
| 2013、       | 9012 Protocol .....                   | 565 |
| 2014、       | NEC Protocol .....                    | 566 |
| 2015、       | RC5 Protocol.....                     | 568 |
| 2016、       | RC6 Protocol.....                     | 569 |
| 2017、       | 红外放大解调模块.....                         | 571 |
| 11.4.3      | Module Description .....              | 572 |
| 11.4.4      | Operation Manual .....                | 573 |
| 11.4.5      | IRC Register List .....               | 574 |
| 11.4.6      | Register Description .....            | 574 |
| 11. 4. 6. 1 | IR_CTL .....                          | 574 |
| 11. 4. 6. 2 | IR_STA .....                          | 575 |
| 11. 4. 6. 3 | IR_LUC .....                          | 576 |
| 11. 4. 6. 4 | IR_HUC .....                          | 576 |
| 11. 4. 6. 5 | IR_KDC .....                          | 576 |
| 11. 4. 6. 6 | IR_CONFIG8 .....                      | 576 |
| 11.4.7      | TESTMODE .....                        | 577 |

---

|             |  |            |
|-------------|--|------------|
| 11. 4. 7. 1 | BLOCK DIAGRAM .....                            | 577        |
| 11. 4. 7. 2 | SIGNAL DESCRIPTION.....                        | 577        |
| 11. 4. 7. 3 | TEST METHOD .....                              | 577        |
| 11.4.8      | Timing fault-tolerant parameter .....          | 577        |
| <b>12</b>   | <b>Audio.....</b>                              | <b>577</b> |
| 12.1        | Audio DAC (林立, 彭洪, 何积军, 丁家平) .....             | 577        |
| 12.1.1      | Feature .....                                  | 578        |
| 12.1.2      | Function Description.....                      | 579        |
| 12.1.2.1    | Audio DAC/PA.....                              | 579        |
| 12.1.3      | Module Description .....                       | 579        |
| 12.1.3.1    | Block Diagram.....                             | 579        |
| 12.1.3.2    | Signal Flow .....                              | 580        |
| 12.1.3.3    | Clock Description .....                        | 581        |
| 12.1.3.4    | PIN Description and Electrical Parameters..... | 581        |
| 12.1.3.5    | AC Parameters .....                            | 582        |
| 12.1.1      | Operation Manual .....                         | 582        |
| 12.1.1.1    | 软件控制流程 .....                                   | 583        |
| 12.1.2      | DAC Register List .....                        | 586        |
| 12.1.1      | Register Description .....                     | 587        |
| 12.1.1.1    | DAC_CTL .....                                  | 587        |
| 12.1.1.2    | DAC_VOLUME0.....                               | 588        |
| 12.1.1.3    | DAC_VOLUME1.....                               | 589        |
| 12.1.1.4    | DAC_CH0_FIFO_CTL0 .....                        | 590        |
| 12.1.1.5    | DAC_CH0_FIFO_CTL1 .....                        | 591        |
| 12.1.1.6    | DAC_CH0_PCML.....                              | 591        |
| 12.1.1.7    | DAC_CH0_PCMM.....                              | 592        |
| 12.1.1.8    | DAC_CH0_PCMH .....                             | 592        |
| 12.1.1.9    | DAC_CH1_FIFO_CTL0 .....                        | 592        |
| 12.1.1.10   | DAC_CH1_FIFO_CTL1 .....                        | 593        |
| 12.1.1.11   | DAC_CH1_PCML.....                              | 593        |
| 12.1.1.12   | DAC_CH1_PCMH .....                             | 593        |
| 12.1.1.13   | I2S_CTL0 .....                                 | 594        |
| 12.1.1.14   | I2S_CTL1 .....                                 | 594        |
| 12.1.1.15   | DAC_ANALOG0 .....                              | 595        |
| 12.1.1.16   | DAC_ANALOG1 .....                              | 596        |
| 12.1.1.17   | DAC_TUNE0 .....                                | 596        |
| 12.1.1.18   | DAC_TUNE1 .....                                | 597        |
| 12.1.1.19   | PA_VOLUME.....                                 | 597        |
| 12.1.1.20   | PA_CTL .....                                   | 598        |
| 12.1.1.21   | PA_APCTL .....                                 | 598        |
| 12.1.1.22   | DDV_CTL0 .....                                 | 599        |
| 12.1.1.23   | DAC_CH0_SR_CTL .....                           | 599        |
| 12.1.1.24   | DAC_CH0_SR_GAIN.....                           | 600        |
| 12.1.1.25   | DAC_CH0_SRFT_CTL0 .....                        | 600        |
| 12.1.1.26   | DAC_CH0_SRFT_CTL1 .....                        | 601        |

|           |   |     |
|-----------|---|-----|
| 12.1.1.27 | DAC_CH0_SRFT_CTL2 .....                         | 601 |
| 12.1.1.28 | DAC_CH0_SRFT_CTL3 .....                         | 601 |
| 12.1.1.29 | DAC_CH0_SRFT_CTL4 .....                         | 602 |
| 12.1.1.30 | DAC_CH1_SR_CTL .....                            | 602 |
| 12.1.1.31 | DAC_CH1_SR_GAIN .....                           | 602 |
| 12.1.1.32 | DAC_CH1_SRFT_CTL0 .....                         | 603 |
| 12.1.1.33 | DAC_CH1_SRFT_CTL1 .....                         | 603 |
| 12.1.1.34 | DAC_CH1_SRFT_CTL2 .....                         | 604 |
| 12.1.1.35 | DAC_CH1_SRFT_CTL3 .....                         | 604 |
| 12.1.1.36 | DAC_CH1_SRFT_CTL4 .....                         | 604 |
| 12.1.2    | TESTMODE .....                                  | 605 |
| 12.1.2.1  | BLOCK DIAGRAM .....                             | 605 |
| 12.1.2.2  | SIGNAL DESCRIPTION .....                        | 605 |
| 12.1.2.3  | TEST METHOD .....                               | 605 |
| 12.2      | Audio ADC (林立, 彭洪, 何积军, 唐晓) .....               | 605 |
| 12.2.1    | Features .....                                  | 606 |
| 12.2.2    | Function Description .....                      | 606 |
| 12.2.2.1  | Audio ADC .....                                 | 606 |
| 12.2.3    | Module Description .....                        | 607 |
| 12.2.3.1  | Block Diagram .....                             | 607 |
| 12.2.3.2  | Signal Flow .....                               | 608 |
| 12.2.3.3  | Clock Description .....                         | 608 |
| 12.2.3.4  | PIN Description and Electrical Parameters ..... | 608 |
| 12.2.3.5  | AC Parameters .....                             | 609 |
| 12.2.1    | Operation Manual .....                          | 609 |
| 12.2.1.1  | 软件控制流程 .....                                    | 610 |
| 12.2.2    | ADC Register List .....                         | 612 |
| 12.2.3    | Register Description .....                      | 612 |
| 12.2.3.1  | ADC_CTL0 .....                                  | 612 |
| 12.2.3.2  | AINOP_CTL .....                                 | 613 |
| 12.2.3.3  | ADC_GAIN0 .....                                 | 614 |
| 12.2.3.4  | ADC_GAIN1 .....                                 | 614 |
| 12.2.3.5  | ADC_TUNE0 .....                                 | 615 |
| 12.2.3.6  | ADC_TUNE1 .....                                 | 616 |
| 12.2.3.7  | ADC_FIFO_DAT .....                              | 616 |
| 12.2.3.8  | ADC_FIFOCTL0 .....                              | 616 |
| 12.2.3.9  | ADC_FIFOCTL1 .....                              | 617 |
| 12.2.3.10 | ADC_CTL1 .....                                  | 618 |
| 12.2.3.11 | ADC_SRFT_CTL0 .....                             | 618 |
| 12.2.3.12 | ADC_SRFT_CTL1 .....                             | 619 |
| 12.2.3.13 | ADC_SRFT_CTL2 .....                             | 619 |
| 12.2.3.14 | ADC_SRFT_CTL3 .....                             | 619 |
| 12.2.3.15 | ADC_SRFT_CTL4 .....                             | 619 |
| 12.2.4    | TESTMODE .....                                  | 620 |
| 12.2.4.1  | BLOCK DIAGRAM .....                             | 620 |

|                 |   |            |
|-----------------|---|------------|
| 12.2.4.2        | SIGNAL DESCRIPTION .....                        | 620        |
| 12.2.4.3        | TEST METHOD .....                               | 620        |
| 12.3            | SPDIF (林立, 彭洪, 孙睿) .....                        | 621        |
| 12.3.1          | Features .....                                  | 621        |
| 12.3.2          | Function Description .....                      | 622        |
| 12.3.3          | Module Description .....                        | 623        |
| 12.3.3.1        | Block Diagram .....                             | 623        |
| 12.3.3.2        | Signal Flow .....                               | 623        |
| 12.3.3.3        | Clock Description .....                         | 624        |
| 12.3.3.4        | PIN Description and Electrical Parameters ..... | 625        |
| 12.3.1          | Operation Manual .....                          | 625        |
| 12.3.1.1        | 软件控制流程 .....                                    | 625        |
| 12.3.2          | SPDIFRX Register List .....                     | 628        |
| 12.3.1          | Register Description .....                      | 629        |
| 12.3.1.1        | SPDIFRX_CTL .....                               | 629        |
| 12.3.1.2        | SPDIFRX_STAT .....                              | 630        |
| 12.3.1.3        | SPDIFRX_CSTAT .....                             | 630        |
| 12.3.1.4        | SPDIFRX_DEBUG .....                             | 631        |
| 12.3.1.5        | SPDIFRX_DAT .....                               | 631        |
| 12.3.1.6        | SPDIFRX_CNT_WR0 .....                           | 632        |
| 12.3.1.7        | SPDIFRX_CNT_WR1 .....                           | 632        |
| 12.3.1.8        | SPDIFRX_CNT_WR2 .....                           | 632        |
| 12.3.2          | TESTMODE .....                                  | 632        |
| 12.3.2.1        | BLOCK DIAGRAM .....                             | 632        |
| 12.3.2.2        | SIGNAL DESCRIPTION .....                        | 632        |
| 12.3.2.3        | TEST METHOD .....                               | 632        |
| <b>13</b>       | <b>UI.....</b>                                  | <b>633</b> |
| 13.1            | LCD &LED (赵天亮、黃俏) .....                         | 633        |
| 13.1.1          | Features .....                                  | 633        |
| 13.1.2          | Function Description .....                      | 633        |
| 13.1.2.1        | Seg & Com Driving Timing .....                  | 633        |
| <b>13.1.2.2</b> | <b>SEG/COM LCD and Digit-LED Blinking .....</b> | <b>638</b> |
| 13.1.3          | Module Description .....                        | 639        |
| 13.1.3.1        | Block Diagram .....                             | 639        |
| 13.1.3.2        | Clock Description .....                         | 639        |
| 13.1.3.3        | Analog setting .....                            | 640        |
| 13.1.3.4        | PAD Driving Capacity .....                      | 640        |
| 13.1.3.5        | The LCD Debug Signals .....                     | 641        |
| 13.1.4          | Operation Manual .....                          | 641        |
| 13.1.4.1        | GL5115 新增內容 .....                               | 641        |
| 13.1.4.2        | Operation Flow for Software .....               | 641        |
| 13.1.5          | LCD Register List .....                         | 643        |
| 13.1.6          | Register Description .....                      | 643        |
| 13.1.6.1        | LCD_MODE .....                                  | 643        |
| 13.1.6.2        | LCD_DATA0 .....                                 | 644        |

|              |                                      |            |
|--------------|--------------------------------------|------------|
| 13.1.6.3     | LCD_DATA1 .....                      | 644        |
| 13.1.6.4     | LCD_DATA2 .....                      | 645        |
| 13.1.6.5     | LCD_DATA3 .....                      | 646        |
| 13.1.6.6     | LCD_DATA4 .....                      | 646        |
| 13.1.6.7     | LCD_DATA5 .....                      | 647        |
| 13.1.6.8     | LCD_DATA6 .....                      | 647        |
| 13.1.6.9     | LCD_DATA7 .....                      | 648        |
| 13.1.6.10    | LCD_DATA8 .....                      | 648        |
| 13.1.6.11    | LCD_DATA9 .....                      | 649        |
| 13.1.6.12    | LCD_DATA10 .....                     | 649        |
| 13.1.6.13    | LCD_DATA11 .....                     | 649        |
| 13.1.6.14    | LCD_DATA12 .....                     | 649        |
| 13.1.6.15    | LCD_DATA13 .....                     | 649        |
| 13.1.6.16    | LCD_DATA14 .....                     | 650        |
| 13.1.6.17    | LCD_DATA15 .....                     | 650        |
| 13.1.7       | TESTMODE .....                       | 650        |
| 13.1.7.1     | BLOCK DIAGRAM .....                  | 650        |
| <b>14</b>    | <b>GPIO and I/O Multiplexer.....</b> | <b>650</b> |
| 14.1         | GPIO （李克伟、蔡瑞仁） .....                 | 650        |
| 14.1.1       | Features .....                       | 653        |
| 14.1.2       | Function Description .....           | 653        |
| 14.1.3       | Module Description .....             | 654        |
| 14. 1. 3. 1  | Block Diagram .....                  | 654        |
| 14. 1. 3. 2  | PAD .....                            | 654        |
| 14. 1. 3. 3  | Analog and Digital PAD .....         | 657        |
| 14. 1. 3. 4  | Multi-Function Bonding Option .....  | 657        |
| 14. 1. 3. 5  | EJTAG Multi-Function .....           | 659        |
| 14. 1. 3. 6  | PAD 的上下拉电阻 .....                     | 659        |
| 14. 1. 3. 7  | P_TEST、P_RESETB .....                | 659        |
| 14. 1. 3. 8  | 模块上下拉电阻 .....                        | 660        |
| 14. 1. 3. 9  | PAD 的施密特触发器 .....                    | 660        |
| 14. 1. 3. 10 | PWM .....                            | 660        |
| 14.1.4       | Operation Manual .....               | 660        |
| 14.1.4.1     | Operation Manual .....               | 660        |
| 14.1.4.2     | GPIO Output .....                    | 661        |
| 14.1.4.3     | GPIO Input .....                     | 661        |
| 14.1.4.4     | GPIO Output/Input Loop Test .....    | 662        |
| 14.1.5       | GPIO Register List .....             | 662        |
| 14.1.6       | Register Description .....           | 664        |
| 14. 1. 6. 1  | GPIOAOUTEN .....                     | 664        |
| 14. 1. 6. 2  | GPIOAINEN .....                      | 664        |
| 14. 1. 6. 3  | GPIOADAT .....                       | 664        |
| 14. 1. 6. 4  | GPIOAPUEN .....                      | 664        |
| 14. 1. 6. 5  | GPIOAPDEN .....                      | 664        |
| 14. 1. 6. 6  | GPIOBOUTEN .....                     | 665        |

---

|              |                  |     |
|--------------|------------------|-----|
| 14. 1. 6. 7  | GPIOBINEN .....  | 665 |
| 14. 1. 6. 8  | GPIOBDAT .....   | 665 |
| 14. 1. 6. 9  | GPIOBPUEN .....  | 665 |
| 14. 1. 6. 10 | GPIOBPDEN .....  | 665 |
| 14. 1. 6. 11 | GPIOCOUTEN ..... | 665 |
| 14. 1. 6. 12 | GPIOCINEN .....  | 666 |
| 14. 1. 6. 13 | GPIOCDAT .....   | 666 |
| 14. 1. 6. 14 | GPIOCPUEN .....  | 666 |
| 14. 1. 6. 15 | GPIOCPDEN .....  | 666 |
| 14. 1. 6. 16 | GPIODOUTEN ..... | 666 |
| 14. 1. 6. 17 | GPIODINEN .....  | 667 |
| 14. 1. 6. 18 | GPIODDAT .....   | 667 |
| 14. 1. 6. 19 | GPIODPUEN .....  | 667 |
| 14. 1. 6. 20 | GPIODPDEN .....  | 667 |
| 14. 1. 6. 21 | GPIOEOUTEN ..... | 667 |
| 14. 1. 6. 22 | GPIOEINEN .....  | 668 |
| 14. 1. 6. 23 | GPIOEDAT .....   | 668 |
| 14. 1. 6. 24 | GPIOEPUEN .....  | 668 |
| 14. 1. 6. 25 | GPIOEPDEN .....  | 668 |
| 14. 1. 6. 26 | GPIOFOUTEN ..... | 668 |
| 14. 1. 6. 27 | GPIOFINEN .....  | 669 |
| 14. 1. 6. 28 | GPIOFDAT .....   | 669 |
| 14. 1. 6. 29 | GPIOFPUEN .....  | 669 |
| 14. 1. 6. 30 | GPIOFPDEN .....  | 669 |
| 14. 1. 6. 31 | GPIOGOUTEN ..... | 669 |
| 14. 1. 6. 32 | GPIOGINEN .....  | 670 |
| 14. 1. 6. 33 | GPIOGDAT .....   | 670 |
| 14. 1. 6. 34 | GPIOGPUEN .....  | 670 |
| 14. 1. 6. 35 | GPIOGPDEN .....  | 670 |
| 14. 1. 6. 36 | MFP_CTL0 .....   | 670 |
| 14. 1. 6. 37 | MFP_CTL1 .....   | 671 |
| 14. 1. 6. 38 | MFP_CTL2 .....   | 672 |
| 14. 1. 6. 39 | MFP_CTL3 .....   | 672 |
| 14. 1. 6. 40 | MFP_CTL4 .....   | 672 |
| 14. 1. 6. 41 | MFP_CTL5 .....   | 673 |
| 14. 1. 6. 42 | MFP_CTL6 .....   | 674 |
| 14. 1. 6. 43 | MFP_CTL7 .....   | 674 |
| 14. 1. 6. 44 | MFP_CTL8 .....   | 675 |
| 14. 1. 6. 45 | AD_Select0 ..... | 675 |
| 14. 1. 6. 46 | AD_Select1 ..... | 676 |
| 14. 1. 6. 47 | PADPUPD .....    | 676 |
| 14. 1. 6. 48 | PADDRV0 .....    | 677 |
| 14. 1. 6. 49 | PADDRV1 .....    | 677 |
| 14. 1. 6. 50 | PADDRV2 .....    | 678 |
| 14. 1. 6. 51 | PADDRV3 .....    | 678 |

---

|           |                                  |            |
|-----------|----------------------------------|------------|
| 14.1.6.52 | PADDRV4 .....                    | 679        |
| 14.1.6.53 | DBGSEL .....                     | 679        |
| 14.1.6.54 | DBGAOE .....                     | 680        |
| 14.1.6.55 | DBGAIE .....                     | 681        |
| 14.1.6.56 | DBGBOE .....                     | 681        |
| 14.1.6.57 | DBGBIE .....                     | 682        |
| 14.1.6.58 | DBGCOE .....                     | 682        |
| 14.1.6.59 | DBGDOE .....                     | 682        |
| 14.1.6.60 | LED_SEG_RC_EN .....              | 683        |
| 14.1.6.61 | LED_SEG_BIAS_EN .....            | 683        |
| <b>15</b> | <b>BROM (黄晓佳) .....</b>          | <b>684</b> |
| 15.1      | Features.....                    | 684        |
| 15.2      | Function Description.....        | 684        |
| 15.3      | Module Description .....         | 685        |
| 15.3.1    | POWER ON.....                    | 685        |
| 15.3.2    | BREC Launcher.....               | 685        |
| 15.3.3    | NorFlash 引导.....                 | 686        |
| 15.3.4    | SDMMC 引导 .....                   | 687        |
| 15.3.5    | ADFULauncher 流程 .....            | 689        |
| 15.4      | BROM 中个物理模块的接口说明.....            | 691        |
| 15.4.1    | SDMMC 物理驱动接口.....                | 691        |
| 15.4.1.1  | 概述 .....                         | 691        |
| 15.4.1.2  | 文件名说明 .....                      | 691        |
| 15.4.1.3  | 函数接口 .....                       | 691        |
| 15.4.1.4  | 函数接口说明 .....                     | 691        |
| 15.4.2    | SPI 物理驱动接口.....                  | 692        |
| 15.4.2.1  | 概述 .....                         | 692        |
| 15.4.2.2  | 文件名说明 .....                      | 692        |
| 15.4.2.3  | 函数接口 .....                       | 692        |
| 15.4.2.4  | 函数接口说明 .....                     | 693        |
| <b>16</b> | <b>TEST MODE (李克伟、蔡瑞仁) .....</b> | <b>693</b> |
| 16.1      | Features.....                    | 693        |
| 16.2      | Function Description.....        | 694        |
| 16.2.1    | TEST MODE 1 .....                | 694        |
| 16.2.1.1  | TESE MODE 1 简介 .....             | 694        |
| 16.2.1.2  | TEST MODE 1 所需 PAD.....          | 694        |
| 16.2.2    | TEST MODE 3 .....                | 696        |
| 16.2.3    | TEST MODE 4 .....                | 696        |
| 16.2.3.1  | SCAN 基本原理.....                   | 696        |
| 16.2.3.2  | TEST MODE 4 所需 PAD.....          | 696        |
| 16.2.3.3  | SCAN 测试过程.....                   | 697        |
| 16.2.3.4  | SCAN 注意事项.....                   | 698        |
| 16.2.4    | TEST MODE 优先级 .....              | 698        |
| 16.3      | Module Description .....         | 698        |
| 16.3.1    | CMU Module TEST .....            | 699        |

---

|           |   |            |
|-----------|---|------------|
| 16.3.2    | AUDIO Module TEST.....                        | 699        |
| 16.4      | 时序注意事项 .....                                  | 699        |
| 16.5      | Operation Manual .....                        | 700        |
| <b>17</b> | <b>Electrical Characteristics .....</b>       | <b>701</b> |
| 17.1      | Absolute Maximum Ratings .....                | 701        |
| 17.2      | DC Characteristics .....                      | 701        |
| 17.3      | Recommended Power Supply .....                | 702        |
| 17.4      | Initialization Parameter.....                 | 702        |
| 17.5      | USB DC Electrical Characteristics .....       | 702        |
| 17.6      | SPI Interface Electrical Parameter .....      | 704        |
| 17.7      | UART Timing Fault-tolerant Parameter .....    | 705        |
| 17.8      | IR Timing fault-tolerant parameter .....      | 706        |
| 17.9      | SD Card Interface AC Parameter.....           | 707        |
| 17.10     | Audio channel Characteristics Parameter ..... | 710        |
| 17.10.1.1 | DAC+PA channel Characteristics Table.....     | 710        |
| 17.10.1.2 | ADC channel Characteristics Table .....       | 712        |
| 17.11     | GPIO Interface Parameter.....                 | 713        |
| 17.12     | LCM Driver Parameter .....                    | 714        |
| 17.13     | External System Bus Parameter.....            | 715        |
| <b>18</b> | <b>Pin Assignment.....</b>                    | <b>716</b> |
| <b>19</b> | <b>Package and Drawing.....</b>               | <b>717</b> |
| 19.1      | ATS2510(数码管 boombox, LQFP-48).....            | 717        |
| 19.2      | ATS2516(点阵屏 boombox, LQFP-48).....            | 718        |
| 19.3      | ATS2885(soundbar, LQFP-64).....               | 719        |
| 19.4      | ESC5115(for IC verify, LQFP-128) .....        | 720        |



# Revision History

| Version | Module      | Revision No                       | Revision Content   | Date       | IC Version |
|---------|-------------|-----------------------------------|--|------------|------------|
| V1.00   |             | initial                           | initial  | 2012-07-24 | GL5115A    |
| V1.01   | USB         | GL5115-V1.0<br>0-20120725-0<br>01 | 1. 修改 Endpoint and memory configuration<br>2. 将 NTIRQ 寄存器删除，NTIRQ 移到 Usbirq_hcusbirq 寄存器中<br>3. 增加 EP1STADDRH/L 和 EP2STADDRH/L 4 个寄存器  | 2012-08-09 | GL5115A    |
| V1.01   | USB         | GL5115-V1.0<br>0-20120804-0<br>07 | 4. 增加 USB 与 DMA 接口 FIFO 的 reset 控制 (0x80_bit[3:2])<br>5. 将 EP1、EP2 DMA start 控制位分开分别控制 (0x80_bit[1:0])<br>6. 将原先 DMAlength 寄存器修改为 EP1DMAlength 寄存器，增加 EP2DMAlength 寄存器<br>7. 删除 SoftVBUS 控制位，VBUS 一直有效。<br>8. 删除 OTGIRQ 和 OTGIEN bit3 和 bit1，删除 Otgctrl bit[6:1]<br>9. 删除 Otgstatus bit0<br>10. 删除 OTG 时间相关寄存器 (0xdf,e1,e2,e3 , f4)<br>11. 删除 <u>BKDOOR</u> bit3 | 2012-08-09 | GL5115A    |
| V1.01   | cmu_digital | GL5115-V1.0<br>0-20120729-0<br>02 | 1、修改了 FIR_MPX_RAM 的输入信号描述<br>2、增加了 SPDIF 时钟的分频电路，因此修改了配图和输入信号列表<br>3、在 AECLKCTL 寄存器的 bit7:5 增加了控制 SPDIF 分频的寄存器位<br>4、增加了 DAC/TISTX 时钟分频的描述，特别注意无缝切换的描述   | 2012-08-09 | GL5115A    |
| V1.01   | cmu_digital | GL5115-V1.0<br>0-20120801-0<br>06 | 5、为了选择到 DAC 的时钟，在 FMRDCLKCTL 寄存器中增加 DACIISTXCLSRCSEL bit<br>6、修改了 DAC 的时钟配图，有两点改动：1 是 DAC 的时钟选择，2 是 DAC 的时钟 gating<br>7、修改 SPDIF 的时钟分频寄存器<br>8、因为 SPDIF 时钟分频的变化修改了 ADC 的配图   | 2012-08-09 | GL5115A    |
| V1.01   | cmu_digital | GL5115-V1.0<br>0-20120806-0<br>09 | 9、修改了 FMclk 的时钟描述寄存器<br>10、修改了 FM 的时钟配图以及后面的文字描述   | 2012-08-09 | GL5115A    |

|       |             |                                   |   |            |         |
|-------|-------------|-----------------------------------|---|------------|---------|
|       |             |                                   | 11、修改了 DAC 时钟描述配图中 DAC/IIS TX CH1 FIFO 的时钟选择位<br>12、将原来 GPIO 中的 PWMDUTY 寄存器移到 CMU 模块中<br>13、在寄存器列表中增加了 PWMDUTY 寄存器描述，并且将 PWMCLKSEL 从 PWMCLKCTL 寄存器挪动到 PWMDUTY 寄存器中，PWMCLKCTL 的分频比有变化，默认值也有变化<br>14、修改了 PWM 的寄存器配图描述<br>15、修改了 PCMRAM 的时钟描述，增加 URAMclk 的选择<br>16、修改了 PCMRAM 的时钟配图描述<br>17、修改了 ADC and SPDIF rx controller 配图中的 SPDIF 的时钟分频描述 |            |         |
| V1.01 | CMU_analog  | GL5115-V1.0<br>0-20120730-0<br>03 | 按照 isp spec 检查工具的要求，更改 register list 的描述  | 2012-08-09 | GL5115A |
| V1.01 | INTC        | GL5115-V1.0<br>0-20120731-0<br>04 | 更改 register list 的描述  | 2012-08-09 | GL5115A |
| V1.01 | FMRX        | GL5115-V1.0<br>0-20120801-0<br>05 | 1. 修改 FMRX_LNA_MIXER_LDO_CTL 为 FMRX_APLL_MIXER_LDO_CTL;<br>2、修改 FMRX_ADC_APLL_LDO_CTL 为 FMRX_LNA_ADC_LDO_CTL  | 2012-08-09 | GL5115A |
| V1.01 | FMRX        | GL5115-V1.0<br>0-20120807-0<br>13 | 3、修改后的 FMRX 录音通路框图<br>4、根据曹志平 RDS 位宽的调整，对 RDS 寄存器进行了重大变更，由于寄存器增多，影响到后面的 test mode 寄存器的位置；   | 2012-08-09 | GL5115A |
| V1.01 | FMRX        | GL5115-V1.0<br>0-20120808-0<br>15 | 5、将 FMRDS_CPTL 计数器位宽由 8bits 修 改为 10 bits，增加了一个寄存器； 将 FMRX_RDS_LP_G1L、 6 、 FMRX_RDS_LP_G1H 、 FMRX_RDS_BS_M 寄存器位置往前移 1 位，原来的 reserved 寄存器被取消  | 2012-08-09 | GL5115A |
| V1.01 | ChipVersion | GL5115-V1.0<br>0-20120804-0<br>08 | 1、与 MFP 定义冲突，修改为 0x06ff   | 2012-08-09 | GL5115A |
| V1.01 | ChipVersion | GL5115-V1.0<br>0-20120804-0<br>10 | 2、原先 EXBUS DATA 寄存器定义在 0x06f4，但是实际设计却是在 0x06ff，与 ChipVersion 冲突，修改 EXBUS 比较麻烦，故将 ChipVersion 寄存器修改为 0x06f4  | 2012-08-09 | GL5115A |
| V1.01 | EXTM        | GL5115-V1.0<br>0-20120807-0<br>11 | 为了保证和 CPU 设计一致，将 EM_DL 寄存器定义地址从 0x06f4 改到 0x06ff  | 2012-08-09 | GL5115A |

|       |               |                                   |   |            |         |
|-------|---------------|-----------------------------------|---|------------|---------|
| V1.01 | PMU           | GL5115-V1.0<br>0-20120807-0<br>12 | <ul style="list-style-type: none"> <li>1. 根据 analog 的做法修改 LDOPD 寄存器</li> <li>2. 修改 SYSTEM_ONOFF 寄存器：增加 UVLOB_SEL 使能位，增加 DC5V_DOWN_SEL 下拉选择位，增加 EN_VDD_SL 偏置电流选择位</li> <li>3. 增加断码屏关闭时的描述</li> <li>4. 根据 analog 需求，修改 debug 信号分布</li> <li>5. 根据 digital 做法修改 system on 部分描述</li> <li>6. 根据 analog 参数，小幅修改 onoff 电路框图</li> <li>7. 修改 onoff 待机及唤醒描述</li> <li>8. 修改 PMUADC_CTL 寄存器，将 LRADC 的使能位改成 1 个</li> <li>9. 修改 TEST_CTL 寄存器</li> <li>10. 修改 FSOURCE 负载能力描述</li> <li>11. 增加 RTCVDD 负载能力 1mA</li> </ul> | 2012-08-09 | GL5115A |
| V1.01 | DAC           | GL5115-V1.0<br>0-20120808-0<br>14 | <ul style="list-style-type: none"> <li>1、去掉 PAEN 注释，PA 不需加 MIC boost.</li> <li>2、更改 PA_VOLUME 描述，pa volume total 41 level 改为 8 level。</li> <li>3、DAC clock description 框图删除，改为描述参考 CMU digital SPEC.</li> </ul>   | 2012-08-09 | GL5115A |
| V1.01 | ADC           | GL5115-V1.0<br>0-20120808-0<br>14 | <ul style="list-style-type: none"> <li>1、更改 ADCFSS 默认寄存器配置，由 10 改为 0</li> <li>2、ADC clock description 框图删除，改为描述参考 CMU digital SPEC</li> </ul>   | 2012-08-09 | GL5115A |
| V1.01 | SPDIF         | GL5115-V1.0<br>0-20120808-0<br>14 | <ul style="list-style-type: none"> <li>1、更改 SPDIFRX 寄存器位 RXFR 的描述。</li> <li>2、增加描述 SPDIFRX_CSTAT 同时作传送脉宽 counter 用。同时删除 SPDIFRX_CNT_RD，及同时更改后面寄存器的地址。</li> <li>3、更改描述 SRDIFRX Block IRQ 为 Channel Statue IRQ，同时更改 IRQ 框图。</li> <li>4、更改 PA_VOLUME 描述，pa volume total 41 level 改为 8 level。</li> <li>5、SPDIFRX clock description 框图删除，改为描述参考 CMU digital SPEC.</li> <li>6、由于 SPDIFRX 设计更改，删除采样率检测 SAMRD 及采样率 error pending。增加 CSFES 状态位。</li> </ul>   | 2012-08-09 | GL5115A |
| V1.01 | Register List | GL5115-V1.0<br>0-20120809-0<br>16 | <ul style="list-style-type: none"> <li>1、SPDIF 增加 4 个寄存器</li> <li>2、增减 CMU ANALOG 寄存器列表</li> <li>3、修改 FMRX 模块寄存器</li> <li>4、修改 EM 模块寄存器</li> </ul>  | 2012-08-09 | GL5115A |

|       |            |                                   |  |            |         |
|-------|------------|-----------------------------------|--|------------|---------|
|       |            |                                   | 5、修改 USB 模块寄存器   |            |         |
| V1.01 | FMRX       | GL5115-V1.0<br>0-20120809-0<br>17 | 1、修改寄存器命名笔误 (exit→exist)<br>2、更新频道对应频率的计算公式  | 2012-08-09 | GL5115A |
| V1.02 | UART       | GL5115-V1.0<br>1-20120810-0<br>18 | 1、UART_CLK 增加支持 6M 波特率   | 2012-08-10 | GL5115A |
| V1.02 | CMU_analog | GL5115-V1.0<br>1-20120813-0<br>19 | 1、将 HOSC/LOSC 的框图 0 端和 I 端修改过来，同时修改 test 模式下外灌 24M 的端口为 HOSCI；<br>2、高校低电路输出的日历时钟修改为 4Hz；<br>3、test 外灌 24M 时钟，从 HOSCI 端外灌   | 2012-08-13 | GL5115A |
| V1.02 | EM         | GL5115-V1.0<br>1-20120814-0<br>20 | 1、数据线回复到 16bit16bit<br>2、添加控制信号输出使能位   | 2012-08-14 | GL5115A |
| V1.02 | LCD        | GL5115-V1.0<br>1-20120814-0<br>21 | 2012、寄存器 LCD_mode 添加 bit4:<br>LCD_OUT_EN   | 2012-08-14 | GL5115A |
| V1.02 | FMRX       | GL5115-V1.0<br>1-20120816-0<br>22 | 1. 将软件计算 channel 的公式，修改为：<br>50K*chan+57. 7M<br>2. 由于计算公式发生变化，因此 FMRX_CHN 的默认值也需要由 0x2b4 变更为 0x2a4，这样才能够让默认频点为 87M；<br>NK 值不需要调整，已经是 87M；<br>FMRX_CHN_RD 默认值也需要修改为 0x2a4；<br>3. 为了节省 DFF 资源，删除 7 个 reserved 寄存器，只保留 3 个；<br>4. 在 FMRX_RDS_STA 寄存器中加入 group 同步指示位，为此调整其他 bit 的位置，向前移动一位；<br>5. 为了节省 DFF 资源，删除 7 个 reserved 寄存器，只保留 3 个；<br>6. 在 FMRX_RDS_STA 寄存器中加入 group 同步指示位，为此调整其他 bit 的位置，向前移动一位 | 2012-08-16 | GL5115A |
| V1.02 | PMU        | GL5115-V1.0<br>1-20120820-0<br>23 | 1. FSOURCE 负载能力修正为 30mA  | 2012-08-20 | GL5115A |
| V1.02 | GPIO       | GL5115-V1.0<br>1-20120820-0<br>24 | 1. MFP_CTL2 增加 SPIBT 模式切换；<br>2. 增加 DBGA0E, DBGAIE, DBGBOE,<br>DBGBIE, DBGC0E, DBGCIE,<br>DBGD0E, DBGDIE 寄存器；  | 2012-08-20 | GL5115A |

|       |                   |                                   |   |            |         |
|-------|-------------------|-----------------------------------|---|------------|---------|
|       |                   |                                   | 3. SPDIF0, SPDIF1 增加施密特触发器;<br>4. 在 AD_Select 增加一个 VDD 输出控制 bit;<br>5. Reserved 掉 MFP 寄存器中模拟复用的 bit;<br>6. 将 DBGSEL 中的 pmu_debug 信号改为一组;<br>7. 修改 PADDRV0—PADDRV4 寄存器;<br>8. 修改 PAD_SMIT0—PAD_SMIT2 寄存器复位默认值;<br>10. 去掉 SPI BOOT 说明;<br>11. MFP 复用 GPIOA3, GPIOA4 改动, 去掉 EM_CEB1, 增加 PWM, MFP_CTL1 做相应改动;<br>12. MFP 复用 GPIOE 增加 EM_[D8:D15], MFP_CTL 做相应改动;<br>13. 各寄存器地址重新排序。 |            |         |
| V1.02 | CMU_digital       | GL5115-V1.0<br>1-20120823-0<br>25 | 1、修改 FMRDS 时钟配图   | 2012-08-23 | GL5115A |
| V1.02 | RMU_digital       | GL5115-V1.0<br>1-20120823-0<br>26 | 1、增加 LCD 模块的复位控制  | 2012-08-23 | GL5115A |
| V1.02 | Memory Controller | GL5115-V1.0<br>1-20120824-0<br>27 | 1、增加 PAGEMISS 的输入 ENTRY<br>2、增加 PageAddr8H , PageAddr8L , PageAddr9H, PageAddr9L, PageAddrMask8, PageAddrMask9 , RedirectAddr8 , RedirectAddr9<br>3、修改了 CodeReplaceEntry 的默认值<br>4、修改了 PageMissEntry 的默认值   | 2012-08-24 | GL5115A |
| V1.02 | GPIO              | GL5115-V1.0<br>1-20120828-0<br>28 | 1. PAD 增加一个 FM_VCC;<br>2. 增加 LED_SEG_RC_EN, LED_SEG_BIAS_EN 寄存器;<br>3. 增加 GPIOA0, GPIOA1, GPIOA2, GPIOA3, GPIOA4, GPIOA5, GPIOA6, GPIOA7 驱动档位;<br>增加 SPIBOOT 驱动能力寄存器设置。   | 2012-08-28 | GL5115A |
| V1.02 | CMU_digital       | GL5115-V1.0<br>1-20120903-0<br>29 | 1、修改 DAC 模块的配图, 在选择 write DAC CH0 FIFO write clock 时的 FMRDS_CLK1 对应的编码应该是 10 而不是 00<br>2、修改 FMRDSCLKCTL , 将 bit 1 DACIISTXCLKSRCSEL 挪到 bit 4;<br>3、修改 FMRDSCLKCTL , 拓展 FMRDSCLKSEL 为 2bits, 可以选择  | 2012-09-03 | GL5115A |

|       |             |                                   |   |            |         |
|-------|-------------|-----------------------------------|---|------------|---------|
|       |             |                                   | FMRDS_DPLL_CLK0 、 HOSC 、<br>38M_FROM_DEBUG_B6;<br>4、修改 FMRDSCLKCTL，将 bit 3:2 修改为 FMRX_ADC_CLK_SEL，可以选择 FMRDS_DPLL_CLK1、FMRDS_CLK1、19M_FROM_DEBUG_B7;<br>5、根据前三点，修改 FMRDS 的时钟配图；<br>6、根据 FMRDS 时钟配图，修改 MEMCLKSELCTL0 、<br>MEMCLKSELCTL1，将 FIR_CS_RAM、<br>FIR_AA_RAM、FIR_MPX_RAM_LM、<br>FIR_RDS_RAM 的 FMRDS_CLK0 修改为 FMRDS_CLK0_GT;<br>7、根据 FMRDS 时钟配图，修改 MEMCLKSELCTL1，将 PCMRAM 的 FMRDS_CLK0 修改为 FMRX_ADC_CLK_GT;<br>8、根据 FMRDS 时钟配图，修改 MEMCLKSELCTL0，将 MURAN1 的 FMRDS_CLK0 修改为 FMRX_ADC_CLK<br>9、将 DAC/IIS-TX 中的 FMRDS_CLK1 选择修改为 FMRDS_CLK1_GT，此时 FMRX 模块写 DAC FIFO 的时钟变更为 gating 后的时钟；<br>10、修改 Clock Requirements 中的 FMRDS 时钟说明，增加 IIS、LCD/LED 时钟需求的说明 |            |         |
| V1.02 | CMU_digital | GL5115-V1.0<br>1-20120904-0<br>30 | 修改 IR 时钟配图，将 IR 内部的 1KHz 测试时钟源头由 LOSC 修改为 200KHz，这是因为 GL5115 ELOSC 是不封装出来的；   | 2012-09-04 | GL5115A |
| V1.02 | Audioio     | GL5115-V1.0<br>1-20120904-0<br>31 | 1、更改 bank 。 Register List 与 Description 不对应。<br>2、设计改为三种半空半满产生方式，且内部产生方式 counter 可配。<br>3、设计改为二种半空半满产生方式，且内部产生方式 counter 可配。<br>4、增加内部 FM 从 DAC 输出不经过 PA 直接到到 ADC<br>5、根据设计结果将 R/W Reserve 寄存器改为 R。<br>6、增加 3 个 bit 作 debug 用。<br>7、SPDIFRX_CSTAT 和 SPDIFRX_DAT: 描述   | 2012-09-04 | GL5115A |
| V1.02 | CMU_analog  | GL5115-V1.0<br>1-20120905-0       | 1、删除高校低电路中，日历时钟源选择：来自于 LOSC 电路还是高校低电路；因   | 2012-09-05 | GL5115A |

|       |      |                                   |  |            |         |
|-------|------|-----------------------------------|--|------------|---------|
|       |      | 32                                | <p>为 RTC 控制器中已经有同样的选择了；</p> <p>2、三个 RTCVDD 电压域下的寄存器：</p> <p>3、HOSC_CTL/LOSC_CTL/CALENDAR_CLK_CTL，都做了密码保护，对寄存器进行操作时需要注意；</p> <p>4、由于原有 LOSC 晶振电路设计中，有 LOSC_OK 信号，目前阶段删除此信号会影响顶层设计，给整合者造成麻烦，所以 spec 当中还需要保留这个信号</p> <p>5、将 cmu analog 寄存器中原来定义 reserved bit 修改成 R/W 形式，因为 analog 那边是按照 R/W 设计的，并没有注意到 spec 定义的只读类型；</p> <p>6、高校低说明中的 2Hz 修改成 4Hz；</p> <p>7、原来 HOSC_CTL 寄存器中 HOSC_CAP_SEL 默认值有笔误，应该是 15pF，也就是 011</p> |            |         |
| V1.02 | FMRX | GL5115-V1.0<br>1-20120905-0<br>33 | 1、对 digital/analog 的一些信号进行了重排，而且删除了 digital 的两个 debug gpio，digital debug gpio 数量由 19 个减少到 17 个   | 2012-09-05 | GL5115A |
| V1.02 | RTC  | GL5115-V1.0<br>1-20120905-0<br>34 | 1、修正 RTC_CTL0 和 RTC_CTL1 的 reserved 位为可使用<br>2、修正 RTCTimeD 和 RTCTimeMon 的默认值为 01<br>3、修改 RTCTimeMint 寄存器名为 RTCTimeMin<br>4、增加 1 路 CTC，及三个相关寄存器   | 2012-09-05 | GL5115A |
| V1.02 | PMU  | GL5115-V1.0<br>1-20120905-0<br>35 | 1、FSOURCE 负载能力修正为 30mA<br>2、FSOURCE 下拉修正为 2mA<br>3、AVDD 下拉修正为 400uA<br>4、增加描述，上电后软件关闭下拉<br>5、VCC 的负载能力修正为 drop 5%<br>6、修改 BANDGAP 寄存器默认值<br>7、修改 bandgap 烧写描述<br>8、修正 BDG_CTL 寄存器的 reserved 位为模拟用可读写<br>9、EFUSE_CTL 的 reserved 位改为只读(即不做)<br>10、修正 CP_CTL1 的 reserved 位，将 bit0 去<br>11、修改 MULTI_USD 寄存器，断码屏默认值关<br>12、修改 TEST_CTL 寄存器，增加数码管偏置位   | 2012-09-05 | GL5115A |

|       |             |                                   |  |            |         |
|-------|-------------|-----------------------------------|--|------------|---------|
| V1.02 | FMRX        | GL5115-V1.0<br>1-20120907-0<br>36 | 1、上一版变更中, 将 FMRX digital test mode 下的 19M 和 38M 时钟输入移到 GPIO_B6/B5 了, 删除了 GPIO_B7; 现在恢复成 19M 从 GPIO_B7 输入, 38M 从 GPIO_B6 输入, 改成删除 GPIO_B5   | 2012-09-07 | GL5115A |
| V1.02 | GPIO        | GL5115-V1.0<br>1-20120907-0<br>37 | 2012、 MFP_CTL8 BIT5, 4 改为:<br>00: I2S0_MCLK<br>01: LCD_SEG26<br>10: EM_D10<br>11: Reserved<br>2、 PAGE13 “URAT ”改为 “UART ”;<br>3、 PAGE13 GPIOE3 复用改为 I2S_DO;<br>4、 PADDRV2 bit[5:0]默认值改为 0x01,<br>PADDRV3 bit[7:2]默认值改为 0x01;<br>5、 更改 2.13.2 PAD 名称, 增加 CPAGND<br>PIN;<br>6、 LED SEG BIAS EN 寄存器 bit2 改为<br>LED_Cathode/Anode Mode.<br>7、 更改 2.1.32 PAD, 把一个 UVCC 去掉,<br>同时把 UVCC 改为 VCC<br>8、 增加两个 FM_AGND_LNA<br>9、 把 SPIBT_SCLK 改为 SPIBT_CLK; 把<br>FM_RF_MATCH 改为 FM_RF_MATCHB。 | 2012-09-07 | GL5115A |
| V1.02 | SPI         | GL5115-V1.0<br>1-20120907-0<br>38 | 1、 SPI_RAND[5:4]用于设置 SPI 3 线<br>模式, 同时删除 SPI_BCH.2 功能, 用<br>于 reserved。  | 2012-09-07 | GL5115A |
| V1.02 | SFR         | GL5115-V1.0<br>1-20120910-0<br>39 | 1、 修改 RTC 模块寄存器列表<br>2、 修改 MEMORY Controller 模块寄存器<br>列表   | 2012-09-10 | GL5115A |
| V1.02 | AUDIOI<br>P | GL5115-V1.0<br>1-20120911-0<br>40 | 1、增加 MP3 编码模式<br>2、在寄存器列表中增加 MP3 编码的寄存器<br>3、修改了 fade in 和 fade out 的时长<br>4、删除了 WAVCtl 寄存器中 bit4 的描述<br>5、修改了 WAVCtl 寄存器中 bit5:4 的描述,<br>当值为 2' b11 时, 为正常播放<br>6、在 wave decoder 中增加 ChannelSel bit<br>来选择左右声道<br>7、增加 input FIFO1 的专用 Reset bit,<br>直通模式专用<br>8、删除 EQ 各频带的增益寄存器, 将其该到<br>RAM 中<br>9、删除寄存器列表中的 EQ 寄存器描述   | 2012-09-11 | GL5115A |
| V1.02 | GPIO        | GL5115-V1.0<br>1-20120911-0<br>41 | 1. LED SEG BIAS EN bit2 寄 存 器<br>LED_Cathode/Anode Mode 改 为<br>LED_Cathode_Anode_Mode;   | 2012-09-11 | GL5115A |

|       |      |                                   |   |            |         |
|-------|------|-----------------------------------|---|------------|---------|
|       |      |                                   | 2. 所有的 Reserved 位定义为 R/W;<br>3. 去掉<br>PAD_SMIT0, PAD_SMIT1, PAD_SMIT2 寄存器;<br>4. 去掉 DBGBIE, DBGDIE 寄存器。   |            |         |
| V1.02 | GPIO | GL5115-V1.0<br>1-20120911-0<br>42 | 1. 去掉 DBGCIE 寄存器, 增加 DBGBIN 寄存器。  | 2012-09-11 | GL5115A |
| V1.02 | SFR  | GL5115-V1.0<br>1-20120912-0<br>43 | 1、修改 AUIP 模块寄存器列表   | 2012-09-12 | GL5115A |
| V1.02 | FMRX | GL5115-V1.0<br>1-20120912-0<br>44 | 1、register list 中 reserved8 修改为 FMRX_STA, 地址修改为 0xCE, 在 V1.08 中的修改疏忽了, 在 list 中误删了 FMRX_STA 寄存器;  | 2012-09-12 | GL5115A |
| V1.02 | GPIO | GL5115-V1.0<br>1-20120912-0<br>45 | 1. 所有 Reserved 位的 name 改为 “Reserved”  | 2012-09-12 | GL5115A |
| V2.00 | INTC | GL5115-V1.0<br>2-20120917-0<br>46 | 1. 寄存器 AIE bit4 EAUDIO 描述修改, AUDIO 中断由 DAC_IRQ、ADC_IRQ、SPDIF_IRQ 三个中断组成;  | 2012-09-17 | GL5115A |
| V2.00 | PMU  | GL5115-V1.0<br>2-20120918-0<br>47 | 1. 增加 TEST_CTL 寄存器, HDSWDET 的描述<br>2. 修改 EFUSE0 bit7-5 的功能及默认值<br>3. 修改 EFUSE1 bit6,7 Reserved 位的描述, 改为 for future use<br>4. 修改 TEST_CTL 寄存器第4位的冗余定义  | 2012-09-18 | GL5115A |
| V2.00 | AUIP | GL5115-V1.0<br>2-20120926-0<br>48 | 1、修改了 EQ 的 memory 占用空间<br>2、增加了 SRS 的描述<br>3、修改了 MUROM1 的大小需求<br>4、修改了 BEPCTL1 寄存器的 EffectEn bit 描述, 5、该 bit 只会控制 SRS 和 karaoke 音效的使能<br>6、修改了原来的 EffectMode 寄存器位, 将起变成 EQEn 和 EffectMode 位<br>7、修改了寄存器列表中的 SRS 寄存器<br>8、修改了 Efuse Option 的描述<br>9、修改信号列表中关于 SRS 使能控制 bit 0x05a7 的 bit6 的描述 | 2012-09-26 | GL5115A |
| V2.00 | GPIO | GL5115-V1.0<br>2-20120928-0<br>49 | 1. 更改<br>GPIOAPUEN, GPIOAPDEN, GPIOCPUEN, GPIOCPDEN 描述: GPIOA6, GPIOC6 下拉电阻改为 100K;<br>2. 更 改 2.1.3.2 PAD , 去 掉   | 2012-09-28 | GL5115A |

|       |      |                                   |   |            |         |
|-------|------|-----------------------------------|---|------------|---------|
|       |      |                                   | <p>FM, SPIBOOT, IR_Analog, I2C;</p> <p>3. 去掉 2.1.3.3 SPIBOOT 的描述;</p> <p>4. 更改 Analog and Digital PAD 表格, 修改 MFP;</p> <p>5. 更改 Multi-Function Bounding option 表格, 更改 MFP;</p> <p>6. 更改 2.1.3.9 PAD 的施密特触发器的描述;</p> <p>7. MFP_CTL6 寄存器 bit5, bit6 改为 Reserved;</p> <p>8. MFP_CTL8 寄存器 bit3:2 增加 11: UART_TX;<br/>Bit1:0 增加 11: UART_RX;</p> <p>9. AD_Select 寄存器 bit5 IR_RX_Analog 改为 Reserved; 同时 GPIOC4 复用的 AVCC 去掉, 增加 GPIOA5 和 GPIOA6 的 AVCC 复用, bit 位排序改变;</p> <p>10. PADPUPD 寄存器 bit1 改为 Reserved; DBGSEL 寄存器 bit4:0: 00100 I2C Debug Signals 改为 SPDIF Debug Signals ; 00111:AUIP Debug Signals 改为 Reserved; 01011: MJPEG Debug Signals 改为 Reserved; 10000:B1B2 Debug Signals 改为 Reserved;</p> <p>10100:SPIBOOT Debug Signals 改为 Reserved;</p> <p>10101: FM Digital Debug Signals 改为 Reserved;</p> <p>10110: FM Analog Debug Signals 改为 Reserved。</p> |            |         |
| V2.00 | RTC  | GL5115-V1.0<br>2-20121008-0<br>50 | 1. 将 RTCTimeMon 修改为 bit3:0  | 2012-10-08 | GL5115A |
| V2.00 | GPIO | GL5115-V1.0<br>2-20121008-0<br>51 | <p>1. 更改 Multi-Function Bounding option 表格, 更改 MFP; 在 Priority 栏, 去掉 EJ_TRST;</p> <p>2. 更改 2.1.3.2 PAD 中的 MFP, 和 GPIOA5 复用的 AVCC 改为 AVCC0, 和 GPIOC6 复用的 AVCC 改为 AVCC1;</p> <p>3. AD_Select 寄存器 bit2 名称改为 AVCC0_OUT, bit1 名称改为 AVCC1_OUT;</p> <p>4. P_RESET 改为 P_RESETB;</p> <p>5. 增加各模块 DEBUG GPIO 列表;</p> <p>6. MFP_CTL2 bit1 改为 Reserved;</p> <p>7. PADDRV4 bit3, bit2 改为 Reserved。</p>   | 2012-10-08 | GL5115A |

|       |            |                                   |   |            |         |
|-------|------------|-----------------------------------|---|------------|---------|
| V2.00 | IRC        | GL5115-V1.0<br>2-20121011-0<br>52 | 删除 IR 模拟部分相关寄存器   | 2012-10-11 | GL5115A |
| V2.00 | GPIO       | GL5115-V1.0<br>2-20121012-0<br>53 | <p>1. 更改 2.1.3.2 PAD, REM_CON 增加 GPIOG2 复用;</p> <p>GPIOG2, GPIOC4, GPIOC3, GPIOA6, GPIOA5, GPIOC7 分别增加 TK0, TK1, TK2, TK3, TK4, TK5 复用;</p> <p>1. 更改 2.1.3.3 表格, REM_CON 增加 GPIOG2 复用;</p> <p>GPIOG2, GPIOC4, GPIOC3, GPIOA6, GPIOA5, GPIOC7 分别增加 TK0, TK1, TK2, TK3, TK4, TK5 复用;</p> <p>3. 更改 2.1.3.4 表格, REM_CON 增加 GPIOG2 复用;</p> <p>GPIOG2, GPIOC4, GPIOC3, GPIOA6, GPIOA5, GPIOC7 分别增加 TK0, TK1, TK2, TK3, TK4, TK5 复用;</p> <p>4. GPIOGOUTEN, GPIOGINEN, GPIOGDAT, GPIOGPUEN, GPIOGPDEN 寄存器增加 bit2, GPIOG2;</p> <p>5. AD_Select 寄存器名称改为 AD_Select0;</p> <p>6. AD_Select0 寄存器增加 bit6, 用于 REM_CON;</p> <p>7. 增加 AD_Select1 寄存器, 用于选择 Touch Key;</p> <p>8. PADDRV4 寄存器增加 bit3, bit2, 用于调节 GPIOG2 驱动档位。</p> | 2012-10-12 | GL5115A |
| V2.00 | GPIO       | GL5115-V1.0<br>2-20121012-0<br>54 | <p>1. 更改 2.1.3.2 PAD, GPIOC6 增加 TK6 复用;</p> <p>2. 更改 2.1.3.3 表格, GPIOC6 增加 TK6 复用;</p> <p>3. 更改 2.1.3.4 表格, GPIOC6 增加 TK6 复用;</p> <p>4. AD_Select1 寄存器, 增加 bit6, 用于选择 TK6。</p>  | 2012-10-12 | GL5115A |
| V2.00 | GPIO       | GL5115-V1.0<br>2-20121015-0<br>55 | <p>1. LED_SEG_BIAS_EN 寄存器增加说明;</p> <p>2. AD_Select0, AD_Select1 寄存器重新合并修改。</p>  | 2012-10-15 | GL5115A |
| V2.00 | CMU_analog | GL5115-V1.0<br>2-20121016-0<br>56 | <p>1. 将 debug 后门输出 GPIO 从 GPIO_C4 修改为 GPIO_D0, 这是因为 GPIO_C4 不确定是否复用做 EJTAG_TRST, 干脆直接避开, 而且 GPIO_D0 各个封装都出 pin 了;</p> <p>2. 高校低电路默认使能, calendar 默认选</p>   | 2012-10-16 | GL5115A |

|       |              |                                   |   |            |         |
|-------|--------------|-----------------------------------|---|------------|---------|
|       |              |                                   | <p>择高校低送过来的 4Hz 时钟；</p> <p>3、修改 HOSC 框图，将电源域修改为 BAT</p> <p>4、修改时钟树以及电源域说明，将 HOSC 变更到 BAT 电压域，并且增加高校低电路到 BAT 电压域的控制关系；</p>   |            |         |
| V2.00 | CMU_digitala | GL5115-V1.0<br>2-20121015-0<br>57 | <p>1、删除 FMRDS 模块的说明以及配置，将 FMRDS 时钟控制位 reserved；</p> <p>2、修改 FIR_CS_RAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替；</p> <p>3、修改 FIR_CS_RAM 的时钟配图以及说明；</p> <p>4、修改 FIR_AA_RAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替；</p> <p>5、修改 FIR_AA_RAM 的时钟配图以及说明；</p> <p>6、修改 FIR_MPX_RAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替；</p> <p>7、修改 FIR_MPX_RAM 的时钟配图以及说明；</p> <p>8、修改 FIR_RDS_RAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替；</p> <p>9、修改 FIR_RDS_RAM 的时钟配图以及说明；</p> <p>10、修改 MURAM1 的时钟选择，FMRDS 的时钟选择用 reserved 代替；</p> <p>11、修改 MURAM1 的时钟配图以及说明；</p> <p>12、修改 PCMRAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替；</p> <p>13、修改 PCMRAM 的时钟配图以及说明；</p> <p>14、修改 DAC_IIS_TX 的时钟配图，删除当中 FMRDS_19M_GT 的时钟选择；</p> <p>15、删除 IIC 的时钟选择以及配置说明，将 IIC 时钟控制位 reserved；</p> <p>16、删除 SPIBOOT 的时钟选择以及配置说明，将 SPIBOOT 时钟控制位 reserved；</p> <p>17、修改 IR 模块时钟框图，删除 IR_TEST_1KHz 信号，因为 IR_ANALOG 电路已经删除；</p> <p>18、修改 CLKENCTL2，增加 TK 模块的时钟使能；</p> <p>19、增加 TK 模块的时钟框图和说明；</p> | 2012-10-16 | GL5115A |

|       |          |                                   |  |            |         |
|-------|----------|-----------------------------------|--|------------|---------|
|       |          |                                   | 20、 根据 1 到 19 点的修改, 更新 Block Diagram 框图;<br>根据 1 到 19 点的修改, 更新 clock requirements 表格;  |            |         |
| V2.00 | FMRX     | GL5115-V1.0<br>2-20121015-0<br>58 | 1、删除 FMRX 模块   | 2012-10-16 | GL5115A |
| V2.00 | RMU      | GL5115-V1.0<br>2-20121015-0<br>59 | 1. 修改 MRCR1, 将 FMRDS 和 IIC 的复位控制 reserved;<br>2. 修改 MRCR3, 增加 TK 的复位控制;<br>3. 根据第 1 和第 2 点, 修改 Block Diagram;  | 2012-10-16 | GL5115A |
| V2.00 | GPIO     | GL5115-V1.0<br>2-20121018-0<br>60 | 1. DBGSEL 寄存器 bit[4:0] 01010: Reserved 改为: 01010:TK analog debug;<br>2. 各模块使用的 DEBUG 信号, 增加 TK 信号。   | 2012-10-18 | GL5115A |
| V2.00 | AUDIOI O | GL5115-V1.0<br>2-20121018-0<br>61 | 1、因去掉内部 FM, 修改主框图<br>2、修改以下寄存器:<br><br>DAC_CH0_SRFT_CTL0、<br>DAC_CH1_SRFT_CTL1、<br>DAC_CH0_SRFT_CTL5、<br>DAC_CH1_SRFT_CTL5、<br>DAC_CH0_SR_GAIN、<br>DAC_CH1_SR_GAIN、<br>DAC_CH1_PCMH、<br>AINOP_CTL、<br>DAC_CH0_FIFO_CTL、<br>DAC_CH1_FIFO_CTL<br>3、修改以下寄存器:<br>ADC_SRFT_CTL0、<br>ADC_SRFT_CTL5 | 2012-10-18 | GL5115A |
| V2.00 | TK       | GL5115-V1.0<br>2-20121018-0<br>62 | 1、新增 touch-key 模块  | 2012-10-18 | GL5115A |
| V2.00 | UART     | GL5115-V1.0<br>2-20121019-0<br>63 | 1、增加 UART 时序错误状态位  | 2012-10-19 | GL5115A |
| V2.00 | TK       | GL5115-V1.0<br>2-20121019-0<br>64 | 1、Debug 模式下, TK_EN 和 TK_ANA_EN 由同一个 GPIO 控制  | 2012-10-19 | GL5115A |
| V2.00 | INTC     | GL5115-V1.0<br>2-20121019-0<br>65 | 1、删除 IIC 中断, 更替为 TK 中断   | 2012-10-19 | GL5115A |
| V2.00 | GPIO     | GL5115-V1.0<br>2-20121019-0<br>67 | 1、所有寄存器 SFR address 和 Bank 地址互换。   | 2012-10-19 | GL5115A |

|       |             |                                   |  |            |         |
|-------|-------------|-----------------------------------|--|------------|---------|
| V2.00 | SFR         | GL5115-V1.0<br>2-20121019-0<br>67 | 2、删除 IIC 模块寄存器列表<br>3、修改 IRC 模块寄存器列表<br>4、增加 TK 模块寄存器列表<br>5、修改 GPIO 模块寄存器列表   | 2012-10-19 | GL5115A |
| V2.01 | GPIO        | GL5115-V2.0<br>0-20121019-0<br>68 | 1、CMU debug 信号改为 P_GPIOD0  | 2012-10-19 | GL5115A |
| V2.01 | RTC         | GL5115-V2.0<br>0-20121023-0<br>69 | 1、修正 RTCTimeY 的有效位为 bit[6:0]   | 2012-10-23 | GL5115A |
| V2.01 | RMU         | GL5115-V2.0<br>0-20121023-0<br>70 | 1、修改 MRCR3 bit1 为 lcd reset<br>2、修改 MRCR3 bit2 为 TK reset  | 2012-10-23 | GL5115A |
| V2.01 | AUDIOI<br>P | GL5115-V2.0<br>0-20121031-0<br>71 | 1、SRS 的控制寄存器与 karaoke 的控制寄存器地址相同,因此修改 karaoke 的默认值<br>2、删除 SRSCTL2 寄存器中的 DefinitionREn 和 DefinitionLEn 位   | 2012-10-31 | GL5115A |
| V2.01 | TK          | GL5115-V2.0<br>0-20121106-0<br>72 | 1. 因为 PWM 的 N 可调,因此在 TK 电流自动调节模块中需要增加 TK1 和 TK2 可调,在 spec 当中加入了 4 个寄存器来设置 TK1 和 TK2:<br><br>TK_AUTO_LOWTH_H<br>TK_AUTO_LOWTH_L<br>TK_AUTO_HIGHTH_H<br>TK_AUTO_HIGHTH_L<br>2. 根据第 1 点,调整了寄存器的地址,更新了 register list;<br>3. 调整了 TK_PRESS_TH 各个档位的阈值,并将默认值修改为 01101;<br>将 TK_NOISE_TH 的默认值修改为 1000;<br>4、调整了 debug 信号的排列,这样 coding 时好例化 | 2012-11-06 | GL5115A |
| V2.01 | GPIO        | GL5115-V2.0<br>0-20121106-0<br>73 | 1、DBGSEL 寄存器 bit[4:0] 10000: Reserved 改为: EFUSE Debug Signals; 2、各模块使用的 DEBUG 信号增加 EFUSE GPIOA[5:0]。   | 2012-11-06 | GL5115A |
| V2.01 | USB         | GL5115-V2.0<br>0-20121106-0<br>74 | 1、USB_PHYCTRL 寄存器描述修改  | 2012-11-06 | GL5115A |
| V2.01 | SFR         | GL5115-V2.0<br>0-20121106-0<br>75 | 1、修改 TK 寄存器  | 2012-11-06 | GL5115A |
| V2.01 | AUDIOI      | GL5115-V2.0                       | 1、增加 SPDIFRX 电路, 增加寄存器   | 2012-11-06 | GL5115A |

|       |           |                                   |   |            |         |
|-------|-----------|-----------------------------------|---|------------|---------|
|       | O         | 0-20121106-0<br>76                | RXPBDC, CSFU, RXCP.<br>2、增加 PWCTOP, 以支持 SPDIF 拔线 time out 场景.     |            |         |
| V2.01 | USB       | GL5115-V2.0<br>0-20121106-0<br>77 | 1、Usbirq_hcusbirq 的 bit6 默认值改为 1                                  | 2012-11-07 | GL5115A |
| V2.02 | TK        | GL5115-V2.0<br>1-20121115-0<br>01 | 1、TK 增加 25 个寄存器<br>2、TK 的 key 在使能信号为 0 时, 将 KEY 接地, 以避免浮空时容易受干扰。  | 2012-11-15 | GL5115A |
| V2.02 | USB       | GL5115-V2.0<br>1-20121123-0<br>02 | 1、完善 operation manual 章节  | 2012-11-23 | GL5115A |
| V2.02 | DMA       | GL5115-V2.0<br>1-20121123-0<br>03 | 1、完善 operation manual 章节  | 2012-11-23 | GL5115A |
| V2.02 | SD        | GL5115-V2.0<br>1-20121203-0<br>04 | 1、完善 operation manual 章节  | 2012-12-03 | GL5115A |
| V2.02 | GPIO      | GL5115-V2.0<br>1-20121203-0<br>05 | 1、完善 operation manual 章节<br>2、触摸按键 TK6 名称改为 SHIELD                | 2012-12-03 | GL5115A |
| V2.02 | LCD       | GL5115-V2.0<br>1-20121203-0<br>06 | 1、完善 operation manual 章节  | 2012-12-03 | GL5115A |
| V2.02 | TEST MODE | GL5115-V2.0<br>1-20121203-0<br>07 | 1、新增 test mode 章节   | 2012-12-03 | GL5115A |
| V2.02 | UART      | GL5115-V2.0<br>1-20121203-0<br>08 | 1、完善 operation manual 章节  | 2012-12-03 | GL5115A |
| V2.02 | SPI       | GL5115-V2.0<br>1-20121203-0<br>09 | 1、完善 operation manual 章节  | 2012-12-03 | GL5115A |
| V2.02 | PMU       | GL5115-V2.0<br>1-20121203-0<br>10 | 1、完善 operation manual 章节  | 2012-12-03 | GL5115A |
| V2.02 | IRC       | GL5115-V2.0<br>1-20121203-0<br>11 | 1、完善 operation manual 章节  | 2012-12-03 | GL5115A |
| V2.02 | RTC       | GL5115-V2.0<br>1-20121203-0<br>12 | 1、完善 operation manual 章节<br>2、将 RTC debug 信号 GPIOD7 修改为高较低的高频使能信号 | 2012-12-03 | GL5115A |
| V2.02 | PMU       | GL5115-V2.0<br>1-20121203-0       | 1. 修改 EFUSE0 bit7-5 的功能及默认值<br>2. 修改 EFUSE1 bit6,7 Reserved 位的描   | 2012-12-03 | GL5115A |

|       |          |                                   |   |            |         |
|-------|----------|-----------------------------------|---|------------|---------|
|       |          | 13                                | 述, 改为 for future use  |            |         |
| V2.02 | DAC/ADC  | GL5115-V2.0<br>1-20121203-0<br>14 | 1、删除 SRC Fine Tune 硬件 counter 相关寄存器及配置<br>2、增加 operation manual 章节  | 2012-12-03 | GL5115A |
| V2.02 | SPDIF    | GL5115-V2.0<br>1-20121203-0<br>15 | 1、增加完善 opertaion manual 章节  | 2012-12-03 | GL5115A |
| V2.02 | BROM     | GL5115-V2.0<br>1-20121204-0<br>16 | 1、新增 BROM 章节  | 2012-12-04 | GL5115A |
| V2.02 | TK       | GL5115-V2.0<br>1-20121204-0<br>17 | 1、完善 operation manual 章节  | 2012-12-04 | GL5115A |
| V2.02 | DMA      | GL5115-V2.0<br>1-20121204-0<br>18 | 1、删除 DMAXSADDR1 和<br>DMAXDADDR1 寄存器及 Symbols<br>and abbreviations 中相关描述   | 2012-12-04 | GL5115A |
| V2.02 | RMU      | GL5115-V2.0<br>1-20121204-0<br>19 | 1、将 SFR 总线访问 USB 控制器的 wait cycle 位宽由 3bits 扩宽到 4bits, 将 PCON bit3 定义为 SFR_wait_bit3.  | 2012-12-04 | GL5115A |
| V2.02 | CMU      | GL5115-V2.0<br>1-20121206-0<br>20 | 1、增加软件 operation manual 说明部分;<br>2、根据高校低 RTC 时间测试实验, 修改 CALENDAR_CLK_CTL 寄存器, 增加了 HOSC 基准振荡时间选择, 和 HCL 校准间隔的选择;<br>3、为了降低功耗, 增加 CALENDAR_CLK_CTL 增加 RTCVDD 电压域下 24M 到 4Hz 分频电路的 gating 控制;<br>4、修改 HCL 电路框图, 将图中的 2Hz 修改为 4Hz;<br>5、删除高校低内部信号说明, 只是以框图形式说明; | 2012-12-06 | GL5115A |
| V2.02 | RTC      | GL5115-V2.0<br>1-20121206-0<br>21 | 1、在 RTC_CTL0 增加高频 4HZ 分频选择 bit。<br>2、修改高频 4HZ 相关时钟图, operation manul  | 2012-12-06 | GL5115A |
| V2.02 | RTC      | GL5115-V2.0<br>1-20121207-0<br>22 | 1、在 RTCVDD_RESERVED 寄存器增加 SYS_FAULT 指示位   | 2012-12-07 | GL5115A |
| V2.02 | AUDIOIOP | GL5115-V2.0<br>1-20121207-0<br>23 | 1、删除 SRSTRL0 的 bit0 SRSEn, 将该 bit 修改成 SurroundEn<br>2、SRSTRL0 的 bit0 变成 SurroundEn 控制位<br>3、SRSTRL0 的 bit6 变成 SRSTruBassProcessMode 控制位   | 2012-12-07 | GL5115A |

|       |             |                                   |  |            |         |
|-------|-------------|-----------------------------------|--|------------|---------|
|       |             |                                   | 4、由于 SRSCTL0 寄存器的默认值变化，并且 LCHGain 与 SRSCTL0 的寄存器的复用的。因此 LCHGain 的默认值变成 0x71。   |            |         |
| V2.02 | PMU         | GL5115-V2.0<br>1-20121210-0<br>24 | 1、增加 RTCVDD2_LDO, 给 RC 振荡器供电，并修改电源框图。<br>2、根据仿真结果，将 RTCVDD 电压修改为 1.8V，波动 10%。<br>3、增加 bandgap 中可校准范围的描述  | 2012-12-10 | GL5115A |
| V2.02 | SFR         | GL5115-V2.0<br>1-20121211-0<br>26 | 1、修改 TK 模块寄存器列表<br>2、修改 DMA 模块寄存器列表<br>3、修改 SDC 模块寄存器列表寄存器名称<br>4、增加 EM 模块寄存器 EXTMEM_DH<br>5、修改 SRS 模块寄存器列表<br>6、修改 CMU Digital 模块寄存器列表  | 2012-12-11 | GL5115A |
| V2.03 | Memory      | GL5115-V2.0<br>2-20121214-0<br>01 | 1、删除 FM/RDS 相关描述   | 2012-12-14 | GL5115A |
| V2.03 | AudioIP     | GL5115-V2.0<br>2-20121215-0<br>02 | 1、增加 operation manual 描述<br>2、修改了 MP3CTIMEH 地址，应该是 0xb2 而不是 0xa2（笔误）<br>3、修改了 WMACTIMEH 地址，应该是 0xb2 而不是 0xa2（笔误）<br>4、修改了 WAVCTIMEH 地址，应该是 0xb2 而不是 0xa2（笔误）                                     | 2012-12-15 | GL5115A |
| V2.03 | CMU_digital | GL5115-V2.0<br>2-20121224-0<br>03 | 1、将原来的 CALENDAR_CLK_CTL 寄存器名称变更为 HCL_CLK_CTL，寄存器地址不变；<br>2、在 HCL_CLK_CTL 中，将 HOSC_OSC_TIME 默认值设置为 500ms，增加 HCL 误差精度选择位；<br>3、增加 HCL_INTERVAL_CTL 控制寄存器，地址为 0xb5；<br>4、针对前面 3 点，修改 register list； | 2012-12-24 | GL5115A |
| V2.03 | AudioIP     | GL5115-V2.0<br>2-20121226-0<br>04 | 1、修改 PCM 的 almost empty/almost full 的门限值设定   | 2012-12-26 | GL5115A |
| V2.03 | AudioIO     | GL5115-V2.0<br>2-20120107-0<br>05 | 1、ASRC 模块新增 DA0DE2EN, DA1DE2EN, ADDE2EN 控制位。   | 2013-01-07 | GL5115A |
| V2.03 | CMU_digital | GL5115-V2.0<br>2-20130108-0<br>06 | 1、对 CPU 时钟框图的 L OSC 源头选择进行说明；<br>2、对 RTC 时钟框图中的 L OSC 源头选择进行说明；<br>3、对 DAC/IIS-TX 时钟框图中的 L OSC 源   | 2013-01-08 | GL5115A |

|       |             |                                   |   |            |         |
|-------|-------------|-----------------------------------|---|------------|---------|
|       |             |                                   | 头选择进行说明;<br>4、对 FMCLK 时钟框图中的 L OSC 源头选择进行说明;<br>5、增加 PWM 的时钟源无缝切换描述;<br>6、对 LED&SEG LCD 时钟框图中的 L OSC 源头选择进行说明;<br>7、增加 ASRC 的时钟源无缝切换描述;<br>8、将 CLKENCTL0 的 bit2 由只读位修改为可读写位;<br>9、将 CLKENCTL1 的 bit4 和 bit1 由只读位修改为可读写位                           |            |         |
| V2.03 | RTC         | GL5115-V2.0<br>2-20130109-0<br>07 | 1、将 operation manual 里高频 4HZ 切回高较低等待时间修改为 800ms。  | 2013-01-09 | GL5115A |
| V2.03 | CMU_digital | GL5115-V2.0<br>2-20130115-0<br>08 | 1、根据设计预期，修改 DAC/IIS-TX 的时钟框图，对写 DAC/IIS TX ch0 FIFO 的时钟增加 ADC_CLK2 的选项；<br>2、根据设计预期，修改 ADC 时钟框图，将写 ADC/IIS RX FIFO 的时钟源，由 ADCIFclk 修改到 ADC_CLK2；<br>3、删除 IIS-RX 中信号列表中的 ADCIFclk 输出项，同时将 IIS_rx_clock 修改为输出项；这是因为 IIS-RX 的输出，只有 IIS_rx_clock 这一项； | 2013-01-15 | GL5115A |
| V2.03 | PMU         | GL5115-V2.0<br>2-20130116-0<br>09 | 1、根据模拟实际所做，修改 VDDOK 到 preok 的时间为 16mS，原 16-32mS   | 2013-01-16 | GL5115A |
| V2.03 | USB         | GL5115-V2.0<br>2-20130117-0<br>10 | 1、增加 PHY 相关描述。<br>2、增加 PHY 描述<br>3、增加 APHY 寄存器描述<br>4、增加 DPHY 寄存器描述<br>5、增加 PHY BUDEG 寄存器描述   | 2013-01-17 | GL5115A |
| V2.03 | GPIO        | GL5115-V2.0<br>2-20130117-0<br>11 | 1、FMINL 改成和 GPIOF2 复用，FMINR 改成和 GPIOF3 复用，BTINL 改成和 GPIOF4 复用，BTINR 改成和 GPIOF5 复用。  | 2013-01-18 | GL5115A |
| V2.03 | SFR         | GL5115-V2.0<br>2-20130124-0<br>12 | 1、修改 CMU_analog 模块寄存器列表   | 2013-01-24 | GL5115A |
| V2.04 | LED&LCD     | GL5115-V20<br>3-20130524-0<br>01  | 1、模块使用指导部分加上 IC 验证后的说明：clk distribution 部分修改图片和频率设置说明。<br>2、增加模拟部分的设置说明：Seg_lcd 的 1/3, 2/3 电压使能，Led 模块的恒流源使能。   | 2013-5-24  | GL5115B |
| V2.04 | DAC         | GL5115-V20                        | 增加比特位 I2SDAOSEN 控制 I2S 和 DAC  | 2013-06-24 | GL5115B |

|       |                |                                  |   |            |         |
|-------|----------------|----------------------------------|---|------------|---------|
|       |                | 3-20130624-0<br>02               | 是否同时输出.   |            |         |
| V2.04 | DAC            | GL5115-V20<br>3-20130624-0<br>03 | 更改 PA_VOLUME.   | 2013-06-24 | GL5115B |
| V2.04 | SPDIF          | GL5115-V20<br>3-20130624-0<br>04 | 增加状态比特位 CSPCM 和 CSCSM 软件可以查询，硬件自动检测非支持源且丢 0.  | 2013-06-24 | GL5115B |
| V2.04 | CARD           | GL5115-V20<br>3-20130626-0<br>05 | Operation manual 添加了新的内容:添加了说明项 8   | 2013-06-26 | GL5115B |
| V2.04 | CMU            | GL5115-V20<br>3-20130626-0<br>06 | 1、将 HOSC 起振时间由 1ms 修改为 1.5ms，这是实测值；<br>2、将 MCU_PLL_CTL bit6 描述修改为 MCUPLL_PMD，表示 MCUPLL 的相位锁定检测，这是为了使得描述更加准确；<br>3、将 AUDIO_PLL_CTL bit7 描述修改为 AUDIO_PLL_PMD，表示 AUDIOPLL 的相位锁定检测，这是为了使得描述更加准确；<br>4、增加 debug signal 的信号描述，使得描述更加准确； | 2013-06-26 | GL5115B |
| V2.04 | RTC            | GL5115-V20<br>3-20130626-0<br>07 | 在 operation manul 里增加上电使用说明   | 2013-06-26 | GL5115B |
| V2.04 | TK             | GL5115-V20<br>3-20130626-0<br>08 | 1、在 operation manual 部分，增加按键使能控制的说明，这是由于实际设计中按键组合逻辑存在问题导致的；   | 2013-06-26 | GL5115B |
| V2.04 | ChipVersion    | GL5115-V20<br>3-20130524-0<br>09 | 1、修改 chip Version 寄存器默认值  | 2013-06-27 | GL5115B |
| V2.05 | Pin_assignment | GL5115-V20<br>4-20130909-0<br>01 | 1、ATS2509 名字改为 ATS2885；<br>2、增加 ATS2511 和 ATS2513 的 PIN 排布  | 2013-09-09 | GL5115B |

# Acronyms, and Abbreviations

| Abbreviations | Descriptions   |
|---------------|--|
| ADC           | Analog-to-Digital-Converter, IC 内将模拟信号转化为数字信号的电路                                   |
| BIRD          | BUILT-IN REAL-TIME DEBUGGER  |
| CP            | charge pump  |
| CTC           | Counter/ Timer channels  |
| DAC           | Digital-to-Analog-Converter, IC 内将数字信号转化为模拟信号的电路                                   |
| DMA           | Direct Memory Access   |
| GPIO          | General Purpose Input Output   |
| HOSC          | High Frequency OSC (24MHz)   |
| I2C           | Inter IC bus—两线式串行总线   |
| I2S           | Inter IC Sound Bus, Philips 公司制定的一种音频数据传输标准  |
| INTC          | Interrupt Controller   |
| IR            | infrared ray—红外接收协议  |
| IRQ           | Interrupt Request  |
| LDO           | low dropout regulator  |
| LOSC          | Low Frequency OSC, include internal RC OSC (about 32K) and external LOSC (32.768K) |
| MCU           | Microprocessor Control Unit  |
| NMI           | Nonmaskable Interrupt  |
| PA            | Power Amplifier, 将 DAC 转化的模拟信号, 放大一定的功率以驱动耳机负载                                     |
| PMU           | POWER MANAGEMENT UNIT  |
| Rlink-BIRD    | USB-JTAG adapter   |
| RTC           | Real time clock  |
| SDC           | SD card controller   |
| SIE           | Serial Interface Engine  |
| SPDIF         | Sony Philips Digital Interconnect Format, Sony & Philips 公司联合制定的一种音频数据传输标准         |
| SPI           | Serial Peripheral Interface—串行外设接口   |
| TTS           | Text to Speech, 语音播报   |
| UART          | Universal Asynchronous Receiver/Transmitter—通用异步接收/发送装置                            |
| UTMI          | USB Transceiver Macro Interface  |

# 1 Introduction

The GL5115 is a new generation single-chip highly-integrated digital multimedia solution for device such as audio players. It includes an audio decoder with embedded RAM and ROM, ADPCM or MP2 record capability, and USB interface for downloading music and uploading voice recordings. GL5115 also provides an interface to SD/MMC flash card,SPI,SPDIF,I2C,UART,IR,LED/LCD,button and switch inputs, headphones and microphone. Its embedded audio codec supports the WAV, MP3, and WMA. For devices like USB-Disk, it can act as a USB mass storage slave device to personal computer system. The Chip has low power consumption to allow long battery life. The built-in Sigma-Delta D/A includes a headphone driver to directly drive low impedance headphone. The A/D includes input for both Microphone and Analog Audio in to support voice recording, external audio source and FM radio integration features. Thus, it provides a true “ALL-IN-ONE” solution that is ideally suited for highly optimized digital audio players.

## 1.1 Features

### Flip80251 Core

- Integrated MCU with EJTAG Debug Unit, the instruction set is compatible with 80C51/8xC251
- A pipelined architecture providing single cycle execution for most of the instruction when the pipeline is full
- MCU run at 24MHz(TBD),F/W can program from DC up to 72MHz(TBD) transparently

### Audio Engine

- WMA Decoder, bit rate 32-384Kbps,sample rate8-48KHz
- Digital Voice Recording(ADPCM or MP2)

### Internal memory

- On-chip PRAM251(22K\*8) is mapped to MCU program memory space.
- On-chip DRAM251A(10.5K\*8) is mapped to MCU data and Stack memory space.
- On-chip Multi-Use RAM(MURAM1:15K\*8) that can be switched to be MCU program or data memory space or Audio hardware codec memory.
- On-chip USB RAM(URAM:384\*8) that can be switched to be MCU data memory space or USB controller memory.
- Some On-chip RAM(DAC\_RAM:96\*16,ADCRAM1:32\*16,ADCRAM2:44\*31) can only be used by DAC/ADC
- Other On-chip RAM (FIR\_CS\_RAM:256\*24, FIR\_AA\_RAM:64\*24, FIR\_MPX\_RAM1.75K\*24, FIR\_RDS\_RAM: 192\*16) that can be switched to be MCU data memory space or FM&RDS demodulator.
- Internal BROM(25K\*8) build in Boot up,USB upgrade firmware and part of firmware such as exfat.
- Internal ROM(ROM1:4096\*24,ROM2:7104\*16,ROM3:2880\*8) for Audio hardware codec memory.
- Internal SRAM(ADC/DAC RAM excepted) and BROM can run up to 150MHz, Audio ROM and ADC/DAC RAM can run up to 50MHz.

### External memory

- Support SPI NorFlash(1/4 bit mode)
- Support following memory card interface
- Multi Media Card Specification Version 4.3(1/4 bit mode)

- Secure Digital Card Specification Version 2.0(1/4 bit mode)
- Support Booting up directly from SD/MMC/Emmc/SPI Norflash.

**Highly-Integrated System peripherals**

- Embedded FM&RDS receiver
- Embedded ChargePump for 5V supply
- Support 24MHz OSC with on-chip PLL
- Support external 32.768KHz OSC
- Support internal 32KHz RC Oscillator
- USB 2.0 device and host controllers with HOST support
- A variety of serial controllers supporting I2C,SPI,SPDIF,UART,IR,I2S
- Support LCM with 8bit CPU Interface

**Audio**

- Built-in stereo Sigma-delta DAC of 92Db SNR, SNR(A-WEIGHTING)>95Db, THD<-75Db; Support Multi-Sample-Rate of 8k/12k/11.025k/16k/22.05k/24k/32k/44.1/48/96kHz
- Built-in TTS circuit to add speech to music
- Support I2S Transmitter and Receiver
- Built-in stereo Sigma-delta ADC of 84Db SNR, Support Multi-Sample-Rate of 8k/12k/11.025k/16k/22.05k/24k/32k/44.1/48k
- Support Microphone/FM/AUX/BT input, which can mix two of four to ADC. This input PIN can be used as GPIO
- DAC Support Sample Rate Convert
- DAC and ADC Support Sample Rate Fine Tune
- SPDIF supports receiver mode only, Supports sampling rate 192k,96k,48k,44.1k,32k
- Built-in stereo Power Amplifier (PA) for 16/32 ohm Headphones, output power : 2×20Mw@16ohm, Built-in anti-pop circuit for ‘click and pop’ suppressing
- PA output supports traditional mode and direct drive mode. Direct drive mode PIN VRO and VROS can be used as GPIO.

**Power**

- Operating voltage: I/O 3.3V, Core 1.8V
- Standby Leakage Current:<50Ua(Whole System)
- Low Power Consumption:<40Mw@1.6V at typical MP3 decoder solution(TBD);  
<40Mw@1.6V at typical WMA decoder solution(TBD)

**Manufacture**

- 0.144um 1P6M MIM CMOS process
- Package at LQFP-48(7mmx7mm),LQFP-64(7mmx7mm)

## 1.2 Application Diagram

## 1.3 Block Diagram

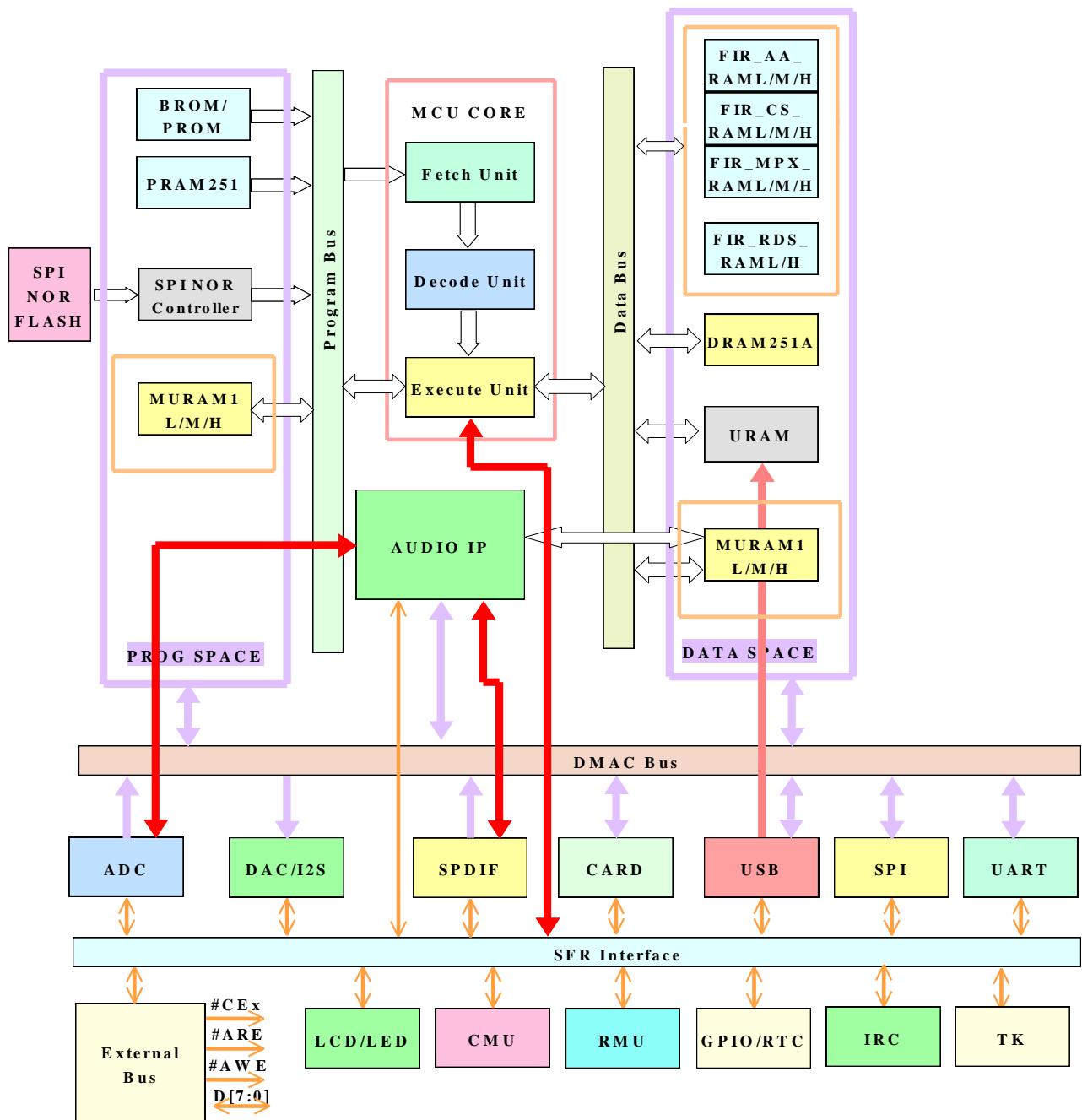


Figure 1-1 block diagram of GL5115

## 2 Register List

| 日期         | 版本    | 描述  | 修订人 |
|------------|-------|---|-----|
| 2012-07-24 | V1.00 | initial   | 黃少彬 |
| 2012-08-09 | V1.01 | 1、SPDIF 增加 4 个寄存器<br>2、增减 CMU ANALOG 寄存器列表<br>3、修改 FMRX 模块寄存器<br>4、修改 EM 模块寄存器<br>5、修改 USB 模块寄存器  | 黃少彬 |
| 2012-09-10 | V1.02 | 1、修改 RTC 模块寄存器列表<br>2、修改 MEMORY Controller 模块寄存器列表<br>3、修改 AUIP 模块寄存器列表   | 黃少彬 |
| 2012-10-20 | V2.00 | 1、删除 IIC 模块寄存器列表<br>2、修改 IRC 模块寄存器列表<br>3、增加 TK 模块寄存器列表<br>4、修改 GPIO 模块寄存器列表  | 黃少彬 |
| 2012-11-07 | V2.01 | 1、修改 TK 寄存器   | 黃少彬 |
| 2012-12-10 | V2.02 | 1、修改 TK 模块寄存器列表<br>2、修改 DMA 模块寄存器列表<br>3、修改 SDC 模块寄存器列表寄存器名称<br>4、增加 EM 模块寄存器 EXTMEM_DH<br>5、修改 SRS 模块寄存器列表<br>6、修改 CMU Digital 模块寄存器列表 | 黃少彬 |
| 2013-01-25 | V2.03 | 1、修改 CMU_analog 模块相关寄存器列表   | 黃少彬 |

下表中红色的部分是为了兼容 dolphin80251 和 Intel80251 占用的 All bank 寄存器

|      | 0x0 | 0x1      | 0x2      | 0x3      | 0x4 | 0x5 | 0x6 | 0x7 |
|------|-----|----------|----------|----------|-----|-----|-----|-----|
| 0Xf8 |     | MbankCtl | reserved | reserved |     |     |     |     |
| 0Xf0 | B   |          |          |          |     |     |     |     |
| 0Xe8 | AIE |          |          |          |     |     |     |     |
| 0Xe0 | ACC |          |          |          |     |     |     |     |
| 0Xd8 |     |          |          |          |     |     |     |     |
| 0Xd0 | PSW | PSW1     |          |          |     |     |     |     |
| 0Xc8 |     |          |          |          |     |     |     |     |
| 0Xc0 | AIF |          |          |          |     |     |     |     |
| 0Xb8 |     |          |          |          |     | SPH |     |     |
| 0Xb0 |     |          |          |          |     |     |     |     |
| 0Xa8 | IE0 |          |          |          |     |     |     |     |

|      |         |       |     |     |      |         |        |        |
|------|---------|-------|-----|-----|------|---------|--------|--------|
| 0xa0 | P2      | MPAGE |     |     |      |         |        |        |
| 0x98 |         |       |     |     |      |         |        |        |
| 0x90 |         |       |     |     |      |         | ExWait |        |
| 0x88 | IF0     |       |     |     |      |         | CCMCON | CCMVAL |
| 0x80 | SFRBANK | SPL   | DPL | DPH | DPXL | BIRDINF |        | PCON   |

| bank number | Bank 地址 | Mnemonic   | Description                                 | 总功能               |
|-------------|---------|------------|---|-------------------|
| all bank    | 0xbe    | SPH        | Stack Pointer high byte                     | 内核寄存器             |
| all bank    | 0x81    | SPL        | Stack Pointer low byte                      |                   |
| all bank    | 0x84    | DPXL       | Data Pointer extended byte                  |                   |
| all bank    | 0x83    | DPH        | Data Pointer high byte                      |                   |
| all bank    | 0x82    | DPL        | Data Pointer low byte                       |                   |
| all bank    | 0xd0    | PSW        | Program Status Word                         |                   |
| all bank    | 0xd1    | PSW1       | Program Status Word1                        |                   |
| all bank    | 0xe0    | ACC        | Accumulator                                 |                   |
| all bank    | 0xf0    | B          | B Register                                  |                   |
| all bank    | 0x96    | ExWait     | External bus access wait cycle register     | ExBUS             |
| all bank    | 0x87    | PCON       | Power Control Register                      | Power Control     |
| all bank    | 0x8e    | CCMCON     | Communication Control Register              | 用于调试              |
| all bank    | 0x8f    | CCMVAL     | Communication Value Register                |                   |
| all bank    | 0x85    | BIRDINFO   | Bird information                            |                   |
| all bank    | 0x80    | SFR_BANK   | SFR Banking control Register                | Cut Bank          |
| all bank    | 0xa0    | P2         | Port2 Register                              | IA                |
| all bank    | 0xa1    | MPAGE      | 与 Port2 寄存器的内容相同                            | IA                |
| all bank    | 0xa8    | IE0        | Interrupt Enable register 0                 | 中断控制              |
| all bank    | 0xe8    | AIE        | Additional Interrupt Enable register        |                   |
| 0x00        | 0xb7    | IPH0       | Interrupt Priority High register 0          |                   |
| 0x00        | 0xb8    | IPL0       | Interrupt Priority Low register 0           |                   |
| 0x00        | 0xf7    | AIPH       | Additional Interrupt Priority High register |                   |
| 0x00        | 0xf8    | AIPL       | Additional Interrupt Priority Low register  |                   |
| all bank    | 0x88    | IF0        | Interrupt Flag register 0                   |                   |
| all bank    | 0xc0    | AIF        | Additional interrupt flag register          |                   |
| 0x06        | 0xd8    | EXTINT     | External Interrupt Control                  |                   |
| 0x01        | 0xf5    | IFDebug0   | Interrupt Flag Debug register 0             |                   |
| 0x01        | 0xf6    | IFDebug1   | Interrupt Flag Debug register 0             |                   |
| all bank    | 0xf9    | MemBankCtl | Memory banking control register             | Memory controller |
| 0x00        | 0x90    | PageAddr0H | High byte of Page Address register 0        |                   |
| 0x00        | 0x91    | PageAddr0L | Low byte of Page Address register 0         |                   |

|      |      |                      |                                       |  |
|------|------|----------------------|---------------------------------------|--|
| 0x00 | 0x92 | <b>PageAddr1H</b>    | High byte of Page Address register 1  |  |
| 0x00 | 0x93 | <b>PageAddr1L</b>    | Low byte of Page Address register 1   |  |
| 0x00 | 0x94 | <b>PageAddr2H</b>    | High byte of Page Address register 2  |  |
| 0x00 | 0x95 | <b>PageAddr2L</b>    | Low byte of Page Address register 2   |  |
| 0x00 | 0x98 | <b>PageAddr3H</b>    | High byte of Page Address register 3  |  |
| 0x00 | 0x99 | <b>PageAddr3L</b>    | Low byte of Page Address register 3   |  |
| 0x00 | 0x9a | <b>PageAddr4H</b>    | High byte of Page Address register 4  |  |
| 0x00 | 0x9b | <b>PageAddr4L</b>    | Low byte of Page Address register 4   |  |
| 0x00 | 0x9c | <b>PageAddr5H</b>    | High byte of Page Address register 5  |  |
| 0x00 | 0x9d | <b>PageAddr5L</b>    | Low byte of Page Address register 5   |  |
| 0x00 | 0x9e | <b>PageAddr6H</b>    | High byte of Page Address register 6  |  |
| 0x00 | 0x9f | <b>PageAddr6L</b>    | Low byte of Page Address register 6   |  |
| 0x00 | 0xa2 | <b>PageAddr7H</b>    | High byte of Page Address register 7  |  |
| 0x00 | 0xa3 | <b>PageAddr7L</b>    | Low byte of Page Address register 7   |  |
| 0x00 | 0xd8 | <b>PageAddr8H</b>    | High byte of Page Address register 8  |  |
| 0x00 | 0xd9 | <b>PageAddr8L</b>    | Low byte of Page Address register 8   |  |
| 0x00 | 0xda | <b>PageAddr9H</b>    | High byte of Page Address register 9  |  |
| 0x00 | 0xdb | <b>PageAddr9L</b>    | Low byte of Page Address register 9   |  |
| 0x00 | 0xb0 | <b>PageAddrMask0</b> | Page 0 Address Mask register          |  |
| 0x00 | 0xb1 | <b>PageAddrMask1</b> | Page 1 Address Mask register          |  |
| 0x00 | 0xb2 | <b>PageAddrMask2</b> | Page 2 Address Mask register          |  |
| 0x00 | 0xb3 | <b>PageAddrMask3</b> | Page 3 Address Mask register          |  |
| 0x00 | 0xb4 | <b>PageAddrMask4</b> | Page 4 Address Mask register          |  |
| 0x00 | 0xb5 | <b>PageAddrMask5</b> | Page 5 Address Mask register          |  |
| 0x00 | 0xb6 | <b>PageAddrMask6</b> | Page 6 Address Mask register          |  |
| 0x00 | 0xd2 | <b>PageAddrMask7</b> | Page 7 Address Mask register          |  |
| 0x00 | 0xd4 | <b>PageAddrMask8</b> | Page 8 Address Mask register          |  |
| 0x00 | 0xd5 | <b>PageAddrMask9</b> | Page 9 Address Mask register          |  |
| 0x00 | 0xc8 | <b>RedirectAddr0</b> | Redirect Address 0                    |  |
| 0x00 | 0xc9 | <b>RedirectAddr1</b> | Redirect Address 1                    |  |
| 0x00 | 0xca | <b>RedirectAddr2</b> | Redirect Address 2                    |  |
| 0x00 | 0xcb | <b>RedirectAddr3</b> | Redirect Address 3                    |  |
| 0x00 | 0xcc | <b>RedirectAddr4</b> | Redirect Address 4                    |  |
| 0x00 | 0xcd | <b>RedirectAddr5</b> | Redirect Address 5                    |  |
| 0x00 | 0xce | <b>RedirectAddr6</b> | Redirect Address 6                    |  |
| 0x00 | 0xcf | <b>RedirectAddr7</b> | Redirect Address 7                    |  |
| 0x00 | 0xd6 | <b>RedirectAddr8</b> | Redirect Address 8                    |  |
| 0x00 | 0xd7 | <b>RedirectAddr9</b> | Redirect Address 9                    |  |
| 0x00 | 0xd3 | <b>FixAddr0H</b>     | Hight byte of Fix Address register 0  |  |
| 0x00 | 0xb9 | <b>FixAddr0M</b>     | Middle byte of Fix Address register 0 |  |
| 0x00 | 0xba | <b>FixAddr0L</b>     | Low byte of Fix Address register 0    |  |

|      |      |                           |   |                |
|------|------|---------------------------|---|----------------|
| 0x00 | 0xbb | <b>FixAddr1H</b>          | Hight byte of Fix Address register 1      |                |
| 0x00 | 0xbc | <b>FixAddr1M</b>          | Middle byte of Fix Address register 1     |                |
| 0x00 | 0xbd | <b>FixAddr1L</b>          | Low byte of Fix Address register 1        |                |
| 0x00 | 0xc2 | <b>FixAddr2H</b>          | Hight byte of Fix Address register 2      |                |
| 0x00 | 0xc3 | <b>FixAddr2M</b>          | Middle byte of Fix Address register 2     |                |
| 0x00 | 0xc4 | <b>FixAddr2L</b>          | Low byte of Fix Address register 2        |                |
| 0x00 | 0xc5 | <b>FixAddr3H</b>          | Hight byte of Fix Address register 3      |                |
| 0x00 | 0xc6 | <b>FixAddr3M</b>          | Middle byte of Fix Address register 3     |                |
| 0x00 | 0xc7 | <b>FixAddr3L</b>          | Low byte of Fix Address register 3        |                |
| 0x00 | 0xf4 | <b>PageMissEntryEx</b>    | Page Miss Entry Address Extension byte    |                |
| 0x00 | 0xf5 | <b>PageMissEntryH</b>     | Page Miss Entry Address High byte         |                |
| 0x00 | 0xf6 | <b>PageMissEntryL</b>     | Page Miss Entry Address Low byte          |                |
| 0x00 | 0xa5 | <b>CodeReplaceEntryEx</b> | Code Replace Entry Address Extension byte |                |
| 0x00 | 0xa6 | <b>CodeReplaceEntryH</b>  | Code Replace Entry Address High byte      |                |
| 0x00 | 0xa7 | <b>CodeReplaceEntryL</b>  | Code Replace Entry Address Low byte       |                |
| 0x00 | 0xfc | <b>TestCounterByte0</b>   | Test Counter byte register 0              |                |
| 0x00 | 0xfd | <b>TestCounterByte1</b>   | Test Counter byte register 1              |                |
| 0x00 | 0xfe | <b>TestCounterByte2</b>   | Test Counter byte register 2              |                |
| 0x00 | 0xff | <b>TestCounterByte3</b>   | Test Counter byte register 3              |                |
| 0x00 | 0xf1 | <b>TestCounterByte4</b>   | Test Counter byte register 4              |                |
| 0x01 | 0xe1 | <b>bist_en1</b>           | Bist enable register1                     | bist           |
| 0x01 | 0xe2 | <b>bist_en2</b>           | Bist enable register2                     |                |
| 0x01 | 0xe4 | <b>bist_fin1</b>          | Bist finish flag register1                |                |
| 0x01 | 0xe5 | <b>bist_fin2</b>          | Bist finish flag register2                |                |
| 0x01 | 0xe7 | <b>bist_info1</b>         | Bist information register1                |                |
| 0x01 | 0xe9 | <b>bist_info2</b>         | Bist information register2                |                |
| 0x01 | 0x86 | <b>MRCR1</b>              | Module Reset Control Register 1           | RMU            |
| 0x01 | 0x87 | <b>PCON</b>               | Power Control Register                    |                |
| 0x01 | 0x89 | <b>MRCR2</b>              | Module Reset Control Register 2           |                |
| 0x01 | 0x9f | <b>MRCR3</b>              | Module Reset Control Register 3           |                |
| 0x01 | 0xa9 | <b>CLKENCTL0</b>          | Clock Enable Control Register 0           | CMU<br>digital |
| 0x01 | 0xaa | <b>CLKENCTL1</b>          | Clock Enable Control Register 1           |                |
| 0x01 | 0xab | <b>CLKENCTL2</b>          | Clock Enable Control Register 2           |                |
| 0x01 | 0xac | <b>SDCLKCTL</b>           | SD Card Control Register                  |                |
| 0x01 | 0xc2 | <b>CPUCLKCTL</b>          | CPU Clock Control Register                |                |
| 0x01 | 0xc3 | <b>PWMCLKCTL</b>          | PWM Clock Control Register                |                |
| 0x01 | 0xaf | <b>PWMDUTY</b>            | PWM Duty Control Register                 |                |
| 0x01 | 0xc4 | <b>AECLKCTL</b>           | Audio Codec Clock Control Register        |                |
| 0x01 | 0xc8 | <b>FMCLKCTL</b>           | FM Clock Control Register                 |                |
| 0x01 | 0xc9 | <b>LED_SEGLCDCLKCTL</b>   | LED & SEG LCD clock Control Register      |                |
| 0x01 | 0xca | <b>SPICLKCTL</b>          | SPI controller Clock Control Register     |                |

|                  |      |                          |   |               |
|------------------|------|--------------------------|---|---------------|
| 0x01             | 0xbd | <b>ADC_DAC_CLK_CTL</b>   | ADC & DAC Clock Control Register                |               |
| 0x01             | 0xad | <b>MEMCLKSELCTL0</b>     | Memory Clock Selection Control Register 0       |               |
| 0x01             | 0xae | <b>MEMCLKSELCTL1</b>     | Memory Clock Selection Control Register 1       |               |
| 0x01             | 0xb4 | <b>MEMCLKENCTL0</b>      | Memory Clock Enable Control Register 0          |               |
| 0x01             | 0xb3 | <b>MCU_PLL_CTL</b>       | MCUPLL control register                         | CMU<br>analog |
| 0x01             | 0xb6 | <b>HOSC_CTL</b>          | HOSC control register                           |               |
| 0x01             | 0xb7 | <b>LOSC_CTL</b>          | LOSC control register                           |               |
| 0x01             | 0xb8 | <b>HCL_CLK_CTL</b>       | HCL clock control register                      |               |
| 0x01             | 0xb5 | <b>HCL_INTERVAL_CTL</b>  | HCL calibration interval control register       |               |
| 0x01             | 0xb9 | <b>AUDIO_PLL_CTL</b>     | AUDIO PLL control register                      |               |
| 0x01             | 0xba | <b>MCU_PLL_SSC_CTL</b>   | MCU PLL spread spectrum control register        |               |
| 0x01             | 0xbb | <b>MCU_PLL_SSC_FSTEP</b> | MCU PLL spread spectrum frequency step register |               |
| 0x01             | 0xbc | <b>MCU_PLL_DEBUG_CTL</b> | MCU PLL debug control register                  |               |
| e/f/10/1<br>1/12 | 0x93 | <b>DMA01234DBG</b>       | DMA0/1/2/3 debug register                       | DMA           |
| 0x0e             | 0x91 | <b>DMA0IP</b>            | DMA0 interrupt pending register                 | DMA0          |
| 0x0e             | 0x92 | <b>DMA0IE</b>            | DMA0 interrupt enable register                  |               |
| 0x0e             | 0x94 | <b>DMA0CTL0</b>          | DMA0 control register 0                         |               |
| 0x0e             | 0x95 | <b>DMA0CTL1</b>          | DMA0 control register 1                         |               |
| 0x0e             | 0x97 | <b>DMA0SADDR0H</b>       | DMA0 source address register 0 high byte        |               |
| 0x0e             | 0x98 | <b>DMA0SADDR0L</b>       | DMA0 source address register 0 low byte         |               |
| 0x0e             | 0x9b | <b>DMA0SADDR0M</b>       | DMA0 source address register 0 middle byte      |               |
| 0x0e             | 0x9d | <b>DMA0DADDR0H</b>       | DMA0 destination address register 0 high byte   |               |
| 0x0e             | 0x9e | <b>DMA0DADDR0L</b>       | DMA0 destination address register 0 low byte    |               |
| 0x0e             | 0x9f | <b>DMA0DADDR0M</b>       | DMA0 destination address register 0 middle byte |               |
| 0x0e             | 0xa4 | <b>DMA0FrameLenH</b>     | DMA0 frame length register high byte            |               |
| 0x0e             | 0xa5 | <b>DMA0FrameLenL</b>     | DMA0 frame length register low byte             |               |
| 0x0f             | 0x91 | <b>DMA1IP</b>            | DMA1 interrupt pending register                 | DMA1          |
| 0x0f             | 0x92 | <b>DMA1IE</b>            | DMA1 interrupt enable register                  |               |
| 0x0f             | 0x94 | <b>DMA1CTL0</b>          | DMA1 control register 0                         |               |
| 0x0f             | 0x95 | <b>DMA1CTL1</b>          | DMA1 control register 1                         |               |
| 0x0f             | 0x97 | <b>DMA1SADDR0H</b>       | DMA1 source address register 0 high byte        |               |
| 0x0f             | 0x98 | <b>DMA1SADDR0L</b>       | DMA1 source address register 0 low byte         |               |
| 0x0f             | 0x9b | <b>DMA1SADDR0M</b>       | DMA1 source address register 0 middle byte      |               |
| 0x0f             | 0x9d | <b>DMA1DADDR0H</b>       | DMA1 destination address register 0 high byte   |               |
| 0x0f             | 0x9e | <b>DMA1DADDR0L</b>       | DMA1 destination address register 0 low byte    |               |
| 0x0f             | 0x9f | <b>DMA1DADDR0M</b>       | DMA1 destination address register 0 middle byte |               |
| 0x0f             | 0xa4 | <b>DMA1FrameLenH</b>     | DMA1 frame length register high byte            |               |
| 0x0f             | 0xa5 | <b>DMA1FrameLenL</b>     | DMA1 frame length register low byte             |               |
| 0x10             | 0x91 | <b>DMA2IP</b>            | DMA2 interrupt pending register                 | DMA2          |

|      |      |                        |   |                       |
|------|------|------------------------|---|-----------------------|
| 0x10 | 0x92 | <b>DMA2IE</b>          | DMA2 interrupt enable register                  |                       |
| 0x10 | 0x94 | <b>DMA2CTL0</b>        | DMA2 control register 0                         |                       |
| 0x10 | 0x95 | <b>DMA2CTL1</b>        | DMA2 control register 1                         |                       |
| 0x10 | 0x97 | <b>DMA2SADDR0H</b>     | DMA2 source address register 0 high byte        |                       |
| 0x10 | 0x98 | <b>DMA2SADDR0L</b>     | DMA2 source address register 0 low byte         |                       |
| 0x10 | 0x9b | <b>DMA2SADDR0M</b>     | DMA2 source address register 0 middle byte      |                       |
| 0x10 | 0x9d | <b>DMA2DADDR0H</b>     | DMA2 destination address register 0 high byte   |                       |
| 0x10 | 0x9e | <b>DMA2DADDR0L</b>     | DMA2 destination address register 0 low byte    |                       |
| 0x10 | 0x9f | <b>DMA2DADDR0M</b>     | DMA2 destination address register 0 middle byte |                       |
| 0x10 | 0xa4 | <b>DMA2FrameLenH</b>   | DMA2 frame length register high byte            |                       |
| 0x10 | 0xa5 | <b>DMA2FrameLenL</b>   | DMA2 frame length register low byte             |                       |
| 0x11 | 0x91 | <b>DMA3IP</b>          | DMA3 interrupt pending register                 | DMA3                  |
| 0x11 | 0x92 | <b>DMA3IE</b>          | DMA3 interrupt enable register                  |                       |
| 0x11 | 0x94 | <b>DMA3CTL0</b>        | DMA3 control register 0                         |                       |
| 0x11 | 0x95 | <b>DMA3CTL1</b>        | DMA3 control register 1                         |                       |
| 0x11 | 0x97 | <b>DMA3SADDR0H</b>     | DMA3 source address register 0 high byte        |                       |
| 0x11 | 0x98 | <b>DMA3SADDR0L</b>     | DMA3 source address register 0 low byte         |                       |
| 0x11 | 0x9b | <b>DMA3SADDR0M</b>     | DMA3 source address register 0 middle byte      |                       |
| 0x11 | 0x9d | <b>DMA3DADDR0H</b>     | DMA3 destination address register 0 high byte   |                       |
| 0x11 | 0x9e | <b>DMA3DADDR0L</b>     | DMA3 destination address register 0 low byte    |                       |
| 0x11 | 0x9f | <b>DMA3DADDR0M</b>     | DMA3 destination address register 0 middle byte |                       |
| 0x11 | 0xa4 | <b>DMA3FrameLenH</b>   | DMA3 frame length register high byte            |                       |
| 0x11 | 0xa5 | <b>DMA3FrameLenL</b>   | DMA3 frame length register low byte             |                       |
| 0x12 | 0x91 | <b>DMA4IP</b>          | DMA4 interrupt pending register                 | DMA4                  |
| 0x12 | 0x92 | <b>DMA4IE</b>          | DMA4 interrupt enable register                  |                       |
| 0x12 | 0x94 | <b>DMA4CTL0</b>        | DMA4 control register 0                         |                       |
| 0x12 | 0x95 | <b>DMA4CTL1</b>        | DMA4 control register 1                         |                       |
| 0x12 | 0x97 | <b>DMA4SADDR0H</b>     | DMA4 source address register 0 high byte        |                       |
| 0x12 | 0x98 | <b>DMA4SADDR0L</b>     | DMA4 source address register 0 low byte         |                       |
| 0x12 | 0x9b | <b>DMA4SADDR0M</b>     | DMA4 source address register 0 middle byte      |                       |
| 0x12 | 0x9d | <b>DMA4DADDR0H</b>     | DMA4 destination address register 0 high byte   |                       |
| 0x12 | 0x9e | <b>DMA4DADDR0L</b>     | DMA4 destination address register 0 low byte    |                       |
| 0x12 | 0x9f | <b>DMA4DADDR0M</b>     | DMA4 destination address register 0 middle byte |                       |
| 0x12 | 0xa4 | <b>DMA4FrameLenH</b>   | DMA4 frame length register high byte            |                       |
| 0x12 | 0xa5 | <b>DMA4FrameLenL</b>   | DMA4 frame length register low byte             |                       |
| 0x04 | 0x90 | <b>AuCodecCtl</b>      | Audio Codec Control Register                    | Audio<br>Codec<br>TOP |
| 0x04 | 0x91 | <b>AuDebugLength</b>   | Audio Codec Debug Length Register               |                       |
| 0x04 | 0x92 | <b>AuCodecDebug</b>    | Audio Codec Debug Register                      |                       |
| 0x04 | 0x93 | <b>AuCodecFIFOCtl</b>  | Audio Codec FIFO control Register               |                       |
| 0x04 | 0x94 | <b>AuCodecFIFOData</b> | Audio Codec FIFO Data Register                  |                       |
| 0x04 | 0x95 | <b>AuCodecDebug2</b>   | Audio Codec Debug Register 2                    |                       |

|      |      |                         |   |                |
|------|------|-------------------------|---|----------------|
| 0x04 | 0x97 | <b>AuCodecDecStateH</b> | Decoding State Register High byte           |                |
| 0x04 | 0x98 | <b>AuCodecDecStateL</b> | Decoding State Register Low byte            |                |
| 0x04 | 0xa9 | <b>MP3IE</b>            | MP3 Decoder Interrupt Enable Register       | MP3<br>Decoder |
| 0x04 | 0xaa | <b>MP3IP</b>            | MP3 Decoder Interrupt Pending Register      |                |
| 0x04 | 0xab | <b>MP3Ctl</b>           | MP3 Decoder Control Register                |                |
| 0x04 | 0xac | <b>MP3HeaderInfo</b>    | Header information register                 |                |
| 0x04 | 0xad | <b>MP3BitRateH</b>      | High byte of bit rate index register        |                |
| 0x04 | 0xae | <b>MP3BitRateL</b>      | Low byte of bit rate index register         |                |
| 0x04 | 0xaf | <b>MP3TtimeH</b>        | Total time hours register                   |                |
| 0x04 | 0xb0 | <b>MP3TtimeM</b>        | Total time minutes register                 |                |
| 0x04 | 0xb1 | <b>MP3TtimeS</b>        | Total time seconds register                 |                |
| 0x04 | 0xb2 | <b>MP3CtimeH</b>        | Current time hours register                 |                |
| 0x04 | 0xb3 | <b>MP3CtimeM</b>        | Current time minutes register               |                |
| 0x04 | 0xb4 | <b>MP3CtimeS</b>        | Current time seconds register               |                |
| 0x04 | 0xb5 | <b>MP3FrameNumH</b>     | High byte of Frame Number register          |                |
| 0x04 | 0xb6 | <b>MP3FrameNumM</b>     | Middle byte of Frame Number register        |                |
| 0x04 | 0xb7 | <b>MP3FrameNumL</b>     | Low byte of Frame Number register           |                |
| 0x04 | 0xb8 | <b>MP3FileLen3</b>      | Byte 3 of File Length register              |                |
| 0x04 | 0xb9 | <b>MP3FileLen2</b>      | Byte 2 of File Length register              |                |
| 0x04 | 0xba | <b>MP3FileLen1</b>      | Byte 1 of File Length register              |                |
| 0x04 | 0xbb | <b>MP3HeaderSynCtl</b>  | MP3 header synchronization control register |                |
| 0x04 | 0xbc | <b>SynHeaderData1</b>   | MP3 header synchronization Data register 1  |                |
| 0x04 | 0xbd | <b>SynHeaderData2</b>   | MP3 header synchronization Data register 2  |                |
| 0x04 | 0xa9 | <b>WMAIE</b>            | WMA Decoder Interrupt Enable Register       | WMA<br>Decoder |
| 0x04 | 0xaa | <b>WMAIP</b>            | WMA Decoder Interrupt Pending Register      |                |
| 0x04 | 0xab | <b>WMACtl</b>           | WMA Decoder Control Register                |                |
| 0x04 | 0xac | <b>WMAHeaderInfo</b>    | Header information register                 |                |
| 0x04 | 0xad | <b>WMABitRateH</b>      | High byte of bit rate index register        |                |
| 0x04 | 0xae | <b>WMABitRateL</b>      | Low byte of bit rate index register         |                |
| 0x04 | 0xaf | <b>WMATTimeH</b>        | Total time hours register                   |                |
| 0x04 | 0xb0 | <b>WMATTimeM</b>        | Total time minutes register                 |                |
| 0x04 | 0xb1 | <b>WMATTimeS</b>        | Total time seconds register                 |                |
| 0x04 | 0xb2 | <b>WMACTimeH</b>        | Current time hours register                 |                |
| 0x04 | 0xb3 | <b>WMACTimeM</b>        | Current time minutes register               |                |
| 0x04 | 0xb4 | <b>WMACTimeS</b>        | Current time seconds register               |                |
| 0x04 | 0xb5 | <b>WMAPackNumH</b>      | High byte of Packet Number register         |                |
| 0x04 | 0xb6 | <b>WMAPackNumM</b>      | Middle byte of Packet Number register       |                |
| 0x04 | 0xb7 | <b>WMAPackNumL</b>      | Low byte of Packet Number register          |                |
| 0x04 | 0xb8 | <b>WMAPackSizeH</b>     | High byte of Packet Size register           |                |
| 0x04 | 0xb9 | <b>WMAPackSizeM</b>     | Middle byte of Packet Size register         |                |
| 0x04 | 0xba | <b>WMAPackSizeL</b>     | Low byte of Packet Size register            |                |

|      |      |                       |  |                  |
|------|------|-----------------------|--|------------------|
| 0x04 | 0xbb | <b>VirtualAddrH</b>   | High byte of virtual address of current frame  |                  |
| 0x04 | 0xbc | <b>VirtualAddrL</b>   | Low byte of virtual address of current frame   |                  |
| 0x04 | 0xbd | <b>PhysicalAddrH</b>  | High byte of physical address of current frame |                  |
| 0x04 | 0xbf | <b>PhysicalAddrL</b>  | Low byte of physical address of current frame  |                  |
| 0x04 | 0xc1 | <b>DiscardLen</b>     | Discard length of current subframe             |                  |
| 0x04 | 0xa9 | <b>WAVIE</b>          | WAV Decoder Interrupt Enable Register          | WAV Decoder      |
| 0x04 | 0xaa | <b>WAVIP</b>          | WAV Decoder Interrupt Pending Register         |                  |
| 0x04 | 0xab | <b>WAVCtl</b>         | WAV Decoder Control Register                   |                  |
| 0x04 | 0xac | <b>WAVHeaderInfo1</b> | Header information register 1                  |                  |
| 0x04 | 0xad | <b>WAVHeaderInfo2</b> | Header information register 2                  |                  |
| 0x04 | 0xaf | <b>WAVTTimeH</b>      | Total time hours register                      |                  |
| 0x04 | 0xb0 | <b>WAVTTimeM</b>      | Total time minutes register                    |                  |
| 0x04 | 0xb1 | <b>WAVTTimeS</b>      | Total time seconds register                    |                  |
| 0x04 | 0xb2 | <b>WAVCTimeH</b>      | Current time hours register                    |                  |
| 0x04 | 0xb3 | <b>WAVCTimeM</b>      | Current time minutes register                  |                  |
| 0x04 | 0xb4 | <b>WAVCTimeS</b>      | Current time seconds register                  |                  |
| 0x04 | 0xb5 | <b>WAVBlockNumH</b>   | High byte of Block Number register             |                  |
| 0x04 | 0xb6 | <b>WAVBlockNumM</b>   | Middle byte of Block Number register           |                  |
| 0x04 | 0xb7 | <b>WAVBlockNumL</b>   | Low byte of Block Number register              |                  |
| 0x04 | 0xab | <b>WAVEncCtl</b>      | WAV Encoder Control Register                   | WAV Encoder      |
| 0x04 | 0xac | <b>WAVEncInfo</b>     | WAV Encoder Information Register               |                  |
| 0x04 | 0xab | <b>MP3EncCtl</b>      | MP3 Encoder Control Register                   | MP3 Encoder      |
| 0x04 | 0xac | <b>MP3EncInfo</b>     | MP3 Encoder Information Register               |                  |
| 0x04 | 0xa9 | <b>SoftIE</b>         | Software Decoder Interrupt Enable Register     | Software Decoder |
| 0x04 | 0xaa | <b>SoftIP</b>         | Software Decoder Interrupt Pending Register    |                  |
| 0x04 | 0xab | <b>SoftCtl</b>        | Software Decoder Control Register              |                  |
| 0x04 | 0xb8 | <b>FrameLen1</b>      | Frame Length Register 1                        |                  |
| 0x04 | 0xb9 | <b>FrameLen0</b>      | Frame Length Register 0                        |                  |
| 0x04 | 0xc2 | <b>BEPCtl1</b>        | Post Processor Control Register 1              | Post Processor   |
| 0x04 | 0xc3 | <b>BEPCtl2</b>        | Post Processor Control Register 2              |                  |
| 0x04 | 0xc4 | <b>BEPCtl3</b>        | Post Processor Control Register 3              |                  |
| 0x04 | 0xc5 | <b>BEPCtl4</b>        | Post Processor Control Register 4              |                  |
| 0x04 | 0xc6 | <b>BEPCtl5</b>        | Post Processor Control Register 5              |                  |
| 0x04 | 0xc7 | <b>GlobalGainH</b>    | Global Gain High Byte Register                 | Karaoke          |
| 0x04 | 0xc8 | <b>GlobalGainM</b>    | Global Gain Middle Byte Register               |                  |
| 0x04 | 0xc9 | <b>GlobalGainL</b>    | Global Gain Low Byte Register                  |                  |
| 0x04 | 0xca | <b>CurrentEnergy</b>  | Current Energy Register                        |                  |
| 0x04 | 0xcb | <b>KaraokeCtl</b>     | Karaoke Control Register                       | Karaoke          |
| 0x04 | 0xcc | <b>LCHGain</b>        | Left channel gain                              |                  |
| 0x04 | 0xcd | <b>RCHGain</b>        | Right channel gain                             |                  |
| 0x04 | 0xce | <b>LPFGain</b>        | Low pass filter gain                           |                  |

|         |      |                          |  |            |
|---------|------|--------------------------|--|------------|
| 0x04    | 0xcf | <b>HPFGain</b>           | High pass filter gain                                  |            |
| 0x04    | 0xcc | <b>SRSCtl0</b>           | SRS Control Register 0                                 | SRS<br>WOW |
| 0x04    | 0xcd | <b>SRSCtl1</b>           | SRS Control Register 1                                 |            |
| 0x04    | 0xce | <b>SRSCtl2</b>           | SRS Control Register 2                                 |            |
| 0x04/13 | 0xd5 | <b>DAC_CTL</b>           | DAC Control Register                                   | DAC_PA     |
| 0x04/13 | 0xd6 | <b>DAC_VOLUME0</b>       | DAC Volume Control register 0                          |            |
| 0x04/13 | 0xd7 | <b>DAC_VOLUME1</b>       | DAC Volume Control register 1                          |            |
| 0x04/13 | 0xd8 | <b>DAC_CH0_FIFO_CTL0</b> | DAC Channel 0 FIFO Control Register 0                  |            |
| 0x04/13 | 0xd9 | <b>DAC_CH0_FIFO_CTL1</b> | DAC Channel 0 FIFO Control Register 1                  |            |
| 0x04/13 | 0xda | <b>DAC_CH0_PCML</b>      | DAC Channel 0 PCM DATA Low byte                        |            |
| 0x04/13 | 0xdb | <b>DAC_CH0_PCMM</b>      | DAC Channel 0 PCM DATA Middle byte                     |            |
| 0x04/13 | 0xdc | <b>DAC_CH0_PCMH</b>      | DAC Channel 0 PCM DATA High byte                       |            |
| 0x04/13 | 0xdd | <b>DAC_CH1_FIFO_CTL0</b> | DAC Channel 1 FIFO Control Register 0                  |            |
| 0x04/13 | 0xde | <b>DAC_CH1_FIFO_CTL1</b> | DAC Channel 1 FIFO Control Register 1                  |            |
| 0x04/13 | 0xdf | <b>DAC_CH1_PCML</b>      | DAC Channel 1 PCM DATA Low byte                        |            |
| 0x04/13 | 0xe1 | <b>DAC_CH1_PCMH</b>      | DAC Channel 1 PCM DATA High byte                       |            |
| 0x04/13 | 0xe2 | <b>I2S_CTL0</b>          | I2S Control Register 0                                 |            |
| 0x04/13 | 0xe3 | <b>I2S_CTL1</b>          | I2S Control Register 1                                 |            |
| 0x04/13 | 0xe4 | <b>DAC_ANALOG0</b>       | DAC Analog Register 0                                  |            |
| 0x04/13 | 0xe5 | <b>DAC_ANALOG1</b>       | DAC Analog Register 1                                  |            |
| 0x04/13 | 0xe6 | <b>DAC_TUNE0</b>         | DAC tuning Control Register 0                          |            |
| 0x04/13 | 0xe7 | <b>DAC_TUNE1</b>         | DAC tuning Control Register 1                          |            |
| 0x04/13 | 0xe9 | <b>PA_VOLUME</b>         | PA VOLUME Control Register                             | DAC_PA     |
| 0x04/13 | 0xea | <b>PA_CTL</b>            | PA Control Register                                    |            |
| 0x04/13 | 0xeb | <b>PA_APCTL</b>          | PA anti-pop Control Register                           |            |
| 0x04/13 | 0xec | <b>DDV_CTL0</b>          | Direct drive Control Register 0                        |            |
| 0x13    | 0x90 | <b>DAC_CH0_SR_CTL</b>    | DAC Channel 0 Sample Rate Control Register             |            |
| 0x13    | 0x91 | <b>DAC_CH0_SR_GAIN</b>   | DAC Channel 0 Sample Rate Gain Control Register        |            |
| 0x13    | 0x92 | <b>DAC_CH0_SRFT_CTL0</b> | DAC Channel 0 Sample Rate Fine Tune Control Register 0 |            |
| 0x13    | 0x93 | <b>DAC_CH0_SRFT_CTL1</b> | DAC Channel 0 Sample Rate Fine Tune Control Register 1 |            |
| 0x13    | 0x94 | <b>DAC_CH0_SRFT_CTL2</b> | DAC Channel 0 Sample Rate Fine Tune Control Register 2 |            |
| 0x13    | 0x95 | <b>DAC_CH0_SRFT_CTL3</b> | DAC Channel 0 Sample Rate Fine Tune Control Register 3 |            |
| 0x13    | 0x97 | <b>DAC_CH0_SRFT_CTL4</b> | DAC Channel 0 Sample Rate Fine Tune Control Register 4 |            |
| 0x13    | 0x98 | <b>DAC_CH1_SR_CTL</b>    | DAC Channel 1 Sample Rate Control Register             |            |
| 0x13    | 0x99 | <b>DAC_CH1_SR_GAIN</b>   | DAC Channel 1 Sample Rate Gain Control Register        |            |
| 0x13    | 0xa  | <b>DAC_CH1_SRFT_CTL0</b> | DAC Channel 1 Sample Rate Fine Tune Control Register   |            |

|         |      |                          |  |       |
|---------|------|--------------------------|--|-------|
|         |      |                          | Register 0   |       |
| 0x13    | 0x9b | <b>DAC_CH1_SRFT_CTL1</b> | DAC Channel 1 Sample Rate Fine Tune Control Register 1 |       |
| 0x13    | 0x9c | <b>DAC_CH1_SRFT_CTL2</b> | DAC Channel 1 Sample Rate Fine Tune Control Register 2 |       |
| 0x13    | 0x9d | <b>DAC_CH1_SRFT_CTL3</b> | DAC Channel 1 Sample Rate Fine Tune Control Register 3 |       |
| 0x13    | 0x9e | <b>DAC_CH1_SRFT_CTL4</b> | DAC Channel 1 Sample Rate Fine Tune Control Register 4 |       |
| 0x04/13 | 0xf1 | <b>ADC_CTL0</b>          | ADC Control Register 0                                 | ADC   |
| 0x04/13 | 0xf2 | <b>AINOP_CTL</b>         | AnalogIN OP Control Register                           |       |
| 0x04/13 | 0xf3 | <b>ADC_GAIN0</b>         | ADC gain Control Register0                             |       |
| 0x04/13 | 0xf4 | <b>ADC_GAIN1</b>         | ADC gain Control Register1                             |       |
| 0x04/13 | 0xf5 | <b>ADC_TUNE0</b>         | ADC tuning control Register 0                          |       |
| 0x04/13 | 0xf6 | <b>ADC_TUNE1</b>         | ADC tuning control Register 1                          |       |
| 0x04/13 | 0xf7 | <b>ADC_FIFO_DAT</b>      | ADC FIFO data register                                 |       |
| 0x04/13 | 0xf8 | <b>ADC_FIFOCTL0</b>      | ADC FIFO control register 0                            |       |
| 0x04/13 | 0xfc | <b>ADC_FIFOCTL1</b>      | ADC FIFO control register 1                            |       |
| 0x04/13 | 0xfd | <b>ADC_CTL1</b>          | ADC Control Register 1                                 |       |
| 0x13    | 0xa2 | <b>ADC_SRFT_CTL0</b>     | ADC Sample Rate Fine Tune Control Register 0           |       |
| 0x13    | 0xa3 | <b>ADC_SRFT_CTL1</b>     | ADC Sample Rate Fine Tune Control Register 1           |       |
| 0x13    | 0xa4 | <b>ADC_SRFT_CTL2</b>     | ADC Sample Rate Fine Tune Control Register 2           |       |
| 0x13    | 0xa5 | <b>ADC_SRFT_CTL3</b>     | ADC Sample Rate Fine Tune Control Register 3           |       |
| 0x13    | 0xa6 | <b>ADC_SRFT_CTL4</b>     | ADC Sample Rate Fine Tune Control Register 4           |       |
| 0x13    | 0xb0 | <b>SPDIFRX_CTL</b>       | SPDIFRX Control Register                               | SPDIF |
| 0x13    | 0xb1 | <b>SPDIFRX_STAT</b>      | SPDIFRX Statue Register                                |       |
| 0x13    | 0xb2 | <b>SPDIFRX_CSTAT</b>     | SPDIFRX Channel Statue Register                        |       |
| 0x13    | 0xb3 | <b>SPDIFRX_DEBUG</b>     | SPDIFRX Debug Register                                 |       |
| 0x13    | 0xb4 | <b>SPDIFRX_DAT</b>       | SPDIFRX Data Register                                  |       |
| 0x13    | 0xb5 | <b>SPDIFRX_CNT_WR0</b>   | SPDIFRX Counter for Write Register 0                   |       |
| 0x13    | 0xb6 | <b>SPDIFRX_CNT_WR1</b>   | SPDIFRX Counter for Write Register 1                   |       |
| 0x13    | 0xb7 | <b>SPDIFRX_CNT_WR2</b>   | SPDIFRX Counter for Write Register 2                   |       |
| 0x05    | 0x89 | <b>VOUT_CTL</b>          | VCC/VDD voltage set Register                           | PMU   |
| 0x05    | 0x8a | <b>LDOPD_CTL</b>         | Capless LDO pulldown control                           |       |
| 0x05    | 0x8b | <b>BDG_CTL</b>           | Bandgap enable Register                                |       |
| 0x05    | 0x8c | <b>BDG_VOL</b>           | Bandgap voltage Register                               |       |
| 0x05    | 0x8d | <b>MULTI_USED</b>        | GPIO multi-used set Register                           |       |
| 0x05    | 0x90 | <b>PMUADC_CTL</b>        | PMU ADC frequency and enable Register                  |       |
| 0x05    | 0x91 | <b>BATADC_DATA</b>       | BATADC data Register                                   |       |
| 0x05    | 0x92 | <b>LRADC1_DATA</b>       | LRADC1 data Register                                   |       |
| 0x05    | 0x93 | <b>LRADC3_DATA</b>       | LRADC3 data Register                                   |       |

|      |      |                        |  |                              |
|------|------|------------------------|--|------------------------------|
| 0x05 | 0x94 | <b>LRADC4_DATA</b>     | LRADC4 data Register                               |                              |
| 0x05 | 0x95 | <b>LRADC5_DATA</b>     | LRADC5 data Register                               |                              |
| 0x05 | 0x97 | <b>CP_CTL0</b>         | Charge pump control Register                       |                              |
| 0x05 | 0x98 | <b>CP_CTL1</b>         | Charge pump control Register                       |                              |
| 0x05 | 0x99 | <b>VDD_reserved</b>    | reserved   |                              |
| 0x05 | 0x9a | <b>TEST_CTL</b>        | Standby test control Register                      |                              |
| 0x05 | 0x9b | <b>SYSTEM_CTL</b>      | System on/off time set Register                    |                              |
| 0x05 | 0x9c | <b>SYSTEM_ONOFF</b>    | on/off statue & RESET time set Register            |                              |
| 0x05 | 0x9d | <b>RTCVDD_reserved</b> | reserved   |                              |
| 0x05 | 0xa2 | <b>FS_CTL</b>          | Fsource control and EFUSE select Register          |                              |
| 0x05 | 0xa3 | <b>EFUSE_CTL</b>       | EFSUE control Register                             |                              |
| 0x05 | 0xa4 | <b>EFUSE0</b>          | EFUSE0 data Register                               |                              |
| 0x05 | 0xa5 | <b>EFUSE1</b>          | EFUSE1 data Register                               |                              |
| 0x05 | 0xa6 | <b>EFUSE2</b>          | EFUSE2 data Register                               |                              |
| 0x05 | 0xa7 | <b>EFUSE3</b>          | EFUSE3 data Register                               |                              |
| 0x06 | 0x9e | <b>EXTMEM_CTL</b>      | Extended Memory Interface Control Register         | External<br>memory<br>access |
| 0x06 | 0x9f | <b>EXTMEM_WT</b>       | Extended Memory Interface Wait State Register      |                              |
| 0x06 | 0xff | <b>EXTMEM_DL</b>       | Extended Memory Interface Low Byte Register        |                              |
| 0x06 | 0xfe | <b>EXTMEM_DH</b>       | Extended Memory Interface High Byte Register       |                              |
| 0x06 | 0x8a | <b>LCD_MODE</b>        | LCD Mode Control Register                          | LCD                          |
| 0x06 | 0x8b | <b>LCD_DATA0</b>       | COM[3:0] of SEG0 and SEG1; SEG[7:0] of COM0        |                              |
| 0x06 | 0x8c | <b>LCD_DATA1</b>       | COM[3:0] of SEG2 and SEG3; SEG[7:0] of COM1        |                              |
| 0x06 | 0x8d | <b>LCD_DATA2</b>       | COM[3:0] of SEG4 and SEG5; SEG[7:0] of COM2        |                              |
| 0x06 | 0x90 | <b>LCD_DATA3</b>       | COM[3:0] of SEG6 and SEG7; SEG[7:0] of COM3        |                              |
| 0x06 | 0x91 | <b>LCD_DATA4</b>       | COM[3:0] of SEG8 and SEG9; SEG[7:0] of COM4        |                              |
| 0x06 | 0x92 | <b>LCD_DATA5</b>       | COM[3:0] of SEG10 and SEG11; SEG[7:0] of COM5      |                              |
| 0x06 | 0x93 | <b>LCD_DATA6</b>       | COM[3:0] of SEG12 and SEG13; SEG[7:0] of COM6      |                              |
| 0x06 | 0x94 | <b>LCD_DATA7</b>       | COM[3:0] of SEG14 and SEG15; SEG[7:0] of COM7      |                              |
| 0x06 | 0x95 | <b>LCD_DATA8</b>       | COM[3:0] of SEG16 and SEG17                        |                              |
| 0x06 | 0x97 | <b>LCD_DATA9</b>       | COM[3:0] of SEG18 and SEG19                        |                              |
| 0x06 | 0x98 | <b>LCD_DATA10</b>      | COM[3:0] of SEG20 and SEG21                        |                              |
| 0x06 | 0x99 | <b>LCD_DATA11</b>      | COM[3:0] of SEG22 and SEG23                        |                              |
| 0x06 | 0x9a | <b>LCD_DATA12</b>      | COM[3:0] of SEG24 and SEG25                        |                              |
| 0x06 | 0x9b | <b>LCD_DATA13</b>      | COM[3:0] of SEG26 and SEG27                        |                              |
| 0x06 | 0x9c | <b>LCD_DATA14</b>      | COM[3:0] of SEG28 and SEG29                        |                              |
| 0x06 | 0x9d | <b>LCD_DATA15</b>      | COM[3:0] of SEG30 and SEG31                        |                              |
| 0x06 | 0xa2 | <b>GPIOAOUTEN</b>      | General Purpose Input Output Group A Output Enable | GPIO                         |

|      |      |                   |  |  |
|------|------|-------------------|--|--|
| 0x06 | 0xa3 | <b>GPIOAINEN</b>  | General Purpose Input Output Group A Input Enable  |  |
| 0x06 | 0xa4 | <b>GPIOADAT</b>   | General Purpose Input Output Group A Data          |  |
| 0x06 | 0xa5 | <b>GPIOAPUEN</b>  | General Purpose Input Output Group A PU Enable     |  |
| 0x06 | 0xa6 | <b>GPIOAPDEN</b>  | General Purpose Input Output Group A PD Enable     |  |
| 0x06 | 0xa7 | <b>GPIOBOUTEN</b> | General Purpose Input Output Group B Output Enable |  |
| 0x06 | 0xa9 | <b>GPIOBINEN</b>  | General Purpose Input Output Group B Input Enable  |  |
| 0x06 | 0xaa | <b>GPIOBDAT</b>   | General Purpose Input Output Group B Data          |  |
| 0x06 | 0xab | <b>GPIOBPUEN</b>  | General Purpose Input Output Group B PU Enable     |  |
| 0x06 | 0xac | <b>GPIOBPDEN</b>  | General Purpose Input Output Group B PD Enable     |  |
| 0x06 | 0xad | <b>GPIOCOUTEN</b> | General Purpose Input Output Group C Output Enable |  |
| 0x06 | 0xae | <b>GPIOCINEN</b>  | General Purpose Input Output Group C Input Enable  |  |
| 0x06 | 0xaf | <b>GPIOCDAT</b>   | General Purpose Input Output Group C Data          |  |
| 0x06 | 0xb0 | <b>GPIOCPUEN</b>  | General Purpose Input Output Group C PU Enable     |  |
| 0x06 | 0xb1 | <b>GPIOCPDEN</b>  | General Purpose Input Output Group C PD Enable     |  |
| 0x06 | 0xb2 | <b>GPIODOUTEN</b> | General Purpose Input Output Group D Output Enable |  |
| 0x06 | 0xb3 | <b>GPIODINEN</b>  | General Purpose Input Output Group D Input Enable  |  |
| 0x06 | 0xb4 | <b>GPIODDAT</b>   | General Purpose Input Output Group D Data          |  |
| 0x06 | 0xb5 | <b>GPIODPUEN</b>  | General Purpose Input Output Group D PU Enable     |  |
| 0x06 | 0xb6 | <b>GPIODPDEN</b>  | General Purpose Input Output Group D PD Enable     |  |
| 0x06 | 0xb7 | <b>GPIOEOUTEN</b> | General Purpose Input Output Group E Output Enable |  |
| 0x06 | 0xb8 | <b>GPIOEINEN</b>  | General Purpose Input Output Group E Input Enable  |  |
| 0x06 | 0xb9 | <b>GPIOEDAT</b>   | General Purpose Input Output Group E Data          |  |
| 0x06 | 0xba | <b>GPIOEPUEN</b>  | General Purpose Input Output Group E PU Enable     |  |
| 0x06 | 0xbb | <b>GPIOEPDEN</b>  | General Purpose Input Output Group E PD Enable     |  |
| 0x06 | 0xbc | <b>GPIOFOUTEN</b> | General Purpose Input Output Group F Output Enable |  |
| 0x06 | 0xbd | <b>GPIOFINEN</b>  | General Purpose Input Output Group F Input Enable  |  |
| 0x06 | 0xbf | <b>GPIOFDAT</b>   | General Purpose Input Output Group F Data          |  |
| 0x06 | 0xc1 | <b>GPIOFPUEN</b>  | General Purpose Input Output Group F PU Enable     |  |
| 0x06 | 0xc2 | <b>GPIOFPDEN</b>  | General Purpose Input Output Group F PD Enable     |  |
| 0x06 | 0xc3 | <b>GPIOGOUTEN</b> | General Purpose Input Output Group G Output Enable |  |
| 0x06 | 0xc4 | <b>GPIOGINEN</b>  | General Purpose Input Output Group G Input Enable  |  |
| 0x06 | 0xc5 | <b>GPIOGDAT</b>   | General Purpose Input Output Group G Data          |  |
| 0x06 | 0xc6 | <b>GPIOGPUEN</b>  | General Purpose Input Output Group G PU Enable     |  |
| 0x06 | 0xc7 | <b>GPIOGPDEN</b>  | General Purpose Input Output Group G PD Enable     |  |
| 0x06 | 0xc8 | <b>MFP_CTL0</b>   | Multi-Function PAD Control Register 0              |  |

|      |      |                         |   |     |
|------|------|-------------------------|---|-----|
| 0x06 | 0xc9 | <b>MFP_CTL1</b>         | Multi-Function PAD Control Register 1       |     |
| 0x06 | 0xca | <b>MFP_CTL2</b>         | Multi-Function PAD Control Register 2       |     |
| 0x06 | 0xcb | <b>MFP_CTL3</b>         | Multi-Function PAD Control Register3        |     |
| 0x06 | 0xcc | <b>MFP_CTL4</b>         | Multi-Function PAD Control Register4        |     |
| 0x06 | 0xcd | <b>MFP_CTL5</b>         | Multi-Function PAD Control Register5        |     |
| 0x06 | 0xce | <b>MFP_CTL6</b>         | Multi-Function PAD Control Register6        |     |
| 0x06 | 0xcf | <b>MFP_CTL7</b>         | Multi-Function PAD Control Register7        |     |
| 0x06 | 0xd2 | <b>MFP_CTL8</b>         | Multi-Function PAD Control Register8        |     |
| 0x06 | 0xd3 | <b>AD_Select0</b>       | ANALOG/DIGITAL Select0                      |     |
| 0x06 | 0xdc | <b>AD_Select1</b>       | ANALOG/DIGITAL Select1                      |     |
| 0x06 | 0xd4 | <b>PADPUPD</b>          | PADPUPD                                     |     |
| 0x06 | 0xd5 | <b>PADDRV0</b>          | Pad Drive CTL0                              |     |
| 0x06 | 0xd6 | <b>PADDRV1</b>          | Pad Drive CTL1                              |     |
| 0x06 | 0xd7 | <b>PADDRV2</b>          | Pad Drive CTL2                              |     |
| 0x06 | 0xd9 | <b>PADDRV3</b>          | Pad Drive CTL3                              |     |
| 0x06 | 0xda | <b>PADDRV4</b>          | Pad Drive CTL4                              |     |
| 0x06 | 0xdb | <b>DBGSEL</b>           | Debug Select Register                       |     |
| 0x06 | 0xe2 | <b>DBGAOE</b>           | DEBUGAOUTEN                                 |     |
| 0x06 | 0xe3 | <b>DBGAIE</b>           | DEBUGAINEN                                  |     |
| 0x06 | 0xe4 | <b>DBGBOE</b>           | DEBUGBOUTEN                                 |     |
| 0x06 | 0xe5 | <b>DBGBIE</b>           | DEBUGBINEN                                  |     |
| 0x06 | 0xe6 | <b>DBGCOE</b>           | DEBUGCOUTEN                                 |     |
| 0x06 | 0xe9 | <b>DBGDOE</b>           | DEBUGDOUTEN                                 |     |
| 0x06 | 0xeb | <b>LED_SEG_RC_EN</b>    | LED SEG Restrict Current                    |     |
| 0x06 | 0xec | <b>LED_SEG_BIAS_EN</b>  | LED SEG Restrict Current                    |     |
| 0x07 | 0x86 | <b>UDMAM</b>            | USB DMA MODE                                | USB |
| 0x07 | 0x89 | <b>AUTOINTIMER</b>      | Auto in mode in token timer                 |     |
| 0x07 | 0x8a | <b>EP1STADDRH</b>       | EP1 FIFO start address high register        |     |
| 0x07 | 0x8b | <b>EP1STADDRL</b>       | EP1 FIFO start address low register         |     |
| 0x07 | 0x8c | <b>EP1DMALENH</b>       | EP1 DMA transfer length high in normal mode |     |
| 0x07 | 0x8d | <b>EP1DMALENL</b>       | EP1 DMA transfer length low in normal mode  |     |
| 0x07 | 0x90 | <b>OUTPCKCNTH</b>       | Out transaction packet counter high         |     |
| 0x07 | 0x91 | <b>OUTPCKCNTL</b>       | Out transaction packet counter low          |     |
| 0x07 | 0x92 | <b>IDVBUSCTRL</b>       | ID&VBUS control                             |     |
| 0x07 | 0x93 | <b>USBSTATUS</b>        | USB status                                  |     |
| 0x07 | 0x94 | <b>DPDMCTRL</b>         | DP DM control register                      |     |
| 0x07 | 0x95 | <b>USB_PHYCTRL</b>      | PHY control register                        |     |
| 0x07 | 0x97 | <b>Out0bc_hcin0bc</b>   | Endpoint 0 OUT Byte Count                   |     |
| 0x07 | 0x98 | <b>In0bc_hcout0bc</b>   | Endpoint 0 IN Byte Count                    |     |
| 0x07 | 0x99 | <b>Ep0cs_hcep0cs</b>    | Endpoint 0 Control and Status               |     |
| 0x07 | 0x9a | <b>In1bch_hcout1bch</b> | Endpoint 1 IN Byte Count High               |     |

|      |      |   |  |  |
|------|------|---|--|--|
| 0x07 | 0x9b | <a href="#"><u>In1bcl_hcout1bcl</u></a>   | Endpoint 1 IN Byte Count Low                   |  |
| 0x07 | 0x9c | <a href="#"><u>In1ctrl_hcout1ctrl</u></a> | Endpoint 1 IN Control                          |  |
| 0x07 | 0x9d | <a href="#"><u>In1cs_hcout1cs</u></a>     | Endpoint 1 IN Control And Status               |  |
| 0x07 | 0x9e | <a href="#"><u>Out2bch_hcin2bch</u></a>   | Endpoint 2 OUT Byte Count High                 |  |
| 0x07 | 0x9f | <a href="#"><u>Out2bcl_hcin2bcl</u></a>   | Endpoint 2 OUT Byte Count Low                  |  |
| 0x07 | 0xa2 | <a href="#"><u>Out2ctrl_hcin2ctrl</u></a> | Endpoint 2 OUT Control                         |  |
| 0x07 | 0xa3 | <a href="#"><u>Out2cs_hcin2cs</u></a>     | Endpoint 2 OUT Control And Status              |  |
| 0x07 | 0xa4 | <a href="#"><u>In3bc_hcout3bc</u></a>     | Endpoint 3 IN Byte Count                       |  |
| 0x07 | 0xa5 | <a href="#"><u>In3ctrl_hcout3ctrl</u></a> | Endpoint 3 IN Control                          |  |
| 0x07 | 0xa6 | <a href="#"><u>In3cs_hcout3cs</u></a>     | Endpoint 3 IN Control And Status               |  |
| 0x07 | 0xa7 | <a href="#"><u>Fifo1dat</u></a>           | FIFO 1 Data                                    |  |
| 0x07 | 0xa9 | <a href="#"><u>Fifo2dat</u></a>           | FIFO 2 Data                                    |  |
| 0x07 | 0xaa | <a href="#"><u>Fifo3dat</u></a>           | FIFO 3 Data                                    |  |
| 0x07 | 0xab | <a href="#"><u>Ep0indata</u></a>          | EP0 IN DATA                                    |  |
| 0x07 | 0xac | <a href="#"><u>Ep0outdata</u></a>         | EP0 OUT DATA                                   |  |
| 0x07 | 0xad | <a href="#"><u>Usbirq_heusbirq</u></a>    | USB Interrupt                                  |  |
| 0x07 | 0xae | <a href="#"><u>Usbien_heusbien</u></a>    | USB interrupt enable                           |  |
| 0x07 | 0xaf | <a href="#"><u>SHORTPCKIRQ</u></a>        | Short packets Interrupt request and enable     |  |
| 0x07 | 0xb0 | <a href="#"><u>Hcep0ctrl</u></a>          | Endpoint 0 Control                             |  |
| 0x07 | 0xb1 | <a href="#"><u>Hcout0err</u></a>          | Endpoint 0 HC OUT Error                        |  |
| 0x07 | 0xb2 | <a href="#"><u>Hcin0err</u></a>           | Endpoint 0 HC IN Error                         |  |
| 0x07 | 0xb3 | <a href="#"><u>Hcout1ctrl</u></a>         | Endpoint 1 HC OUT Control                      |  |
| 0x07 | 0xb4 | <a href="#"><u>Hcout1err</u></a>          | Endpoint 1 HC OUT Error                        |  |
| 0x07 | 0xb5 | <a href="#"><u>Hcin2ctrl</u></a>          | Endpoint 2 HC IN Control                       |  |
| 0x07 | 0xb6 | <a href="#"><u>Hcin2err</u></a>           | Endpoint 2 HC OUT Error                        |  |
| 0x07 | 0xb7 | <a href="#"><u>EP2STADDRH</u></a>         | EP2 FIFO start address high register           |  |
| 0x07 | 0xb8 | <a href="#"><u>EP2STADDRL</u></a>         | EP2 FIFO start address low register            |  |
| 0x07 | 0xb9 | <a href="#"><u>Hcout3ctrl</u></a>         | Endpoint 3 HC OUT Control                      |  |
| 0x07 | 0xba | <a href="#"><u>Hcout3err</u></a>          | Endpoint 3 HC OUT Error                        |  |
| 0x07 | 0xbb | <a href="#"><u>Setupdat0</u></a>          | SETUP DATA0                                    |  |
| 0x07 | 0xbc | <a href="#"><u>Setupdat1</u></a>          | SETUP DATA1                                    |  |
| 0x07 | 0xbd | <a href="#"><u>Setupdat2</u></a>          | SETUP DATA2                                    |  |
| 0x07 | 0xbf | <a href="#"><u>Setupdat3</u></a>          | SETUP DATA3                                    |  |
| 0x07 | 0xc1 | <a href="#"><u>Setupdat4</u></a>          | SETUP DATA4                                    |  |
| 0x07 | 0xc2 | <a href="#"><u>Setupdat5</u></a>          | SETUP DATA5                                    |  |
| 0x07 | 0xc3 | <a href="#"><u>Setupdat6</u></a>          | SETUP DATA6                                    |  |
| 0x07 | 0xc4 | <a href="#"><u>Setupdat7</u></a>          | SETUP DATA7                                    |  |
| 0x07 | 0xc5 | <a href="#"><u>Ep03irq</u></a>            | Endpoint 0 to 3 Interrupt Request              |  |
| 0x07 | 0xc6 | <a href="#"><u>Ep03ien</u></a>            | Endpoint 0 to 3 Interrupt Enables              |  |
| 0x07 | 0xc7 | <a href="#"><u>Ep03tokirq</u></a>         | Endpoint 0 to 3 Token Interrupt Request        |  |
| 0x07 | 0xc8 | <a href="#"><u>Ep03tokien</u></a>         | Endpoint 0 to 3 Token Interrupt Request Enable |  |

|      |      |                               |   |  |
|------|------|-------------------------------|---|--|
| 0x07 | 0xc9 | <a href="#">IVECT</a>         | Interrupt Vector                            |  |
| 0x07 | 0xca | <a href="#">EPRST</a>         | Endpoint Reset                              |  |
| 0x07 | 0xcb | <a href="#">UsbCTRL_STUS</a>  | USB Control And Status                      |  |
| 0x07 | 0xcc | <a href="#">FrmCNTH</a>       | USB Frame Counter HIGH                      |  |
| 0x07 | 0xcd | <a href="#">FrmCNTL</a>       | USB Frame Counter Low                       |  |
| 0x07 | 0xce | <a href="#">Fnaddr</a>        | Function Address                            |  |
| 0x07 | 0xcf | <a href="#">Clkgate</a>       | Clock Gate                                  |  |
| 0x07 | 0xd2 | <a href="#">Fifoctrl</a>      | FIFO Control                                |  |
| 0x07 | 0xd3 | <a href="#">Hcportctrl</a>    | HC Port Control                             |  |
| 0x07 | 0xd4 | <a href="#">Hcfrmnh</a>       | HC Frame Number high                        |  |
| 0x07 | 0xd5 | <a href="#">Hcfrmnl</a>       | HC Frame Number low                         |  |
| 0x07 | 0xd6 | <a href="#">Hcfrmremainh</a>  | HC Frame Remain high                        |  |
| 0x07 | 0xd7 | <a href="#">Hcfrmremainl</a>  | HC Frame Remain Low                         |  |
| 0x07 | 0xd8 | <a href="#">Hcep03errirq</a>  | HC 0 to 3 Error Interrupt Request           |  |
| 0x07 | 0xd9 | <a href="#">Hcep03errien</a>  | HC 0 to 3 Error Interrupt Enable            |  |
| 0x07 | 0xda | <a href="#">Otgirq</a>        | OTG Interrupt Request                       |  |
| 0x07 | 0xdb | <a href="#">Otgstate</a>      | The OTG FSM State                           |  |
| 0x07 | 0xdc | <a href="#">Otctrl</a>        | OTG Control                                 |  |
| 0x07 | 0xdd | <a href="#">Otgstatus</a>     | OTG Status                                  |  |
| 0x07 | 0xde | <a href="#">Otgien</a>        | OTG Interrupt Enable                        |  |
| 0x07 | 0xe2 | <a href="#">EP2DMALENH</a>    | EP2 DMA transfer length high in normal mode |  |
| 0x07 | 0xe3 | <a href="#">EP2DMALENL</a>    | EP2 DMA transfer length low in normal mode  |  |
| 0x07 | 0xe4 | <a href="#">Hcin0maxpck</a>   | HC IN 0 Max Packet Size                     |  |
| 0x07 | 0xe5 | <a href="#">Hcin2maxpckh</a>  | HC IN 2 max packet high                     |  |
| 0x07 | 0xe6 | <a href="#">Hcin2maxpckl</a>  | HC IN 2 max packet low                      |  |
| 0x07 | 0xe7 | <a href="#">Hcout3maxpck</a>  | HC OUT 3 max packet                         |  |
| 0x07 | 0xe9 | <a href="#">Hcout1maxpckh</a> | HC OUT 1 max packet low                     |  |
| 0x07 | 0xea | <a href="#">Hcout1maxpckl</a> | HC OUT 1 max packet high                    |  |
| 0x07 | 0xeb | <a href="#">USBEIRQ</a>       | USB external Interrupt request              |  |
| 0x07 | 0xec | <a href="#">AUTONAKCTRL</a>   | auto nak control                            |  |
| 0x07 | 0xed | <a href="#">HCINCTRL</a>      | host in control                             |  |
| 0x07 | 0xee | <a href="#">DBGMODE</a>       | debug mode                                  |  |
| 0x07 | 0xef | <a href="#">VDCTRL</a>        | USB PHY vendor control                      |  |
| 0x07 | 0xf1 | <a href="#">VDSTAT</a>        | USB PHY vendor status                       |  |
| 0x07 | 0xf3 | <a href="#">BKDOOR</a>        | Test back door                              |  |
| 0x07 | 0xf5 | <a href="#">OTGTRIEN</a>      | OTR status machine interrupt enable         |  |
| 0x07 | 0xf6 | <a href="#">OTGTRIRQ</a>      | OTR status machine interrupt request        |  |
| 0x07 | 0xf7 | <a href="#">USB_Efuse_Ref</a> | Usb Access Efuse_Ref register               |  |
| 0x07 | 0xfc | <a href="#">FSMPRESTATE</a>   | FSM pre-state register                      |  |
| 0x07 | 0xfd | <a href="#">HCIN2CNTH</a>     | hcin2 packet counter high                   |  |
| 0x07 | 0xfe | <a href="#">HCIN2CNTL</a>     | hcin2 packet counter low                    |  |

|      |      |                            |   |      |
|------|------|----------------------------|---|------|
| 0x08 | 0x89 | <b>ACC_CTRL</b>            | Control Register  | ACC  |
| 0x08 | 0x8a | <b>ACC_BLKLEN_LENHI</b>    | Search block length or the low byte of data length                |      |
| 0x08 | 0x8b | <b>ACC_BLKNUM_LENLO</b>    | Search block number or the high byte of data length               |      |
| 0x08 | 0xf1 | <b>ACC_CMPDATA0_SUMHIE</b> | The 1 <sup>st</sup> byte of data searched or the high byte result |      |
| 0x08 | 0xf2 | <b>ACC_CMPDATA1_SUMHI</b>  | The 2 <sup>nd</sup> byte of data searched or the high byte result |      |
| 0x08 | 0xf3 | <b>ACC_CMPDATA2_SUMLO</b>  | The 3 <sup>rd</sup> byte of data searched or the low byte result  |      |
| 0x08 | 0xf4 | <b>ACC_CMPDATA3_SUMLOE</b> | The 4 <sup>th</sup> byte of data searched or the low byte result  |      |
| 0x08 | 0xf5 | <b>ACC_MATCNTHI</b>        | The high byte of bitmap   |      |
| 0x08 | 0xf6 | <b>ACC_MATCNTLO</b>        | The low byte of match number                                      |      |
| 0x08 | 0xf7 | <b>ACC_BITMAPHI</b>        | The high byte of bitmap   |      |
| 0x08 | 0xf8 | <b>ACC_BITMAPLO</b>        | The low byte of bitmap  |      |
| 0x08 | 0x8c | <b>ACC_FIFO</b>            | ACC FIFO address  |      |
| 0x09 | 0x90 | <b>SD_CMD</b>              | SD/MMC CMD Register   | CARD |
| 0x09 | 0x91 | <b>SD_CMD_ARGRSP</b>       | SD/MMC Argument or RSP Register                                   |      |
| 0x09 | 0x92 | <b>SD_RSP_POIN</b>         | SD/MMC RSP Point Register   |      |
| 0x09 | 0x93 | <b>SD_TF_CTL</b>           | SD/MMC control register   |      |
| 0x09 | 0x94 | <b>SD_STATE</b>            | MMC/SD status Register  |      |
| 0x09 | 0x97 | <b>SD_BLK_SIZE_H</b>       | SD/MMC High Block size Register                                   |      |
| 0x09 | 0x98 | <b>SD_BLK_SIZE_L</b>       | SD/MMC Low Block size Register                                    |      |
| 0x09 | 0x99 | <b>SD_BLK_NUM</b>          | SD/MMC BLOCK number Register                                      |      |
| 0x09 | 0x9a | <b>SD_CLK_CTL</b>          | SD/MMC Clock Control Register                                     |      |
| 0x09 | 0x9b | <b>SD_PAD_CTL</b>          | SD/MMC PAD CONTROL Register                                       |      |
| 0x09 | 0x9c | <b>SD_INT_CTL</b>          | SD/MMC INTERRUPT Control Register                                 |      |
| 0x09 | 0x9d | <b>SD_DATA_FIFO</b>        | SD/MMC Data FIFO Register   |      |
| 0x09 | 0x9e | <b>SD_TIMEOUT_CTL</b>      | Data Timeout Counter Register                                     | SDC  |
| 0x09 | 0xad | <b>SD_TIMING_CTL</b>       | CARD TIMING CONTROL   |      |
| 0x09 | 0xae | <b>SD_DBG_CTL</b>          | SDC debug control   |      |
| 0x0a | 0x99 | <b>SPI_CTL</b>             | SPI Control Register  |      |
| 0x0a | 0x9a | <b>SPI_DRQ</b>             | SPI DMA/IRQ control Register.                                     | SPI  |
| 0x0a | 0x9b | <b>SPI_STA</b>             | SPI Status Register   |      |
| 0x0a | 0x9c | <b>SPI_CLKDIV</b>          | SPI Clock Divide Control Register                                 |      |
| 0x0a | 0x9d | <b>SPI_TXDAT</b>           | SPI tx fifo register  |      |
| 0x0a | 0x9e | <b>SPI_RXDAT</b>           | SPI rx fifo register  |      |
| 0x0a | 0x9f | <b>SPI_BCL</b>             | SPI Bytes Count Low Register                                      |      |
| 0x0a | 0xa2 | <b>SPI_BCH</b>             | SPI Bytes Count high Register                                     |      |
| 0x0a | 0xa3 | <b>SPI_DEBUG</b>           | SPI debug register  |      |
| 0x0a | 0xc8 | <b>SPI_RAND</b>            | SPI Randomizer Control Register                                   |      |
| 0x0a | 0xc9 | <b>SPI_SEED0</b>           | SPI Randomizer test seed Register.                                |      |
| 0x0a | 0xca | <b>SPI_SEED1</b>           | SPI Randomizer test seed Register.                                | UART |
| 0x0a | 0x90 | <b>UART_BR</b>             | UART BAUDRATE Register.   |      |
| 0x0a | 0x91 | <b>UART_MODE</b>           | UART mode setup Register.   |      |

|      |      |                   |                              |         |
|------|------|-------------------|------------------------------|---------|
| 0x0a | 0x92 | <b>UART_CTL</b>   | UART Control Register.       |         |
| 0x0a | 0x93 | <b>UART_DRQ</b>   | UART DRQ/IRQ register        |         |
| 0x0a | 0x94 | <b>UART_STA</b>   | UART Status Register         |         |
| 0x0a | 0x95 | <b>UART_TXDAT</b> | UART TX FIFO register        |         |
| 0x0a | 0x97 | <b>UART_RXDAT</b> | UART RX FIFO register        |         |
| 0x0a | 0x98 | <b>UART_DEBUG</b> | UART debug register.         |         |
| 0x0a | 0xa5 | <b>IR_CTL</b>     | IR Control Register          | IR      |
| 0x0a | 0xa6 | <b>IR_STA</b>     | IR Status Register           |         |
| 0x0a | 0xa7 | <b>IR_LUC</b>     | IR low user code register.   |         |
| 0x0a | 0xa9 | <b>IR_HUC</b>     | IR high user code register.  |         |
| 0x0a | 0xaa | <b>IR_KDC</b>     | IR key data code register.   |         |
| 0x0a | 0xb5 | <b>IR_CONFIG8</b> | IR inner demodulator config8 |         |
| 0x0c | 0xa9 | <b>RTC_CTL0</b>   | RTC Control 0 Register       | RTC/CTC |
| 0x0c | 0xaa | <b>RTC_CTL1</b>   | RTC Control 1 register       |         |
| 0x0c | 0xab | <b>RTCTimeS</b>   | RTC Time Second Register     |         |
| 0x0c | 0xac | <b>RTCTimeMin</b> | RTC Time Minute Register     |         |
| 0x0c | 0xad | <b>RTCTimeH</b>   | RTC Time Hour Register       |         |
| 0x0c | 0xae | <b>RTCTimeD</b>   | RTC Time Day Register        |         |
| 0x0c | 0xaf | <b>RTCTimeMon</b> | RTC Time Month Register      |         |
| 0x0c | 0xb0 | <b>RTCTimeY</b>   | RTC Time Year Register       |         |
| 0x0c | 0xb1 | <b>RTCALMS</b>    | RTC Alarm Second Register    |         |
| 0x0c | 0xb2 | <b>RTCALMM</b>    | RTC Alarm Minute Register    |         |
| 0x0c | 0xb3 | <b>RTCALMH</b>    | RTC Alarm Hour Register      |         |
| 0x0c | 0xb4 | <b>RTCRUPD</b>    | RTC Register update Register |         |
| 0x0c | 0xb5 | <b>TimerLB</b>    | Timer low Byte               |         |
| 0x0c | 0xb6 | <b>TimerMB</b>    | Timer middle Byte            |         |
| 0x0c | 0xb7 | <b>TimerHB</b>    | Timer high Byte              |         |
| 0x0c | 0xb8 | <b>WDCTL</b>      | watch dog control register   |         |
| 0x0c | 0xb9 | <b>CTCCTL</b>     | CTC control register         |         |
| 0x0c | 0xba | <b>CTCCNTL</b>    | CTC counter low register     |         |
| 0x0c | 0xbb | <b>CTCCNTH</b>    | CTC counter high register    |         |
| 0x0c | 0xbc | <b>RTCRDM0</b>    | RTC Random access Register   |         |
| 0x0c | 0xbd | <b>RTCRDM1</b>    | RTC Random access Register   |         |
| 0x0c | 0xbf | <b>RTCRDM2</b>    | RTC Random access Register   |         |
| 0x0c | 0xc1 | <b>RTCRDM3</b>    | RTC Random access Register   |         |
| 0x0c | 0xc2 | <b>RTCRDM4</b>    | RTC Random access Register   |         |
| 0x0c | 0xc3 | <b>RTCRDM5</b>    | RTC Random access Register   |         |
| 0x0c | 0xc4 | <b>RTCRDM6</b>    | RTC Random access Register   |         |
| 0x0c | 0xc5 | <b>RTCRDM7</b>    | RTC Random access Register   |         |
| 0x0c | 0xc6 | <b>RTCRDM8</b>    | RTC Random access Register   |         |
| 0x0c | 0xc7 | <b>RTCRDM9</b>    | RTC Random access Register   |         |

|      |      |                         |   |    |
|------|------|-------------------------|---|----|
| 0x0c | 0xc8 | <b>RTCRDM10</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xc9 | <b>RTCRDM11</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xca | <b>RTCRDM12</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xcb | <b>RTCRDM13</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xcc | <b>RTCRDM14</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xcd | <b>RTCRDM15</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xce | <b>RTCRDM16</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xcf | <b>RTCRDM17</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xd2 | <b>RTCRDM18</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xd3 | <b>RTCRDM19</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xd4 | <b>RTCRDM20</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xd5 | <b>RTCRDM21</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xd6 | <b>RTCRDM22</b>         | RTC Random access Register                                    |    |
| 0x0c | 0xd7 | <b>CTCCTL2</b>          | CTC2 control register   |    |
| 0x0c | 0xd8 | <b>CTCCNTL2</b>         | CTC2 counter low register                                     |    |
| 0x0c | 0xd9 | <b>CTCCNTH2</b>         | CTC3 counter high register                                    |    |
| 0x14 | 0x90 | <b>TK_EN_CTL</b>        | Touch Key Control Register.                                   | TK |
| 0x14 | 0x91 | <b>TK_FUN_CTL</b>       | Touch Key Function Control Register.                          |    |
| 0x14 | 0x92 | <b>TK_IRQ_PD</b>        | Touch Key IRQ Pending Status Register.                        |    |
| 0x14 | 0x93 | <b>TK_PRESS_STA</b>     | Touch Key Press Status Register.                              |    |
| 0x14 | 0x94 | <b>TK_DB_CTL</b>        | Touch Key Debounce Control Register.                          |    |
| 0x14 | 0x95 | <b>TK_PWM_CNTH</b>      | Touch Key PWM Counter High Register.                          |    |
| 0x14 | 0x97 | <b>TK_PWM_CNTL</b>      | Touch Key PWM Counter Low Register.                           |    |
| 0x14 | 0x99 | <b>TK_PRESS_TH</b>      | Touch Key Press Threshold Register.                           |    |
| 0x14 | 0x9A | <b>TK_BL_CAL_TH</b>     | Touch Key Baseline Calibration Threshold Register.            |    |
| 0x14 | 0x9B | <b>TK_NOISE_TH</b>      | Touch Key Noise Threshold Register.                           |    |
| 0x14 | 0x9C | <b>TK_BUCKET_TH</b>     | Touch Key Bucket Threshold And Baseline Step Adjust Register. |    |
| 0x14 | 0x9D | <b>TK_AUTO_LOWTH_H</b>  | Touch Key idac auto adjust low threshold high bits Register.  |    |
| 0x14 | 0x9E | <b>TK_AUTO_LOWTH_L</b>  | Touch Key idac auto adjust low threshold low bits Register.   |    |
| 0x14 | 0xa2 | <b>TK_AUTO_HIGHTH_H</b> | Touch Key idac auto adjust high threshold high bits Register. |    |
| 0x14 | 0xa3 | <b>TK_AUTO_HIGHTH_L</b> | Touch Key idac auto adjust high threshold low bits Register.  |    |
| 0x14 | 0xa4 | <b>TK_KEY0_IDA_CTL</b>  | Touch Key0 charging current Control Register.                 |    |
| 0x14 | 0xa5 | <b>TK_KEY1_IDA_CTL</b>  | Touch Key1 charging current Control Register.                 |    |
| 0x14 | 0xa6 | <b>TK_KEY2_IDA_CTL</b>  | Touch Key2 charging current Control Register.                 |    |
| 0x14 | 0xa9 | <b>TK_KEY3_IDA_CTL</b>  | Touch Key3 charging current Control Register.                 |    |
| 0x14 | 0xaa | <b>TK_KEY4_IDA_CTL</b>  | Touch Key4 charging current Control Register.                 |    |

|      |      |                        |   |  |
|------|------|------------------------|---|--|
| 0x14 | 0Xab | <b>TK_KEY5_IDA_CTL</b> | Touch Key5 charging current Control Register.   |  |
| 0x14 | 0Xad | <b>TK_UPDAT_PD</b>     | Touch Key New Data Update Pending Register.     |  |
| 0x14 | 0Xae | <b>TK_KEY0_DATH</b>    | Touch Key0 current data high bits.              |  |
| 0x14 | 0Xaf | <b>TK_KEY0_DATL</b>    | Touch Key0 current data low bits.               |  |
| 0x14 | 0Xb0 | <b>TK_KEY1_DATH</b>    | Touch Key1 current data high bits.              |  |
| 0x14 | 0Xb1 | <b>TK_KEY1_DATL</b>    | Touch Key1 current data low bits.               |  |
| 0x14 | 0Xb2 | <b>TK_KEY2_DATH</b>    | Touch Key2 current data high bits.              |  |
| 0x14 | 0Xb3 | <b>TK_KEY2_DATL</b>    | Touch Key2 current data low bits.               |  |
| 0x14 | 0Xb4 | <b>TK_KEY3_DATH</b>    | Touch Key3 current data high bits.              |  |
| 0x14 | 0Xb5 | <b>TK_KEY3_DATL</b>    | Touch Key3 current data low bits.               |  |
| 0x14 | 0Xb6 | <b>TK_KEY4_DATH</b>    | Touch Key4 current data high bits.              |  |
| 0x14 | 0Xb7 | <b>TK_KEY4_DATL</b>    | Touch Key40 current data low bits.              |  |
| 0x14 | 0Xb8 | <b>TK_KEY5_DATH</b>    | Touch Key5 current data high bits.              |  |
| 0x14 | 0Xb9 | <b>TK_KEY5_DATL</b>    | Touch Key5 current data low bits.               |  |
| 0x14 | 0Xba | <b>TK_RAW_PD</b>       | Touch Key New RAW Data Update Pending Register. |  |
| 0x14 | 0Xbc | <b>TK_KEY0_RAWH</b>    | Touch Key0 current RAW high bits.               |  |
| 0x14 | 0Xbd | <b>TK_KEY0_RAWL</b>    | Touch Key0 current RAW low bits.                |  |
| 0x14 | 0Xbf | <b>TK_KEY1_RAWH</b>    | Touch Key1 current RAW high bits.               |  |
| 0x14 | 0Xc1 | <b>TK_KEY1_RAWL</b>    | Touch Key1 current RAW low bits.                |  |
| 0x14 | 0Xc2 | <b>TK_KEY2_RAWH</b>    | Touch Key2 current RAW high bits.               |  |
| 0x14 | 0Xc3 | <b>TK_KEY2_RAWL</b>    | Touch Key2 current RAW low bits.                |  |
| 0x14 | 0Xc4 | <b>TK_KEY3_RAWH</b>    | Touch Key3 current RAW high bits.               |  |
| 0x14 | 0Xc5 | <b>TK_KEY3_RAWL</b>    | Touch Key3 current RAW low bits.                |  |
| 0x14 | 0Xc6 | <b>TK_KEY4_RAWH</b>    | Touch Key4 current RAW high bits.               |  |
| 0x14 | 0Xc7 | <b>TK_KEY4_RAWL</b>    | Touch Key4 current RAW low bits.                |  |
| 0x14 | 0Xc8 | <b>TK_KEY5_RAWH</b>    | Touch Key5 current RAW high bits.               |  |
| 0x14 | 0Xc9 | <b>TK_KEY5_RAWL</b>    | Touch Key5 current RAW low bits.                |  |
| 0x14 | 0Xca | <b>TK_KEY0_BL_H</b>    | Touch Key0 current baseline high bits.          |  |
| 0x14 | 0Xcc | <b>TK_KEY0_BL_L</b>    | Touch Key0 current baseline low bits.           |  |
| 0x14 | 0Xcd | <b>TK_KEY1_BL_H</b>    | Touch Key1 current baseline high bits.          |  |
| 0x14 | 0Xce | <b>TK_KEY1_BL_L</b>    | Touch Key1 current baseline low bits.           |  |
| 0x14 | 0Xd2 | <b>TK_KEY2_BL_H</b>    | Touch Key2 current baseline high bits.          |  |
| 0x14 | 0Xd3 | <b>TK_KEY2_BL_L</b>    | Touch Key2 current baseline low bits.           |  |
| 0x14 | 0Xd4 | <b>TK_KEY3_BL_H</b>    | Touch Key3 current baseline high bits.          |  |
| 0x14 | 0Xd5 | <b>TK_KEY3_BL_L</b>    | Touch Key3 current baseline low bits.           |  |
| 0x14 | 0Xd6 | <b>TK_KEY4_BL_H</b>    | Touch Key4 current baseline high bits.          |  |
| 0x14 | 0Xd7 | <b>TK_KEY4_BL_L</b>    | Touch Key4 current baseline low bits.           |  |
| 0x14 | 0Xd8 | <b>TK_KEY5_BL_H</b>    | Touch Key5 current baseline high bits.          |  |
| 0x14 | 0Xd9 | <b>TK_KEY5_BL_L</b>    | Touch Key5 current baseline low bits.           |  |
| 0x14 | 0Xda | <b>TK_DEBUG</b>        | Touch Key Debug Register.                       |  |

## 3 Processor (黃少彬、黃俏)

## 4 Debug Unit (黃少彬、黃俏)

| 日期         | 版本     | 描述      | 修订人 |
|------------|--------|---------|-----|
| 2012-07-24 | V1. 00 | initial | 黃少彬 |

### 4.1 Features

The BIRD (Built-in Real-time Debugger) provides the application program developer with a real-time and comfortable application debugging at a low cost. The BIRD is an embedded debug solution which combines a Virtual Component in the SoC and a Monitor Program, forming the best solution for real-time emulation at SoC level with complex events, trace and quick code loading... The BIRD is driven by the Rlink-BIRD, a low cost adapter which interfaces the SoC with the development environment on a PC.

The BIRD concept is original since it combines a software solution (monitor) and a hardware (breakpoint, trace modules) solution, and provides high-end debug features whilst minimizing the silicon cost.

The BIRD has following features:

- (1) Real time ffuse r , no CPU frequency limitation
- (2) communication through USB 1.1 with the host PC and JTAG with the SoC.
- (3) Source level debugging such as single step, C-line to line, step into, step over, go to address, step out.
- (4) Unlimited software breakpoints.
- (5) 4 real-time hardware breakpoint (PC comparator).
- (6) an access to DATA, SFR or CODE memory space
- (7) a read, write or read/write at a given address
- (8) a read, write or read/write of a given data
- (9) a combination of the 1 data event
- (10) a sequence of the 1 data event

### 4.2 Function Description

When the BIRD is integrated with the Flip80251, it provides a new operating mode to the Flip80251, the debug mode. The debug mode is activated by a JTAG instruction: when the user starts a debug session on the host PC, the BIRD plug-in sends automatically the activation command through the JTAG interface. Then, the BIRD is activated and the debug functions become available. The debug mode is exited when the CPU is reset (when the input corerst is asserted) After a reset, the CPU is in normal mode and the BIRD is off. That means the clock connected to BIRD modules is stopped and debug functions are de-activated. In

normal mode, the BIRD does not impact the dynamic power consumption of the Flip80251. In debug mode, the total dynamic power consumption (BIRD+CPU) is just slightly higher than the power consumption in CPU normal mode.

## 4.3 Module Description

### 4.3.1 Block Diagram

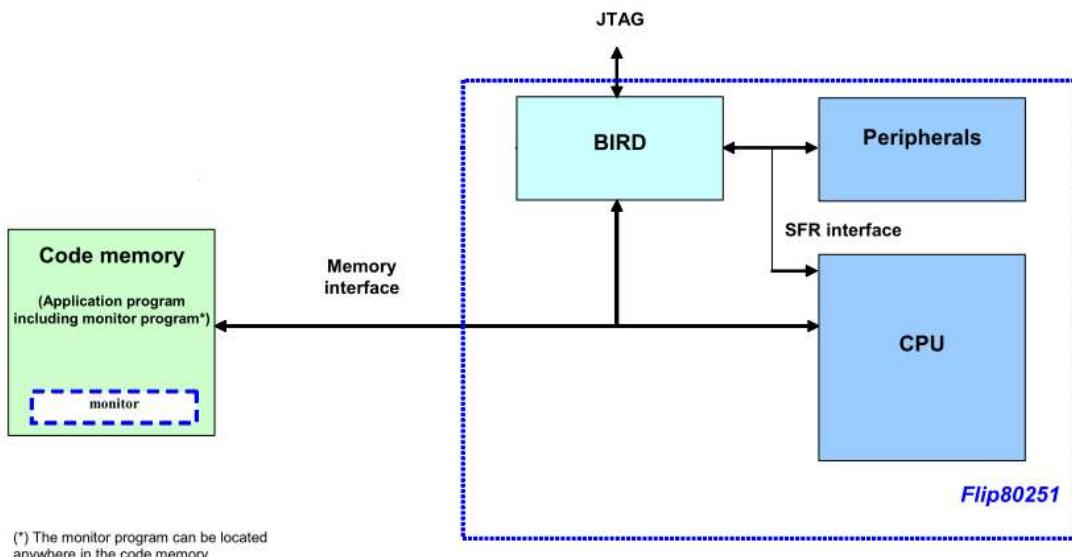


Figure 4-1 block diagram of BIRD

### 4.3.2 TAP state machine

The process of serial test and debug is best explained in conjunction with the JTAG state machine. Figure below shows the state transitions that occur in the TAP controller. The TAP state machine is from IEEE Std 1149.1-1990. Copyright 1999 IEEE. All rights reserved.

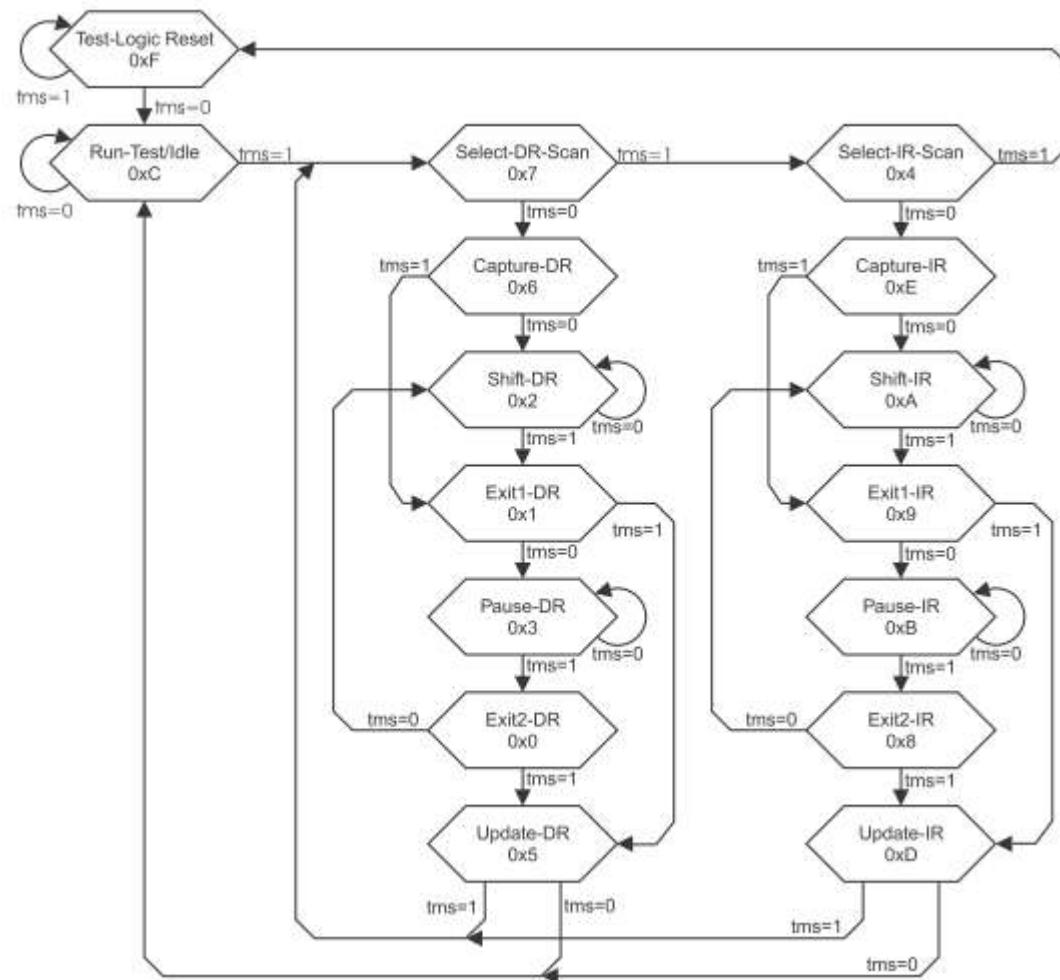


Figure 4-2 TAP state machine

### 4.3.3 JTAG Instruction

The Bird can execute the following instructions with is identical with the standard of JTAG communication IEEE 1149.1.

| Code | Name            | Chosen DR<br>(Source of data) | Description   |
|------|-----------------|-------------------------------|---|
| 001  | IR_IDCODE       | ID Register                   | 32 bit ID register. Identifies device and accessed processor in the device. |
| 010  | IR_SetAddr      | IM_AD                         | Address Register  |
| 011  | IR_RdData       | IM_DR                         | Read data register  |
| 100  | IR_WrData       | IM_DW                         | Write data register   |
| 101  | IR_StatusAccess | BIRD_BUS_STATUS               | Bird bus status register  |
| 111  | IR_BYPASS       | BYPASS register               | Select Bypass register, performing a BYPASS                                 |

Table 1 implemented Instructions

The description of ID instruction:

| Bits | Value | Description | Access |
|------|-------|-------------|--------|
|      |       |             |        |

|       |             |  |   |
|-------|-------------|--|---|
| 31:28 | 4'b1001     | Information on Core  | R |
| 27:12 | 16'Hffff    | Not used bits  | R |
| 11:5  | 7'b00000110 | Delivry number. This number has to be increased for each BIRD Delivery | R |
| 4:3   | 2'b01       | BIRD configuration   | R |
| 2     | 1'b1        | CRC, parity bit of ID[31:17]   | R |
| 1     | 1'b0        | CRC, parity bit of ID[16:3]  | R |
| 0     | 1'b1        | CRC, parity bit of ID[31:1]  | R |

Table 2 ID instruction

ID=32'b9FFFF0CD

The description of IM\_AD instruction:

| Bits | Name    | Description                          | Access | Reset    |
|------|---------|--------------------------------------|--------|----------|
| 4:0  | address | Bird Register Address used to access | R      | 5'b00000 |

Table 3 IM\_AD instruction

The description of IM\_DR instruction:

| Bits | Name | Description                    | Access | Reset |
|------|------|--------------------------------|--------|-------|
| 30:0 | data | The data from Register Address | R      | 31'b0 |

Table 4 IM\_DR instruction

The description of IM\_DW instruction:

| Bits | Name | Description                  | Access | Reset |
|------|------|------------------------------|--------|-------|
| 30:0 | data | The data to Register Address | R      | 8'b0  |

Table 5 IM\_DW instruction

The description of BIRD\_BUS\_STATUS instruction:

| Bits | Name            | Description  | Access | Reset |
|------|-----------------|--|--------|-------|
| 8:6  | reserve         | Reserved   | R      | 3'h0  |
| 5    | rd_cmd          | Reading JTAG register, low is active                       | R      | 1'b1  |
| 4    | wr_cmd          | Writing JTAG register, low is active                       | R      | 1'b1  |
| 3    | rdy_for_rd      | Ready to read JTAG register, high is active                | R      | 1'b0  |
| 2    | rdy_for_wr      | Ready to write JTAG register, high is active               | R      | 1'b0  |
|      | wr_finished     | Writing is finished  | R      | 1'b1  |
| 1    | RST             | Initiate reset(write 0 to this bit will enable bird clock) | W      | 1'b1  |
| 0    | clk_on          | Bird clock is present                                      | R      | 1'b0  |
|      | INIT_CLK_DETECT | Detect the presence of the core clock                      | W      | 1'b1  |

Table 6 BIRD\_BUS\_STATUS instruction

The description of BYPASS instruction:

| Bits | Name | Description                              | Access | Reset |
|------|------|--|--------|-------|
| 0    | -    | Ignored on writes; returns zero on reads | R      | 1'b0  |

Table 7 BYPASS instruction

## 4.4 BIRD Register Group

| Function Block    | Register name   | BIRD address | Description                       |
|-------------------|-----------------|--------------|-----------------------------------|
| Memory Access     | R_AD_MAM_ADDR   | 5'b00000     | Read or write address Register    |
|                   | R_AD_MAM_DATA   | 5'b00001     | Read or write data Register       |
|                   | R_AD_MAM_CTRL   | 5'b00010     | Read or write control Register    |
| MCU Communication | R_AD_CCM_CTRL   | 5'b00100     | Communication Register            |
|                   | R_AD_CCM_STATUS | 5'b00100     | Communication Status Register     |
|                   | R_AD_CCM_COMM   | 5'b00101     | Communication Data Register       |
| Event Manage      | R_AD_EM_CTRL1   | 5'b01001     | Event manage control Register 1   |
|                   | R_AD_EM_BASICEV | 5'b01010     | Event manage basic event Register |
|                   | R_AD_EM_EVCTRL0 | 5'b01100     | Event Control Register 0          |
|                   | R_AD_EM_EVCTRL1 | 5'b01101     | Event Control Register 1          |
|                   | R_AD_EM_EVCTRL2 | 5'b01110     | Event Control Register 2          |
| Breakpoint Manage | R_AD_BM_CTRL    | 5'b10000     | Breakpoint Control Register       |
|                   | R_AD_BM_CMP0    | 5'b10100     | Breakpoint Compare Register 0     |
|                   | R_AD_BM_CMP1    | 5'b10101     | Breakpoint Compare Register 1     |
|                   | R_AD_BM_CMP2    | 5'b10110     | Breakpoint Compare Register 2     |
|                   | R_AD_BM_CMP3    | 5'b10111     | Breakpoint Compare Register 3     |
| MCU Manage        | B_AD_CPU_INFO   | 5'b11000     | CPU information Register          |

Table 8 BIRD Register

## 4.5 BIRD Register Description

### 4.5.1 Memory Access Register Description

**R\_AD\_MAM\_ADDR**(Memory Access Address register, BIRD address 5'b00000)

| Bit Number | Bit Mnemonic | Function   | Access | Reset   |
|------------|--------------|--|--------|---------|
| 23:0       | address      | The 24-bit address to access mcu. The address should write 3 times for only 8-bit can be shifted in each time. | R/W    | 000000h |

**R\_AD\_MAM\_DATA**(Memory Access Data register, BIRD address 5'b00001)

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | M_DW         | Data to write. Write these bit can write program memory & data memory. | R      | 00h   |
|            | M_DR         | Data to read.  | W      | 00h   |

**R\_AD\_MAM\_CTRL(Memory Access Control register, BIRD address 5'b00010)**

| Bit Number | Bit Mnemonic   | Function   | Access | Reset |
|------------|----------------|--|--------|-------|
| 7:1        | Reserved       | Be read as '0000000'   | -      | -     |
| 0          | BIRDDmaRequest | 0: Bird does not require access memory<br>1: Bird requires access memory | R      | 0     |

**4.5.2 MCU Communication Register Description****R\_AD\_CCM\_CTRL(Communication register, BIRD address 5'b00100)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:3        | Reserved     | Be write as '00000'  | -      | -     |
| 2          | code_repl_on | 0: Use the instruction from program memory;<br>1: Use the instruction from R_AD_CCM_COMM register. | W      | 0     |
| 1          | PSTOP        | 0: No hardware interrupt;<br>1: Generate hardware interrupt.                                       | W      | 0     |
| 0          | FRST         | 0: Active MCU;<br>1: Reset MCU.  | W      | 0     |

**R\_AD\_CCM\_STATUS(Communication Status register, BIRD address 5'b00100)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:3        | Reserved     | Be write as '00000'  | -      | -     |
| 2          | code_repl_on | 0: the instruction from program memory;<br>1: the instruction is from R_AD_CCM_COMM register.                        | R      | 0     |
| 1          | HALT         | 0: MCU is in normal mode;<br>1: MCU is in debug mode, a interrupt service routine is generated by trap instruction.. | R      | 0     |
| 0          | CORE_RST     | 0: MCU is active;<br>1: MCU is reset.  | R      | 0     |

**4.5.3 Event Manage Register Description****R\_AD\_EM\_CTRL1(Event manage control Register 1, BIRD address 5'b01001)**

| Bit Number | Bit Mnemonic | Function | Access | Reset |
|------------|--------------|----------|--------|-------|
|            |              |          |        |       |

---

|     |          |                         |   |   |
|-----|----------|-------------------------|---|---|
| 7:1 | Reserved | Be write as '0000000'   | - | - |
| 0   | EN_EV0   | Enable event 0 trigger. | W | 0 |

**R\_AD\_EM\_BASICEV(Event manage basic event Register, BIRD address 5'b01010)**

| Bit Number | Bit Mnemonic  | Function                       | Access | Reset |
|------------|---------------|--------------------------------|--------|-------|
| 7:1        | Reserved      | Be read as '0000000'           | -      | -     |
| 0          | BASIC_EV_RS T | Reset all event trigger module | W      | 0     |

**R\_AD\_EM\_EVCTRL0(Event Control Register 0, BIRD address 5'b01100)**

| Bit Number | Bit Mnemonic | Function                                     | Access | Reset      |
|------------|--------------|--|--------|------------|
| 47:46      | Reserved     | Be write as '00'                             | -      | -          |
| 45:23      | addr_ref     | Reference address                            | W      | 0xffffffff |
| 22:0       | addr_mask    | Address mask, '1' means address mask active. | W      | 0x0000 00  |

**R\_AD\_EM\_EVCTRL1(Event Control Register 1, BIRD address 5'b01101)**

| Bit Number | Bit Mnemonic | Function                                  | Access | Reset |
|------------|--------------|---|--------|-------|
| 15:8       | data_ref     | Reference data                            | W      | 0xff  |
| 7:0        | data _mask   | Data mask, '1' means address mask active. | W      | 0x00  |

**R\_AD\_EM\_EVCTRL2(Event Control Register 2, BIRD address 5'b01110)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | cnt_end      | The event match while cnt_end times of event occurrence.   | W      | 0001  |
| 3:2        | bus_select   | The bus selected to match the event.<br>00: program memory<br>01: data memory<br>10: reserved<br>11: sfr | W      | 00    |
| 1          | wr_access    | 0: write access disable<br>1: write access enable  | W      | 0     |
| 0          | rd_access    | 0: read access disable<br>1: read access enable  | W      | 0     |

#### 4.5.4 Breakpoint Manage Register Description

**R\_AD\_BM\_CTRL**(Breakpoint Control Register, BIRD address 5'b10000)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:5        | Reserved     | Be write as '000'   | -      | -     |
| 4          | EN_BEVENT    | 0: Event will trigger trace function;<br>1: Event will trigger hardware breakpoint interrupt. | W      | 0     |
| 3          | EN_CMP3      | 0: disable breakpoint 3<br>1: enable breakpoint 3   | W      | 0     |
| 2          | EN_CMP0      | 0: disable breakpoint 2<br>1: enable breakpoint 2   | W      | 0     |
| 1          | EN_CMP1      | 0: disable breakpoint 1<br>1: enable breakpoint 1.  | W      | 0     |
| 0          | EN_CMP0      | 0: disable breakpoint 0<br>1: enable breakpoint 0   | W      | 0     |

**R\_AD\_BM\_CMP0**(Breakpoint Compare Register 0, BIRD address 5'b10100)

| Bit Number | Bit Mnemonic | Function                          | Access | Reset |
|------------|--------------|-----------------------------------|--------|-------|
| 23:0       | CMP_PC       | The program counter may be match. | W      | 0     |

**R\_AD\_BM\_CMP1**(Breakpoint Compare Register 1, BIRD address 5'b10101)

| Bit Number | Bit Mnemonic | Function                          | Access | Reset |
|------------|--------------|-----------------------------------|--------|-------|
| 23:0       | CMP_PC       | The program counter may be match. | W      | 0     |

**R\_AD\_BM\_CMP2**(Breakpoint Compare Register 2, BIRD address 5'b10110)

| Bit Number | Bit Mnemonic | Function                          | Access | Reset |
|------------|--------------|-----------------------------------|--------|-------|
| 23:0       | CMP_PC       | The program counter may be match. | W      | 0     |

**R\_AD\_BM\_CMP3**(Breakpoint Compare Register 3, BIRD address 5'b10111)

| Bit Number | Bit Mnemonic | Function                          | Access | Reset |
|------------|--------------|-----------------------------------|--------|-------|
| 23:0       | CMP_PC       | The program counter may be match. | W      | 0     |

#### 4.5.5 MCU Manage Register Description

**B\_AD\_CPU\_INFO**(CPU information Register, BIRD address 5'b11000)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:6        | Reserved     | Be read as '00'   | -      | -     |
| 5          | i_moninuser  | 0: special debug ram is used for monitor program;<br>1: program memory is used for monitor program. | R      | 0     |
| 4          | Reserved     | Be read as '0'  | -      | -     |
| 3          | isolintrmode | 0: 2 bytes stack mode;<br>1: 4 bytes stack mode t.  | R      | 0     |
| 2:0        | Reserved     | Be read as '000'  | -      | -     |

## 4.6 SFR accessed BIRD Register

These register can be accessed by MCU:

| Register name   | SFR address | SFR bank address | Description                    |
|-----------------|-------------|------------------|--------------------------------|
| C_AD_CCM_CTRL   | 0x8e        | All bank         | Communication Control Register |
| C_AD_CCM_STATUS | 0x8e        |                  | Communication Status Register  |
| C_AD_CCM_COMM   | 0x8f        |                  | Communication Data Register    |
| C_AD_CCM_REG    | 0x8f        |                  | Communication manage Register  |
| C_AD_CCM_REPL   | 0x8f        |                  | Code replace Register          |
| BIRDINFO        | 0x85        |                  | BIRD information Register      |

Table 9 BIRD Register for MCU access

## 4.7 MCU accessed Register Description

**C\_AD\_CCM\_CTRL(Communication Control Register, SFR Address 0x8e, SFR all bank)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:2        | Reserved     | Be write as '000000'  | -      | -     |
| 1:0        | feature_sel  | 00: read or wirte sfr address 0x8f will select R_AD_CCM_COMM ffuse r<br>01: read or wirte sfr address 0x8f will select C_AD_CCM_REG register<br>10: read or wirte sfr address 0x8f will select C_AD_CCM_REPL register<br>11: reserved | W      | 00    |

**C\_AD\_CCM\_STATUS(Communication Control Register, SFR Address 0x8e, SFR all bank)**

| Bit Number | Bit Mnemonic  | Function  | Access | Reset |
|------------|---------------|---|--------|-------|
| 7:2        | Reserved      | Be read as '000000'   | -      | -     |
| 1          | C_Wr_COMM_RDY | 0: B_AD_CCM_COMM is not empty<br>1: B_AD_CCM_COMM is empty<br><br>The MCU can read the information of BIRD from the B_AD_CCM_COMM if it (COMM_C_TO_B) is empty. | R      | 0     |
| 0          | C_Rd_COMM_RDY | 0: B_AD_CCM_COMM is not empty<br>1: B_AD_CCM_COMM is empty<br><br>The MCU can write the information of BIRD to the B_AD_CCM_COMM if it (COMM_B_TO_C) is empty.  | R      | 0     |

**C\_AD\_CCM\_COMM(Communication Data Register, SFR Address 0x8f, SFR all bank)**

| Bit Number | Bit Mnemonic | Function                                     | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | RW_DATA      | Used for exchange data between CPU and BIRD. | R/W    | 0     |

**C\_AD\_CCM\_REG(Communication manage Register, SFR Address 0x8f, SFR all bank)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7          | Soft_breakpoint | 0: disable software breakpoint<br>1: enable software breakpoint                | R/W    | 0     |
| 6:1        | Reserved        | Be read as '000000'  | -      | -     |
| 0          | enj2user        | 0: the MCU can return to normal mode if the reti instruction is been executed. | W      | 0     |
|            | isolintrmode    | 0: 2 bytes stack mode<br>1: 4 bytes stack mode                                 | R      | 0     |

**C\_AD\_CCM REPL(Code Replace Register, SFR Address 0x85, SFR all bank)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | Code_replace | The instruction can be replaced by the data in this register. | W      | 0     |

**BIRDINFO(Bird information Register, SFR Address 0x8f, SFR all bank)**

| Bit Number | Bit Mnemonic | Function                                      | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | BIRDinfo     | User defined BIRD information for debug mode. | R/W    | -     |

## 5 Memory Controller(彭洪、黄俏)

| 日期         | 版本     | 描述  | 修订人 |
|------------|--------|---|-----|
| 2012-07-24 | V1. 00 | initial   | 彭洪  |
| 2012-09-12 | V1. 02 | 1、增加 PAGEMISS 的输入 ENTRY<br>2、增加 PageAddr8H, PageAddr8L, PageAddr9H, PageAddr9L, PageAddrMask8, PageAddrMask9, RedirectAddr8, RedirectAddr9<br>3、修改了 CodeReplaceEntry 的默认值<br>4、修改了 PageMissEntry 的默认值 | 彭洪  |
| 2013-01-25 | V2. 03 | 1、删除 FM/RDS 的描述   | 彭洪  |

### 5.1 Features

- (1) Full synchronous design with operation clock rate up to 50MHz.
- (2) It is accessible for all the ram blocks through DMA0/1/2/3.
- (3) The page miss control mechanism can support 8 different pages at the same time.
- (4) Seamless clock switching is supported.
- (5) Off Mode for (RAM Retention): 1  $\mu$ A

### 5.2 Function Description

#### 5.2.1 Memory controller overview

GL5115 使用 dolphin 的 80251 内核，没有 ICACHE 和 DCACHE。大部分 RAM 和 ROM 都是通过 CPU 的程序和数据总线直接访问。其访问 memory 可以用下图表示：

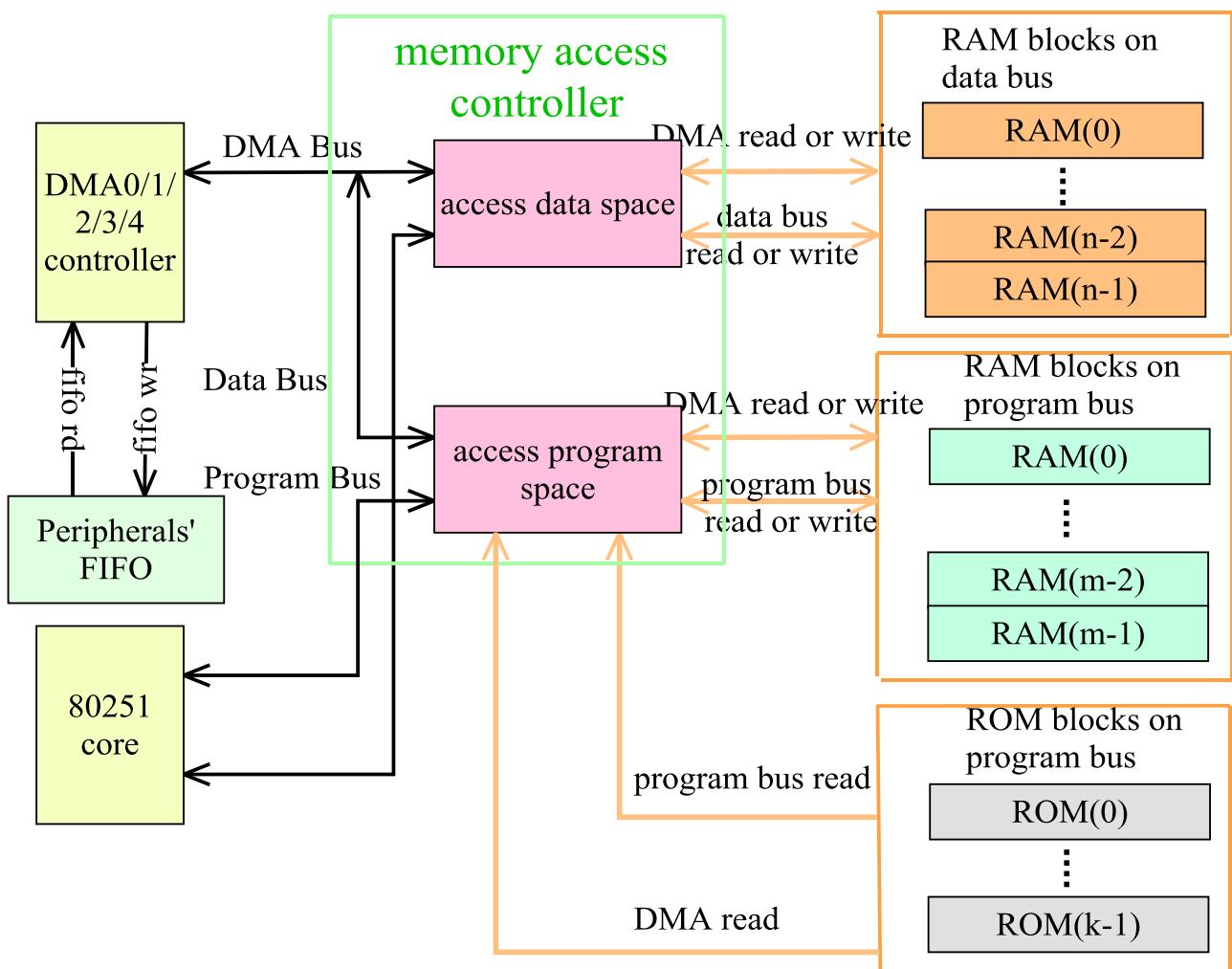


Figure 5-1 GL5115 memory controller overview

注:

- (1) ROM 只能通过 CPU 的程序总线访问 (read only)，不能通过 DMA 总线访问
- (2) DMA 总线 (以及控制器) 始终与 MCU 总线同频率、同相 (也就是说来自于 MCU clock)
- (3) DMA 控制器在同一时刻只能处理一个 DMA 通道的 DMA 请求。

## 5.2.2 Memory access rule

Memory block 要么被放在程序空间，要么被放在数据空间，因此同一时刻不会出现 MCU 的 program bus 和 data bus 同时访问。下面给出几种常见的情况，来描述 memory 的访问优先级。

### 5.2.2.1 MCU/DMA access different memory

这种情况下，系统的效率最高。如下例中，MCU 的程序或数据总线访问 RAM2。DMA 的访问 RAM1。这种情况下各信号完全独立。

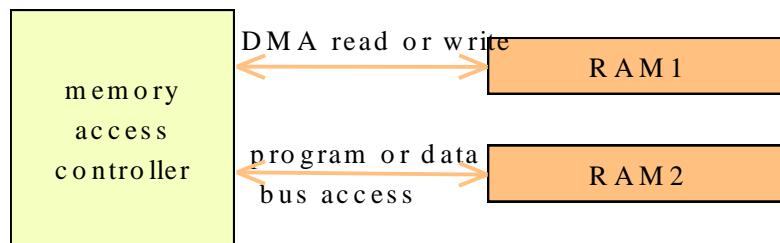


Figure 5-2 MCU/DMA 同时访问不同 RAM

注:

- (1) 此时的指令读和数据读均会在对应地址的读写命令之后的 1cycle 后获得。
- (2) DMA 控制器内部有寄存器保存 dma\_read\_bus 中读出的数据, 再从该寄存器写到 dma\_write\_bus 上, 这样做增加了 1 个 cycle 的读写 latency, 但是减少 DMA 读写的 loading, 有利于提高时钟频率。

### 5.2.2.2 MCU/DMA read access same memory

MCU 访问 RAM1, 同时也访问 RAM1。

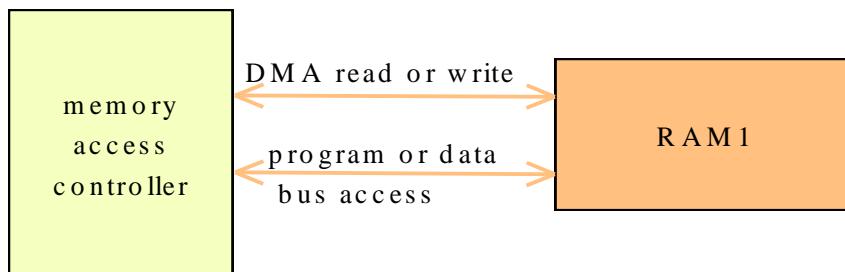


Figure 5-3 MCU/DMA 同时访问同一块 RAM

这种情况下, 首先保证 DMA 访问, MCU 被 hold 住, 直到 DMA 访问完成。在 JC3320 和 GL5115 中的 DMA 控制器做了一定简化, 将所有的 memory 到 memory 的传输都看成一块 memory 内部的传输。

### 5.2.2.3 Memory access rules conclusion

(2012) 两种 Memory 访问源的优先级:

DMA>MCU 数据或指令读写

2013、程序空间访问优先级:

Memory controller 将程序空间的所有 ram/rom 都看成一块 memory, 因此只需要检测 80251 发过来的程序空间的读写信号和 DMA 发过来的程序空间读写信号, 就能判断是否应该将 80251 stall。而不需要做地址比较和判断。

2014、数据空间访问优先级:

Memory controller 将数据空间的所有 ram 都看成多块 memory, 因此除了检测 80251 发过来的数据空间的读写信号和 DMA 发过来的数据空间读写信号, 还需要做地址比较, 才能判断是否应该将 80251 stall。

(4) 对于 DMA 内部来说, 只有高优先级的 DMA 通道, 并且在该通道的 DRQ 信号有效时才传输如下图, DMA 各通道的传输规则如图:

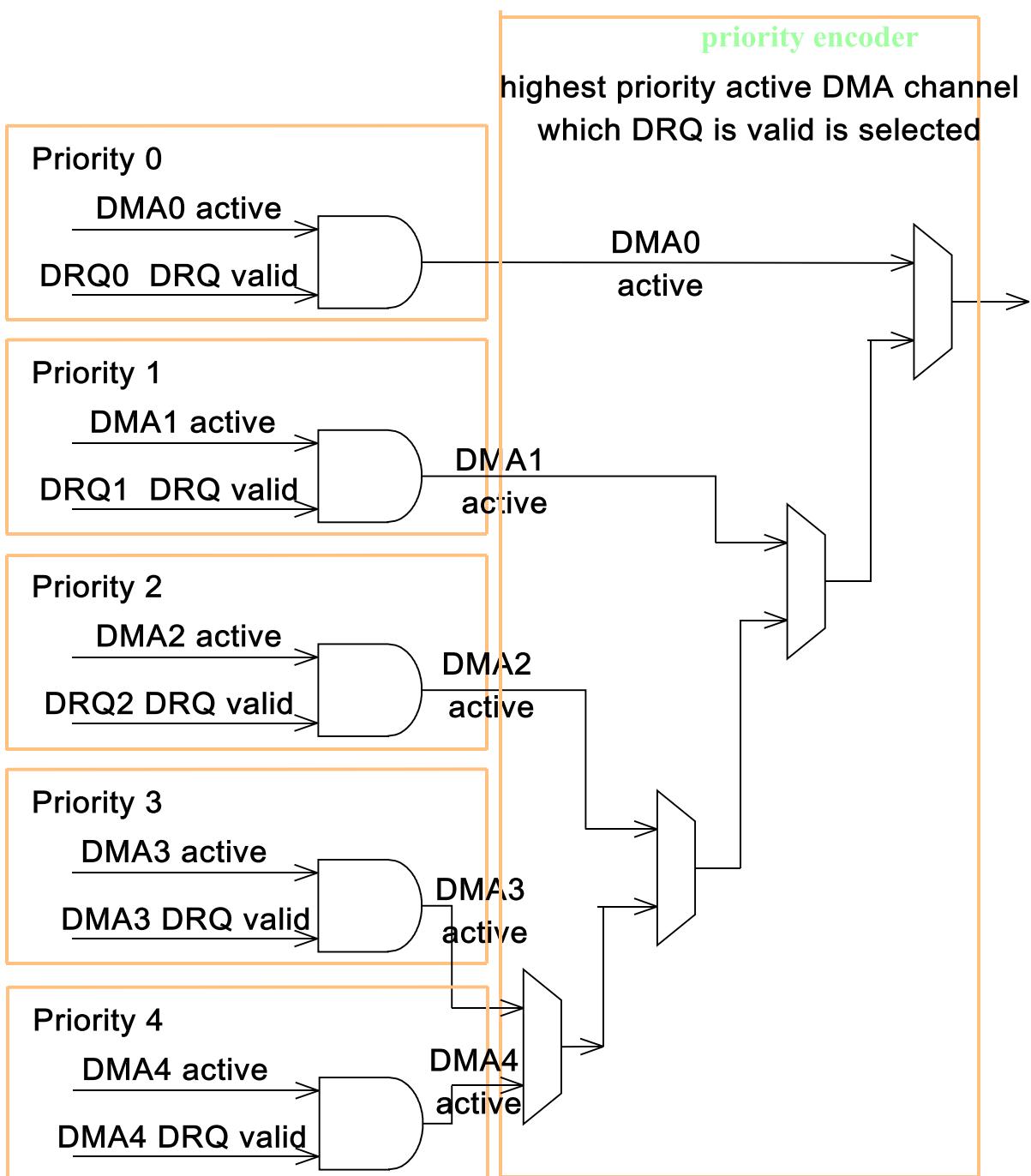


Figure 5-4 Priority Control of DMA01234

DMA 各个通道的优先级固定，DMA0 最高，DMA4 最低，但是都高于 CPU，如下：

| 通道<br>优先级 | 通道名称 |
|-----------|------|
| 0 (最高)    | DMA0 |
| 1         | DMA1 |
| 2         | DMA2 |
| 3         | DMA3 |
| 4         | DMA4 |
| 5 (最低)    | CPU  |

### 5.2.3 Program Address Redirection

The memory controller contains a simplified MMU to perform a translation to map the program addresses from virtual addresses to physical address. The MMU provides a Page Miss based Mapping Translation (PMMT) mechanism that is smaller and simpler than a full Translation Lookaside Buffer (TLB) found in other cores. The figure below shows how the PMMT is implemented in the memory controller.

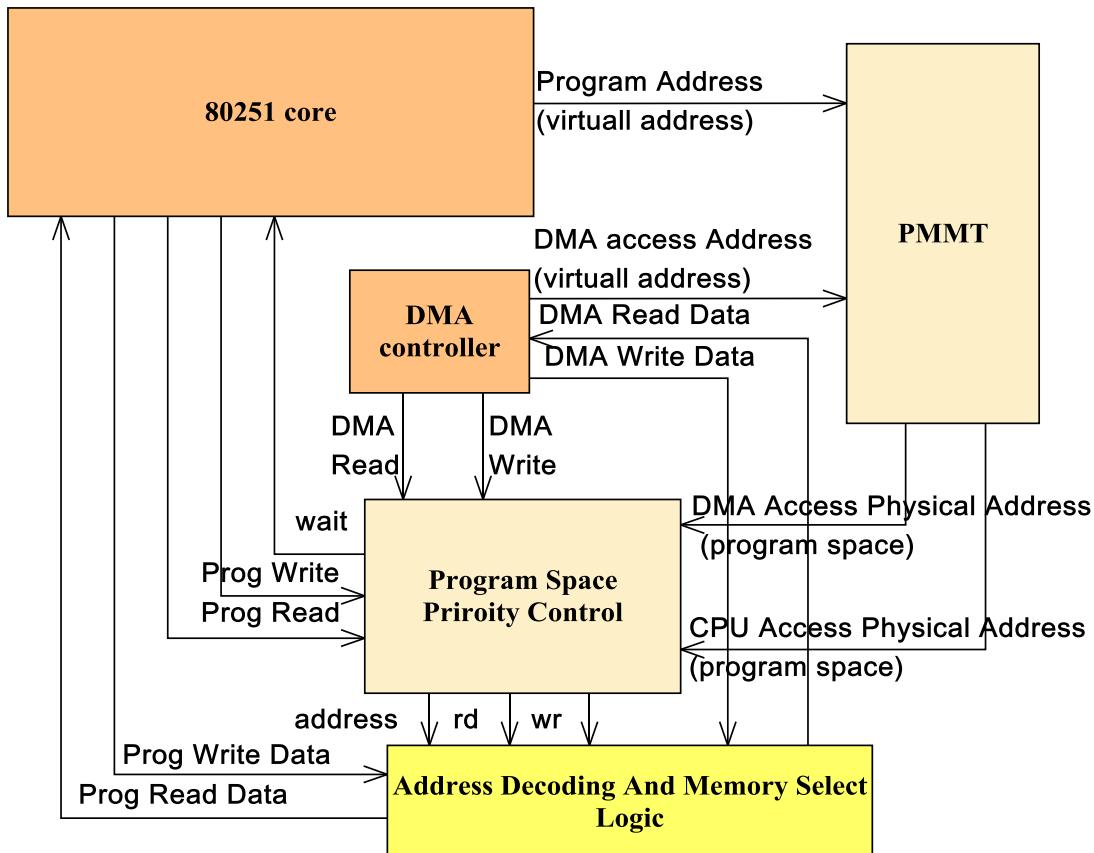


Figure 5-5 PMMT of Memory Controller

The program address redirection is based on the content of Page Miss Address register.

The PMMT performs a simple translation to map from virtual addresses to physical addresses. Both the 80251 core and DMA controller uses the same Program Space Address Mapping Logic. This mapping is shown in the figure below:

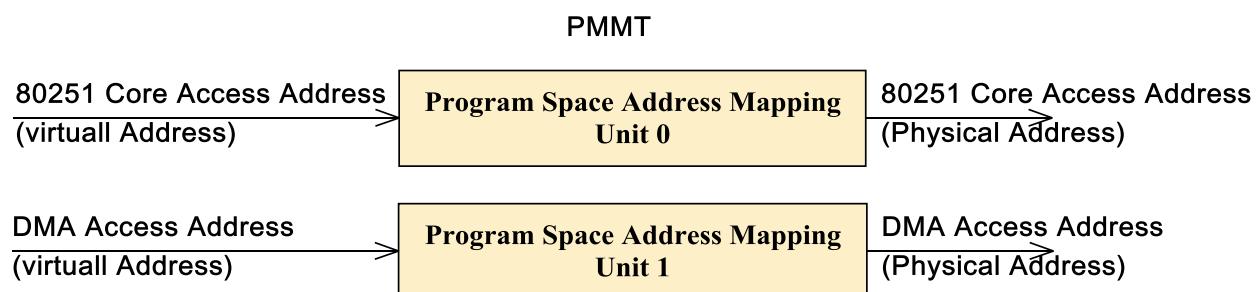


Figure 5-6 Program Space Address Mapping Unit of PMMT

The detail structure of each Program Space Address Mapping Unit is in the figure below:

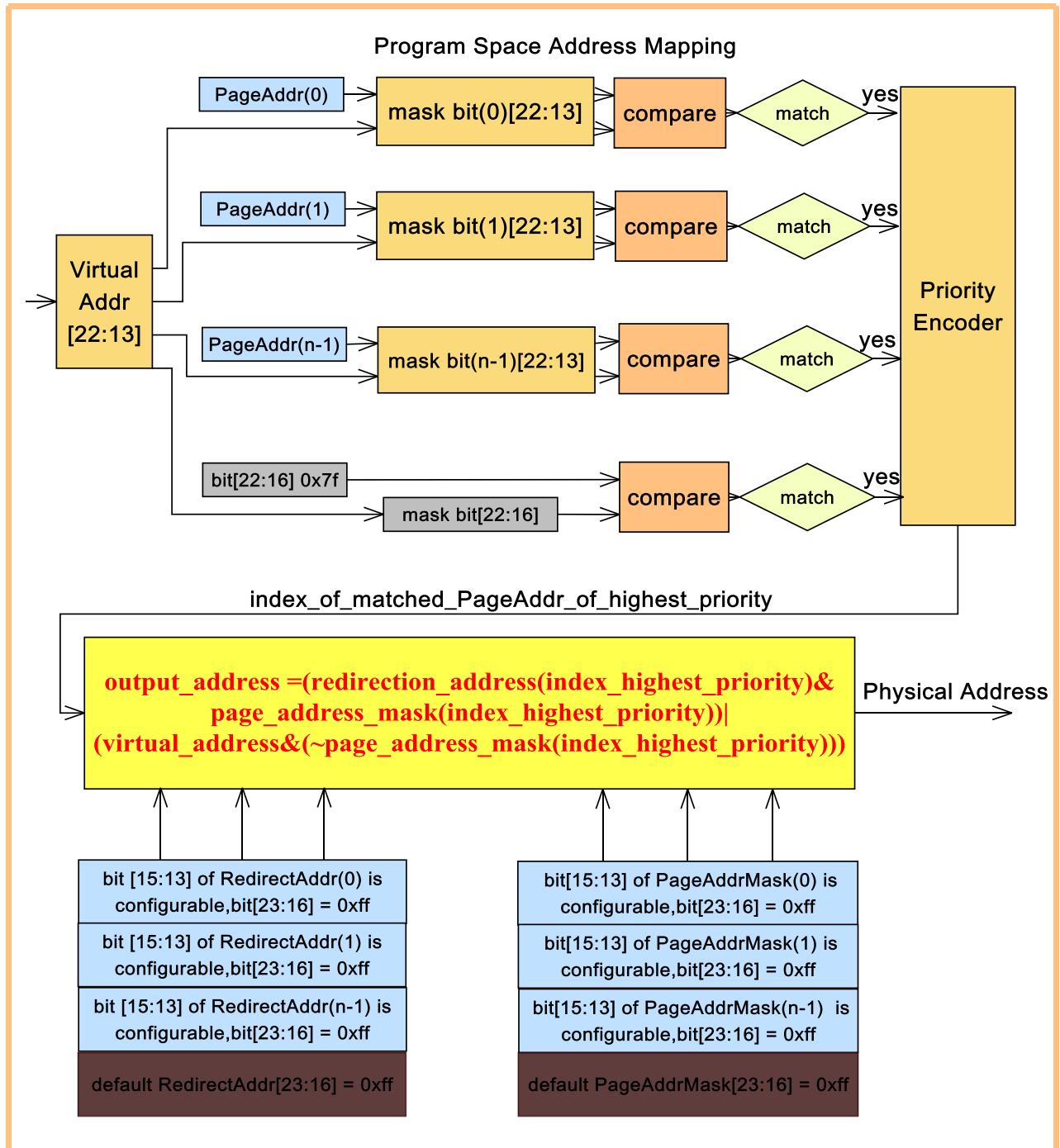


Figure 5-7 Program Space Address Mapping unit

Note:

- b) The error pending is set if one virtual address is mapped to more than two different physical addresses.  
The data read/write from program space is selected by the priority decoder of PMMT.
- c) The error pending is not set if no match of any page address registers nor the access address is from 0xff0000 to 0xffff. The CPU data read/write from program space is mapped to 0xff0000 to 0xffff.
- d) The error pending is set if no match of any page address registers nor the access address is from 0xff0000 to 0xffff. The DMA data read from program space is mapped to 0xff0000 to 0xffff.

The different virtual address can be mapped to one physical address page. But one virtual address can not be mapped to more than two different physical addresses. The error bit is set in this case. The physical address is

selected by priority encoder of the PMMT controller, which is demonstrated in the figure below:

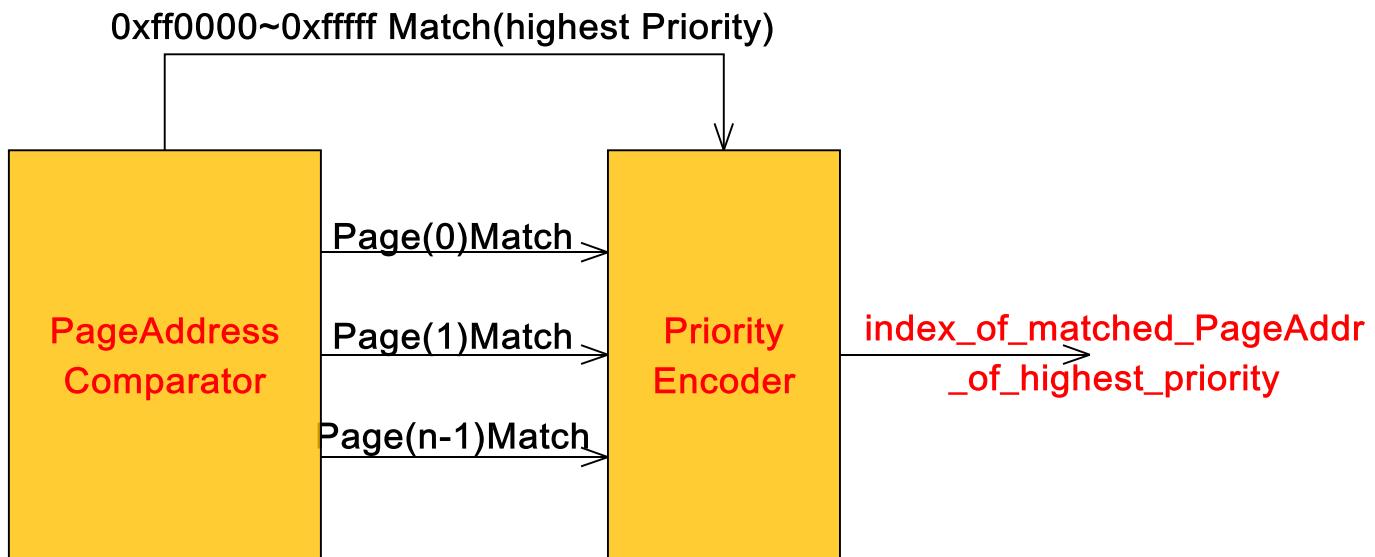


Figure 5-8 the output of priority encoder

#### 5.2.4 Page Miss Control Mechanism

80251 中无 TLB，为了实现换页功能，在 memory controller 中增加了一个简化的控制器，帮助实现换页功能。功能框图如下：

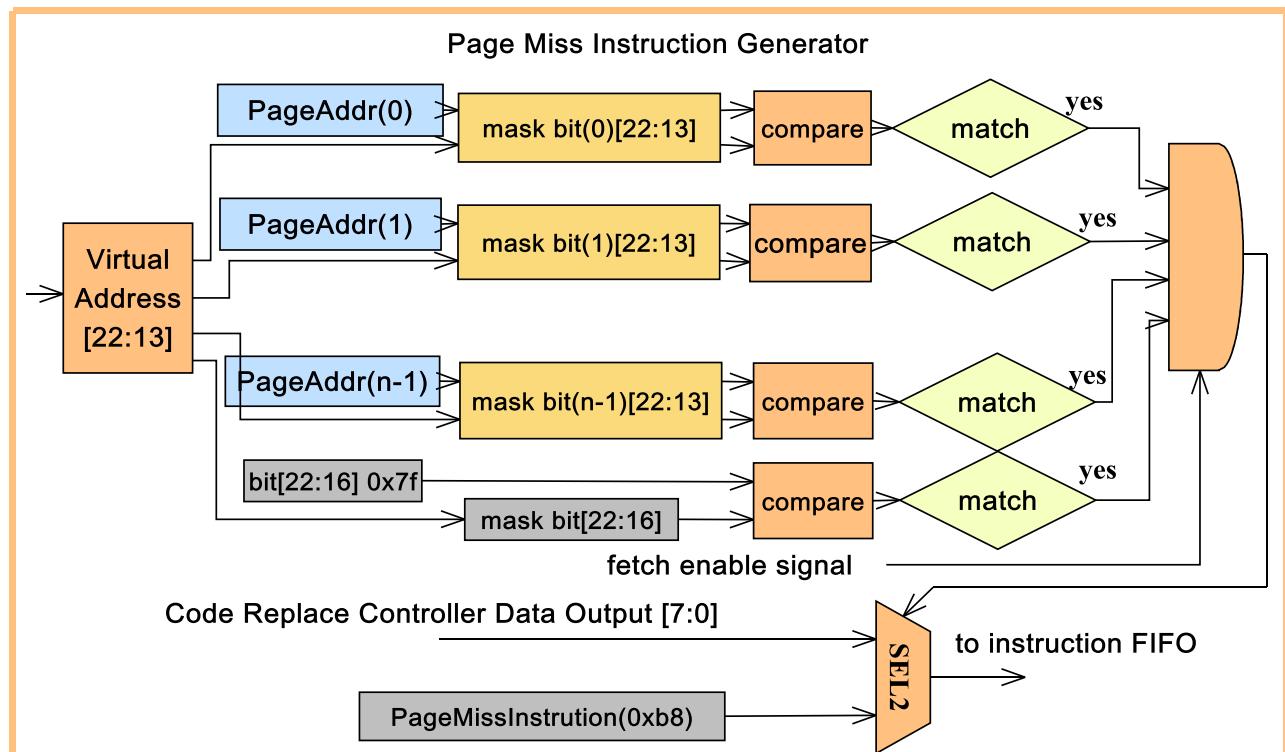


Figure 5-9 Pass Miss Instruction Generator

GL5115 的 Page Miss Instruction Generator 是这样解析 CPU 发过来的地址的：

- (1) 只有地址范围处于 0x80000~0xfffffff 的空间的取指令操作才有 Page Miss 的功能。属于这段地址范围的程序通过 Memory Controller 映射地址范围属于 0xff0000~0xfffffff 的 memory。如果是物理地址范围属于 0x80000~0xfffffff 的 data access (比如通过间接寻址从程序空间取数据)，会被 memory controller 重新映射到地址为 0xff0000~0xfffffff 的空间。
- (2) 地址范围属于 0xff0000~0xfffffff 的 memory 始终是被匹配的，访问这段空间不会产生 Page Miss 指令。

其中 MCU 访问程序空间时，地址的 bit22:16 没有实际的物理地址对应，因此页地址的为 64kBytes。PageMissInstruction 目前是 0xb8。这是我们为 MCU 增加了一条新指令，它的指令代码是 0Xb8 (如果为 binary mode 时则为 0xa5 0xb8)。80251 在运行程序时，指令首先被送入到 FIFO 中，保证取指令的时候能被取到完整的指令。这些指令是从程序空间读出的。通过分析程序空间的地址线，比较指令读取地址的高 8bit (PAGE 地址) 是否能与 page address 匹配。因此会遇到两种情况：

- A、找到了与之匹配的页地址，则直接从该页中读取程序代码
- B、未找到与之匹配的页地址，则向指令 FIFO 发送软件断点指令：如果当前是 Binary Mode 则发送 0XA5, 0Xb8，如果是 Source Mode 则发送 0Xb8 进入指令 FIFO

如果以后程序运行到换页指令 (TRAP2)，这条指令类似于 ECALL，它会跳转到 PageMissEntryH 和 PageMissEntryL 寄存器指向的地址 (还是在 FF 页面)，换页程序从该地址执行，执行完以后用 RET 返回。

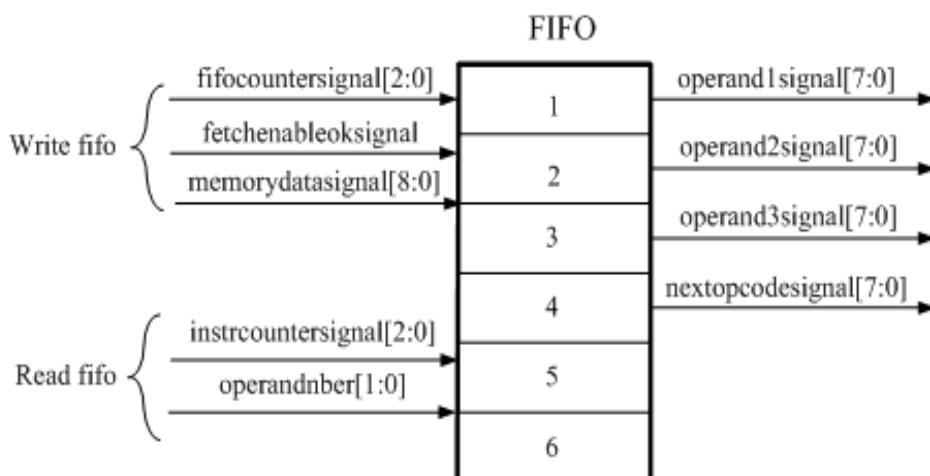


Figure 5-10 指令在被 predecode 前首先进入指令 FIFO 中

当程序检测到换页指令的时候，执行以下动作：

- A、旧 PC 被 push 到堆栈，堆栈指针+3，注意这个时候保护的 pc 值是换页指令执行完后的 pc 值。
- B、保存新页面的数值到 NewPageAddr 寄存器中（有 8bit）
- C、将 PC 值指向换页程序的入口地址（来自 PageMissEntry，默认为 0Xff :0083h）
- D、清空指令 FIFO

后面的操作将由软件完成（从 FF 页面的 PageMissEntry 开始，执行换页程序）：

- A、修改堆栈指针返回的地址，即新的返回地址=原返回地址-1。
- B、从 NAND FLASH、SD/MMC 卡，nor flash 中读出当前页面对应的程序代码，并复制到一块 RAM 中，修改该 RAM 块对应的页号（从 PageAddrx 获得新的页面号码）。
- C、程序返回（用 eret），堆栈指针-3

### 5.2.5 Hardware Code Replace Mechanism

功能框图如下：

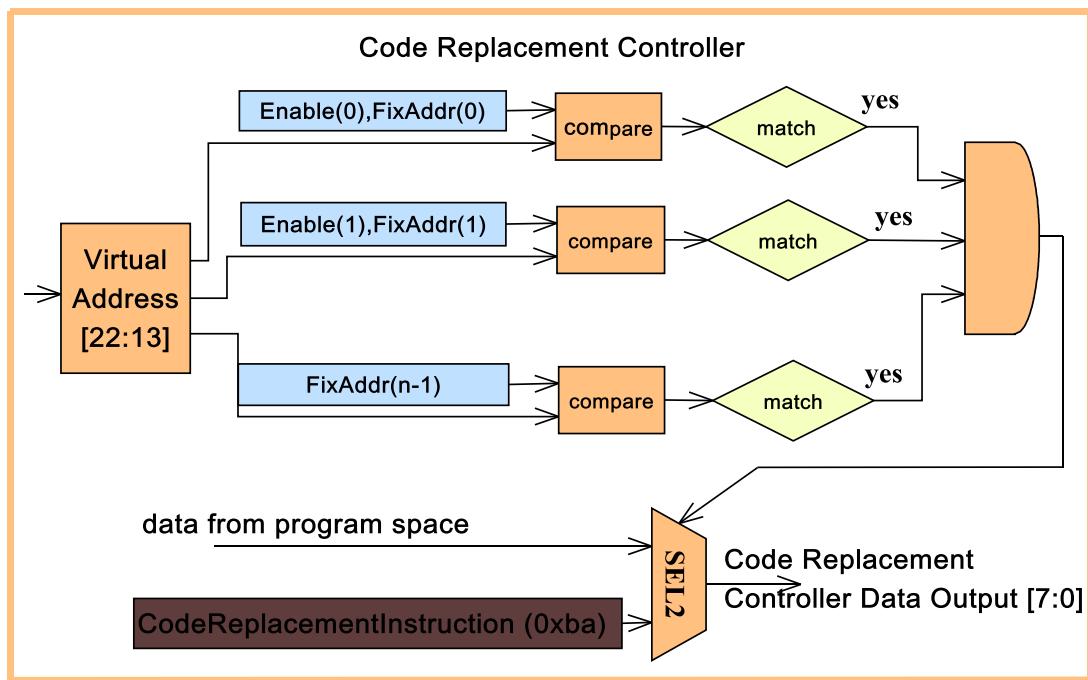


Figure 5-11 code replacement controller

其中蓝色的框是要配置的寄存器，也就是在修改代码前需要填入的值，它包括

- (1) 使能修改点，当虚地址 bit23 为 1 则使能。
- (2) 1~n 个修改点的位置（也就是修改点的程序计数器的值）

上述电路实现的功能为：

分析地址线，判断读信号使能时的当前地址与修改点的位置是否匹配，如果匹配，则将“备份”数据替换从 memory 读出的数据送给 CPU，如果不满足，则不替换从 memory 读到的数据，送给 CPU。

指令的修改的方式比 GL5106 的修改方式简单很多：

- (1) 地址采用虚地址匹配，要比较地址线的 bit23:0，其中 bit23 兼做使能位。
- (2) 不再区分是取指令还是取数据，这样初始化数据错误也能修改。
- (3) 不再区分单个指令还是多个指令的替换，一律使用 0xba（代码替换指令）。该指令类似于 ecall 指令，执行时首先将替换点的位置放入堆栈，再将 PC 值修改到指令替换服务程序的起始位置。如下例：

|          |             |
|----------|-------------|
| 地 址      | 指 令         |
| ⋮        |             |
| 0x81a2bc | ecall @ dr4 |
| ⋮        |             |

Figure 5-12 pre-replaced instruction

则需要将修改点的位置设置成 0x81a2bc，要替换的代码写成 trap3。CPU 运行到该地址时执行的就是 target 中的代码。

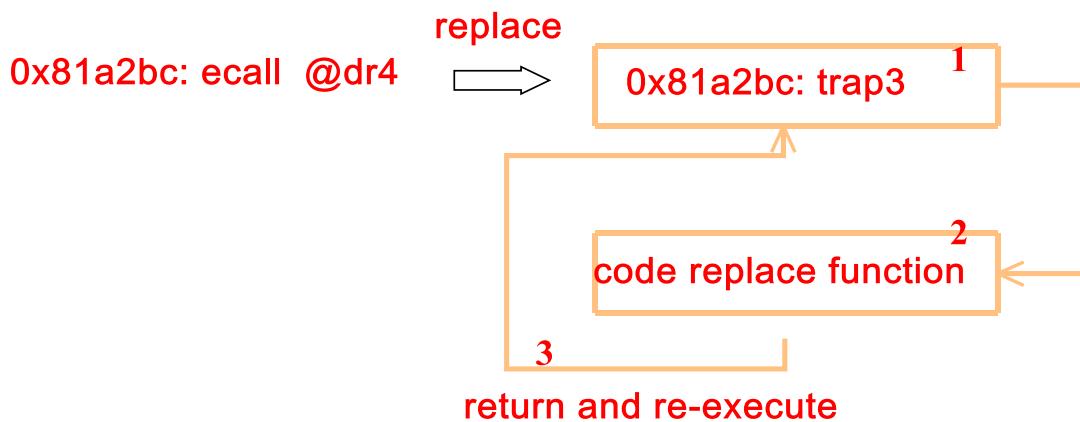


Figure 5-13 code replace and re-execute

图中的执行顺序为 1、2、3，粗体字描述。这样改动就非常灵活。

软硬件工作的流程如下：

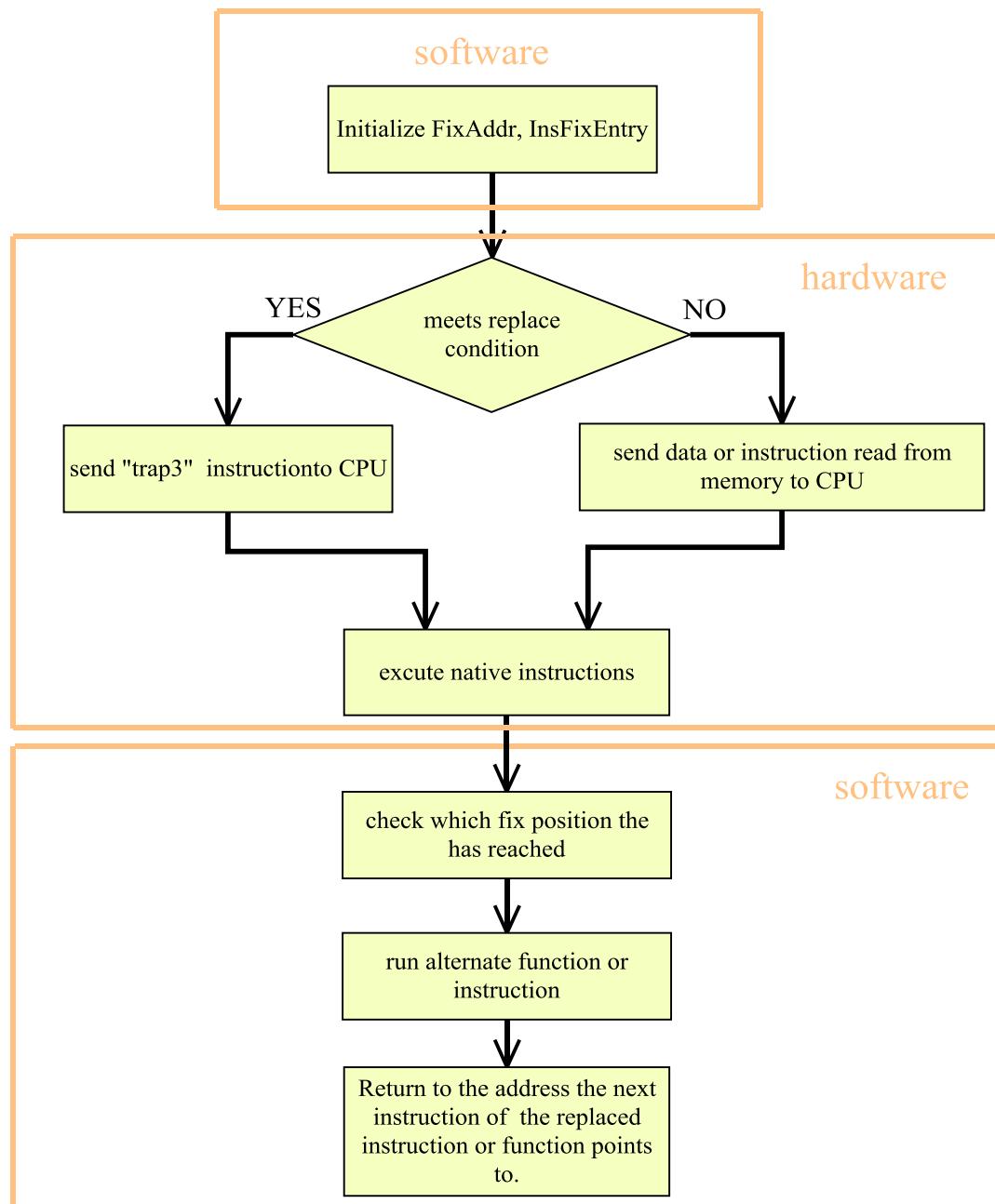


Figure 5-14 code replacement flow chart

将代码替换和换页管理的控制电路合并到一起的图如下：

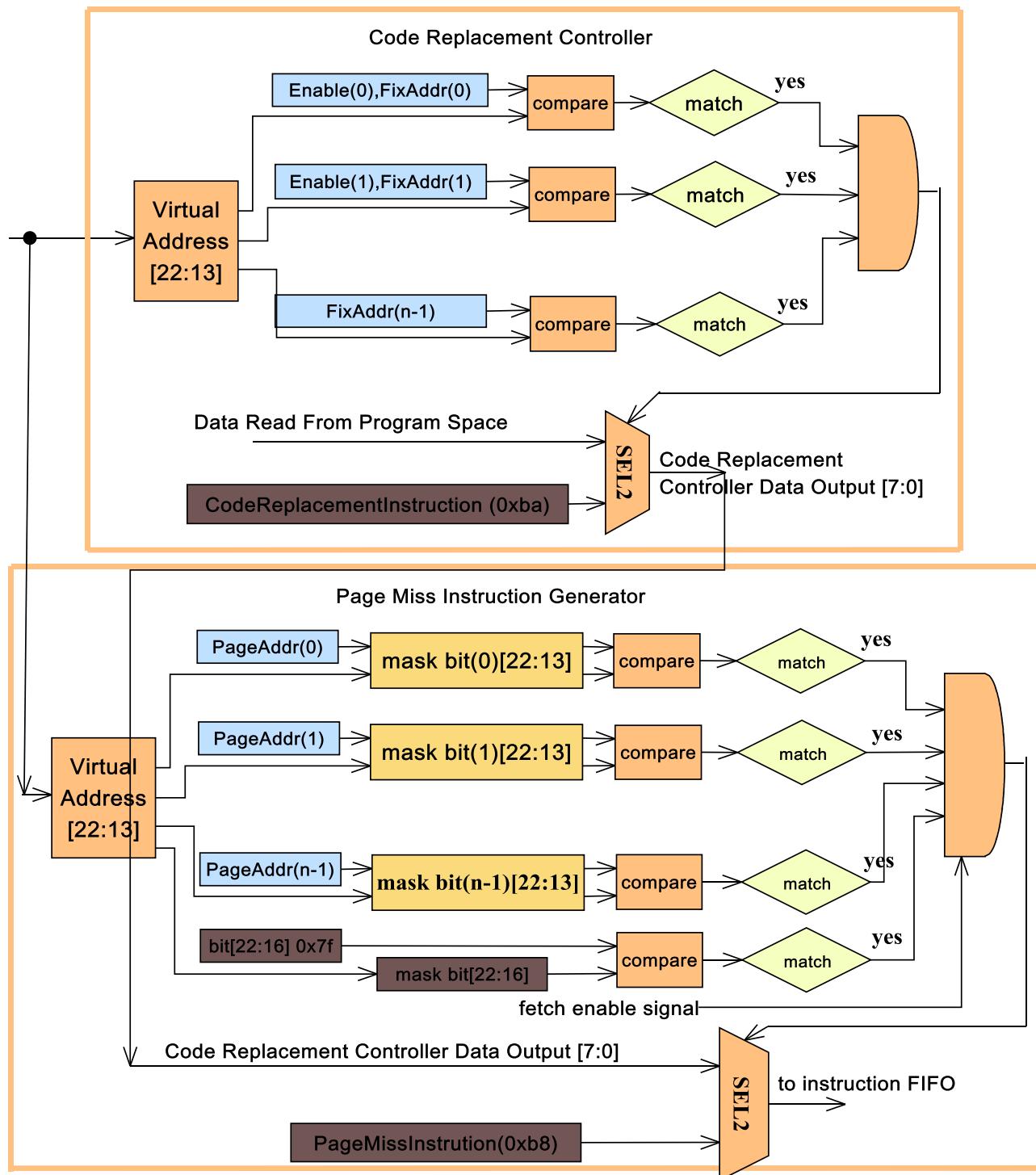


Figure 5-15 the cascade of code replacement controller and page switch controller

### 5.2.6 Content protection mechanism

The programmable code protection is set by security fuse. There are two security fuse bits for different levels of code protection. These control bits are in the ffuse controller specification. See description of Efuse\_Img register.

### 5.2.6.1 bit jtag\_disable

The input signal tck and tms of jtag interface is disable by setting this bit to '1'

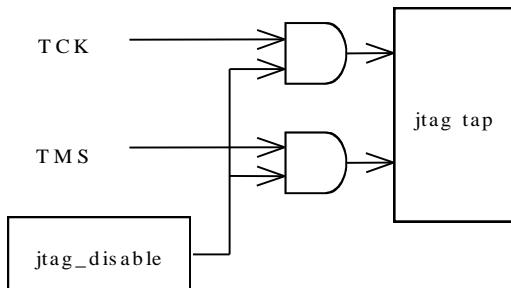


Figure 5-16 jtag\_disable = '0' bit to enable the input signal 'tck' & 'tms'

The JTAG is activated by the image register of EFUSE0.

| Efuse bit name | Description  | Efuse image regists state                             |
|----------------|--|---|
| jtag_disable   | 0:The EFUSE read and TCK/TMS of JTAG is enable if the corresponding register bit of EFUSE is clear.<br>1: The EFUSE read and TCK/TMS of JTAG is enable if the corresponding register bit of EFUSE is set | Same as the bit 6 of image register of Efuse0(0x05a6) |

### 5.2.7 BUS extension

The GL5115 can extend its bus though SPI-BOOT controller. The address between 0xff9c00 to 0xfffffff ROM of CPU can be mapped to internal 2 blocks of ROM as separated memory blocks or be mapped to SPI-BOOT controller as a block of ROM. The SPI-BOOT or internal ROM mapping is controlled by ROM0\_SEL, ROM1\_SEL which are generated by SPI-BOOT controller.

The pseudo-code below shows how these ROMx\_SEL signals are used by controller memory:

```

if(ROM0_SEL==1) // default value is 0
{
    the address from ff9c00h~ffbfhh is mapped to SPI-BOOT controller (SPI NOR flash internal address is
    from 0x0000 to 0x3fff)
}
else
{
    the address from ff9c00h ~ ffbffh is mapped to ROM0
}

if(ROM1_SEL==1) // default value is 0
{
    the address from ffc000h~fffffh is mapped to SPI-BOOT controller (SPI NOR flash internal address is
    from 0x4000 to 0x7fff)
}
    
```

```

else
{
    the address from ffc000h~fffffh is mapped to ROM1
}

```

## 5.2.8 Signal List

This section describes the signal interface of the memory controller of GL5115.

The pin direction key for the signal descriptions is shown in table below.

| Dir | Description  |
|-----|--|
| I   | <b>Input to the memory controller sampled on the rising edge of the appropriate CLK signal.</b>  |
| O   | <b>Output of the memory controller, unless otherwise noted, driven at the rising edge of the appropriate CLK signal..</b>                                    |
| A   | <b>Asynchronous inputs that are synchronized by the core.</b>  |
| S   | <b>Static input to the M4K core. These signals are normally tied to either power or ground and should not change state while SI_ColdReset is deasserted.</b> |

Table 1 Memory Controller signal type

The memory controller signals are listed in table below. Note that the signals are grouped by logical function, not by expected physical location.

| Signal             | Width | I/O | Clock Domain | Description               |
|--------------------|-------|-----|--------------|---------------------------|
| scan_mode          | 1     | I   |              | scan chain enable         |
| sfr_clk            | 1     | I   | Clk_cpu      | Sfr bus clock             |
| brom_clk           | 1     | I   | Clk_cpu      | brom clock                |
| uram_clk           | 1     | I   | Clk_cpu      | uram clock                |
| muram1_clk         | 1     | I   | Clk_cpu      | muram1 clock              |
| dram251_clk        | 1     | I   | Clk_cpu      | dram251 clock             |
| muram2_clk         | 1     | I   | Clk_cpu      | muram2 clock              |
| pcmram_clk         | 1     | I   | Clk_cpu      | pcmram clock              |
| pram_clk           | 1     | I   | Clk_cpu      | pram clock                |
| fir_cs_ram_clk     | 1     | I   | Clk_cpu      | fir_cs_ram clock          |
| fir_aa_ram_clk     | 1     | I   | Clk_cpu      | fir_aa_ram clock          |
| fir_mpx_ram_lm_clk | 1     | I   | Clk_cpu      | fir_mpx_ram_lm clock      |
| fir_mpx_ram_h_clk  | 1     | I   | Clk_cpu      | fir_mpx_ram_h clock       |
| j0_cpuclk_sel      | 1     | I   | Clk_cpu      | jram0 cpu clock select    |
| j0_veclk_sel       | 1     | I   | Clk_cpu      | jram0 video clock select  |
| j1_cpuclk_sel      | 1     | I   | Clk_cpu      | jram1 cpu clock select    |
| j1_veclk_sel       | 1     | I   | Clk_cpu      | jram1 video clock select  |
| j23_cpuclk_sel     | 1     | I   | Clk_cpu      | jram23 cpu clock select   |
| j23_veclk_sel      | 1     | I   | Clk_cpu      | jram23 video clock select |

|                  |    |   |         |                               |
|------------------|----|---|---------|-------------------------------|
| pcm_cpuclk_sel   | 1  | I | Clk_cpu | Pcmram cpu clock select       |
| pcm_aeclk_sel    | 1  | I | Clk_cpu | Pcmram audio clock select     |
| mu1h_cpuclk_sel  | 1  | I | Clk_cpu | Muram1h cpu clock select      |
| mu1h_aeclk_sel   | 1  | I | Clk_cpu | Muram1h audio clock select    |
| mu1m_cpuclk_sel  | 1  | I | Clk_cpu | Muram1m cpu clock select      |
| mu1m_aeclk_sel   | 1  | I | Clk_cpu | Muram1m audio clock select    |
| mu1m_veclk_sel   | 1  | I | Clk_cpu | Muram1m video clock select    |
| mu1l_cpuclk_sel  | 1  | I | Clk_cpu | Muram1l cpu clock select      |
| mu1l_aeclk_sel   | 1  | I | Clk_cpu | Muram1l audio clock select    |
| mu1l_veclk_sel   | 1  | I | Clk_cpu | Muram1l video clock select    |
| uram_cpuclk_sel  | 1  | I | Clk_cpu | Uram cpu clock select         |
| uram_usbclk_sel  | 1  | I | Clk_cpu | Uram usb clock select         |
| dramb_cpuclk_sel | 1  | I | Clk_cpu | Dramb cpu clock select        |
| dramb_veclk_sel  | 1  | I | Clk_cpu | Dramb video clock select      |
| rst_n            | 1  | I | Clk_cpu | Memory reset                  |
| boot_brom0       | 1  | i | Clk_cpu | Brom0 boot enable             |
| boot_brom1       | 1  | I | Clk_cpu | Brom1 boot enable             |
| rambusy          | 1  | O | Clk_cpu | Data ram busy                 |
| prgbusy          | 1  | O | Clk_cpu | program ram busy              |
| dma3wait         | 1  | O | Clk_cpu | Dma3 read hold                |
| prgaddr          | 23 | i | Clk_cpu | Program address               |
| p_d_sel          | 1  | i | Clk_cpu | Program or data memory maping |
| prgoe_n          | 1  | i | Clk_cpu | Program bus read enable       |
| prgwe_n          | 1  | i | Clk_cpu | Program bus write enable      |
| prgdatatocpu     | 8  | o | Clk_cpu | Program data output to cpu    |
| dramaddr         | 23 | i | Clk_cpu | Data address                  |
| dramoe_n         | 1  | i | Clk_cpu | data bus read enable          |
| dramwe_n         | 1  | i | Clk_cpu | data bus write enable         |
| dramdatatocpu    | 8  | o | Clk_cpu | data output to cpu            |
| dramdatafromcpu  | 8  | i | Clk_cpu | data get from cpu             |
| dma012addrs      | 17 | i | Clk_cpu | Dma012 read bus address       |
| dma012oe_n       | 1  | i | Clk_cpu | Dma012 read enable            |
| datatodma012     | 8  | o | Clk_cpu | Dma012 data output            |
| dma012addrd      | 17 | i | Clk_cpu | Dma012 write bus address      |
| dma012we_n       | 1  | i | Clk_cpu | Dma012 write enable           |
| datafromdma012   | 8  | i | Clk_cpu | Dma012 data input             |
| dma3oe_n         | 1  | i | Clk_cpu | Lcd read enable               |
| dma3addr         | 16 | i | Clk_cpu | Lcd dma address               |
| datatodma3       | 8  | o | Clk_cpu | Lcd dma data output           |
| aumu2en          | 1  | i | Clk_cpu | Audio rw muram2 enable        |
| aumu2we_n        | 1  | i | Clk_cpu | Audio write muram2 enable     |
| aumu2oe_n        | 1  | i | Clk_cpu | Audio read muram2 enable      |

|               |    |   |         |                           |
|---------------|----|---|---------|---------------------------|
| aumu2addr     | 12 | i | Clk_cpu | Audio rw muram2 address   |
| datafromaumu2 | 8  | o | Clk_cpu | Audio data output         |
| datatoaumu2   | 8  | i | Clk_cpu | Audio data input          |
| aupcen        | 1  | i | Clk_cpu | Audio rw muram2 enable    |
| aupcwe_n      | 1  | i | Clk_cpu | Audio write pcmram enable |
| aupcoe_n      | 1  | i | Clk_cpu | Audio read pcmram enable  |
| aupcaddr      | 12 | i | Clk_cpu | Audio rw pcmram address   |
| datafromaupc  | 16 | o | Clk_cpu | Audio data output         |
| datatoaupc    | 16 | i | Clk_cpu | Audio data input          |
| aumuhen       | 1  | i | Clk_cpu | Audio rw muramh enable    |
| aumumen       | 1  | i | Clk_cpu | Audio rw muramm enable    |
| aumulen       | 1  | i | Clk_cpu | Audio rw muraml enable    |
| aumu1we_n     | 1  | i | Clk_cpu | Audio write muram1 enable |
| aumu1oe_n     | 1  | i | Clk_cpu | Audio read muram1 enable  |
| aumu1addr     | 13 | i | Clk_cpu | Audio rw muram1 address   |
| datafromaumu1 | 24 | o | Clk_cpu | Audio data output         |
| datatoaumu1   | 24 | i | Clk_cpu | Audio data input          |
| vemu1en       | 1  | i | Clk_cpu | video rw muram1 enable    |
| vemu1we_n     | 1  | i | Clk_cpu | Video write muram1 enable |
| vemu1addr     | 15 | i | Clk_cpu | Video rw muram1 address   |
| datatovemu1   | 8  | i | Clk_cpu | Video data input          |
| Vej0en        | 1  | i | Clk_cpu | video rw jram0 enable     |
| Vej0oe_n      | 1  | i | Clk_cpu | Video read jram0 enable   |
| Vej0addr      | 7  | i | Clk_cpu | Video rw jram0 address    |
| Datafromvej0  | 32 | o | Clk_cpu | Video data output         |
| Vej1en        | 1  | i | Clk_cpu | video rw jram1 enable     |
| Vej1oe_n      | 1  | i | Clk_cpu | Video read jram1 enable   |
| Vej1addr      | 7  | i | Clk_cpu | Video rw jram1 address    |
| Datafromvej1  | 32 | o | Clk_cpu | Video data output         |
| Vej2en        | 1  | i | Clk_cpu | video rw jram3 enable     |
| Vej2oe_n      | 1  | i | Clk_cpu | Video read jram3 enable   |
| Vej2addr      | 9  | i | Clk_cpu | Video rw jram3 address    |
| Datafromvej2  | 8  | o | Clk_cpu | Video data output         |
| Vej3we_n      | 1  | i | Clk_cpu | Video write jram4 enable  |
| Vej3en        | 1  | i | Clk_cpu | video rw jram4 enable     |
| Vej3oe_n      | 1  | i | Clk_cpu | Video read jram4 enable   |
| Vej3addr      | 8  | i | Clk_cpu | Video rw jram4 address    |
| Datafromvej3  | 8  | o | Clk_cpu | Video data output         |
| usburen       | 1  | i | Clk_cpu | Usb rw uram enable        |
| usburwe_n     | 1  | i | Clk_cpu | Usb write uram enable     |
| usburoe_n     | 1  | i | Clk_cpu | Usb read uram enable      |
| usbbyten      | 4  | i | Clk_cpu | Usb byte enable           |

|                   |    |   |         |   |
|-------------------|----|---|---------|---|
| usburaddr         | 10 | i | Clk_cpu | Usb rw uram address                         |
| datafromusbur     | 32 | o | Clk_cpu | Usb data output                             |
| datatousbur       | 32 | i | Clk_cpu | Usb data input                              |
| j_outbuffer_write | 1  | I | Clk_cpu | Connected to output buffer write_en port    |
| j_outbuffer_addr  | 16 | I | Clk_cpu | Connected to output buffer address port     |
| j_outbuffer_wdata | 8  | I | Clk_cpu | JEPG Controller Write Data to output buffer |
| vemu2oe           | 1  | i | Clk_cpu | video rw muram2 enable                      |
| vemu2we           | 1  | i | Clk_cpu | video write muram2                          |
| vemu2addr         | 12 | i | Clk_cpu | Video rw muram2 address                     |
| datatovemu2       | 8  | i | Clk_cpu | Video data input                            |
| datafromvemu2     | 8  | o | Clk_cpu | Video data output                           |
| bist_en1~2        | 8  | i | Clk_cpu | Bist enable                                 |
| bist_fin1~2       | 8  | o | Clk_cpu | Bist operation finish                       |
| bist_info1~2      | 8  | o | Clk_cpu | Bist fail or success                        |

Table 2 Memory Controller signal description

如果 CPU 访问 MURAM1，但是 MURAM1 (clock) 被分配给 AUIP，读出的数值为 0，CPU 写 MURAM1 的地址范围也不能把数据写进去。同样 MURAM3、JRAM0 等也是相同的访问规则。

## 5.3 Module Description

### 5.3.1 Memory Organization

GL5115 中需要以下 memory:

| Memory 块名称 | 组织形式           | 要字节使能? | 访问源     | 速度         |
|------------|----------------|--------|---------|------------|
| ROM0       | 9k*8bit        | 不要     | CPU/DMA | 100MHz     |
| ROM1       | 16k*8bit       | 不要     | CPU/DMA | 100MHz     |
| MUROM1     | 4096*24bit(??) | 不要     | AUIP    | 24MHz      |
| MUROM2     | 7104*16bit     | 不要     | AUIP    | 24MHz      |
| MUROM3     | 2880*24bit     | 不要     | AUIP    | 24MHz      |
| ASRCROM    | 209*16bit      | 不要     | ASRC    | 100MHz     |
| DACRAM     | 96*16bit       | 不要     | DAC     | 24.576 MHz |
| ADCRAM0    | 32*16bit       | 不要     | ADC     | 24.576 MHz |
| ADCRAM1    | 44*31bit       | 不要     | ADC     | 24.576 MHz |
| PRAM251    | 22k*8bit       | 不要     | CPU/DMA | 100MHz     |
| DRAM251    | 10.5k*8bit     | 不要     | CPU/DMA | 100MHz     |
| PCMRAM     | 1k*32bit       | 要      | CPU/DMA | 100MHz     |
|            |                |        | AUIP    | 24MHz      |
|            |                |        | FM/RDS  | 38.912MHz  |
|            |                |        | USB     | 60MHz      |
| MURAM1     | 5k*24bit       | 要      | CPU/DMA | 100MHz     |
|            |                |        | FM/RDS  | 38.912MHz  |
|            |                |        | AUIP    | 24MHz      |

|             |             |   |              |           |
|-------------|-------------|---|--------------|-----------|
| FIR_CS_RAM  | 256*24bit   | 要 | CPU/DMA      | 100MHz    |
|             |             |   | DAC CH1 ASRC | 50MHz     |
|             |             |   | FM/RDS       | 38.912MHz |
|             |             |   | AUIP         | 24 MHz    |
| FIR_AA_RAM  | 64*24bit    | 要 | CPU/DMA      | 100MHz    |
|             |             |   | ADC ASRC     | 50MHz     |
|             |             |   | FM/RDS       | 38.912MHz |
|             |             |   | AUIP         | 24 MHz    |
| FIR_MPX_RAM | 1.75k*24bit | 要 | CPU/DMA      | 100MHz    |
|             |             |   | FM/RDS       | 38.912MHz |
|             |             |   | AUIP         | 24 MHz    |
| FIR_RDS_RAM | 96*32bit    | 要 | CPU/DMA      | 100MHz    |
|             |             |   | DAC CH0 ASRC | 50MHz     |
|             |             |   | FM/RDS       | 38.912MHz |
|             |             |   | USB          | 60MHz     |
| URAM        | 128*32bit   | 要 | CPU/DMA      | 100MHz    |
|             |             |   | URAM         | 60MHz     |

Table 3 Memory Organization

### 5.3.2 Memory Allocation

#### 5.3.2.1 Default memory mapping

为了便于描述不同的硬件模块访问的用不同颜色表示:

- (1) 蓝色: FM/RDS 解调器, 在一些场景下会被分配给 AUIP 或 ADC 用
- (2) 黄色: AUIP, 在一些场景下会被分配给 DAC 用
- (3) 橙色: MCU 独占的 RAM
- (4) 深灰: MCU 独占的 ROM

所有的 RAM/ROM 在默认情况下都可以通过 DMA01234 访问 (ROM 只读), GL5115 默认状态的 memory 映射图如下:

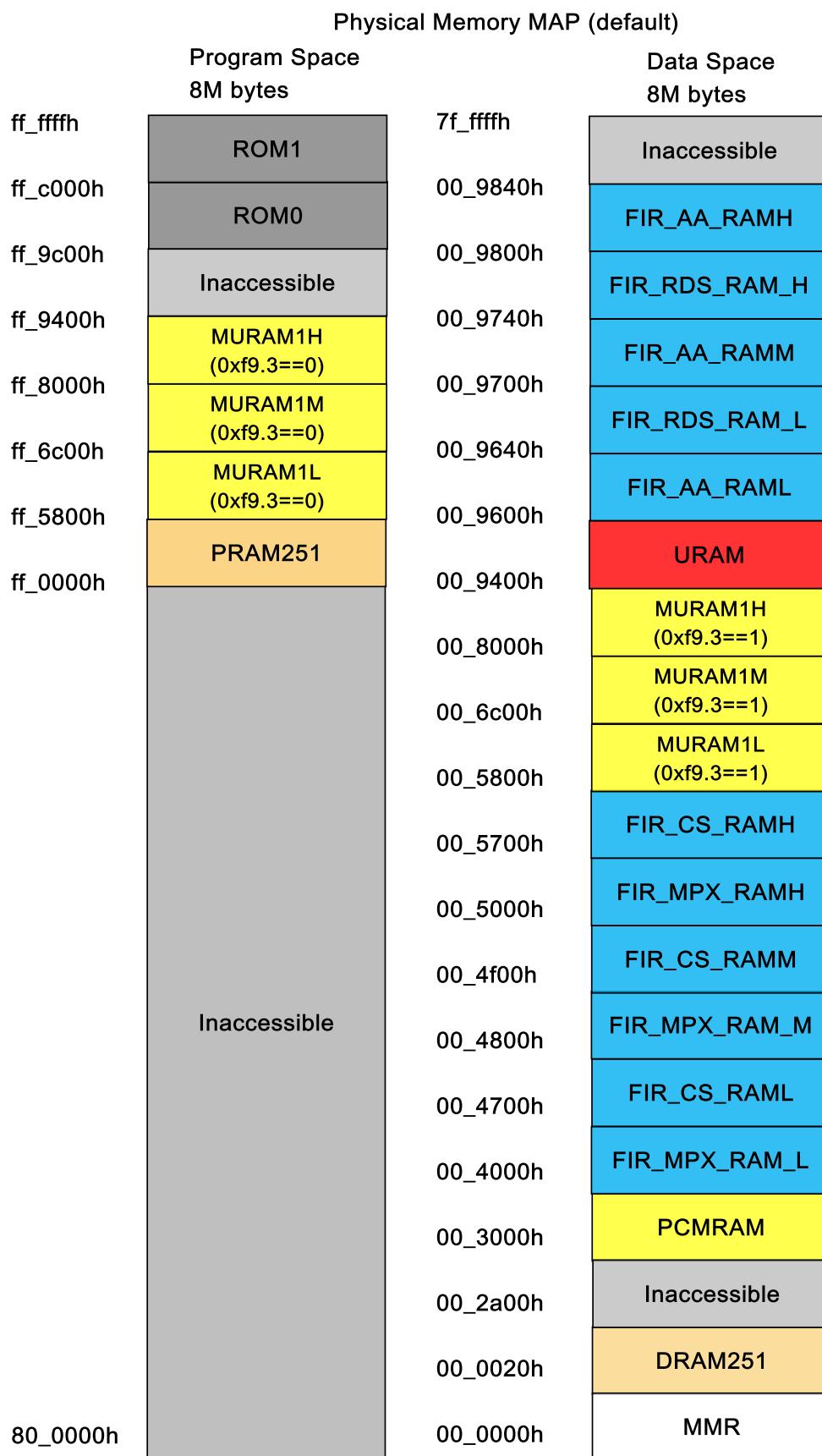


Figure 5-17 default memory mapping

Note:

The first 1k bytes of MURAM1 is overlapped with the last 1k bytes of pram251. Only last 4.75kbytes of

MURAM1L is accessible by CPU while  $0xf9.3 == 0$ .

### 5.3.2.2 MP3 decode Memory mapping

MP3 解码时的 GL5115 的 memory 映射图为：

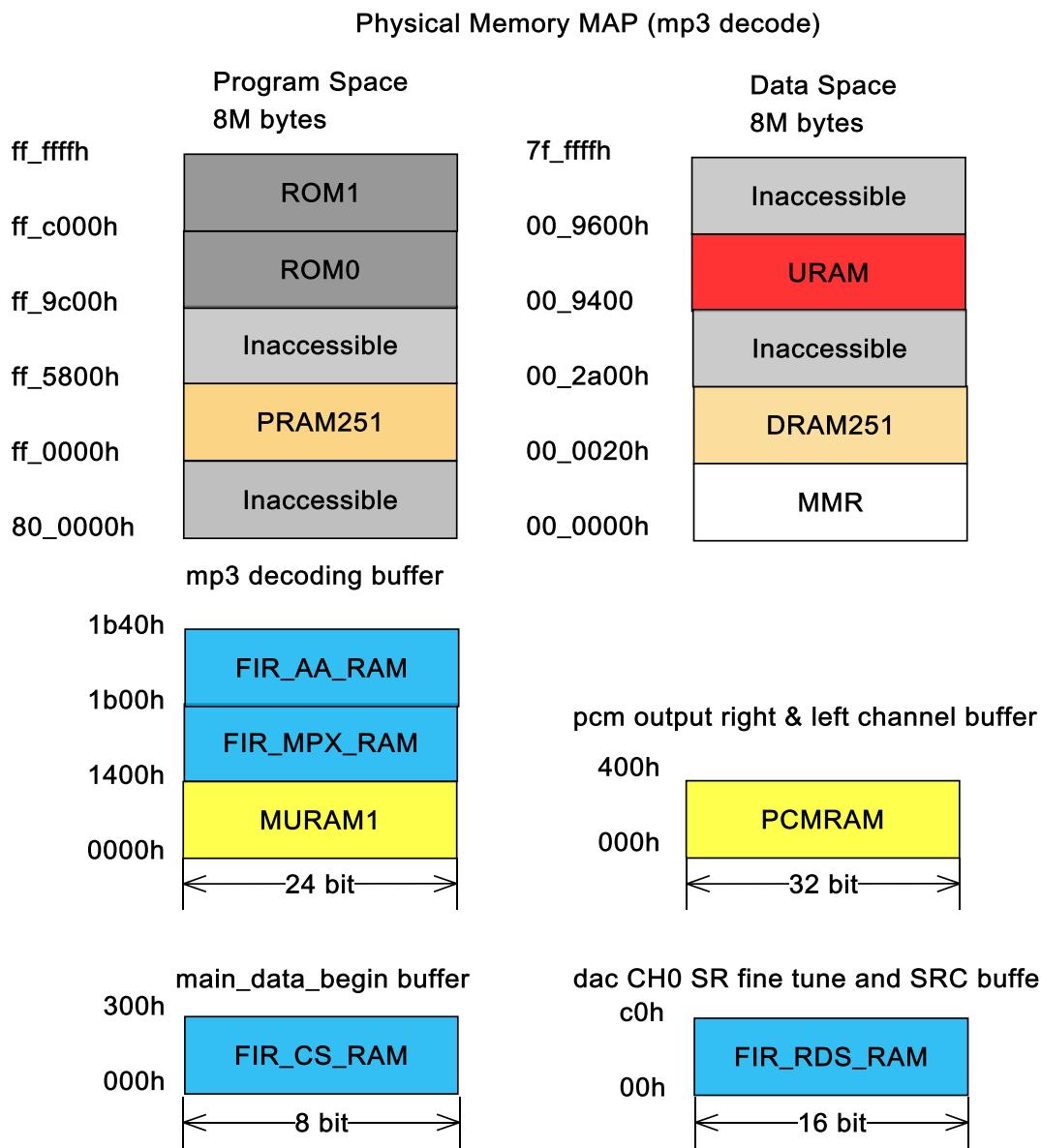


Figure 5-2 mp3 decode memory mapping

(2012) 如果此时还要语音播报，还需要 **FIR\_RDS\_RAM** 用于 **CHO** 的 **DAC** 采样率转换。

### 5.3.2.3 WMA/WAV decode Memory mapping

WMA/WAV 解码时的 GL5115 的 memory 映射图为：

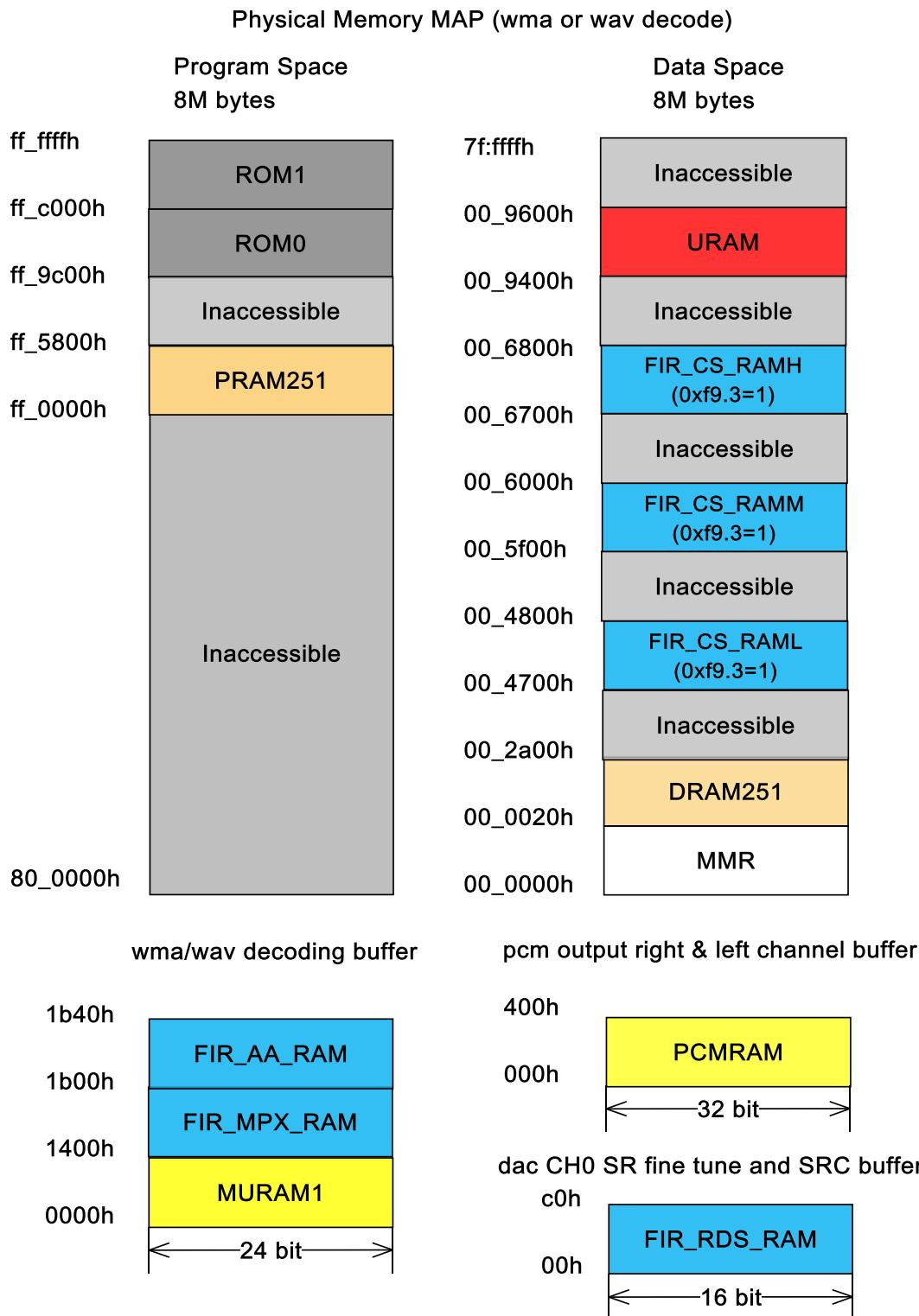


Figure 5-19 wma decode memory mapping

### 5.3.2.4 WAV/MP3 Encode Memory mapping

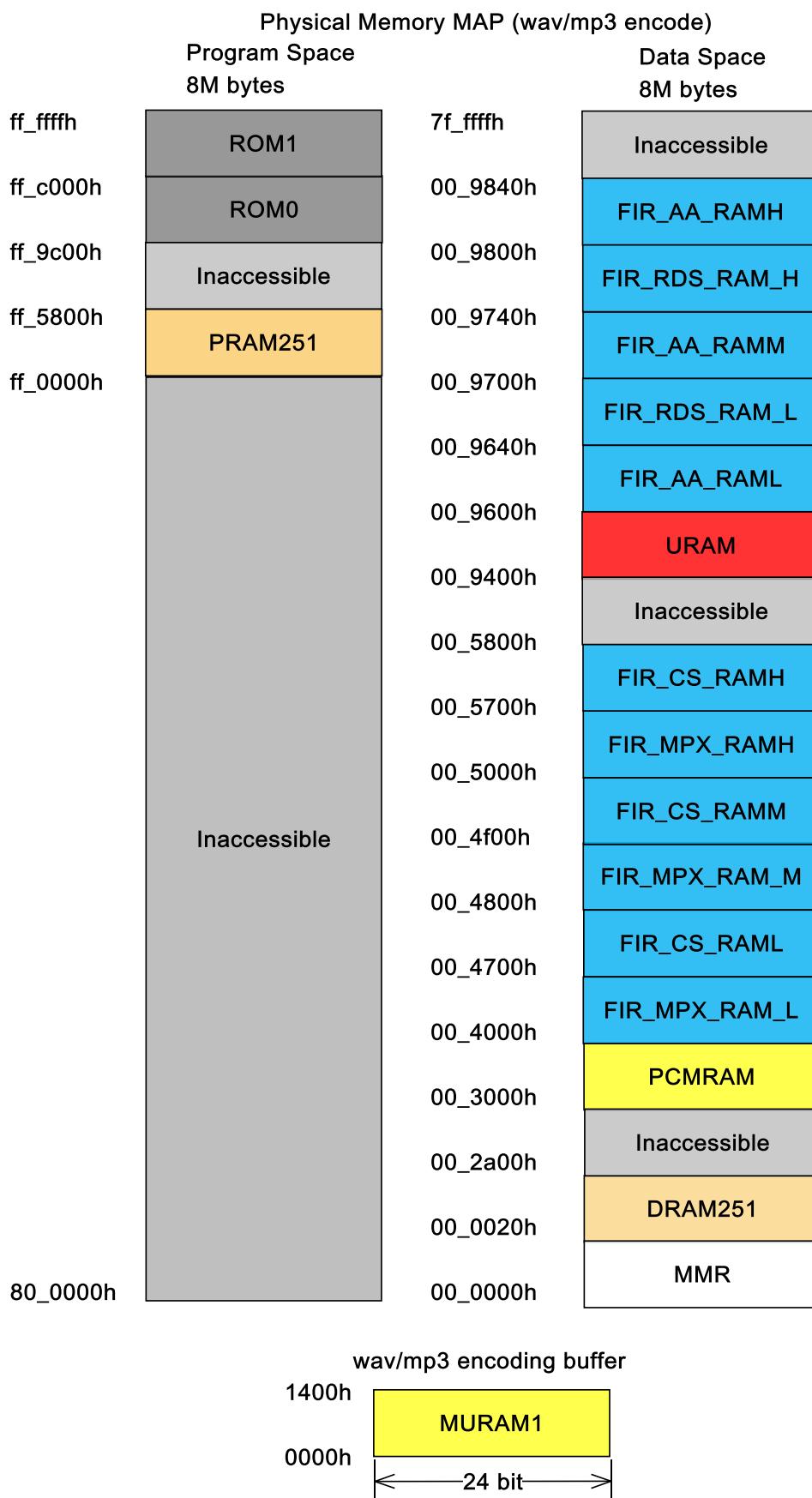


Figure 5-20 wav or mp3 encode memory mapping

### 5.3.2.5 USB speaker & WAV/MP3 encode memory map

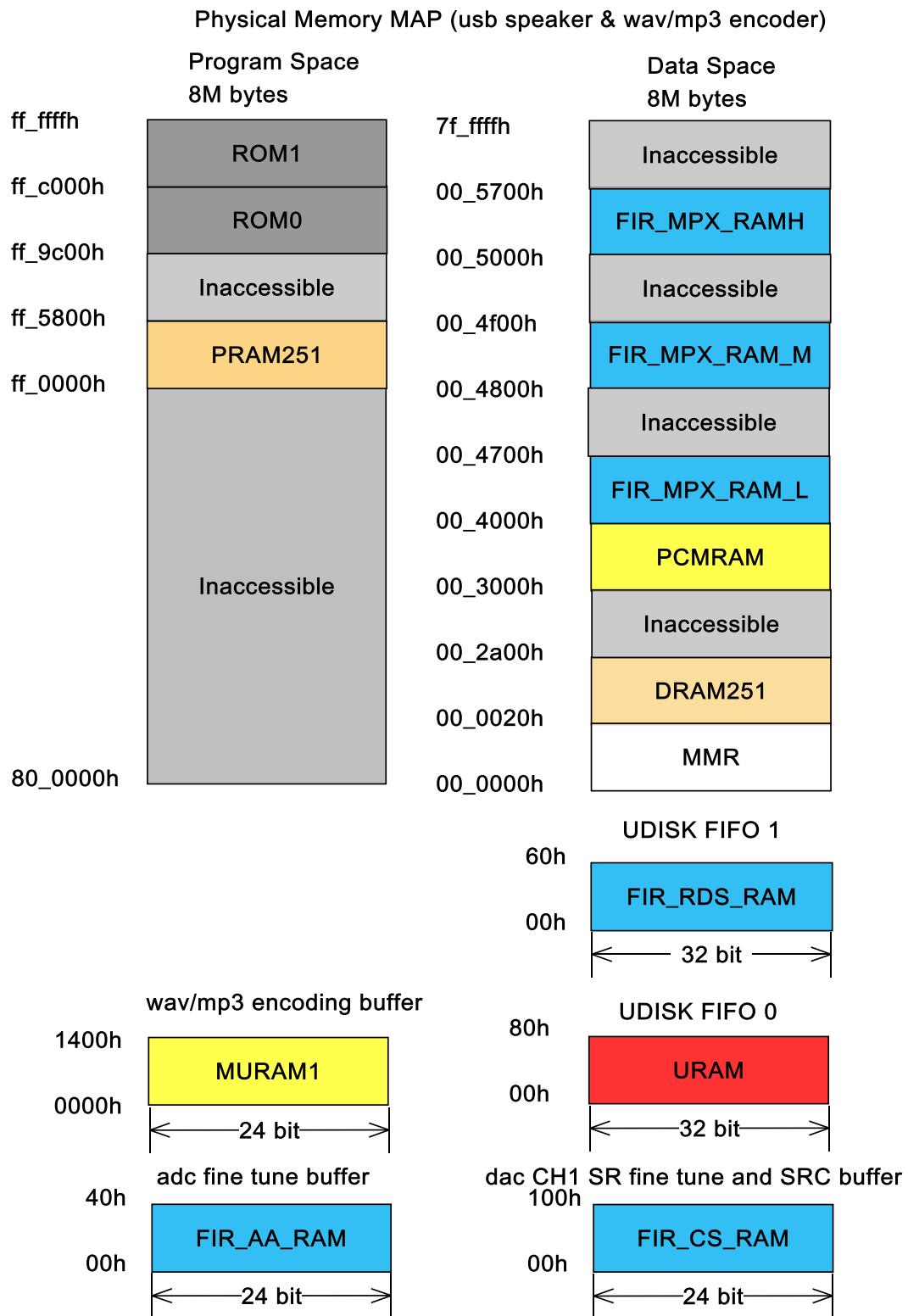


Figure 5-21 USB speaker & wav/mp3 encoding memory mapping

### 5.3.2.6 UDISK memory mapping

**Physical Memory MAP (UDISK)**

| Program Space<br>8M bytes |                        | Data Space<br>8M bytes |                        |
|---------------------------|------------------------|------------------------|------------------------|
| ff_ffffh                  | ROM1                   | 7f_ffffh               | Inaccessible           |
| ff_c000h                  | ROM0                   | 00_9840h               | FIR_AA_RAMH            |
| ff_9c00h                  | Inaccessible           | 00_9800h               | FIR_RDS_RAM_H          |
| ff_9400h                  | MURAM1H<br>(0xf9.3==0) | 00_9740h               | FIR_AA_RAMM            |
| ff_8000h                  | MURAM1M<br>(0xf9.3==0) | 00_9700h               | FIR_RDS_RAM_L          |
| ff_6c00h                  | MURAM1L<br>(0xf9.3==0) | 00_9640h               | FIR_AA_RAML            |
| ff_5800h                  | PRAM251                | 00_9600h               | Inaccessible           |
| ff_0000h                  | Inaccessible           | 00_9400h               | MURAM1H<br>(0xf9.3==1) |
|                           |                        | 00_8000h               | MURAM1M<br>(0xf9.3==1) |
|                           |                        | 00_6c00h               | MURAM1L<br>(0xf9.3==1) |
|                           |                        | 00_5800h               | FIR_CS_RAMH            |
|                           |                        | 00_5700h               | FIR_MPX_RAMH           |
|                           |                        | 00_5000h               | FIR_CS_RAMM            |
|                           |                        | 00_4f00h               | FIR_MPX_RAM_M          |
|                           |                        | 00_4800h               | FIR_CS_RAML            |
|                           |                        | 00_4700h               | FIR_MPX_RAM_L          |
|                           |                        | 00_4000h               | Inaccessible           |
|                           |                        | 00_2a00h               | DRAM251                |
| 80_0000h                  |                        | 00_0020h               | MMR                    |
|                           |                        | 00_0000h               |                        |

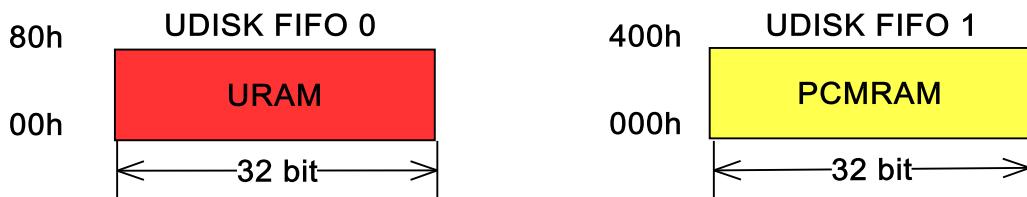


Figure 5-22 UDISK memory map

### 5.3.2.7 Soundbar memory map

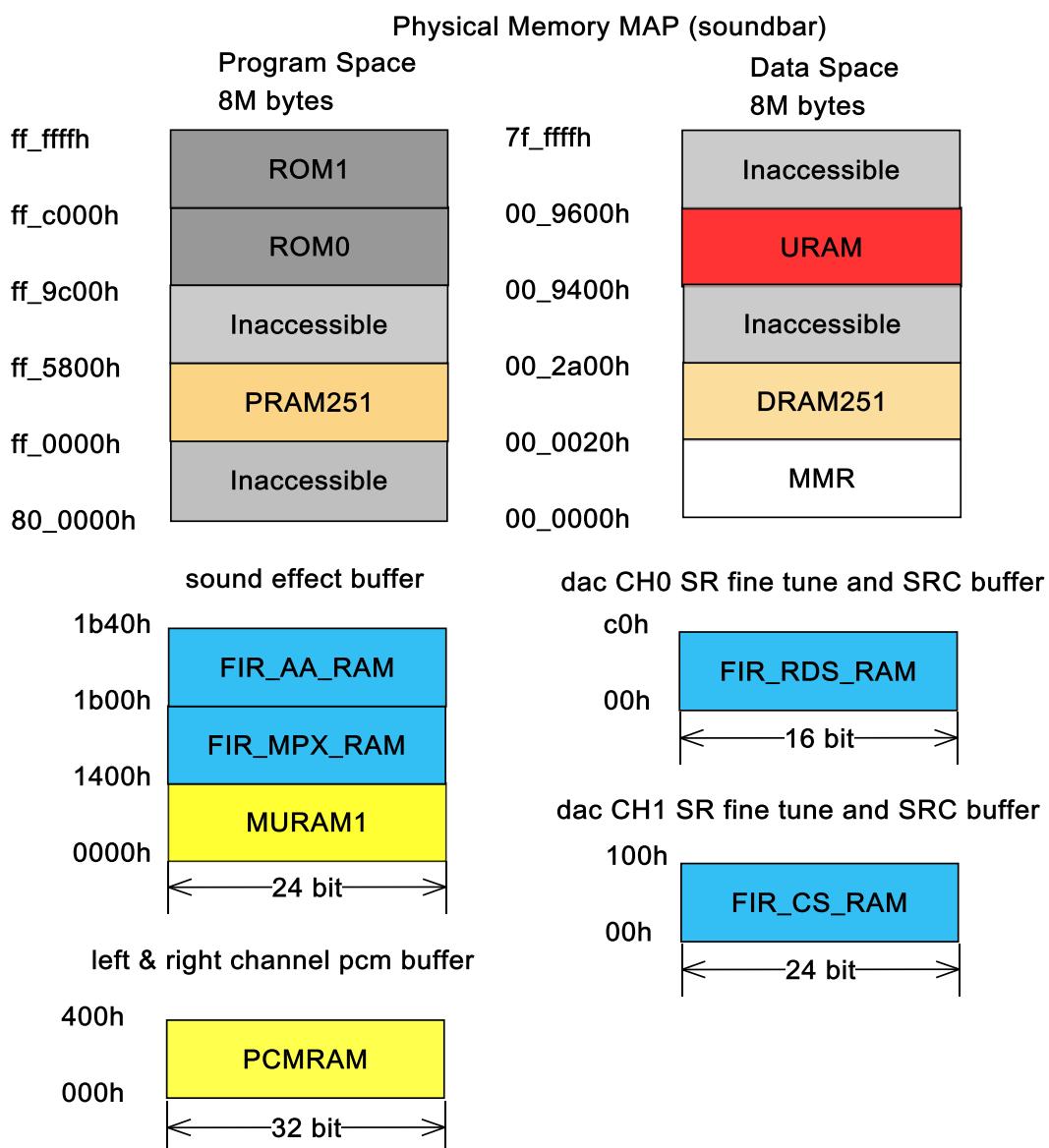


Figure 5-3 sound bar memory mapping

### 5.3.3 Memory blocks & peripherals access channel list

DMA 和 CPU 的时钟是同源的，它们具有相同的频率和相位，USB、AUIP、SD CARD、IIC、SPI、UART、FM/RDS、ADC、DAC 都有独立的时钟源。

| Access path<br>Access target | CPU<br>program bus |   | CPU<br>data bus |   | SFR<br>BUS |   | DMA<br>0/1/2/3<br>/4 |   | USB |   | FM/R<br>DS |   | AUIP |   | DAC |   | ADC |   |   |
|------------------------------|--------------------|---|-----------------|---|------------|---|----------------------|---|-----|---|------------|---|------|---|-----|---|-----|---|---|
|                              | R                  | W | R               | W | R          | W | R                    | W | R   | W | R          | W | R    | W | R   | W | R   | W |   |
| ROM0/1                       | Y                  | 1 |                 |   |            |   | Y                    |   |     |   |            |   |      |   |     |   |     |   |   |
| PRAM251                      | Y                  | Y |                 |   |            |   | Y                    | Y |     |   |            |   |      |   |     |   |     |   |   |
| DRAM251                      |                    |   | Y               | Y |            |   | Y                    | Y |     |   |            |   |      |   |     |   |     |   |   |
| URAM                         |                    |   | Y               | Y |            |   | Y                    | Y | Y   | Y |            |   |      |   |     |   |     |   |   |
| MURAM10xf9.3=1               |                    |   | Y               | Y |            |   | Y                    | Y |     |   | Y          |   |      |   |     |   |     |   |   |
| MURAM10xf9.3=0               | Y                  | Y |                 |   |            |   | Y                    | Y |     |   | Y          |   | Y    | Y |     |   |     |   |   |
| PCMRAM                       |                    |   | Y               | Y |            |   | Y                    | Y | Y   | Y |            | Y | Y    | Y |     |   |     |   |   |
| FIR_CS_RAM                   |                    |   | Y               | Y |            |   | Y                    | Y |     |   | Y          | Y | Y    | Y | Y   | Y |     |   |   |
| FIR_AA_RAM                   |                    |   | Y               | Y |            |   | Y                    | Y |     |   | Y          | Y | Y    | Y |     |   | Y   | Y |   |
| FIR_MPX_RAM                  |                    |   | Y               | Y |            |   | Y                    | Y |     |   | Y          | Y | Y    | Y |     |   |     |   |   |
| FIR_RDS_RAM                  |                    |   | Y               | Y |            |   | Y                    | Y | Y   | Y | Y          | Y |      |   |     |   |     |   |   |
| DACRAM                       |                    |   |                 |   |            |   |                      |   |     |   |            |   |      |   |     | Y | Y   |   |   |
| ADCRAM                       |                    |   |                 |   |            |   |                      |   |     |   |            |   |      |   |     |   |     | Y | Y |
| USB FIFO                     |                    |   |                 |   | Y          | Y | Y                    | Y |     |   |            |   |      |   |     |   |     |   |   |
| CARD FIFO                    |                    |   |                 |   | Y          | Y | Y                    | Y |     |   |            |   |      |   |     |   |     |   |   |
| AUIP TX FIFO                 |                    |   |                 |   |            | Y |                      | Y |     |   |            |   |      |   |     |   |     |   |   |
| AUIP RX FIFO                 |                    |   |                 |   | Y          |   | Y                    |   |     |   |            |   |      |   |     |   |     |   |   |
| SPI FIFO                     |                    |   |                 |   | Y          | Y | Y                    | Y |     |   |            |   |      |   |     |   |     |   |   |
| UART FIFO                    |                    |   |                 |   | Y          | Y | Y                    | Y |     |   |            |   |      |   |     |   |     |   |   |
| ADC FIFO                     |                    |   |                 |   | Y          |   | Y                    |   |     |   |            |   |      |   |     |   |     |   |   |
| DAC FIFO                     |                    |   |                 |   |            | Y |                      | Y |     |   |            |   |      |   |     |   |     |   |   |
| ACC FIFO                     |                    |   |                 |   |            | Y |                      | Y |     |   |            |   |      |   |     |   |     |   |   |
| Peripherals' regs            |                    |   |                 |   | Y          | Y |                      |   |     |   |            |   |      |   |     |   |     |   |   |

Table 1 memory blocks and peripherals access channel list

注意：

- (1) ROM0/1 不能被 CPU program bus 写，但是 FPGA 平台上是可写的。

### 5.3.4 Memory management

下面列出各种场景下的 memory 分配：

| Memory mode | ROM 0/1 | PRA M251,<br>DRA M251 | MUR AM1 | PCM RAM | FIR_CS_R AM | FIR_AA_R AM | FIR_MPX_R AM | FIR_R DS_R AM | URA M |
|-------------|---------|-----------------------|---------|---------|-------------|-------------|--------------|---------------|-------|
|-------------|---------|-----------------------|---------|---------|-------------|-------------|--------------|---------------|-------|

| Default         | R | R/W |
|-----------------|---|-----|-----|-----|-----|-----|-----|-----|-----|
| MP3 decode      | R | R/W |
| WMA/WA V decode | R | R/W |
| WAV/MP3 encode  | R | R/W |
| USB speaker     | R | R/W |
| UDISK           | R | R/W |
| FM/RDS          | R | R/W |
| Soundbar        | R | R/W |

Table 2 memory allocation list

主要访问源有以下，分别用不同的颜色表示

|        |                |
|--------|----------------|
| Yellow | AUIP           |
| Blue   | FM/RDS         |
| Pink   | USB            |
| Orange | MCU/DMA0/1/2/3 |
| Green  | DAC CH0        |
| Red    | DAC CH1        |
| Grey   | ADC            |

## 5.4 Memory\_Controller register list

| Index | Mnemonic   | Description                          | BANK     |
|-------|------------|--------------------------------------|----------|
| 0xf9  | MemBankCtl | Memory banking control register      | all bank |
| 0x90  | PageAddr0H | High byte of Page Address register 0 | 0x00     |
| 0x91  | PageAddr0L | Low byte of Page Address register 0  | 0x00     |
| 0x92  | PageAddr1H | High byte of Page Address register 1 | 0x00     |
| 0x93  | PageAddr1L | Low byte of Page Address register 1  | 0x00     |
| 0x94  | PageAddr2H | High byte of Page Address register 2 | 0x00     |
| 0x95  | PageAddr2L | Low byte of Page Address register 2  | 0x00     |
| 0x98  | PageAddr3H | High byte of Page Address register 3 | 0x00     |
| 0x99  | PageAddr3L | Low byte of Page Address register 3  | 0x00     |
| 0x9a  | PageAddr4H | High byte of Page Address register 4 | 0x00     |
| 0x9b  | PageAddr4L | Low byte of Page Address register 4  | 0x00     |
| 0x9c  | PageAddr5H | High byte of Page Address register 5 | 0x00     |
| 0x9d  | PageAddr5L | Low byte of Page Address register 5  | 0x00     |
| 0x9e  | PageAddr6H | High byte of Page Address register 6 | 0x00     |

|      |                    |   |      |
|------|--------------------|---|------|
| 0x9f | PageAddr6L         | Low byte of Page Address register 6       | 0x00 |
| 0xa2 | PageAddr7H         | High byte of Page Address register 7      | 0x00 |
| 0xa3 | PageAddr7L         | Low byte of Page Address register 7       | 0x00 |
| 0xd8 | PageAddr8H         | High byte of Page Address register 8      | 0x00 |
| 0xd9 | PageAddr8L         | Low byte of Page Address register 8       | 0x00 |
| 0xda | PageAddr9H         | High byte of Page Address register 9      | 0x00 |
| 0xdb | PageAddr9L         | Low byte of Page Address register 9       | 0x00 |
| 0xb0 | PageAddrMask0      | Page 0 Address Mask register              | 0x00 |
| 0xb1 | PageAddrMask1      | Page 1 Address Mask register              | 0x00 |
| 0xb2 | PageAddrMask2      | Page 2 Address Mask register              | 0x00 |
| 0xb3 | PageAddrMask3      | Page 3 Address Mask register              | 0x00 |
| 0xb4 | PageAddrMask4      | Page 4 Address Mask register              | 0x00 |
| 0xb5 | PageAddrMask5      | Page 5 Address Mask register              | 0x00 |
| 0xb6 | PageAddrMask6      | Page 6 Address Mask register              | 0x00 |
| 0xd2 | PageAddrMask7      | Page 7 Address Mask register              | 0x00 |
| 0xd4 | PageAddrMask8      | Page 8 Address Mask register              | 0x00 |
| 0xd5 | PageAddrMask9      | Page 9 Address Mask register              | 0x00 |
| 0xc8 | RedirectAddr0      | Redirect Address 0                        | 0x00 |
| 0xc9 | RedirectAddr1      | Redirect Address 1                        | 0x00 |
| 0xca | RedirectAddr2      | Redirect Address 2                        | 0x00 |
| 0xcb | RedirectAddr3      | Redirect Address 3                        | 0x00 |
| 0xcc | RedirectAddr4      | Redirect Address 4                        | 0x00 |
| 0xcd | RedirectAddr5      | Redirect Address 5                        | 0x00 |
| 0xce | RedirectAddr6      | Redirect Address 6                        | 0x00 |
| 0xcf | RedirectAddr7      | Redirect Address 7                        | 0x00 |
| 0xd6 | RedirectAddr8      | Redirect Address 8                        | 0x00 |
| 0xd7 | RedirectAddr9      | Redirect Address 9                        | 0x00 |
| 0xd3 | FixAddr0H          | Hight byte of Fix Address register 0      | 0x00 |
| 0xb9 | FixAddr0M          | Middle byte of Fix Address register 0     | 0x00 |
| 0xba | FixAddr0L          | Low byte of Fix Address register 0        | 0x00 |
| 0xbb | FixAddr1H          | Hight byte of Fix Address register 1      | 0x00 |
| 0xbc | FixAddr1M          | Middle byte of Fix Address register 1     | 0x00 |
| 0xbd | FixAddr1L          | Low byte of Fix Address register 1        | 0x00 |
| 0xc2 | FixAddr2H          | Hight byte of Fix Address register 2      | 0x00 |
| 0xc3 | FixAddr2M          | Middle byte of Fix Address register 2     | 0x00 |
| 0xc4 | FixAddr2L          | Low byte of Fix Address register 2        | 0x00 |
| 0xc5 | FixAddr3H          | Hight byte of Fix Address register 3      | 0x00 |
| 0xc6 | FixAddr3M          | Middle byte of Fix Address register 3     | 0x00 |
| 0xc7 | FixAddr3L          | Low byte of Fix Address register 3        | 0x00 |
| 0xf4 | PageMissEntryEx    | Page Miss Entry Address Extension byte    | 0x00 |
| 0xf5 | PageMissEntryH     | Page Miss Entry Address High byte         | 0x00 |
| 0xf6 | PageMissEntryL     | Page Miss Entry Address Low byte          | 0x00 |
| 0xa5 | CodeReplaceEntryEx | Code Replace Entry Address Extension byte | 0x00 |

|      |                   |                                      |      |
|------|-------------------|--------------------------------------|------|
| 0xa6 | CodeReplaceEntryH | Code Replace Entry Address High byte | 0x00 |
| 0xa7 | CodeReplaceEntryL | Code Replace Entry Address Low byte  | 0x00 |
| 0xfc | TestCounterByte0  | Test Counter byte register 0         | 0x00 |
| 0xfd | TestCounterByte1  | Test Counter byte register 1         | 0x00 |
| 0xfe | TestCounterByte2  | Test Counter byte register 2         | 0x00 |
| 0xff | TestCounterByte3  | Test Counter byte register 3         | 0x00 |
| 0xf1 | TestCounterByte4  | Test Counter byte register 4         | 0x00 |
| 0xe1 | bist_en1          | Bist enable register1                | 0x01 |
| 0xe2 | bist_en2          | Bist enable register2                | 0x01 |
| 0xe4 | bist_fin1         | Bist finish flag register1           | 0x01 |
| 0xe5 | bist_fin2         | Bist finish flag register2           | 0x01 |
| 0xe7 | bist_info1        | Bist information register1           | 0x01 |
| 0xe9 | bist_info2        | Bist information register2           | 0x01 |

NOTE: Registers in the list of different colors represent different functions

- (1) 红色表示总功能控制。
- (2) 黄色是用于 memory banking
- (3) 粉红色是用于 code replace
- (4) 绿色是计数器用于测试 Mcu 的性能
- (5) 蓝色是 memory bist 的寄存器

## 5.5 Register Description

### 5.5.1 MemBankCtl

MemBankCtl (Memory banking control register, SFR address 0xf9, all bank)

| Bit Number | Bit Mnemonic        | Function   | Access | Reset |
|------------|---------------------|--|--------|-------|
| 7:6        | TestCounterMode     | 00: No Operations<br>01: Run cycle mode<br>10: Interrupt trigger count mode<br>11: Executed interrupt trigger count mode | R/W    | 00    |
| 5          | RstTestCounter      | 0: test counter active<br>1: test counter reset  | R/W    | 0     |
| 4          | CountEnable         | 0: Count disable<br>1 : Count enable   | R/W    | 0     |
| 3          | MURAM_ADD_R         | The start address of MURAM1:<br>0: FF5800H~FF93FFH<br>1: 005800H~0093FFH   | R/W    | 0     |
| 2          | InterruptVectorPage | The Interrupt vector's start address is 0ffc000h as a default value. The address 0ff0000h is this bit is set.            | R/W    | 0     |

|   |                         |  |     |   |
|---|-------------------------|--|-----|---|
|   |                         | 0: the interrupt vector start address is<br>0ffc000h<br><br>1: the interrupt vector start address is<br>0ff0000h   |     |   |
| 1 | PMMT_ERR_C<br><br>PU_IP | Error pending bit to indicate the Page<br>Miss based Mapping Translation error<br>while CPU accessing Program Space <sup>(2)</sup><br><br>0: no error<br>1: error<br><br>Writing '1' to this bit will clear the error<br>pending.  | R/W | 0 |
| 0 | PMMT_ERR_D<br><br>MA_IP | Error pending bit to indicate the Page<br>Miss based Mapping Translation error<br>while DMA accessing Program Space: <sup>(1)</sup><br><br>0: no error<br>1: error<br><br>Writing '1' to this bit will clear the error<br>pending. | R/W | 0 |

Note:

- e) The error pending is set if one virtual address is mapped to more than two different physical addresses. The data read/write from program space is selected by the priority decoder of PMMT.
- f) The error pending is not set if no match of any page address registers nor the access address is from 0xff0000 to 0xfffff. The CPU data read/write from program space is mapped to 0xff0000 to 0xfffffff.
- g) The error pending is set if no match of any page address registers nor the access address is from 0xff0000 to 0xfffff. The DMA data read from program space is mapped to 0xff0000 to 0xfffff.

## 5.5.2 PageAddr0H

PageAddr0H (High byte of Page Address register 0, SFR address 0x90, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.3 PageAddr0L

PageAddr0L (Low byte of Page Address register 0, SFR address 0x91, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.4 PageAddr1H

**PageAddr1H (High byte of Page Address register 1, SFR address 0x92, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.5 PageAddr1L

**PageAddr1L (Low byte of Page Address register 1, SFR address 0x93, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.6 PageAddr2H

**PageAddr2H (High byte of Page Address register 2, SFR address 0x94, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.7 PageAddr2L

**PageAddr2L (Low byte of Page Address register 2, SFR address 0x95, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.8 PageAddr3H

**PageAddr3H (High byte of Page Address register 3, SFR address 0x98, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.9 PageAddr3L

**PageAddr3L (Low byte of Page Address register 3, SFR address 0x99, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.10 PageAddr4H

**PageAddr4H (High byte of Page Address register 4, SFR address 0x9a, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.11 PageAddr4L

**PageAddr4L (Low byte of Page Address register 4, SFR address 0x9b, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.12 PageAddr5H

**PageAddr5H (High byte of Page Address register 5, SFR address 0x9c, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.13 PageAddr5L

**PageAddr5L (Low byte of Page Address register 5, SFR address 0x9d, bank: 00)**

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.14 PageAddr6H

PageAddr6H (High byte of Page Address register 6, SFR address 0x9e, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.15 PageAddr6L

PageAddr6L (Low byte of Page Address register 6, SFR address 0x9f, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.16 PageAddr7H

PageAddr7H (High byte of Page Address register 7, SFR address 0xa2, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.17 PageAddr7L

PageAddr7L (Low byte of Page Address register 7, SFR address 0xa3, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.18 PageAddr8H

PageAddr8H (High byte of Page Address register 8, SFR address 0xd8, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.19 PageAddr8L

PageAddr8L (Low byte of Page Address register 8, SFR address 0xd9, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.20 PageAddr9H

PageAddr9H (High byte of Page Address register 9, SFR address 0xda, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7          | Reserved     | Be read as '1'                   | -      | 1     |
| 6:0        | PageAddr     | The bit 22:16 of program address | R/W    | 7fh   |

## 5.5.21 PageAddr9L

PageAddr9L (Low byte of Page Address register 9, SFR address 0xdb, bank: 00)

| Bit Number | Bit Mnemonic | Function                         | Access | Reset |
|------------|--------------|----------------------------------|--------|-------|
| 7:5        | PageAddr     | The bit 15:13 of program address | R/W    | 0h    |
| 4:0        | Reserved     | Be read as 5 zeros               | -      | -     |

## 5.5.22 PageAddrMask0

PageAddrMask0 (Page 0 Address Mask register, SFR address 0xb0, bank: 00)

| Bit Number | Bit Mnemonic           | Function   | Access | Reset |
|------------|------------------------|--|--------|-------|
| 7          | PageAddressMask_bit_15 | Bit 15 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 6          | PageAddressMask_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 5          | PageAddressMask_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1: Address Mask enable  | R/W    | 0     |
| 4:0        | Reserved               | Be read as 5 zeros   | -      | -     |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

## 5.5.23 PageAddrMask1

**PageAddrMask1 (Page 1 Address Mask register, SFR address 0xb1, bank: 00)**

| Bit Number | Bit Mnemonic           | Function   | Access | Reset |
|------------|------------------------|--|--------|-------|
| 7          | PageAddressMask_bit_15 | Bit 15 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 6          | PageAddressMask_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 5          | PageAddressMask_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 4:0        | Reserved               | Be read as 5 zeros   | -      | -     |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

## 5.5.24 PageAddrMask2

**PageAddrMask2 (Page 2 Address Mask register, SFR address 0xb2, bank: 00)**

| Bit Number | Bit Mnemonic           | Function   | Access | Reset |
|------------|------------------------|--|--------|-------|
| 7          | PageAddressMask_bit_15 | Bit 15 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 6          | PageAddressMask_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 5          | PageAddressMask_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 4:0        | Reserved               | Be read as 5 zeros   | -      | -     |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

## 5.5.25 PageAddrMask3

**PageAddrMask3 (Page 3 Address Mask register, SFR address 0xb3, bank: 00)**

| Bit Number | Bit Mnemonic           | Function   | Access | Reset |
|------------|------------------------|--|--------|-------|
| 7          | PageAddressMask_bit_15 | Bit 15 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |

|     |                        |  |     |   |
|-----|------------------------|--|-----|---|
| 6   | PageAddressMask_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W | 0 |
| 5   | PageAddressMask_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W | 0 |
| 4:0 | Reserved               | Be read as 5 zeros   | -   | - |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

### 5.5.26 PageAddrMask4

**PageAddrMask4 (Page 4 Address Mask register, SFR address 0xb4, bank: 00)**

| Bit Number | Bit Mnemonic           | Function   | Access | Reset |
|------------|------------------------|--|--------|-------|
| 7          | PageAddressMask_bit_15 | Bit 15 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 6          | PageAddressMask_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 5          | PageAddressMask_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 4:0        | Reserved               | Be read as 5 zeros   | -      | -     |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

### 5.5.27 PageAddrMask5

**PageAddrMask5 (Page 5 Address Mask register, SFR address 0xb5, bank: 00)**

| Bit Number | Bit Mnemonic           | Function   | Access | Reset |
|------------|------------------------|--|--------|-------|
| 7          | PageAddressMask_bit_15 | Bit 15 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 6          | PageAddressMask_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 5          | PageAddressMask_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 4:0        | Reserved               | Be read as 5 zeros   | -      | -     |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

## 5.5.28 PageAddrMask6

**PageAddrMask6 (Page 6 Address Mask register, SFR address 0xb0, bank: 00)**

| Bit Number | Bit Mnemonic           | Function   | Access | Reset |
|------------|------------------------|--|--------|-------|
| 7          | PageAddressMask_bit_15 | Bit 15 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 6          | PageAddressMask_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 5          | PageAddressMask_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 4:0        | Reserved               | Be read as 5 zeros   | -      | -     |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

## 5.5.29 PageAddrMask7

**PageAddrMask7 (Page 7 Address Mask register, SFR address 0xd2, bank: 00)**

| Bit Number | Bit Mnemonic           | Function   | Access | Reset |
|------------|------------------------|--|--------|-------|
| 7          | PageAddressMask_bit_15 | Bit 15 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 6          | PageAddressMask_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 5          | PageAddressMask_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 4:0        | Reserved               | Be read as 5 zeros   | -      | -     |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

## 5.5.30 PageAddrMask8

**PageAddrMask8 (Page 8 Address Mask register, SFR address 0xd4, bank: 00)**

| Bit Number | Bit Mnemonic    | Function                    | Access | Reset |
|------------|-----------------|-----------------------------|--------|-------|
| 7          | PageAddressMask | Bit 15 of Page Address Mask | R/W    | 0     |

|     |                            |  |     |   |
|-----|----------------------------|--|-----|---|
|     | sk_bit_15                  | 0:Address Mask disable<br>1 : Address Mask enable                                |     |   |
| 6   | PageAddressMa<br>sk_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W | 0 |
| 5   | PageAddressMa<br>sk_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W | 0 |
| 4:0 | Reserved                   | Be read as 5 zeros   | -   | - |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

### 5.5.31 PageAddrMask9

**PageAddrMask9 (Page 9 Address Mask register, SFR address 0xd5, bank: 00)**

| Bit Number | Bit Mnemonic               | Function   | Access | Reset |
|------------|----------------------------|--|--------|-------|
| 7          | PageAddressMa<br>sk_bit_15 | Bit 15 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 6          | PageAddressMa<br>sk_bit_14 | Bit 14 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 5          | PageAddressMa<br>sk_bit_13 | Bit 13 of Page Address Mask<br>0:Address Mask disable<br>1 : Address Mask enable | R/W    | 0     |
| 4:0        | Reserved                   | Be read as 5 zeros   | -      | -     |

Note: The bits 23:16 of Page Address Mask are all ones (always enable).

### 5.5.32 RedirectAddr0

**RedirectAddr0 (Redirect Address register 0, SFR address 0xc8, bank: 00)**

| Bit Number | Bit Mnemonic                    | Function                      | Access | Reset |
|------------|---------------------------------|-------------------------------|--------|-------|
| 7:5        | RedirectAddress<br>bit_15_to_13 | Bit 15:13 of Redirect Address | R/W    | 0     |
| 4:0        | Reserved                        | Be read as 5 zeros            | -      | -     |

### 5.5.33 RedirectAddr1

**RedirectAddr1 (Redirect Address register 1, SFR address 0xc9, bank: 00)**

| Bit Number | Bit Mnemonic | Function | Access | Reset |
|------------|--------------|----------|--------|-------|
|            |              |          |        |       |

|     |                                 |                               |     |   |
|-----|---------------------------------|-------------------------------|-----|---|
| 7:5 | RedirectAddress<br>bit_15_to_13 | Bit 15:13 of Redirect Address | R/W | 0 |
| 4:0 | Reserved                        | Be read as 5 zeros            | -   | - |

### 5.5.34 RedirectAddr2

**RedirectAddr2 (Redirect Address register 2, SFR address 0xca, bank: 00)**

| Bit Number | Bit Mnemonic                    | Function                      | Access | Reset |
|------------|---------------------------------|-------------------------------|--------|-------|
| 7:5        | RedirectAddress<br>bit_15_to_13 | Bit 15:13 of Redirect Address | R/W    | 0     |
| 4:0        | Reserved                        | Be read as 5 zeros            | -      | -     |

### 5.5.35 RedirectAddr3

**RedirectAddr3 (Redirect Address register 3, SFR address 0xcb, bank: 00)**

| Bit Number | Bit Mnemonic                    | Function                      | Access | Reset |
|------------|---------------------------------|-------------------------------|--------|-------|
| 7:5        | RedirectAddress<br>bit_15_to_13 | Bit 15:13 of Redirect Address | R/W    | 0     |
| 4:0        | Reserved                        | Be read as 5 zeros            | -      | -     |

### 5.5.36 RedirectAddr4

**RedirectAddr4 (Redirect Address register 4, SFR address 0xcc, bank: 00)**

| Bit Number | Bit Mnemonic                    | Function                      | Access | Reset |
|------------|---------------------------------|-------------------------------|--------|-------|
| 7:5        | RedirectAddress<br>bit_15_to_13 | Bit 15:13 of Redirect Address | R/W    | 0     |
| 4:0        | Reserved                        | Be read as 5 zeros            | -      | -     |

### 5.5.37 RedirectAddr5

**RedirectAddr5 (Redirect Address register 5, SFR address 0xcd, bank: 00)**

| Bit Number | Bit Mnemonic                    | Function                      | Access | Reset |
|------------|---------------------------------|-------------------------------|--------|-------|
| 7:5        | RedirectAddress<br>bit_15_to_13 | Bit 15:13 of Redirect Address | R/W    | 0     |
| 4:0        | Reserved                        | Be read as 5 zeros            | -      | -     |

## 5.5.38 RedirectAddr6

**RedirectAddr6 (Redirect Address register 6, SFR address 0xce, bank: 00)**

| Bit Number | Bit Mnemonic                 | Function                      | Access | Reset |
|------------|------------------------------|-------------------------------|--------|-------|
| 7:5        | RedirectAddress bit_15_to_13 | Bit 15:13 of Redirect Address | R/W    | 0     |
| 4:0        | Reserved                     | Be read as 5 zeros            | -      | -     |

## 5.5.39 RedirectAddr7

**RedirectAddr7 (Redirect Address register 7, SFR address 0xcf, bank: 00)**

| Bit Number | Bit Mnemonic                 | Function                      | Access | Reset |
|------------|------------------------------|-------------------------------|--------|-------|
| 7:5        | RedirectAddress bit_15_to_13 | Bit 15:13 of Redirect Address | R/W    | 0     |
| 4:0        | Reserved                     | Be read as 5 zeros            | -      | -     |

## 5.5.40 RedirectAddr8

**RedirectAddr8 (Redirect Address register 8, SFR address 0xd6, bank: 00)**

| Bit Number | Bit Mnemonic                 | Function                      | Access | Reset |
|------------|------------------------------|-------------------------------|--------|-------|
| 7:5        | RedirectAddress bit_15_to_13 | Bit 15:13 of Redirect Address | R/W    | 0     |
| 4:0        | Reserved                     | Be read as 5 zeros            | -      | -     |

## 5.5.41 RedirectAddr9

**RedirectAddr9 (Redirect Address register 9, SFR address 0xd7, bank: 00)**

| Bit Number | Bit Mnemonic                 | Function                      | Access | Reset |
|------------|------------------------------|-------------------------------|--------|-------|
| 7:5        | RedirectAddress bit_15_to_13 | Bit 15:13 of Redirect Address | R/W    | 0     |
| 4:0        | Reserved                     | Be read as 5 zeros            | -      | -     |

## 5.5.42 FixAddr0H

**FixAddr0H (High byte of Fix Address register 0, SFR address 0xd3, bank: 00)**

| Bit | Bit Mnemonic | Function | Access | Reset |
|-----|--------------|----------|--------|-------|
|     |              |          |        |       |

| Number |         |                                  |     |   |
|--------|---------|----------------------------------|-----|---|
| 7:0    | FixAddr | The bit 23:16 of program address | R/W | - |

### 5.5.43 FixAddr0M

**FixAddr0M (Middle byte of Fix Address register 0, SFR address 0xb9, bank: 00)**

| Bit Number | Bit Mnemonic | Function                        | Access | Reset |
|------------|--------------|---------------------------------|--------|-------|
| 7:0        | FixAddr      | The bit 15:8 of program address | R/W    | -     |

### 5.5.44 FixAddr0L

**FixAddr0L (Low byte of Fix Address register 0, SFR address 0xba, bank: 00)**

| Bit Number | Bit Mnemonic | Function                       | Access | Reset |
|------------|--------------|--------------------------------|--------|-------|
| 7:0        | FixAddr      | The bit 7:0 of program address | R/W    | -     |

### 5.5.45 FixAddr1H

**FixAddr1H (High byte of Fix Address register 1, SFR address 0xbb, bank: 00)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | FixEn        | 0 :Code replace disable<br>1 :Code replace enable | R/W    | 0     |
| 6:0        | FixAddr      | The bit 22:16 of program address                  | R/W    | -     |

### 5.5.46 FixAddr1M

**FixAddr1M (Middle byte of Fix Address register 1, SFR address 0xbc, bank: 00)**

| Bit Number | Bit Mnemonic | Function                        | Access | Reset |
|------------|--------------|---------------------------------|--------|-------|
| 7:0        | FixAddr      | The bit 15:8 of program address | R/W    | -     |

### 5.5.47 FixAddr1L

**FixAddr1L (Low byte of Fix Address register 1, SFR address 0xbd, bank: 00)**

| Bit Number | Bit Mnemonic | Function                       | Access | Reset |
|------------|--------------|--------------------------------|--------|-------|
| 7:0        | FixAddr      | The bit 7:0 of program address | R/W    | -     |

## 5.5.48 FixAddr2H

**FixAddr2H (High byte of Fix Address register 2, SFR address 0xc2, bank: 00)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | FixEn        | 0 :Code replace disable<br>1 :Code replace enable | R/W    | 0     |
| 6:0        | FixAddr      | The bit 22:16 of program address                  | R/W    | -     |

## 5.5.49 FixAddr2M

**FixAddr2M (Middle byte of Fix Address register 2, SFR address 0xc3, bank: 00)**

| Bit Number | Bit Mnemonic | Function                        | Access | Reset |
|------------|--------------|---------------------------------|--------|-------|
| 7:0        | FixAddr      | The bit 15:8 of program address | R/W    | -     |

## 5.5.50 FixAddr2L

**FixAddr2L (Low byte of Fix Address register 2, SFR address 0xc4, bank: 00)**

| Bit Number | Bit Mnemonic | Function                       | Access | Reset |
|------------|--------------|--------------------------------|--------|-------|
| 7:0        | FixAddr      | The bit 7:0 of program address | R/W    | -     |

## 5.5.51 FixAddr3H

**FixAddr3H (High byte of Fix Address register 3, SFR address 0xc5, bank: 00)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | FixEn        | 0 :Code replace disable<br>1 :Code replace enable | R/W    | 0     |
| 6:0        | FixAddr      | The bit 22:16 of program address                  | R/W    | -     |

## 5.5.52 FixAddr3M

**FixAddr3M (Middle byte of Fix Address register 3, SFR address 0xc6, bank: 00)**

| Bit Number | Bit Mnemonic | Function                        | Access | Reset |
|------------|--------------|---------------------------------|--------|-------|
| 7:0        | FixAddr      | The bit 15:8 of program address | R/W    | -     |

## 5.5.53 FixAddr3L

**FixAddr3L (Low byte of Fix Address register 3, SFR address 0xc7, bank: 00)**

| Bit Number | Bit Mnemonic | Function                       | Access | Reset |
|------------|--------------|--------------------------------|--------|-------|
| 7:0        | FixAddr      | The bit 7:0 of program address | R/W    | -     |

## 5.5.54 PageMissEntryEx

**PageMissEntryEx (Extension byte of page miss entry register, SFR address 0xf4, bank: 00)**

| Bit Number | Bit Mnemonic    | Function                         | Access | Reset |
|------------|-----------------|----------------------------------|--------|-------|
| 7:0        | PageMissEntryEx | The bit 23:16 of program address | R/W    | ffh   |

## 5.5.55 PageMissEntryH

**PageMissEntryH (High byte of page miss entry register, SFR address 0xf5, bank: 00)**

| Bit Number | Bit Mnemonic   | Function                        | Access | Reset |
|------------|----------------|---------------------------------|--------|-------|
| 7:0        | PageMissEntryH | The bit 15:8 of program address | R/W    | c0h   |

## 5.5.56 PageMissEntryL

**PageMissEntryL (Low byte of page miss entry register, SFR address 0xf6, bank: 00)**

| Bit Number | Bit Mnemonic   | Function                       | Access | Reset |
|------------|----------------|--------------------------------|--------|-------|
| 7:0        | PageMissEntryL | The bit 7:0 of program address | R/W    | 83h   |

## 5.5.57 CodeReplaceEntryEx

**CodeReplaceEntryEx (Code Replace Entry Address Extension byte register, SFR address 0xa5, bank: 00)**

| Bit Number | Bit Mnemonic       | Function                         | Access | Reset |
|------------|--------------------|----------------------------------|--------|-------|
| 7:0        | CodeReplaceEntryEx | The bit 23:16 of program address | R/W    | ffh   |

## 5.5.58 CodeReplaceEntryH

**CodeReplaceEntryH** Code Replace Entry Address High byte register, SFR address 0xa6, bank: 00)

| Bit Number | Bit Mnemonic      | Function                        | Access | Reset |
|------------|-------------------|---------------------------------|--------|-------|
| 7:0        | CodeReplaceEntryH | The bit 15:8 of program address | R/W    | 0c1h  |

## 5.5.59 CodeReplaceEntryL

**CodeReplaceEntryL** (Code Replace Entry Address Low byte register, SFR address 0xa7, bank: 00)

| Bit Number | Bit Mnemonic      | Function                       | Access | Reset |
|------------|-------------------|--------------------------------|--------|-------|
| 7:0        | CodeReplaceEntryL | The bit 7:0 of program address | R/W    | 83h   |

## 5.5.60 TestCounterByte0

**TestCounterByte0** (Test Counter byte 0, SFR address 0xfc, bank 0x00)

| Bit Number | Bit Mnemonic     | Description        | Access | Reset |
|------------|------------------|--------------------|--------|-------|
| 7:0        | TestCounterByte0 | Test counter [7:0] | R      | 0     |

## 5.5.61 TestCounterByte1

**TestCounterByte1** (Test Counter byte 0, sfr address 0xfd, bank 0x00)

| Bit Number | Bit Mnemonic     | Description         | Access | Reset |
|------------|------------------|---------------------|--------|-------|
| 7:0        | TestCounterByte1 | Test counter [15:8] | R      | 0     |

## 5.5.62 TestCounterByte2

**TestCounterByte2** (Test Counter byte 0, sfr address 0xfe, bank 0x00)

| Bit Number | Bit Mnemonic     | Description          | Access | Reset |
|------------|------------------|----------------------|--------|-------|
| 7:0        | TestCounterByte2 | Test counter [23:16] | R      | 0     |

## 5.5.63 TestCounterByte3

**TestCounterByte3** (Test Counter byte 0, sfr address 0xff, bank 0x00)

| Bit Number | Bit Mnemonic     | Description          | Access | Reset |
|------------|------------------|----------------------|--------|-------|
| 7:0        | TestCounterByte3 | Test counter [31:24] | R      | 0     |

## 5.5.64 TestCounterByte4

**TestCounterByte4** (Test Counter byte 0, sfr address 0xf1, bank 0x00)

| Bit Number | Bit Mnemonic     | Description          | Access | Reset |
|------------|------------------|----------------------|--------|-------|
| 7:0        | TestCounterByte4 | Test counter [40:32] | R      | 0     |

## 5.5.65 bist\_en1

**bist\_en1** (Bist enable register1, SFR address: 0xe1, bank: 0x1)

| Bit Number | Bit Mnemonic    | Description                                      | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7          | dram251_bist_en | Dram251 BIST enable bit<br>0:disable<br>1:enable | R/W    | 0     |
| 6          | pram251_bist_en | Pram251 BIST enable bit<br>0:disable<br>1:enable | R/W    | 0     |
| 5          | Pcm_bist_en     | Pcmram BIST enable bit<br>0:disable<br>1:enable  | R/W    | 0     |
| 4          | Reserved        | Be read as 1 zero.                               | -      | -     |
| 3          | Mu1_bist_en     | Muram1 BIST enable bit<br>0:disable<br>1:enable  | R/W    | 0     |
| 2          | URAM_bist_en    | URAM BIST enable bit<br>0:disable<br>1:enable    | R/W    | 0     |
| 1          | ADCRAM_bist_en  | ADCRAM BIST enable bit<br>0:disable<br>1:enable  | R/W    | 0     |
| 0          | DACRAM_bist_en  | DACRAM BIST enable bit<br>0:disable<br>1:enable  | R/W    | 0     |

## 5.5.66 bist\_en2

**bist\_en2** (Bist enable register 2, SFR address: 0xe2, bank: 0x1)

| Bit Number | Bit Mnemonic      | Description   | Access | Reset |
|------------|-------------------|---|--------|-------|
| 7          | FIR_CS_RAM_bit_en | FIR_CS_RAM BIST enable bit<br>0:disable<br>1:enable | R/W    | 0     |
| 6          | FIR_AA_RAM_b      | FIR_AA_RAM BIST enable bit                          | R/W    | 0     |

|     |                     |  |     |   |
|-----|---------------------|--|-----|---|
|     | ist_en              | 0:disable<br>1:enable  |     |   |
| 5   | FIR_MPX_RAM_bist_en | FIR_MPX_RAM BIST enable bit<br>0:disable<br>1:enable               | R/W | 0 |
| 4   | Reserved            | Be read as 1 zero.   | -   | - |
| 3   | FIR_RDS_RAM_bist_en | FIR_RDS_RAM BIST enable bit<br>0:disable<br>1:enable               | R/W | 0 |
| 2:1 | Reserved            | Be read as 2 zeros   | -   | - |
| 0   | Murom_bist_en       | Murom1/2/3 BIST enable bit <sup>(1)</sup><br>0:disable<br>1:enable | R/W | 0 |

Note:

(1) MUROM BIST uses audio codec clock

## 5.5.67 bist\_fin1

**bist\_fin1 (BIST finish flag register 1, SFR address:0xe4, bank:0x1)**

| Bit Number | Bit Mnemonic     | Description  | Access | Reset |
|------------|------------------|--|--------|-------|
| 7          | dram251_bist_fin | Dram251 BIST finish flag bit<br>0:not finished<br>1:finish | R      | 0     |
| 6          | Pram251_bist_fin | Pram251 BIST finish flag bit<br>0:not finished<br>1:finish | R      | 0     |
| 5          | Pcm_bist_fin     | Pcmram BIST finish flag bit<br>0:not finished<br>1:finish  | R      | 0     |
| 4          | Reserved         | Be read as 1 zero.   | -      | -     |
| 3          | Mu1_bist_fin     | Muram1 BIST finish flag bit<br>0:not finished<br>1:finish  | R      | 0     |
| 2          | Uram_bist_fin    | URAM BIST finish flag bit<br>0:not finished<br>1:finish    | R      | 0     |
| 1          | ADCRAM_bist_fin  | ADCRAM BIST finish bit<br>0:not finished<br>1:finished     | R      | 0     |
| 0          | DACRAM_bist_fin  | DACRAM BIST finish bit<br>0:not finished<br>1:finished     | R      | 0     |

## 5.5.68 bist\_fin2

bist\_fin2 (BIST finish flag register 2, SFR address: 0xe5, bank: 0x1)

| Bit Number | Bit Mnemonic         | Description   | Access | Reset |
|------------|----------------------|---|--------|-------|
| 7          | FIR_CS_RAM_bit_fin   | FIR_CS_RAM BIST finish flag bit<br>0:not finished<br>1:finish | R      | 0     |
| 6          | FIR_AA_RAM_bit_fin   | FIR_AA_RAM BIST finish bit<br>0:not finished<br>1:finished    | R      | 0     |
| 5          | FIR_MPX_RAM_bist_fin | FIR_MPX_RAM BIST finish bit<br>0:not finished<br>1:finished   | R      | 0     |
| 4          | Reserved             | Be read as 1 zero.  | -      | -     |
| 3          | FIR_RDS_RAM_bist_fin | FIR_RDS_RAM BIST finish bit<br>0:not finished<br>1:finished   | R      | 0     |
| 2          | Murom1_fin           | Murom1 BIST finish flag bit<br>0:not finished<br>1:finish     | R      | 0     |
| 1          | Murom2_fin           | Murom2 BIST finish flag bit<br>0:not finished<br>1:finish     | R      | 0     |
| 0          | Murom3_fin           | Murom3 BIST finish flag bit<br>0:not finished<br>1:finish     | R      | 0     |

## 5.5.69 bist\_info1

bist\_info1 (BIST information register 1, SFR address:0xe7, bank:0x1)

| Bit Number | Bit Mnemonic      | Description                                      | Access | Reset |
|------------|-------------------|--|--------|-------|
| 7          | dram251_bist_info | Dram251 BIST information bit<br>0:pass<br>1:fail | R      | 0     |
| 6          | pram251_bist_info | Pram251 BIST information bit<br>0:pass<br>1:fail | R      | 0     |
| 5          | Pcm_bist_info     | Pcmram BIST information bit<br>0:pass<br>1:fail  | R      | 0     |
| 4          | Reserved          | Be read as 1 zero.                               | -      | -     |

|   |                  |   |   |   |
|---|------------------|---|---|---|
| 3 | Muram1_bist_info | Muram1 BIST information bit<br>0:pass<br>1:fail | R | 0 |
| 2 | uram_info        | URAM BIST information bit<br>0:pass<br>1:fail   | R | 0 |
| 1 | ADCRAM_bist_info | ADCRAM BIST information bit<br>0:pass<br>1:fail | R | 0 |
| 0 | DACRAM_bist_info | DACRAM BIST information bit<br>0:pass<br>1:fail | R | 0 |

## 5.5.70 bist\_info2

bist\_info2 (BIST information register 2, SFR address:0xe9, bank: 0x1)

| Bit Number | Bit Mnemonic          | Description  | Access | Reset |
|------------|-----------------------|--|--------|-------|
| 7          | FIR_CS_RAM_bist_info  | FIR_CS_RAM BIST information bit<br>0:pass<br>1:fail  | R      | 0     |
| 6          | FIR_AA_RAM_bist_info  | FIR_AA_RAM BIST information bit<br>0:pass<br>1:fail  | R      | 0     |
| 5          | FIR_MPX_RAM_bist_info | FIR_MPX_RAM BIST information bit<br>0:pass<br>1:fail | R      | 0     |
| 4          | Reserved              | Be read as 1 zero.                                   | -      | -     |
| 3          | FIR_RDS_RAM_bist_info | FIR_RDS_RAM BIST information bit<br>0:pass<br>1:fail | R      | 0     |
| 2          | Murom1_info           | Murom1 BIST information bit<br>0:pass<br>1:fail      | R      | 0     |
| 1          | Murom2_info           | Murom2 BIST information bit<br>0:pass<br>1:fail      | R      | 0     |
| 0          | Murom3_info           | Murom3 BIST information bit<br>0:pass<br>1:fail      | R      | 0     |

## 6 PMU 模块（冯崧祥、肖丽荣、彭祥、江力）

| 日期         | 版本     | 描述  | 修订人 |
|------------|--------|---|-----|
| 2012-07-24 | V1. 00 | initial   | 冯崧祥 |
| 2012-08-09 | V1. 01 | <ul style="list-style-type: none"> <li>1. 根据 analog 的做法修改 LDOPD 寄存器</li> <li>2. 修改 SYSTEM_ONOFF 寄存器：增加 UVLOB_SEL 使能位，增加 DC5V_DOWN_SEL 下拉选择位，增加 EN_VDD_SL 偏置电流选择位</li> <li>3. 增加断码屏关闭时的描述</li> <li>4. 根据 analog 需求，修改 debug 信号排布</li> <li>5. 根据 digital 做法修改 system on 部分描述</li> <li>6. 根据 analog 参数，小幅修改 onoff 电路框图</li> <li>7. 修改 onoff 待机及唤醒描述</li> <li>8. 修改 PMUADC_CTL 寄存器，将 LRADC 的使能位改成 1 个</li> <li>9. 修改 TEST_CTL 寄存器</li> <li>10. 修改 FSOURCE 负载能力描述</li> <li>11. 增加 RTCVDD 负载能力 1Ma</li> </ul> | 冯崧祥 |
| 2012-09-12 | V1. 02 | <ul style="list-style-type: none"> <li>1. FSOURCE 负载能力修正为 30Ma</li> <li>2. FSOURCE 下拉修正为 2Ma</li> <li>3. AVDD 下拉修正为 400Ua</li> <li>4. 增加描述，上电后软件关闭下拉</li> <li>5. VCC 的负载能力修正为 drop 5%</li> <li>6. 修改 BANDGAP 寄存器默认值</li> <li>7. 修改 bandgap 烧写描述</li> <li>8. 修正 BDG_CTL 寄存器的 reserved 位为模拟用可读写</li> <li>9. EFUSE_CTL 的 reserved 位改为只读（即不做）</li> <li>10. 修正 CP_CTL1 的 reserved 位，将 bit0 去掉</li> <li>11. 修改 MULTI_USD 寄存器，断码屏默认值关</li> <li>12. 修改 TEST_CTL 寄存器，增加数码管偏置位</li> </ul>                         | 冯崧祥 |
| 2012-10-20 | V2. 00 | <ul style="list-style-type: none"> <li>1. 增加 TEST_CTL 寄存器，<b>HDSWDET</b> 的描述</li> <li>2. 修改 EFUSE0 bit7-5 的功能及默认值</li> <li>3. 修改 EFUSE1 bit6,7 Reserved 位的描述，改为 <b>for future use</b></li> <li>4. 修改 TEST_CTL 寄存器第 4 位的冗余定义</li> </ul>  | 冯崧祥 |
| 2013-01-16 | V2. 03 | <ul style="list-style-type: none"> <li>1. 根据模拟实际所做，修改 VDDOK 到 preok 的时间为 16Ms，原 16-32Ms</li> </ul>  | 冯崧祥 |

## 6.1 Features

The GL5115 integrates a comprehensive power supply subsystem, designed to minimize external component requirements .including the following features:

- Supports Li-Ion battery power supply.
- Charge pump power supply for USB HOST.
- Seven linear regulators supply power. Output VCC, AVCC, VDD, AVDD, RTCVDD, FSOURCE& UVDD.
- Internal bandgap using AVCC power supply.
- Low precision A/D converters for Battery voltage monitor, system monitors for wire-controller.

## 6.2 Function Description

Figure 6-1 shows a functional block diagram of the power supply components including six linear regulators; as well as 2~5 A/D converters for battery monitoring and wire-controller monitoring.

The figure is intended to give a better understanding of the hardware connections between each subsystem; they are not intended to be a complete architecture description.

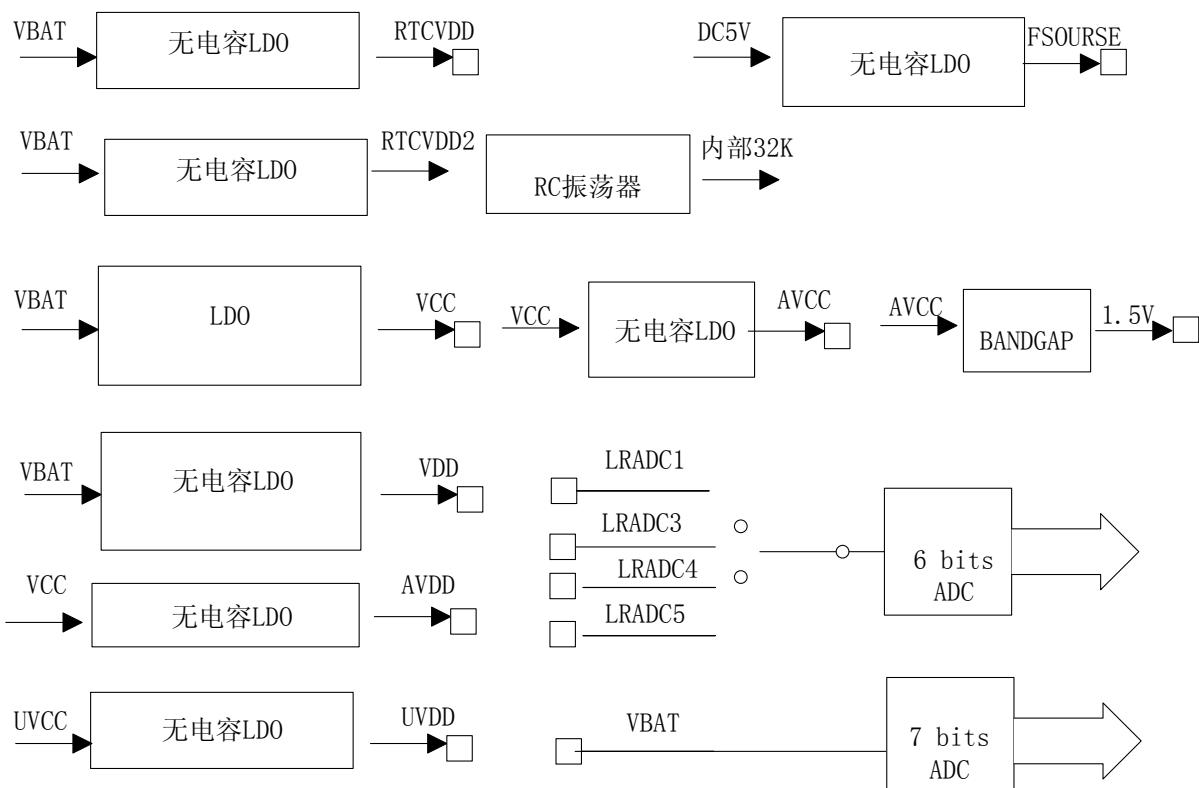


Figure 6-1 Power Supply Block Diagram

## 6.3 Module Description

### 6.3.1 Charge Pump

GL5115 集成了一个 2 倍的 charge pump 作为 HOST 的电源, 输出电压 5V, 在输入电压 3.3V 时负载能力 200mA。CP 框图如下:

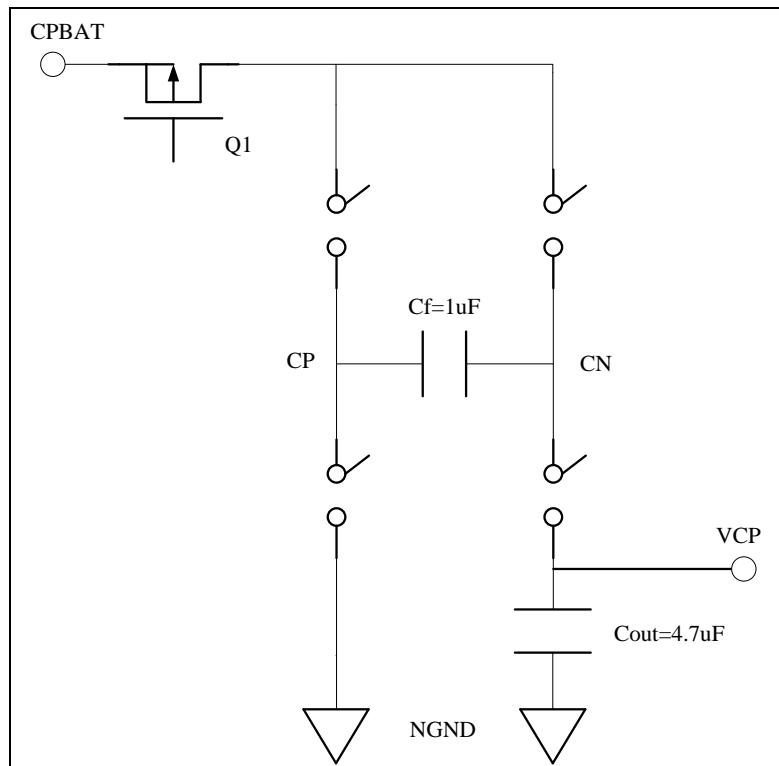


Figure 6-2 charge pump block diagram

CP 具有软启动功能, 在启动时, 输入和输出之间直通, 输出电压达到 0.8VIN 时开始启动电荷泵。整个 CP 启动过程需要保证 CP 的输入端总电流小于 1A。

CP 的效率随输入电压升高而降低。在输入电压比较高时, 为提高效率, 可以开启直通功能。当输入电压大于 4.7V (可寄存器设置) 并持续 debounce 16ms, CP 的输入和输出直通。一旦输入电压小于 4.5V (可寄存器设置), 又立刻启动 charge pump。

注意: 如果要外灌电压, 一定要等到 PWROK 以后, 不然会出现 VCP->CPBAT 的漏电。

### 6.3.2 Linear Regulators

#### 6.3.2.1 Regulators Architectonics

There are six integrated linear regulators. Architecturally, four regulator generates VCC&VDD&RTCVDD&FSOURCE from the VBAT pin, one generates AVCC from VCC (used in analog circuit for fewer ripples), and the other generate UVDD from the UVCC supply. Therefore, all of the current is supplied by VBAT.

VCC/VDD在POWEROK前上电有软启动，VCC另有1K电阻分压成2/3,1/3 VCC来驱动断码屏，各有1Ma驱动能力。

### 6.3.2.2 Regulators Output Voltage Set

When you set VCC, AVCC is the same set. There is a register control drop voltage between VCC and AVCC, default drop voltage is 0.15V.

### 6.3.2.3 Regulators Accurate and Maximum Output Current

The output voltages are highly precise within  $\pm 2\%$ , They provide large currents with a significantly small dropout voltage within  $\pm 5\%$ . [Table1](#) shows data of maximum output current.

| Block name        | Loading   |
|-------------------|---|
| VCC Regulator     | BAT=3.4V, <b>250Ma</b> @ VCC=3.1V 下降 5%(瞬态要求挂 2.2uf 电容, 1us 内, 100Ma 负载波动小于 2%) |
| VDD Regulator     | BAT=3.4V, <b>80Ma</b> @ VDD=1.8V 下降 5%  |
| AVCC Regulator    | VCC=3.1V, <b>70Ma</b> @ AVCC=2.85V 下降 2%  |
| AVDD Regulator    | VCC=3.1V, <b>2Ma</b> @ AVDD=1.7V 下降 2%  |
| UVDD Regulator    | UVCC=3.1V, <b>70Ma</b> @ UVDD=1.7V 下降 2%  |
| RTCVDD Regulator  | VBAT=3.4V, <b>5Ma</b> @ RTCVDD=1.8V 下降 10%                                      |
| RTCVDD2 Regulator | VBAT=3.4V, <b>@ RTCVDD2=1.8V</b>  |
| FSOURCE Regulator | DC5V=4.5V, <b>30Ma</b> @ FSOURCE =3.8V 下降到 3.7V, 整个负载范围内波动时 FSOURCE 不可小于 3.7V   |

Table 1 Regulators Maximum Output Current

### 6.3.2.4 Regulators Power Down

If the system is to operate from an external power supply, then the internal linear regulators are powered down automatically.

## 6.3.3 Reference Voltage

The GL5115 need a precise reference voltage-1.5V qua benchmark of all voltage. This reference voltage is generated by an external linear regulator or internal bandgap , which should have highly accuracy with  $\pm 1\%$ . If the reference voltage has changed, the voltage of VCC, VDD, AVCC, AVDD,FSOURCE and others mentioned before have changed.

### EFUSE

There is a build-in 8bits electrical poly-fuse macro organized in 1-8 liner array, whose standby current

is 5UA, operating current is 4mA. Its block diagram as follow:

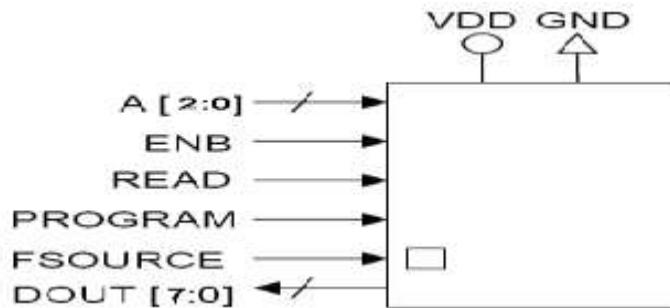


Figure 6-3 EFUSE Block Diagram

**注意：**上电时，在发出PREOK 之后，硬件自动把32位EFUSE 的输出数据加载到内部的寄存器中，再把数据送给各个模块。Efuse数据只是在上电时加载一次，烧写后要重新上电才能生效。BANDGAP的EFUSE上电自动load，控制USB 6.2K电阻的EFUSE，在brom中调用。

#### BANDGAP

There is a build-in 1.5V-reference voltage output—VBG, It is standard of all analog circuit, its range is from 1.38V to 1.6125V, the minimum step is 7.5Mv。可以通过 [REG\[BDG\\_FILTER\]](#)来设定 BANDGAP 的输出端是否接电阻滤波。可以通过将 bandgap 电压设为最低来使用外部 1.5V LDO 供电.

#### 6.3.3.1 EFUSE 与 BANDGAP 的关系及操作过程

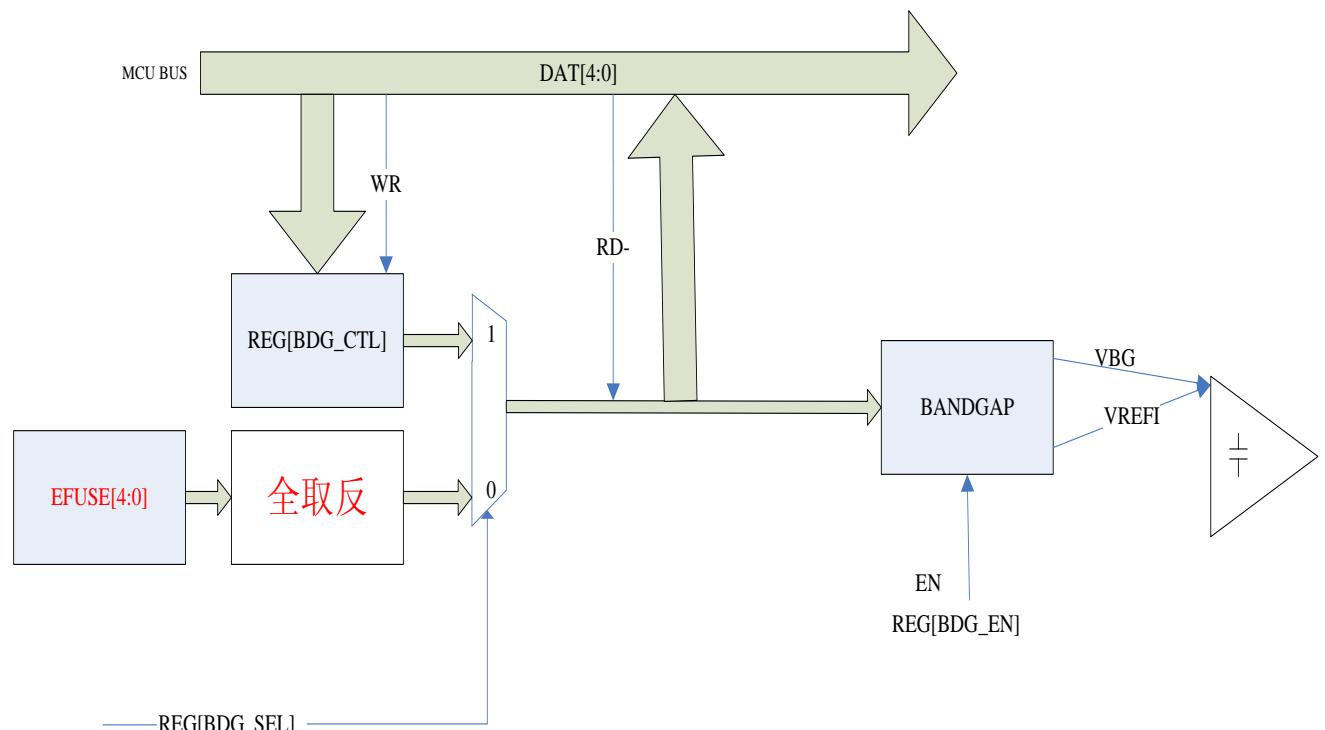


Figure 6-4 EFUSE & Bandgap Controller Diagram

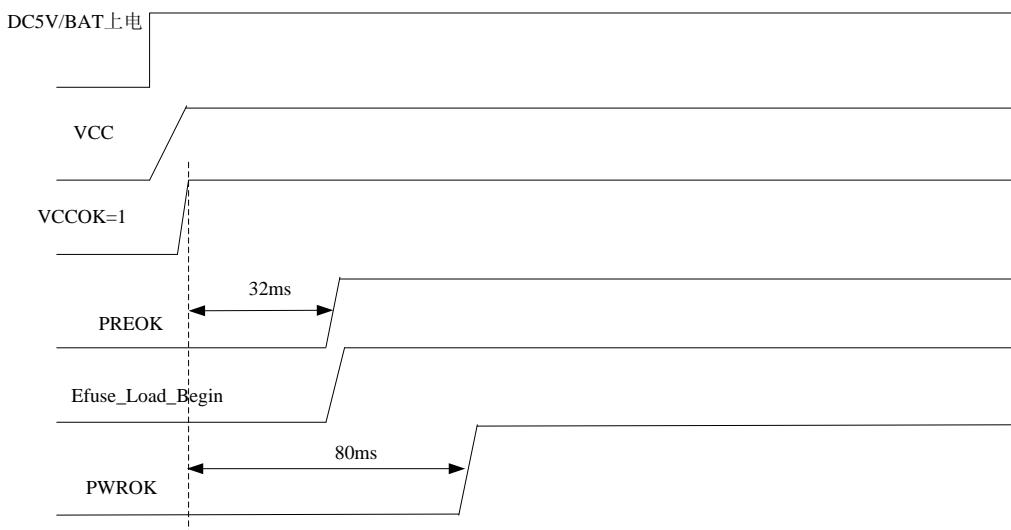


Figure 6-5 上电 EFUSE 自动 LOAD 时序

PREOK 在 VCCOK 后 32ms 发出。

操作流程:

- 1、Wafer 测试时先切为寄存器控制, 测试出 REG[BDG\_CTL.bit4:0]默认值 13 (0DH) 对应的 BANDGAP 输出电压, 记录为 VBG1, VBG1 可校准范围 1.3575V---1.605V;
- 2、根据 VBG1 计算所需写入的数据, 计算方法为: 十进制数据{13+[(1.5V-VBG1)/7.5Mv]}四舍五入后  
再取反所对应的十六进制数据, 将此数据配到 BDG\_VOL 中, 并使能寄存器控制 bandgap, 确认此数据可使 BANDGAP 输出 1.5V±7.5Mv;
- 3、选择烧写 PMU 模块的 EFUSE, 再将 Fsource 拉高到 3.8V, 对 Efuse 写入所需数据, 烧写成功后完全掉电并重启, 测量 VBG, 确认此数据可使 BANDGAP 输出 1.5V±7.5Mv;
- 4、在 BROM 或应用程序中设置 REH[BDG\_SEL]=1 选择 REG[BDG\_CTL]控制 BANDGAP, 并通过设置 EFUSE\_CTL 寄存器关闭 Efuse。

若想在Package IC上测试BANDGAP的其它电压, 可以设置REG[BDG\_CTL.bit4:0]为不同值, 从而得到不同的BANDGAP电压输出。

### 6.3.4 A/D Converters

There is a low resolution 7 bit A/D for Battery monitor, The input voltage range of which is 1.4 to 4.4V at VBAT pin in Li-Ion supply mode.

There is a low resolution 6 bit A/D for wire control.. The input voltage range of which is 0 to VCC at REM\_CON pin.

The all A/D converter's working frequency is 128HZ default, you can reduce the working frequency down to 64HZ by setting [REG\[ADC\\_FS\]=0](#).

The impedance between BAT(or REM\_CON or TEMP) and GND is up to MΩ.

The output data of BATADC can be calculated as the following formula:

LI-ION BATADC:

$$\text{一个 LSB} = \frac{4.4 - 1.4}{2^7} * \frac{Vref}{1.5}, \text{ 当电池电压等于 VBAT 时, ADC 的数据 } n = \frac{\frac{Vbat - 1.4 * \frac{Vref}{1.5}}{4.4 - 1.4} * \frac{Vref}{1.5}}{2^7}$$

其中 Vref 为实际测试到的参考电压的值。

例如 Vref=1.500V, 那么 1LSB=23.44Mv, 那么从 1.4V 到 1.42344V 对应的数据都为 00h, 从 1.42344V 到 1.44688V 对应的数据为 01h。

#### LRADC1/LRADC3/LRADC4/LRADC5:

$$\text{一个 LSB} = \frac{VCC}{2^6}, \text{ 当输入电压等于 } V \text{ 时, 对应的 ADC 数据 } n = \frac{\frac{V}{VCC}}{2^6}$$

例如 VCC=3.1V, 那么 1LSB=48.44Mv, 那么从 0V 到 0.04844V 对应的数据都为 00h, 从 0.04844V 到 0.9688V 对应的数据为 01h。

LRADC1/LRADC3/LRADC4/LRADC5 分时复用 1 个 ADC。并且 LRADC3 与 [GPIO\\_A5](#) 复用, LRADC4 与 [GPIO\\_A6](#) 复用, LRADC5 与 [GPIO\\_C7](#) 复用。

### 6.3.5 SYSTEM ON/OFF

#### 6.3.5.1 System ON/OFF 模块框图

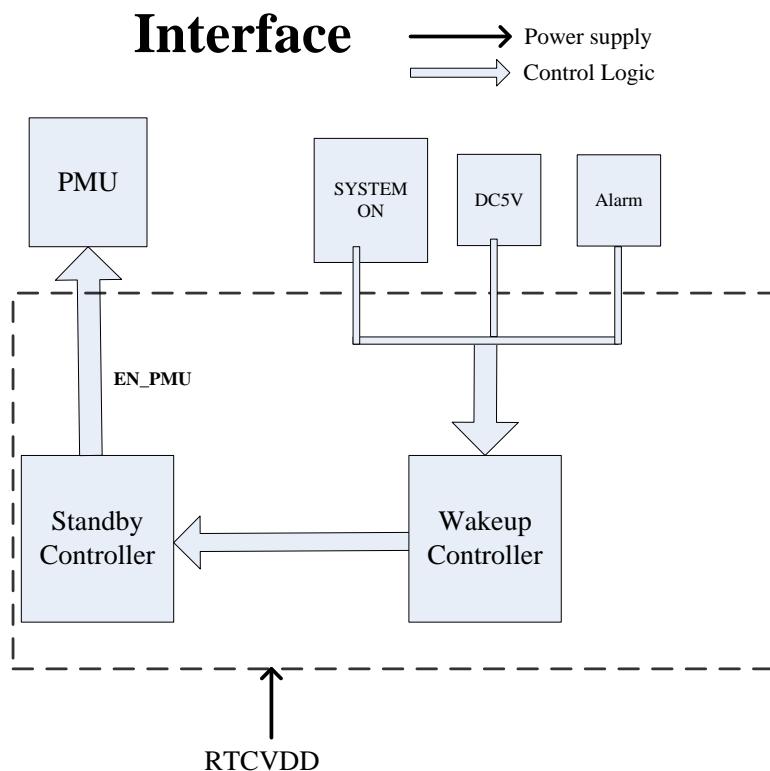


Figure 6-6 System On-Off 模块框图

RTCVDD 电压随工艺、温度和输入电压的变化，范围 1.4-2.0V，典型值 1.7V。

PMU 模块所需的时钟 CK32KD(RTCVDD 域)，由 digital 通过寄存器选择内部 RTCVDD 域的 32K 信号 RTCK32KD 和外部低频信号产生，默认使用内部 32k。

**SYSTEM\_ON 唤醒信号产生说明：**

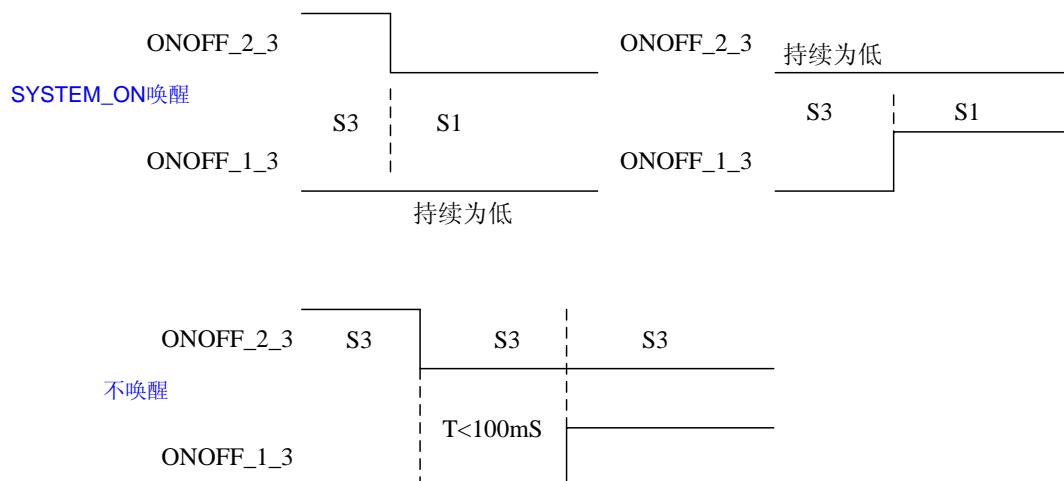


Figure 6-7 SYSTEM\_ON 唤醒信号产生说明

### 6.3.5.2 System ON/OFF 电路框图

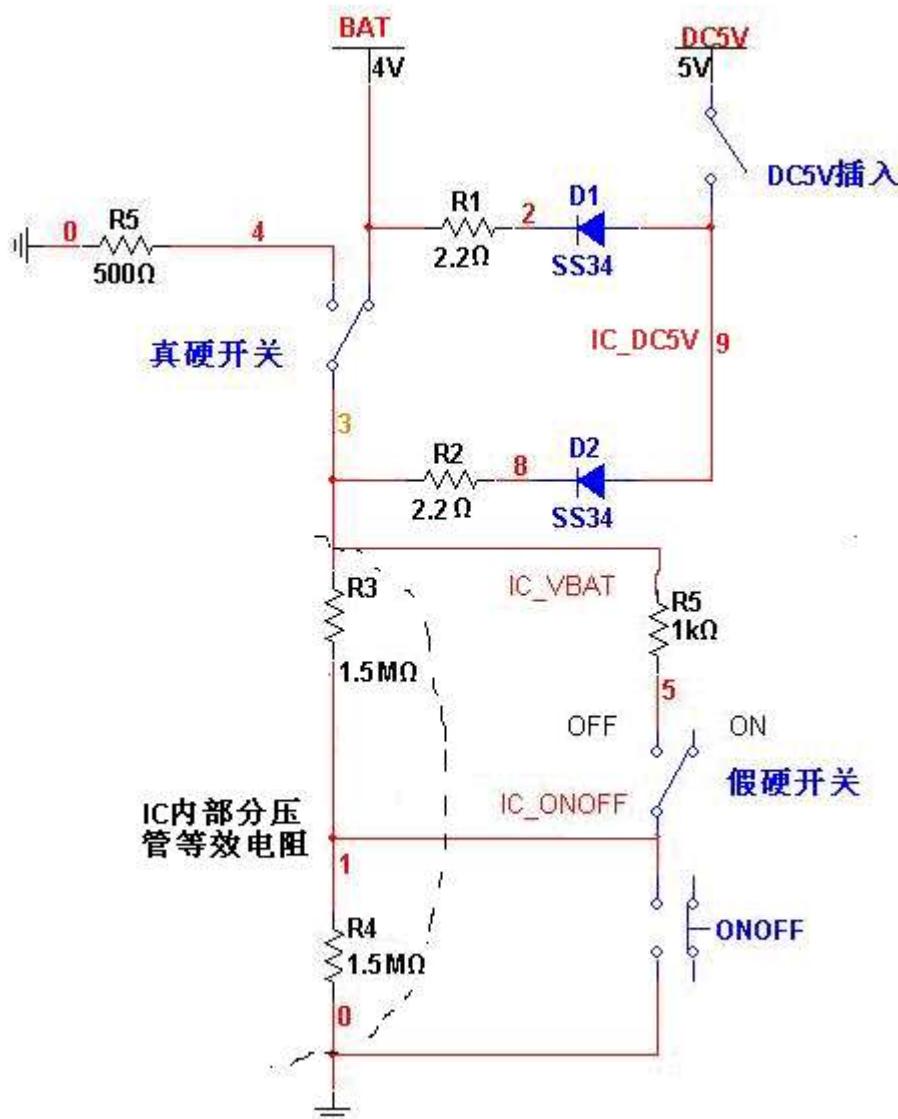


Figure 6-8 System On-Off 电路框图

### 6.3.5.3 ON/OFF 多功能键描述

当 Figure 6-7 中 ONOFF 键被按下时, 由 Analog 电路负责检测按键状态 (上升沿 debounce 16Ms), 送到 digital 经过 4Ms debounce 后给到寄存器, 系统在 S1 时, 按键时间满足软件定义后, 需保存断点, 关闭屏幕, 将各模块至于 reset 状态, 当按键抬起时, 可延时一小段 (保证断点已存完), 写 ENPMU=0; 当按键时间超过 SFR:0x9d, SFR bank 0x05 bit 2:1 定义时间后, 仍没有抬起, 系统 reset。

### 6.3.5.4 System ON/OFF 状态转换描述

#### Standby Mechanism:

在 GL5115 中，会有一个 Standby 单元来管理系统的唤醒工作。当系统由 Runtime 进入 Standby 时，系统的其他模块完全断电，只有和 Standby 相关的电路保持着电源，以便系统能够被唤醒。

为了准确的说明系统不同的状态，定义 4 种不同的状态。

- Runtime(S1)——系统最大耗电状态，系统完全上电，VCC/VDD 都有电，正常运行程序。EN\_PMU=1
- DC5V\_Standby 状态(S2)——系统较小耗电状态，软件处理各个模块进 standby，但 VCC/VDD 都还有电，MCU 可继续低速运行，保持总耗电在 2.5mA 以内即可,EN\_PMU=1
- Standby 状态(S3)——系统最小耗电状态，VCC/VDD 掉电，只有 RTCVDD 有电。可通 DC5V、Alarm、ONOFF 按键、硬开关来唤醒系统。EN\_PMU=0
- 无电状态(S4)——系统无电状态，不存在 RTCVDD。

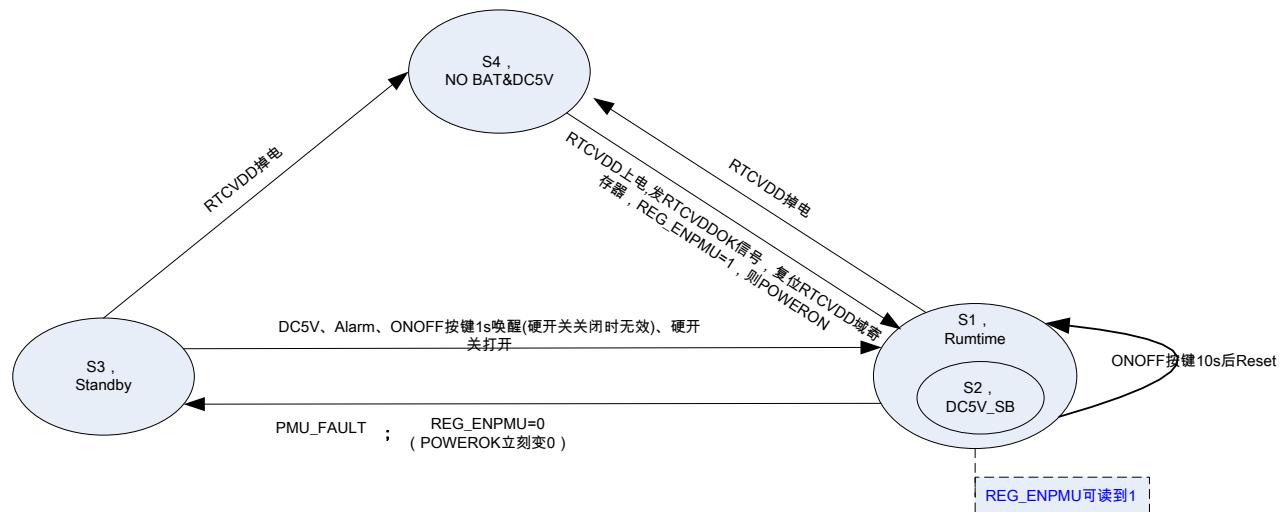


Figure 6-9 状态转换图

1. 系统最开始处于 S4，当接入 DC5V 或者 BAT 时，RTCVDD 上电，上电完成后发出 RTCVDDOK 信号。进入 S1 的 runtime 状态，所有电源均有电，开始运行 BROM 程序。
2. 当系统处于 S3 时，如果此时插入 DC5V、按 ONOFF 按键（假硬开关没有关闭时），或者 Alarm 唤醒信号有效，或者假硬开关打开，系统都会从 S3 转换到 S1。
3. 当系统处于 S1 时。如果没有 DC5V，软件检测到了长按 ONOFF 键，或 HDSWDET 信号（硬开关关闭），软件需要保存断点，然后将 REG\_ENPMU 设置为 0，即 disable VCC/VDD，使系统进入 S3。如果有 DC5V，长按 ONOFF 键，不要将 REG\_ENPMU 设置为 0 而要保持为 1，接下来需要软件进行特殊处理，使系统进入到 S2 状态。在 S2 状态下，MCU 仍然可以工作，VCC/VDD 仍然有电，只是软件降功耗到 2.5mA 以内（认证时需要这样处理）。
4. 当系统处于 S2 时。此时如果长按 SYSON 键，软件可恢复到 S1。当检测到拔掉 DC5V 或 D+/D-(DP,DM) 的动作，则软件可通过 REG\_ENPMU=0 来进入 S3 状态，或者恢复到 S1 状态。
5. 当系统处于 S1 或者 S2 时。如果超长按 ONOFF 键，系统重新启动。
6. 当系统处于 S4 时。ONOFF 不起任何作用，因为此时 RTCVDD 没有电。

7. IC 在从 S3 到 S1 状态转换时有上电复位，把所有的 VDD 电压阈下的寄存器复位到默认值，然后运行 BROM，不能复位 RTCVDD 域下的寄存器。

9. PMU\_FAULT 作为进 standby 的源，高有效。其中包括：1 过流——powerOK 后有效；2 电池低电 LB；3 系统低电 LVPROT——DC5V 下无效，EN\_PMU 后 delay 一段时间后如果 powerOK 依然为 0，则发 Time

protect 信号，受 EN\_PMU=0 来 reset。

2012. UVLOWPD 唤醒信号，包括 UVLOA：当系统处于 S3 时，当 DC5V 大于 VBAT+0.14V，且持续 16ms；UVLOB(可通过 UVLOB\_SEL 关闭)：每 1 秒内对 DC5V 拉 4Ms 负载(由 DC5V\_DOWN\_SEL 设置，默认 4K)，检测到有 DC5V>BAT-0.05 且持续 2Ms 以上，则 UVLOWPD 由 0 变 1，开始唤醒到 S1。

### 6.3.5.5 System ON/OFF 时序图

#### 6.3.5.5.1 RTCVDD 上电时序

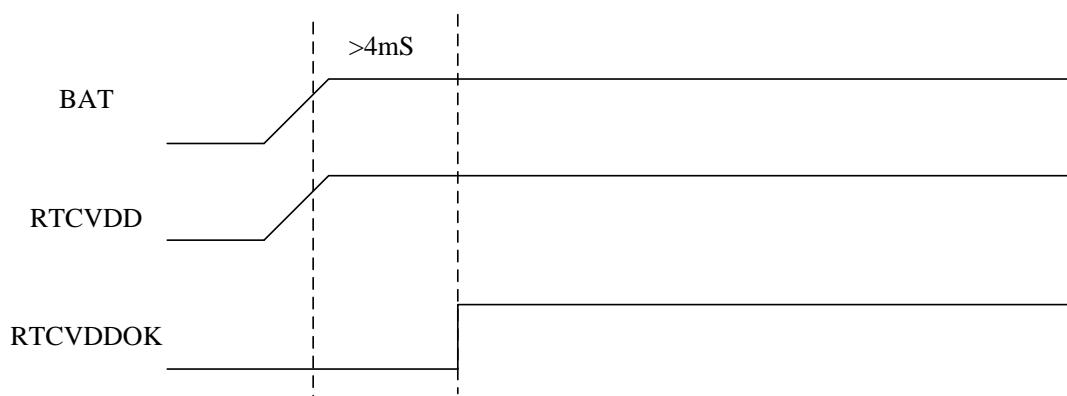


Figure 6-10 RTCVDD 上电时序

#### 6.3.5.5.2 正常上电 VCC/VDD 情况

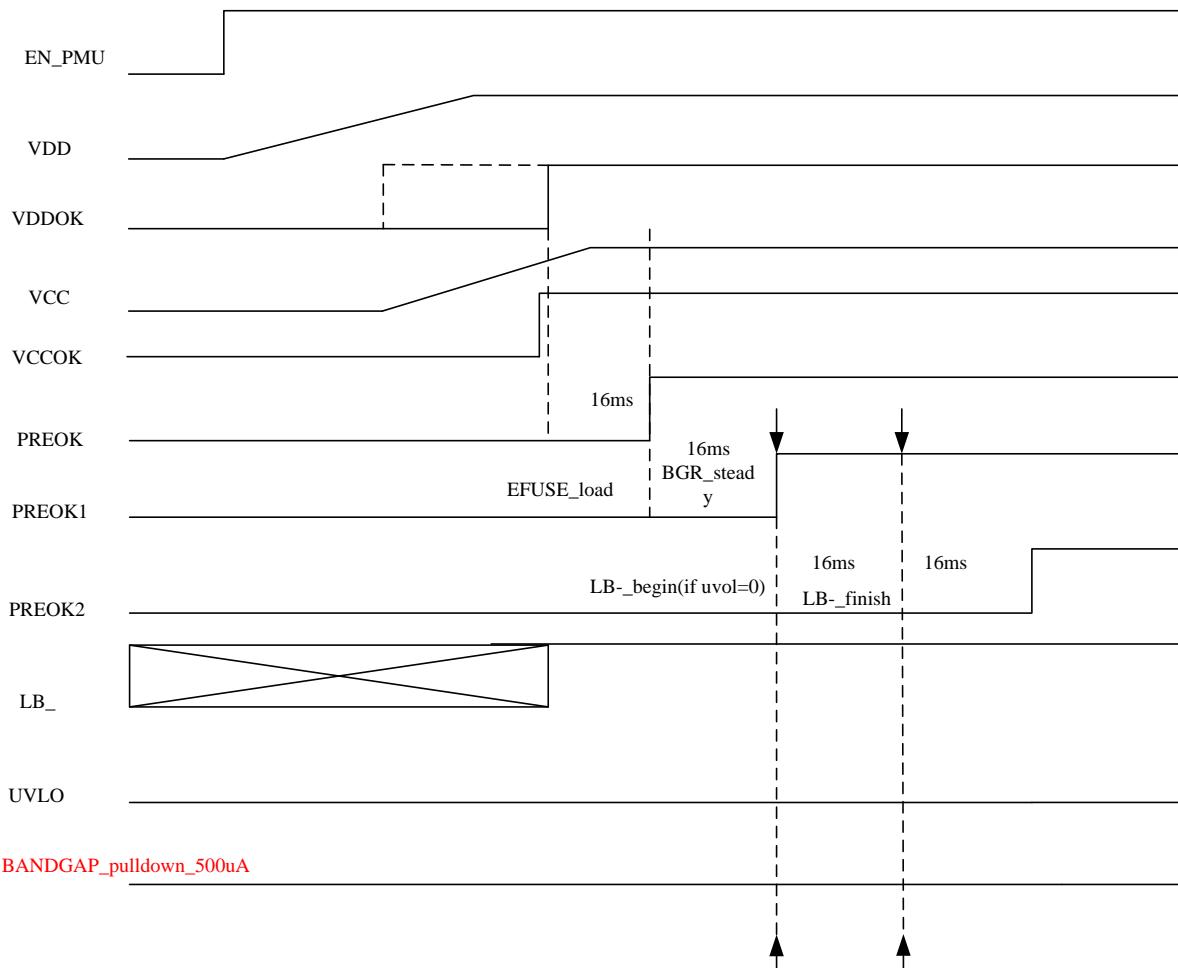


Figure 6-11 VDD/VCC 正常上电

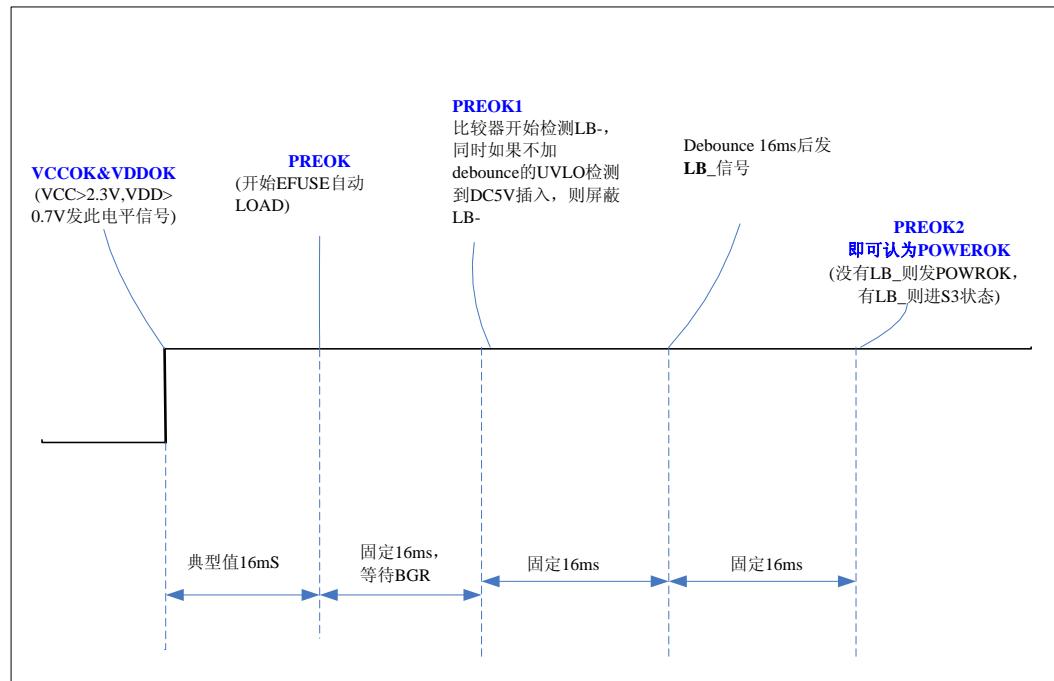


Figure 6-12 PMU\_PWROK 信号关系

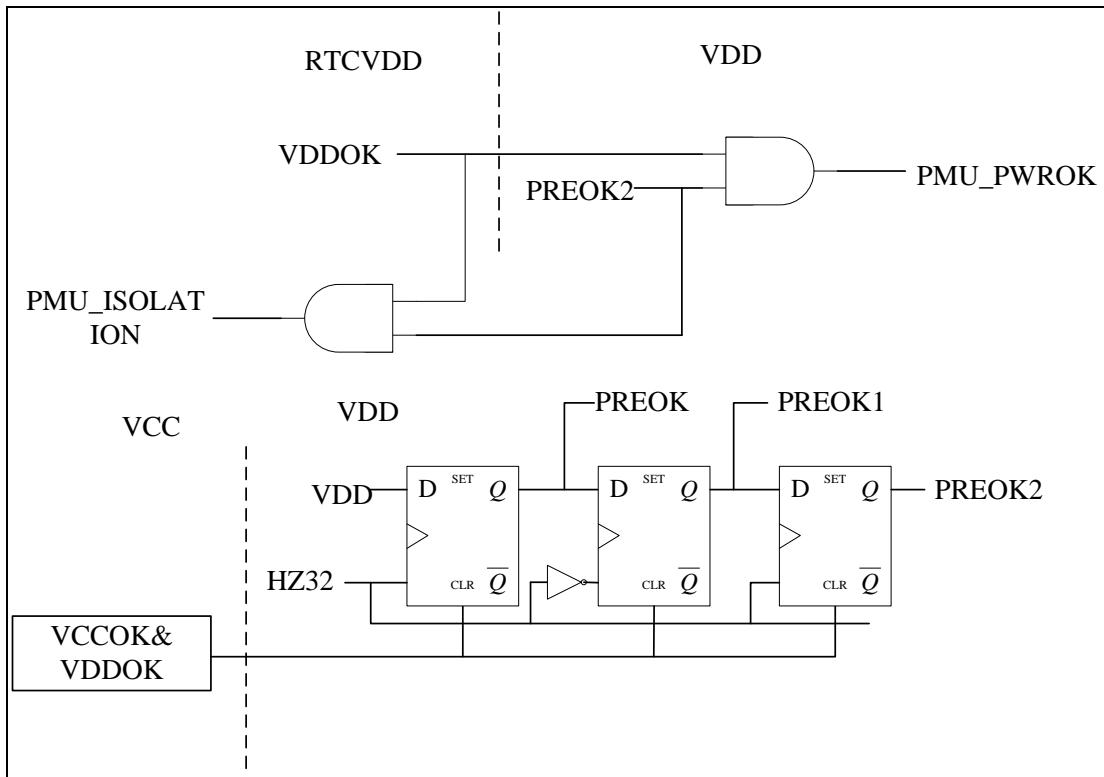


Figure 6-13 PMU\_PWROK 和 ISOLATION (隔离)信号产生逻辑关系

在 PREOK2 后经过几个 ms 发 PREOKA 信号。

Power ok 后到 digital 后用 32k 抓 4 拍。

所以最长的上电时间为  $80\text{ms} + 16\text{ms} = 96\text{ms}$ 。

### 6.3.5.3 外灌 VCC 或多节干电池

外灌电 VCC 时需要能 work, test 以及外挂电源方案。

- 有电池 PIN 包出来, 4 节电池外挂 VCC LDO 的方案可将 VCC 与 VBAT 短接产生 RTCVDD, 那么电池低电电压 LB\_需要设置较低如 2.8V 等。这时候 VCC 一直有电, **VDD 必须有电**。当 VCC/VDD ok 后会发 powerok 信号, 系统 reset 完成后, MCU 开始工作。
- 有电池 pin 包出来, 双节干电池方案时, 外挂 DCDC 产生 VCC, 那么只能将 VCC 与 VBAT 短接后给 RTCVDD 供电, 否则 BAT 浮空等会出现低电复位进 S3。其余同上。

### 6.3.5.6 Reset 复位 PMU 说明描述

The reset mode includes RTCVDDOK reset, POWEROK reset, Watchdog reset, Soft reset.

按照电压域分:

| RESET 分类 | RTCVDDOK | POWEROK | Watchdog | ONOFF |
|----------|----------|---------|----------|-------|
|          |          |         |          |       |

|                                      | reset  | reset | reset | reset  |
|--------------------------------------|--------|-------|-------|--------|
| RESET 信号所处电压域                        | RTCVDD | VDD   | VDD   | RTCVDD |
| 是否复位 VDD 电压域寄存器                      | NO     | YES   | YES   | YES    |
| 是否复位 PMU/CMU/RTC 模块<br>RTCVDD 电压域寄存器 | YES    | NO    | NO    | NO     |
| 调试口                                  | NO     | YES   | NO    | NO     |
| MCU 等                                | NO     | YES   | YES   | YES    |

Table 2 RESET 复位说明

## 6.4 Operation Manual

### 6.4.1 软件进 standby (S2) 流程

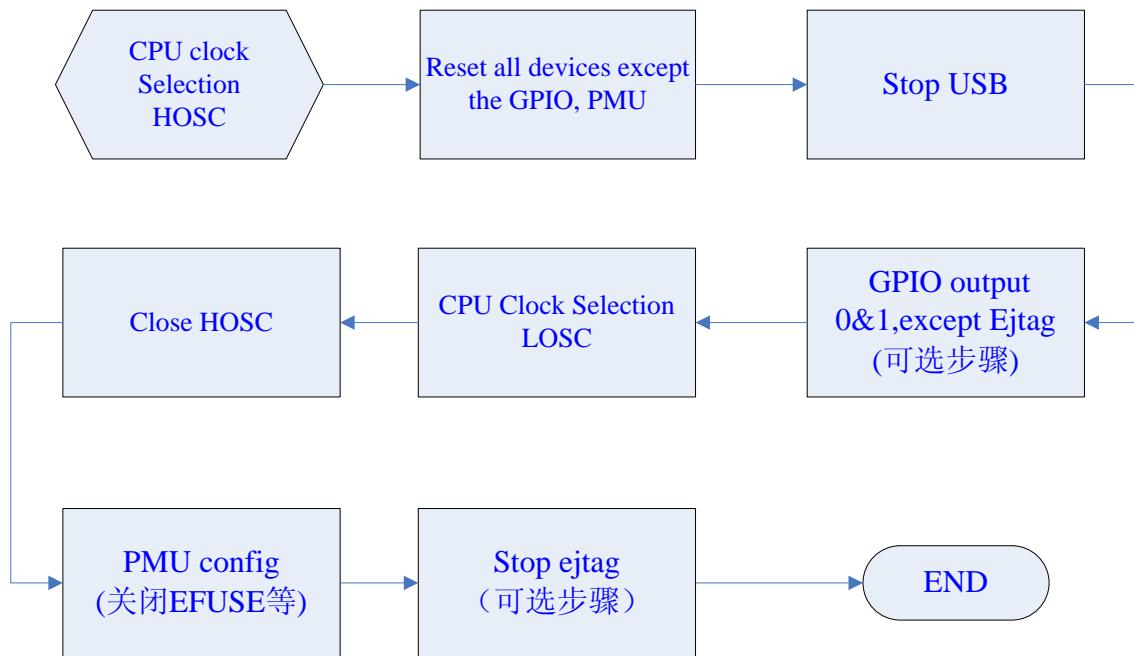


Figure 6-14 STANDBY 软件控制流程图

### 6.4.2 软件关机 (S3) 流程

注意：系统关机都由软件来实现，且 UVLO=1 时不进 S3，只可进 S2

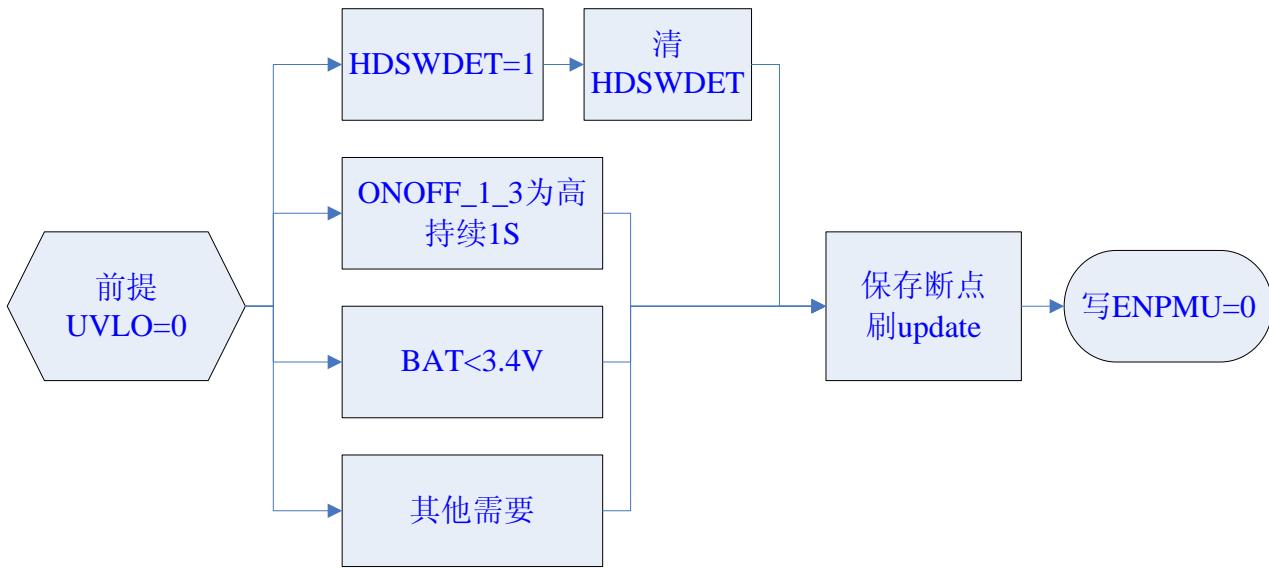


Figure 6-15 软件关机流程图

### 6.4.3 软件 SDK 初始化流程

在 BROM 上电完成，SDK 初始化软件应用界面时，应关闭部分为稳定启动而设的 PMU 寄存器，具体列表如下：(均为 bank: 05)

| SFR  | BIT | NAME    | Description                      | 应设为 |
|------|-----|---------|----------------------------------|-----|
| 0x8a | 3:2 | AVCCPD  | AVCC 无电容 LDO 弱下拉，上电默认值 01，1Ma 下拉 | 00  |
| 0x8a | 1:0 | VDDPD   | VDD 无电容 LDO 弱下拉，上电默认值 01，1Ma 下拉  | 00  |
| 0x8b | 4   | BDG_PDR | BANDGAP 下拉电阻控制，上电默认值 1，有下拉电阻     | 0   |

### 6.4.4 按键与硬开关检测

2015、ONOFF\_1\_3 指示软开关的按下，ONOFF\_2\_3 指示硬开关的断开与闭合；  
 IC 外部限流电阻接法决定两个开关优先级的高低，两个指示位不会同时为高。

2. 硬开关合上过一次，HDSWDET 就会为高，直到手工清除或重启；  
 软件检测到此位，应关机或执行其他操作；

## 6.4.5 各电源的开启方法

| 电源          | 开启方法  |
|-------------|---|
| UVDD        | 0x8d , bit7, UVDD_EN, 设为 1  |
| 断码屏电源       | 0x8d , bit3, SEG_DISP_VCC_EN, 设为 1  |
| 数码管 bias    | 0x9a , bit4, SEG_BIAS_EN, 设为 1, 在使用数码管 LED 前须打开此位   |
| Charge pump | 0x98 , bit5, CL_SEL, 设为 0, 先设置限流为 400Ma;<br>0x98 , bit6, EN_CS, 设为 1, 开启限流;<br>0x98 , bit3, THRO_CP, 设为 1, 开启直通功能<br>0x97 , bit0, EN_2XCP, 设为 1, 最后开启 charge pump |

## 6.5 PMU Register List

The Address of PMU Controller Register Group = SFR:0X89, SFR BANK: 0x05.

| Index | Mnemonic        | Description                               | BANK | 电压域    |
|-------|-----------------|---|------|--------|
| 0x89  | VOUT_CTL        | VCC/VDD voltage set Register              | 0x05 | VDD    |
| 0x8a  | LDOPD_CTL       | Capless LDO pulldown control              | 0x05 | VDD    |
| 0x8b  | BDG_CTL         | Bandgap enable Register                   | 0x05 | VDD    |
| 0x8c  | BDG_VOL         | Bandgap voltage Register                  | 0x05 | VDD    |
| 0x8d  | MULTI_USED      | GPIO multi-used set Register              | 0x05 | VDD    |
| 0x90  | PMUADC_CTL      | PMU ADC frequency and enable Register     | 0x05 | VDD    |
| 0x91  | BATADC_DATA     | BATADC data Register                      | 0x05 | VDD    |
| 0x92  | LRADC1_DATA     | LRADC1 data Register                      | 0x05 | VDD    |
| 0x93  | LRADC3_DATA     | LRADC3 data Register                      | 0x05 | VDD    |
| 0x94  | LRADC4_DATA     | LRADC4 data Register                      | 0x05 | VDD    |
| 0x95  | LRADC5_DATA     | LRADC5 data Register                      | 0x05 | VDD    |
| 0x97  | CP_CTL0         | Charge pump control Register              | 0x05 | VDD    |
| 0x98  | CP_CTL1         | Charge pump control Register              | 0x05 | VDD    |
| 0x99  | VDD_reserved    | Reserved                                  | 0x05 | VDD    |
| 0x9a  | TEST_CTL        | Standby test control Register             | 0x05 | VDD    |
| 0x9b  | SYSTEM_CTL      | System on/off time set Register           | 0x05 | RTCVDD |
| 0x9c  | SYSTEM_ONOFF    | on/off statue & RESET time set Register   | 0x05 | RTCVDD |
| 0x9d  | RTCVDD_reserved | Reserved                                  | 0x05 | RTCVDD |
| 0xa2  | FS_CTL          | Fsource control and EFUSE select Register | 0x05 | VDD    |
| 0xa3  | EFUSE_CTL       | EFSUE control Register                    | 0x05 | VDD    |
| 0xa4  | EFUSE0          | EFUSE0 data Register                      | 0x05 | VDD    |
| 0xa5  | EFUSE1          | EFUSE1 data Register                      | 0x05 | VDD    |
| 0xa6  | EFUSE2          | EFUSE2 data Register                      | 0x05 | VDD    |
| 0xa7  | EFUSE3          | EFUSE3 data Register                      | 0x05 | VDD    |

Table 6-6 PMU 寄存器列表

## 6.6 Register Description

### 6.6.1 VOUT\_CTL

VCC/VDD VOLTAGE SET Register.

(SFR:0x89, SFR bank 0x05)

| Bit(s) | Name      | Description            | R/W | Reset |
|--------|-----------|------------------------|-----|-------|
| 7      | VCC_LDO_I | VCC LDO Current limit: | R/W | 1     |

|     |         |  |     |      |
|-----|---------|--|-----|------|
|     |         | 0: 400Ma, debounce 120us<br>1: 600Ma, debounce 120us   |     |      |
| 6:4 | VCC_SET | VCC voltage level select<br>2016、 2.6V<br>2017、 2.7V<br>010 2.8V<br>011 2.9V<br>100 3.0V<br>***101 3.1V<br>110 3.2V<br>111 3.3V<br><br>在 test mode 下, 此 3 个 BIT 默认值为 000, 可读写, 即 VCC 电压可调。AVCC is set at the same time.<br>由 digital 实现  | R/W | 101  |
| 3:0 | VDD_SET | VDD(DC-DC & Regulator) voltage coarse control<br>2018、 1.3V<br>2019、 1.35V<br>0010 1.4V<br>0011 1.45V<br>0100 1.5V<br>0101 1.55V<br>0110 1.6V<br>0111 1.65V<br>**1000 1.7V<br>1001 1.75V<br>1010 1.8V<br>1011 1.85V<br>1100 1.9V<br>1101 1.95V<br>1110 2.0V<br>1111 2.0V<br><br>You can set VDD at lower voltage 1.6V to at light loading to reduce power dissipation.<br>在 test 模式下, 上电默 VDD=1.3V<br>由 digital 实现 | R/W | 1000 |

### 6.6.2 LDOPD\_CTL

Capless LDO pulldown SET Register。

(SFR:0x8a, SFR bank 0x05)

| Bit(s) | Name   | Description       | R/W | Reset |
|--------|--------|-------------------|-----|-------|
| 7:6    | UVDDPD | UVDD 无电容 LDO 弱下拉: | R/W | 00    |

|     |           |  |     |    |
|-----|-----------|--|-----|----|
|     |           | 00: 无下拉<br>01: 1Ma 下拉<br>10: 10Ma 下拉<br>11: 11Ma 下拉                    |     |    |
| 5   | FSOURCEPD | Fsource 无电容 LDO 弱下拉<br>0: 无下拉<br>1: 2Ma 下拉                             | R/W | 1  |
| 4   | AVDDPD    | AVDD 无电容 LDO 弱下拉<br>0: 无下拉<br>1: 400Ua 下拉                              | R/W | 0  |
| 3:2 | AVCCPD    | AVCC 无电容 LDO 弱下拉:<br>00: 无下拉<br>01: 1Ma 下拉<br>10: 4Ma 下拉<br>11: 5Ma 下拉 | R/W | 01 |
| 1:0 | VDDPD     | VDD 无电容 LDO 弱下拉:<br>00: 无下拉<br>01: 1Ma 下拉<br>10: 4Ma 下拉<br>11: 5Ma 下拉  | R/W | 01 |

注：上电后软件关闭 AVCC 和 VDD 的下拉

### 6.6.3 BDG\_CTL

Bandgap Control Register

(SFR:0x8b, SFR bank 0x05)

| Bit(s) | Name    | Description   | R/W | Reset |
|--------|---------|---|-----|-------|
| 7      | BDG_EN  | BANDGAP ENABLE BIT<br>0: BANDGAP ENABLE<br>1: BANDGAP DISABLE   | R/W | 0     |
| 6      | BDG_SEL | 0: Efuse 的输出直截接到 BANDGAP 的控制电路中，对 BDG_VOL[bit4:0]的写操作将无效。<br>1 : Efuse 的输出被屏蔽，而通过设置寄存器 BDG_VOL[bit4:0]来确定BANDGAP的输出电压值，便于测试BANDGAP电压调节电路的精度测试 | R/W | 0     |

|     |            |  |     |      |
|-----|------------|--|-----|------|
| 5   | BDG_FILTER | BANDGAP filter Control REG<br>0: BANDGAP has no filter resistor<br>1: BANDGAP has filter resistor<br><br>Notes: Make sure this bit is set to 1 before using DAC/ADC, OR IT WILL CAUSE BIG NOISE! | R/W | 0    |
| 4   | BDG_PDR    | BANDGAP 下拉电阻控制<br>0: NO pull down resistor<br>1: have pull down resistor<br><br>默认一直有下拉电阻，上电后软件disable 以节省功耗   | R/W | 1    |
| 3:0 | Reserved   | Reserved for analog future use   | R/W | 0010 |

## 6.6.4 BDG\_VOL

Bandgap voltage DATA Register

(SFR:0x8c, SFR bank 0x05)

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:5    | Reserved | RESERVED  | R   | 0     |
| 4:0    | BDG_VOL  | Bandgap voltage control, Its rang is 1.38~1.6125V.<br>00000: 1.38V<br>00001: 1.3875<br>.....<br>01000: 1.44V<br>.....<br>*01101: 1.4775V<br>01110: 1.485V<br>01111: 1.4925V<br>10000: 1.50V<br>10001: 1.5075<br>.....<br>11111: 1.6125V<br>To Adjust band gap voltage, the minimum step is 7.5Mv. | R/W | 01101 |

## 6.6.5 MULTI\_USED

GPIO multi-used Control Register

(SFR:0X8d, SFR bank 0x05)

| Bit(s) | Name            | Description  | R/W | Reset |
|--------|-----------------|--|-----|-------|
| 7      | UVDD_EN         | USBVDD LDO enable:<br>0: disable<br>1: enable<br>注意：该LDO使能不受USB efuse的bit7控制。  | R/W | 0     |
| 6:4    | UVDD_V          | USBVDD LDO output voltage control:<br>2020、 1.3V<br>2021、 1.4<br>010 1.5<br>011 1.6<br>***100 1.7<br>101 1.8<br>110 1.9<br>111 2.0 | R/W | 100   |
| 3      | SEG_DISP_VCC_EN | 段码屏电源使能控制位，不用时关闭，减少耗电，关闭后电压等于VCC<br>0: disable<br>1: enable  | R/W | 0     |
| 2:1    | AVCC_DROP       | AVCC LDO margin tuning, voltage drop from VCC<br>***00 0.15V<br>2022、 0.20V<br>10 0.25V<br>11 0.30V                                | R/W | 00    |
| 0      | Biased          | AVCC LDO bias control:<br>0: small current<br>1: big current   | R/W | 0     |

## 6.6.6 PMUADC\_CTL

PMU ADC Frequency and enable Control Register

(SFR:0x90, SFR bank 0x05)

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7      | ADC_FS    | The all ADCs Frequency Source Select:<br>The all A/D converter's working frequency is 128HZ default, you can raise the working frequency down to 64HZ by setting this bit 0 to consume little power.<br>0: 64HZ<br>1: 128HZ | R/W | 1     |
| 6      | BATADC_EN | Battery A/D enable.<br>0: Disable.<br>1: Enable<br>注意：使能ADC之后，至少等待15ms 才开始读ADC  | R/W | 1     |

|     |           |  |     |       |
|-----|-----------|--|-----|-------|
|     |           | (ADC 的采样频率128Hz 的情况下，若是64Hz 则加倍）。  |     |       |
| 5   | LRADC1_EN | 6bit LRADC1/3/4/5 A/D enable.<br>0: Disable.<br>1: Enable<br><br>注意，保证在LRADC1/3/4/5 切MFP 之前这一位是关闭的，否则会增加ADC 的稳定时间和干扰其他的ADC 的正常输出。同时在切MFP 和使能ADC 之后，至少等待15ms 才开始读 ADC (ADC 的采样频率 128Hz 的情况下，若是 64Hz 则加倍）。 | R/W | 0     |
| 4:0 | Reserved  | Reserved for analog future use   | R/W | 00011 |

### 6.6.7 BATADC\_DATA

BATADC DATA Register  
 (SFR:0x91, SFR bank 0x05)

| Bit(s) | Name     | Description  | R/W | Reset |
|--------|----------|--|-----|-------|
| 7      | Reserved | Reserved   | R   | x     |
| 6:0    | BATADC   | Battery 7bit Voltage ADC, used to detect Battery voltage。<br><br>Input voltage range is:<br>Li-ion: 1.4-4.4V | R   | xx    |

### 6.6.8 LRADC1\_DATA

LRADC1 DATA Register  
 (SFR:0x92, SFR bank 0x05)

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:6    | Reserved | RESERVED  | R   | x x   |
| 5:0    | LRADC1   | 6bit LRADC1 的数据输出.<br><br>LRADC1 input voltage range is from 0 to VCC | R   | xx    |

### 6.6.9 LRADC3\_DATA

LRADC3 DATA Register  
 (SFR:0x93, SFR bank 0x05)

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |          |   |   |    |
|-----|----------|---|---|----|
| 7:6 | Reserved | RESERVED  | R | xx |
| 5:0 | LRADC3   | 6bit LRADC3 的数据输出.<br>LRADC3 input voltage range is from 0 to VCC | R | xx |

### 6.6.10 LRADC4\_DATA

LRADC4 DATA Register

(SFR:0x94, SFR bank 0x05)

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:6    | Reserved | RESERVED  | R   | xx    |
| 5:0    | LRADC4   | 6bit LRADC4 的数据输出.<br>LRADC4 input voltage range is from 0 to VCC | R   | xx    |

### 6.6.11 LRADC5\_DATA

LRADC5 DATA Register

(SFR:0x95, SFR bank 0x05)

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:6    | Reserved | RESERVED  | R   | xx    |
| 5:0    | LRADC5   | 6bit LRADC5 的数据输出.<br>LRADC5 input voltage range is from 0 to VCC | R   | xx    |

### 6.6.12 CP\_CTL0

Charge pump Control Register

(SFR:0x97, SFR bank 0x05)

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7      | THRO_PMOS | Charge Pump 的输出电压强行为输入电压的2 倍的使能位,<br>用于Charge Pump 的内部环路debug:<br>0: disable<br>1: enable | R/W | 0     |
| 6:5    | CP_VOL    | Charge pump 输出电压设置:<br>00:4.9V<br>01:5.0V<br>10:5.1V                                      | R/W | 01    |

|     |         |  |     |    |
|-----|---------|--|-----|----|
|     |         | 11:5.2V  |     |    |
| 4:3 | SS      | 软启动阈值电压设置:<br>00:0.6 倍VIN<br>01:0.7 倍 VIN<br>10:0.8 倍VIN<br>11:0.9 倍 VIN                   | R/W | 10 |
| 2:1 | MODESEL | 从 CP 模式切到直通模式电压设置, 从直通切回 CP 模式阈值电压比此电压低 100Mv:<br>00:4.7V<br>01:4.8V<br>10:4.9V<br>11:5.0V | R/W | 00 |
| 0   | EN_2XCP | 电荷泵使能:<br>0: disable<br>1: enable  | R/W | 0  |

### 6.6.13 CP\_CTL1

Charge pump control register

(SFR:0x98, SFR bank 0x05)

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7      | OV_STATUS | Charge pump 的输出过压状态指示位:<br>0: 正常<br>1: 发生过压<br>当输出电压高于设定值的 5%, 指示过压。         | R   | X     |
| 6      | EN_CS     | 输入限流功能使能<br>0: disable<br>1: enable  | R/W | 0     |
| 5      | CL_SEL    | 输入电流限流阈值选择:<br>0:400Ma<br>1:600Ma  | R/W | 1     |
| 4      | EN_THSS   | 软启动使能<br>0: disable<br>1: enable   | R/W | 0     |
| 3      | THRO_CP   | Charge Pump 的直通功能使能:<br>0: disable<br>1: enable<br>将 Charge Pump 的输入直接导通到输出。 | R/W | 0     |
| 2:1    | Reserved  | Reserved for analog future use   | R/W | 00    |
| 0      | Reserved  | Reserved   | R   | 0     |

## 6.6.14 VDD\_reserved

Reserved for analog future use

(SFR:0x99, SFR bank 0x05)

| Bit(s) | Name     | Description                    | R/W | Reset |
|--------|----------|--------------------------------|-----|-------|
| 7:0    | Reserved | Reserved for analog future use | R/W | A5    |

## 6.6.15 TEST\_CTL

Standby test control Register

(SFR:0x9a, SFR bank 0x05)

| Bit(s) | Name        | Description  | R/W | Reset |
|--------|-------------|--|-----|-------|
| 7      | UVLO        | 检测到 DC5V 插入时此位为 1  | R   | X     |
| 6:5    | BIAS_CTL    | IC Current bias control<br>Lower 00<br>Low 01<br>**High 10<br>Higher 11<br>These 2bits only used for test. | R/W | 10    |
| 4      | SEG_BIAS_EN | 0 : 关 seg 的 bias 电流<br>1: 开启 seg 的 bias 电流<br>注: 数码管 LED 开启前须打开这个位   | R/W | 0     |
| 3:2    | Reserved    | Reserved for analog future use   | R/W | 00    |
| 1      | HDSWDET     | 假硬开关有关闭动作(ONOFF_2_3由0跳1, 并且在这之前100Ms内没有ONOFF_1_3由1跳0的动作)时digital会将此位置1, 可被软件写1清0                           | R/W | 0     |
| 0      | TEST_STATUS | Test 模式下的状态转换指示位<br>0: S3<br>1: S1   | R   | X     |

/\*RTCVDD 域寄存器\*\*\*/

## 6.6.16 SYSTEM\_CTL

System on/off and play/pause timer Control Register

(SFR:0x9b, SFR bank 0x05) (RTCVDD)

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:6    | SYSON_TIME | SYSON 按键时间长短设置:<br>00: t >=0.5s, 系统启动<br>01: t >=1s, 系统启动 | R/W | 01    |

|     |           |  |     |    |
|-----|-----------|--|-----|----|
|     |           | 10: t >=2s, 系统启动<br>11: t >=3s, 系统启动<br><b>注意: 假硬开关关闭时, SYSON 将无法从 S3 唤醒系统</b>   |     |    |
| 5   | LB_EN     | LB_ 进 standby 使能:<br>0: disable, 屏蔽低电进 standby 功能<br>1: enable, 开启低电进 standby 功能<br>低电进 standby 的电压可以通过 LB_VOL 设置<br><b>注意: (UVLO 不加 debounce)=1, 低电 LB_功能自动 disable。注意低电检测的比较器和 UVLO 检测的先后顺序, 保证 UVLO 先于比较器。</b>  | R/W | 1  |
| 4:3 | LB_VOL    | LB (Low battery) voltage setting<br>Li-ION<br>2023、 2.7V(考虑没有电池而将 VCC 与 BAT 短接的情况)<br>2024、 3.0V(锂电池供电希望 3.0V 保护)<br>1X 3.3V<br>8ms debounce, 采用异步检测的方式, 条件更苛刻。这样 VCC 与 BAT 短接时选择 3.0V 或 2.8V 低电都可以, 若 VCC 应用中波动大, 则在 AP 中将此位设置为 2.8V 也是可行的。<br>当电池电压低于默认的 3.0V, 系统就会发出低电 system off 信号, 从而 VCC/VDD 断电。与 LB_ 进 standby enable 结合使用 | R/W | 01 |
| 2   | OC_EN     | VCC/VDD LDO 过流保护使能<br>0: disable。VCC/VDD 过流保护功能无效<br>1: enable。使能 VCC/VDD 过流保护功能, 当 VCC/VDD 的电流超过设定值时进 standby。 Power ok 后有效   | R/W | 1  |
| 1   | LVPRO_EN  | VCC/VDD 电压过低保护使能<br>0: disable。即使 VCC/VDD 电压过低也不进行任何保护行为。<br>1: enable。当 EN_PMU 后, VCC/VDD 电压低于 2.3/0.7V 时间超过 512ms 时, 进入 standby 状态。  | R/W | 1  |
| 0   | REG_ENPMU | REG_ENPMU:<br>0: disable VCC/VDD<br>1: enable VCC/VDD<br>当状态从 S3 转到 S1 后, 此位被置 1。<br>(只与 digital 有关, 做状态机转参考换用)  | R/W | 1  |

注意: 在对 RTCVDD 域下的寄存器进行写操作后需要等待 3 个 MCUCCLK 周期加 4 个低频周期的时间才能真正的被写入, 才可读到被写入的值。

## 6.6.17 SYSTEM\_ONOFF

System ONOFF register

(SFR:0x9c, SFR bank 0x05) (RTCVDD)

| Bit(s) | Name             | Description  | R/W | Reset |
|--------|------------------|--|-----|-------|
| 7      | UVLOB_SEL        | UVLOB 检测信号的使能:<br>0: disable<br>1: enable  | R/W | 1     |
| 6      | DC5V_DOWN_SE_L   | UVLOB 检测功能中, DC5V 下拉负载的大小:<br>0: 4K<br>1: 600Ω   | R/W | 0     |
| 5      | EN_VDD_SL        | VDD 偏置电流:<br>0: 默认值<br>1: 大偏置电流  | R/W | 0     |
| 4      | ONOFF_2_3        | 假硬开关处于关闭状态时此位为 1 (ONOFF_X_3 两位不会同时为 1) 数字 debounce 4Ms   | R   | X     |
| 3      | ONOFF_1_3        | ONOFF 按键被按下时此位为 1 (ONOFF_X_3 两位不会同时为 1) 数字 debounce 4Ms  | R   | X     |
| 2:1    | ONOFF_RESET_TIME | ONOFF 长按 Reset 时间设置<br>00: 6s<br>01: 8s<br>10: 10s<br>11: 14s<br><b>注意: reset 仅在 S1 时有效, S3 时不 reset</b> | R/W | 10    |
| 0      | Reset_EN         | Reset_EN:<br>0: 屏蔽 ONOFF 长按 reset 功能。<br>1: 开启 ONOFF 长按 reset 功能。  | R/W | 1     |

注意: 在对 RTCVDD 域下的寄存器进行写操作后需要等待 3 个 MCUCLK 周期加 4 个低频周期的时间才能真正的被写入, 才可读到被写入的值。

### 6.6.18 RTCVDD\_reserved

Reserved for analog future use

(SFR:0x9d, SFR bank 0x05) (RTCVDD)

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7      | SYS_FAULT | 发生 pmu_fault 后此位会被置 1, 开机时软件检测到此位, 应写 1 清 0, 并将 RTC 时间和控制恢复成默认值 | R/W | 0     |
| 6:0    | Reserved  | Reserved for analog future use                                  | R/W | 5A    |

## /\*\*EFUSE Control 寄存器\*\*/

### 6.6.19 FS\_CTL

FSOURCE control and EFUSE select Register.

(SFR:0xa2, SFR bank 0x05)

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7      | FS_CTL    | <p>0: 关闭FSOURCE<br/>1: 开启DC5V到FSOURCE的LDO。</p> <p>It is strongly recommended FSOURCE/GND power bus impedance is smaller than 10ohm to prevent unintended voltage drop during programming.</p>  | R/W | 0     |
| 6      | FS_VOL    | <p>FSOURCE_LDO 电压输出:</p> <p>0: 3.8V。<br/>1: 3.9V。</p>  | R/W | 0     |
| 5:4    | EFUSE_SEL | <p>EFUSE SELECT:</p> <p>00: 烧写BANDGAP对应的EFUSE<br/>01: 烧写USB对应的EFUSE<br/>10: 烧写chipID0对应的EFUSE2<br/>11: 烧写chipID1对应的EFUSE3</p> <p>无论烧写哪个EFUSE, 都需要将EFUSE的enable信号置为无效, 即disable它。软件需要特别注意, 且一定要保证做完如下操作才可以软件烧写EFUSE:</p> <p>包括, 将控制bandgap和USB6.2K电阻的efuse数据读出到相应寄存器, 切换到寄存器控制bandgap和USB6.2K电阻, 然后再将EUFSE DISABLE, 然后再加3.8V电压到FSOURCE。</p> | R/W | 00    |
| 3      | EN_PROG   | <p>PROGRAM Enable, high active.</p> <p>0: disable<br/>1: ENABLE, 开始烧写</p> <p>注意PROGRAM时序。</p>  | R/W | 0     |

|     |         |                      |     |   |
|-----|---------|----------------------|-----|---|
| 2:0 | ADDRESS | Address input,A2~A0: | R/W | 0 |
|-----|---------|----------------------|-----|---|

## 6.6.20 EFUSE\_CTL

EFUSE control Register  
 (SFR:0xa3, SFR bank 0x05)

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:3    | Reserved   | Reserved  | R   | 0     |
| 2      | EFUSE23_EN | Efuse2,3 standby control bit<br>0: standby; 1: enable                   | R/W | 0     |
| 1      | EFUSE1_EN  | Efuse1 standby control bit(与 USB 的控制位一起做一个或运算)<br>0: standby; 1: enable | R/W | 0     |
| 0      | EFUSE0_EN  | Efuse0 standby control bit<br>0: standby; 1: enable                     | R/W | 0     |

## 6.6.21 EFUSE0

EFUSE0 data Register  
 (SFR:0xa4, SFR bank 0x05)

| Bit(s) | Name              | Description   | R/W | Reset |
|--------|-------------------|---|-----|-------|
| 7      | JTAG_disable      | 0: CHIP read and JTAG TCK & TMS input from IO PAD is<br><b>disable</b><br>1: CHIP read and JTAG TCK & TMS input from IO PAD is<br><b>enable</b> | R   | 1     |
| 6      | SRS_EN            | SRS WOW HD disable bit:<br>0: disable<br>1:enable   | R   | 1     |
| 5      | Reserved          | Reserved <b>for future use (software use)</b>   | R   | 1     |
| 4:0    | BANDGAP_EF<br>USE | Bandgap4:0  | R   | 00000 |

## 6.6.22 EFUSE1

EFUSE1 data Register  
 (SFR:0xa5, SFR bank 0x05)

| Bit(s) | Name     | Description                            | R/W | Reset  |
|--------|----------|--|-----|--------|
| 7      | Reserved | Reserved for future use (software use) | R   | 1      |
| 6      | Reserved | Reserved for future use (software use) | R   | 1      |
| 5:0    | USBEFUSE | USB 6.25k refer value                  | R   | 111111 |

## 6.6.23 EFUSE2

EFUSE2 data Register

(SFR:0xa6, SFR bank 0x05)

| Bit(s) | Name    | Description | R/W | Reset |
|--------|---------|-------------|-----|-------|
| 7:0    | ChipID0 | ChipID0     | R   | 0     |

## 6.6.24 EFUSE3

EFUSE3 data Register

(SFR:0xa7, SFR bank 0x05)

| Bit(s) | Name    | Description | R/W | Reset |
|--------|---------|-------------|-----|-------|
| 7:0    | ChipID1 | ChipID1     | R   | 0     |

## 6.7 TEST MODE

### 6.7.1 VOUT 电压调节

在 test mode 下, VCC 和 VDD 的电压默认为最低值, 即 2.6V 和 1.3V。它们的电压可以通过 Register——VOUT\_CTL 来调节。

### 6.7.2 Power ON

- 在 test 模式下测试其它模块时——需要外灌 VCC

- 测试 PMU 模块默认电压值时——电池供电时 VCC/VDD 默认电压值。
- 上电后测试电流过大的标准为：VCC 应小于 100Ma, CP 测试时 VDD 应小于 20Ma。(目前是各小于 100Ma)

### 6.7.3 Reset for TEST MODE

通过 P\_reset\_GPIO 按键的次数判断进入 test mode x, reset 按键时间为 100us。

### 6.7.4 状态转换测试

#### 6.7.4.1 S4 到 S1

在 Power On 中已经包含此部分功能，所以不需要再重复测试

#### 6.7.4.2 S1 到 S3 及唤醒

因为增加了 scan 测试，所以只进行 Alarm 唤醒，其余唤醒功能不测：通过配置 REG\_ENPMU ， 测试 S1 到 S3 后，通过配置 alarm 寄存器，测试唤醒。

### 6.7.5 正常工作模式下的后门信号

当使能后门后，可通过 GPIO 释放 PMU 相关信号进行 debug，对应关系如下：

PMU\_DEBUG\_signal:

| GPIO    | 对应的信号                             |
|---------|-----------------------------------|
| GPIO_A0 | VCCOK ( <b>vdd 域</b> )            |
| GPIO_A1 | VDDOK ( <b>vdd 域</b> )            |
| GPIO_A2 | 1KHZ ( <b>vdd 域</b> )             |
| GPIO_A3 | ONOFF_2_3L ( <b>vdd 域</b> )       |
| GPIO_A4 | ONOFF_1_3L ( <b>vdd 域</b> )       |
| GPIO_A5 | VDD 域下 preok2 信号 ( <b>vdd 域</b> ) |
| GPIO_A6 | INTK32 ( <b>PMU 内部 32k</b> )      |

|         |     |
|---------|-----|
| GPIO_A7 | GND |
|---------|-----|

|         |                              |
|---------|------------------------------|
| GPIO    | 对应的信号                        |
| GPIO_D0 | CK32KL (内部和外部选后的 32k, vdd 域) |
| GPIO_D1 | UVLOAL (vdd 域)               |
| GPIO_D2 | UVLOBL (vdd 域)               |
| GPIO_D3 | UVLOWPPL (vdd 域)             |

注：两组 GPIO 信号同时放出，请 digital 注意

## 7 System Control

### 7.1 ChipVersion (黄少彬、蔡瑞仁)

| 日期         | 版本     | 描述  | 修订人 |
|------------|--------|---|-----|
| 2012-07-24 | V1. 00 | initial   | 黄少彬 |
| 2012-08-07 | V1. 01 | 1、与 MFP 定义冲突，修改为 0x06ff<br>2、原先 EXBUS DATA 寄存器定义在 0x06f4，但是实际设计却是在 0x06ff，与 ChipVersion 冲突，修改 EXBUS 比较麻烦，故将 ChipVersion 寄存器修改为 0x06f4 | 黄少彬 |
| 2013-07-05 | V2. 04 | 修改 chip Version 寄存器默认值  | 黄少彬 |

#### 7.1.1 ChipVersion Register group

| Index | Mnemonic    | Description           | BANK |
|-------|-------------|-----------------------|------|
| 0xf4  | ChipVersion | Chip Version Register | 0x06 |

#### 7.1.2 Register Description

ChipVersion(Chip Version Register, SFR Address 0xf4, 0x06)

| Bit Number | Bit Mnemonic | Function        | Access | Reset |
|------------|--------------|-----------------|--------|-------|
| 7:4        | Reserved     | Be read as zero | -      | -     |

|     |             |   |   |     |
|-----|-------------|---|---|-----|
| 3:0 | ChipVersion | The chip revision history is stored in these bits, which is consistent with the chip version.<br>If the chip version is B, the value of ChipVersion is 0001b. | R | 01h |
|-----|-------------|---|---|-----|

## 7.2 RMU Digital (彭洪、刘惠民、蔡瑞仁)

| 日期         | 版本     | 描述   | 修订人 |
|------------|--------|--|-----|
| 2012-07-24 | V1. 00 | initial  | 彭洪  |
| 2012-09-12 | V1. 02 | 1、增加 LCD 模块的复位控制   | 彭洪  |
| 2012-10-20 | V2. 00 | 1. 修改 MRCR1，将 FMRDS 和 IIC 的复位控制 reserved;<br>2. 修改 MRCR3，增加 TK 的复位控制;<br>3. 根据第 1 和第 2 点，修改 Block Diagram; | 刘惠民 |
| 2012-11-07 | V2. 01 | 1. 修改 MRCR3 bit1 为 lcd reset<br>2. 修改 MRCR3 bit2 为 TK reset  | 刘惠民 |
| 2012-12-12 | V2. 02 | 1、将 SFR 总线访问 USB 控制器的 wait cycle 位宽由 3bits 扩宽到 4bits，将 PCON bit3 定义为 SFR_wait_bit3.                        | 刘惠民 |

### 7.2.1 Features

The RMU Controller of GL5115 has following features:

- (1) The RMU (Reset Management Unit) can reset all the peripherals.
- (2) The MCU can enter power-saving mode by setting the registers of RMU .

### 7.2.2 Function Description

The RMU (Reset Management Unit) can reset all the peripherals and can force MCU enter IDLE or Power Down Mode. The wait cycles to access the SFR can be set by the PCON register which some of bits is different from the old 8051/80251

### 7.2.3 Module Description

#### 7.2.3.1 Block Diagram

All modules except the MCU core will be in reset mode after power on reset.

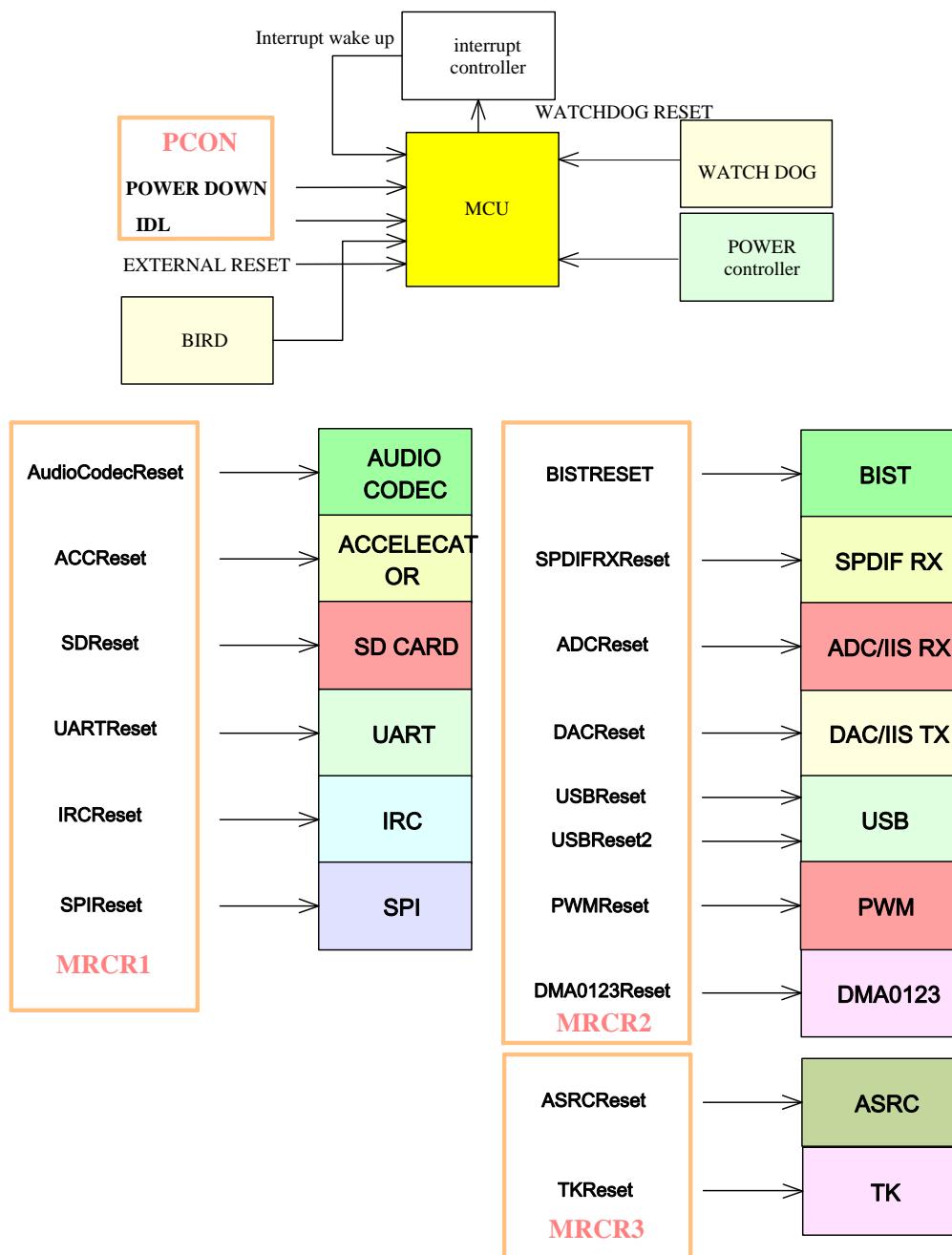


Figure 7-1 Block Diagram of RMU

Note: \* the reset bit of B1/B2 controller and RTC are not listed in the RMU register.

\*\* the reset wire of SPI BOOT controller is connected to the wire of MCU reset. It can be reset while the power on reset, watch dog reset or reset pin of MCU is set low.

## 7.2.1 RMU\_Digital\_Part Register List

| Index | Mnemonic | Description                     | BANK     |
|-------|----------|---------------------------------|----------|
| 0x86  | MRCR1    | Module Reset Control Register 1 | 0x01     |
| 0x87  | PCON     | Power Control Register          | all bank |

|      |        |                                 |          |
|------|--------|---------------------------------|----------|
| 0x89 | MRCR2  | Module Reset Control Register 2 | 0x01     |
| 0x9f | MRCR3  | Module Reset Control Register 3 | 0x01     |
| 0x96 | ExWait | External bus access wait cycle  | all bank |

## 7.2.2 Register Description

### 7.2.2.1 MRCR1

**MRCR1(Module Reset Control Register 1, SFR:0x86, sfr bank 0x01)**

| Bit Number | Bit Mnemonic    | Description   | Access | Reset |
|------------|-----------------|---|--------|-------|
| 7          | AudioCodecReset | Audio Codec Reset<br>0: reset<br>1: normal            | R/W    | 0     |
| 6          | ACCReset        | Accelerator Reset<br>0: reset<br>1: normal            | R/W    | 0     |
| 5          | SDReset         | SD/MMC Card Controller Reset<br>0: reset<br>1: normal | R/W    | 0     |
| 4          | Reserved        | Be read as zero.                                      | -      | -     |
| 3          | UARTReset       | UART Controller Reset<br>0: reset<br>1: normal        | R/W    | 0     |
| 2          | IRCReset        | IRC Controller Reset<br>0: reset<br>1: normal         | R/W    | 0     |
| 1          | SPIReset        | SPI Controller Reset<br>0: reset<br>1: normal         | R/W    | 0     |
| 0          | Reserved        | Be read as zero.                                      | -      | -     |

### 7.2.2.1 PCON

**PCON(Power Control Register, SFR:0x87, all bank)**

| Bit Number | Bit Mnemonic  | Description  | Access | Reset |
|------------|---------------|--|--------|-------|
| 7:5        | SFR_wait[2:0] | SFR access wait cycle control bit[2:0] for USB controller, control with SFR_wait_bit3. *<br>Wait cycle = SFR_wait[3:0] + 1 | R/W    | 111   |

|   |               |  |     |   |
|---|---------------|--|-----|---|
| 4 | POF           | Power Off flag<br>This bit is the image of the input “poweroff”. It is set by hardware as VCC rises above TBD voltage to indicate that power has been off or VCC had fallen below a TBD voltage and that on-chip volatile memory is indeterminate. It can be set or cleared by software. | R/W | 0 |
| 3 | SFR_wait_bit3 | SFR access wait cycle control bit3 for USB controller.*  | R/W | 0 |
| 2 | Reserved      | Be read as ‘0’   | -   | - |
| 1 | PD            | Powerdown mode bit<br>When set, activates powerdown mode<br>Clear by hardware when an enabled external interrupt or a reset occurs.  | W   | 0 |
| 0 | IDL           | Idle mode bit<br>When set, activates idle mode<br>Clear by hardware when an enabled interrupt or a reset occurs.   | W   | 0 |

\* The wait cycles are used for access USB controller registers in 0x07 page.

### 7.2.2.2 MRCR2

**MRCR2(Module Reset Control Register 2, SFR:0x89, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Description  | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | BISTRESET    | RAM&ROM BIST Reset<br>0: reset<br>1: normal            | R/W    | 0     |
| 6          | SPDIFRXReset | SPDIF RX controller Reset<br>0: reset<br>1: normal     | R/W    | 0     |
| 5          | ADCReset     | ADC and IIS RX Reset<br>0: reset<br>1: normal          | R/W    | 0     |
| 4          | DACReset     | DAC and IIS TX Reset<br>0: reset<br>1: normal          | R/W    | 0     |
| 3          | USBReset     | USB Reset<br>0: reset<br>1: normal                     | R/W    | 0     |
| 2          | USBReset2    | This bit should be reset before USBReset bit is reset. | R/W    | 0     |

|   |               |   |     |   |
|---|---------------|---|-----|---|
|   |               | 0: reset<br>1: normal                         |     |   |
| 1 | PWMReset      | PWM Controller Reset<br>0: reset<br>1: normal | R/W | 0 |
| 0 | DMA01234Reset | DMA0/1/2/3/4 Reset<br>0: reset<br>1: normal   | R/W | 0 |

### 7.2.2.3 MRCR3

MRCR3(Module Reset Control Register 3, SFR:0x9f, SFR bank 0x01)

| Bit Number | Bit Mnemonic | Description   | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:3        | Reserved     | Be read as 6 zeros                                  | -      | -     |
| 2          | TKReset      | Touch Key Controller Reset<br>0: reset<br>1: normal | R/W    | 0     |
| 1          | LCDReset     | LCD Controller Reset<br>0: reset<br>1: normal       | R/W    | 0     |
| 0          | ASRCReset    | ASRC Controller Reset<br>0: reset<br>1: normal      | R/W    | 0     |

### 7.2.2.4 ExWait

ExWait(External bus access wait cycle register, SFR:0x96, SFR all bank)

| Bit Number | Bit Mnemonic | Description                    | Access | Reset |
|------------|--------------|--------------------------------|--------|-------|
| 7:5        | Reserved     | Be read as '000'               | --     | -     |
| 4:0        | ExWait       | External bus access wait cycle | R/W    | 01fh  |

The wait cycles are used for accessing the register EXTMEM\_DL (page address 06, sfr address 0xff ).

## 7.3 CMU Analog (刘惠民、邵要华、陈文杰)

| 日期         | 版本     | 描述  | 修订人 |
|------------|--------|---|-----|
| 2012-07-24 | V1. 00 | initial   | 刘惠民 |
| 2012-08-07 | V1. 01 | 按照 isp spec 检查工具的要求, 更改 <a href="#">register list</a> 的描述 | 刘惠民 |

|            |       |   |     |
|------------|-------|---|-----|
| 2012-09-12 | V1.02 | <ul style="list-style-type: none"> <li>1、将 HOSC/LOSC 的框图 0 端和 I 端修改过来，同时修改 test 模式下外灌 24M 的端口为 HOSCI；</li> <li>2、高校低电路输出的日历时钟修改为 4Hz；</li> <li>3、test 外灌 24M 时钟，从 HOSCI 端外灌</li> <li>4、删除高校低电路中，日历时钟源选择：来自于 LOSC 电路还是高校低电路；因为 RTC 控制器中已经有同样的选择了；</li> <li>5、三个 RTCVDD 电压域下的寄存器：</li> <li>6、HOSC_CTL/LOSC_CTL/CALENDAR_CLK_CTL，都做了密码保护，对寄存器进行操作时需要注意；</li> <li>7、由于原有 LOSC 晶振电路设计中，有 LOSC_OK 信号，目前阶段删除此信号会影响顶层设计，给整合者造成麻烦，所以 spec 当中还需要保留这个信号</li> <li>8、将 cmu analog 寄存器中原来定义 reserved bit 修改成 R/W 形式，因为 analog 那边是按照 R/W 设计的，并没有注意到 spec 定义的只读类型；</li> <li>9、高校低说明中的 2Hz 修改成 4Hz；</li> <li>10、原来 HOSC_CTL 寄存器中 HOSC_CAP_SEL 默认值有笔误，应该是 15Pf，也就是 011</li> </ul> | 刘惠民 |
| 2012-10-20 | V2.00 | <ul style="list-style-type: none"> <li>1. 将 debug 后门输出 GPIO 从 GPIO_C4 修改为 GPIO_D0，这是因为 GPIO_C4 不确定是否复用做 EJTAG_TRST，干脆直接避开，而且 GPIO_D0 各个封装都出 pin 了；</li> <li>2. 高校低电路默认使能，calendar 默认选择高校低送过来的 4Hz 时钟；</li> <li>3. 修改 HOSC 框图，将电源域修改为 BAT；</li> <li>4. 修改时钟树以及电源域说明，将 HOSC 变更到 BAT 电压域，并且增加高校低电路到 BAT 电压域的控制关系；</li> </ul>   | 刘惠民 |
| 2012-12-12 | V2.02 | <ul style="list-style-type: none"> <li>1. 增加软件 operation manual 说明部分；</li> <li>2. 根据高校低 RTC 时间测试实验，修改 CALENDAR_CLK_CTL 寄存器，增加了 HOSC 基准振荡时间选择，和 HCL 校准间隔的选择；</li> <li>3. 为了降低功耗，增加 CALENDAR_CLK_CTL 增加 RTCVDD 电压域下 24M 到 4Hz 分频电路的 gating 控制；</li> <li>4. 修改 HCL 电路框图，将图中的 2Hz 修改为 4Hz；</li> <li>5. 删除高校低内部信号说明，只是以框图形式说明；</li> </ul>  | 刘惠民 |
| 2013-01-25 | V2.03 | <ul style="list-style-type: none"> <li>1. 将原来的 CALENDAR_CLK_CTL 寄存器名称变更为 HCL_CLK_CTL，寄存器地址不变；</li> <li>2. 在 HCL_CLK_CTL 中，将 HOSC_OSC_TIME 默认值设置为 500ms，增加 HCL 误差精度选择</li> </ul>   | 刘惠民 |

|            |        |   |     |
|------------|--------|---|-----|
|            |        | 位;<br>3. 增加 HCL_INTERVAL_CTL 控制寄存器, 地址为 0xb5;<br>4. 针对前面 3 点, 修改 register list;   |     |
| 2013-07-05 | V2. 04 | 1、将 HOSC 起振时间由 1ms 修改为 1.5ms, 这是实测值;<br>2 、将 MCU_PLL_CTL bit6 描述修改为 MCUPLL_PMD, 表示 MCUPLL 的相位锁定检测, 这是为了使得描述更加准确;<br>3 、将 AUDIO_PLL_CTL bit7 描述修改为 AUDIO_PLL_PMD, 表示 AUDIOPLL 的相位锁定检测, 这是为了使得描述更加准确;<br>4、增加 debug signal 的信号描述, 使得描述更加准确; | 刘惠民 |

### 7.3.1 Features

- ◆ Support two oscillator inputs: HOSC and LOSC;
- ◆ Supply 2 PLLs and special clocks of all modules. The 2 PLLs is MCU PLL, Audio PLL.
- ◆ Built-in 32K oscillator.
- ◆ Built-in HOSC calibrate LOSC circuit.
- ◆ MCU PLL support spread spectrum.

| 晶振电路          | 起振电压/V | 维持电压/V | 起振时间/s | 负阻/欧姆 |
|---------------|--------|--------|--------|-------|
| HOSC          | >2.6   | >2.6   | <1.5ms | >150  |
| Dual pin LOSC | <0.9   | <0.9   | <1S    | >150K |

### 7.3.2 Function Description

GL5115 support calibrating LOSC using HOSC, realizing RTC function without LOSC oscillator;  
 MCU PLL support spread spectrum in order to improve system EMI performance.

### 7.3.3 Module Description

#### 7.3.3.1 Block Diagram

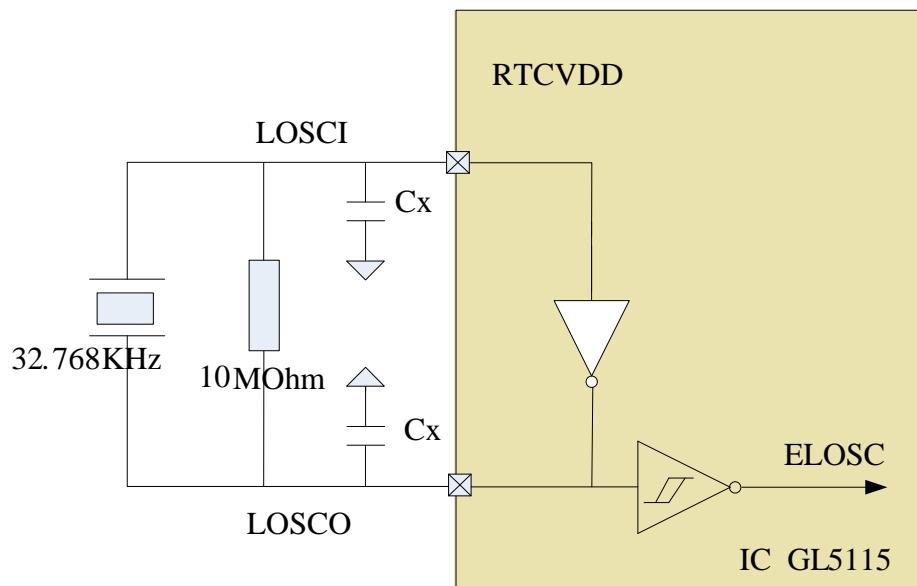


Figure 7-2 dual pin LOSC block Diagram

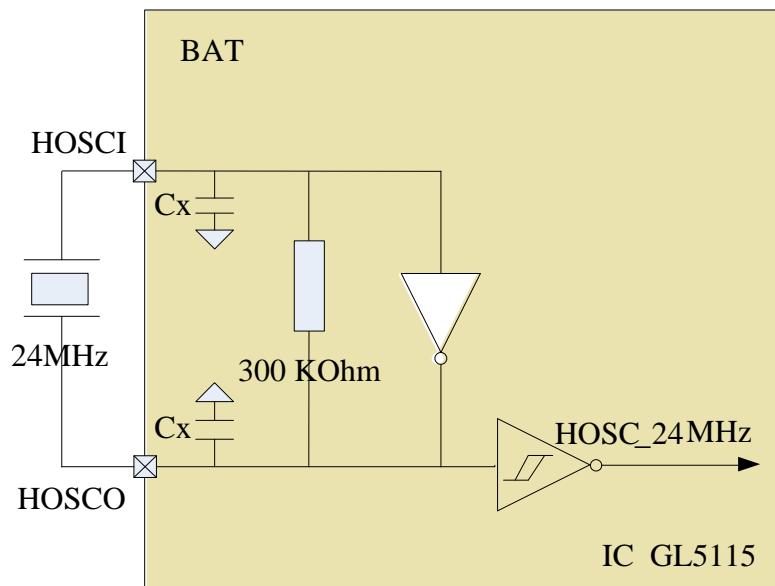


Figure 7-3 HOSC block Diagram

时钟树以及电源域说明：

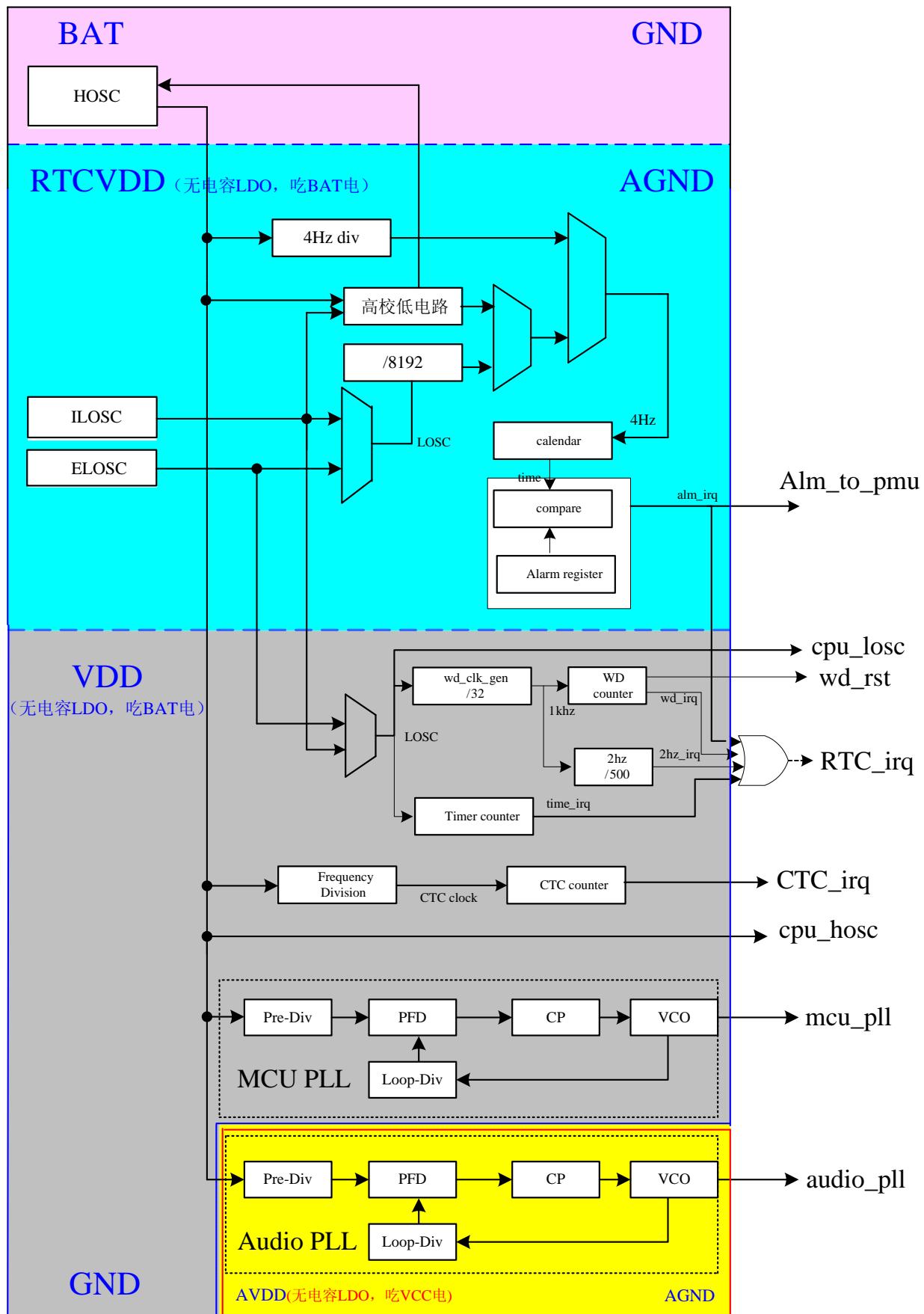


Figure 7-4 CMU 时钟树和电源域关系图

### 7.3.4 Operation Manual

#### 7.3.4.1 高校低电路说明

RTC 电路使用的 4Hz 是从 LOSC 分频得到的；如果使用内部的 RC 电路产生 LOSC，那么得到的 LOSC 时钟是不准确的，为了从不准确的 LOSC 得到精准的 4Hz，那么必须随时调整分频系数；HOSC 校准电路的思路出发点就是，HOSC 时钟是精准的，由 HOSC 和 LOSC 同时计数一段等长的时间，当 HOSC 计数器溢出时，立刻将 LOSC 计数值赋给 Calendar DIV，这样就能够使得 LOSC 分频得到的万年历时钟和 HOSC 的定时值一致；并且调整 HOSC 的计数器，就能够从 LOSC 中得到不同的万年历时钟；高校低电路框图如下所示：

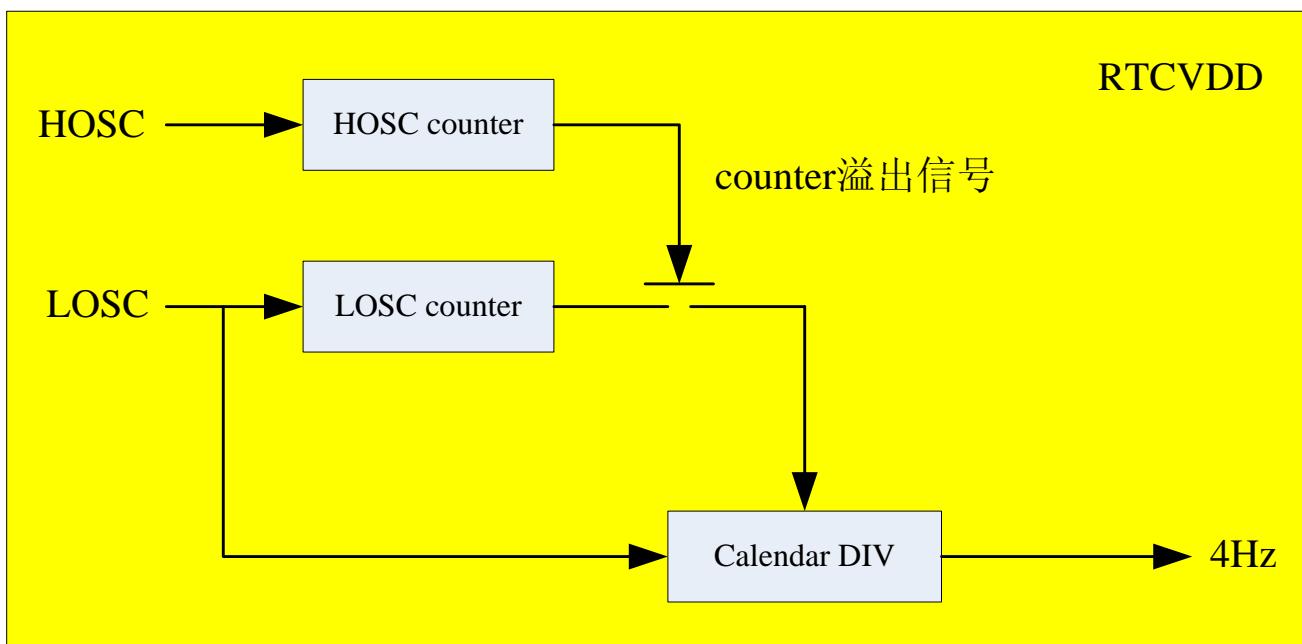
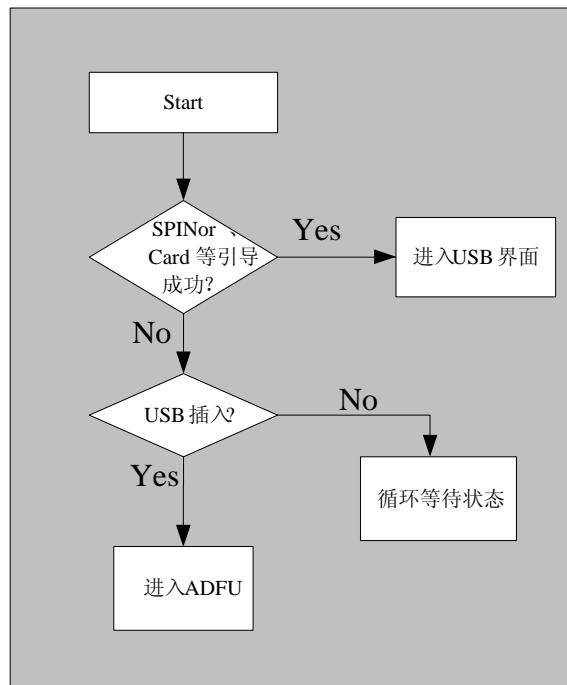


Figure 7-5 高校低电路说明框图

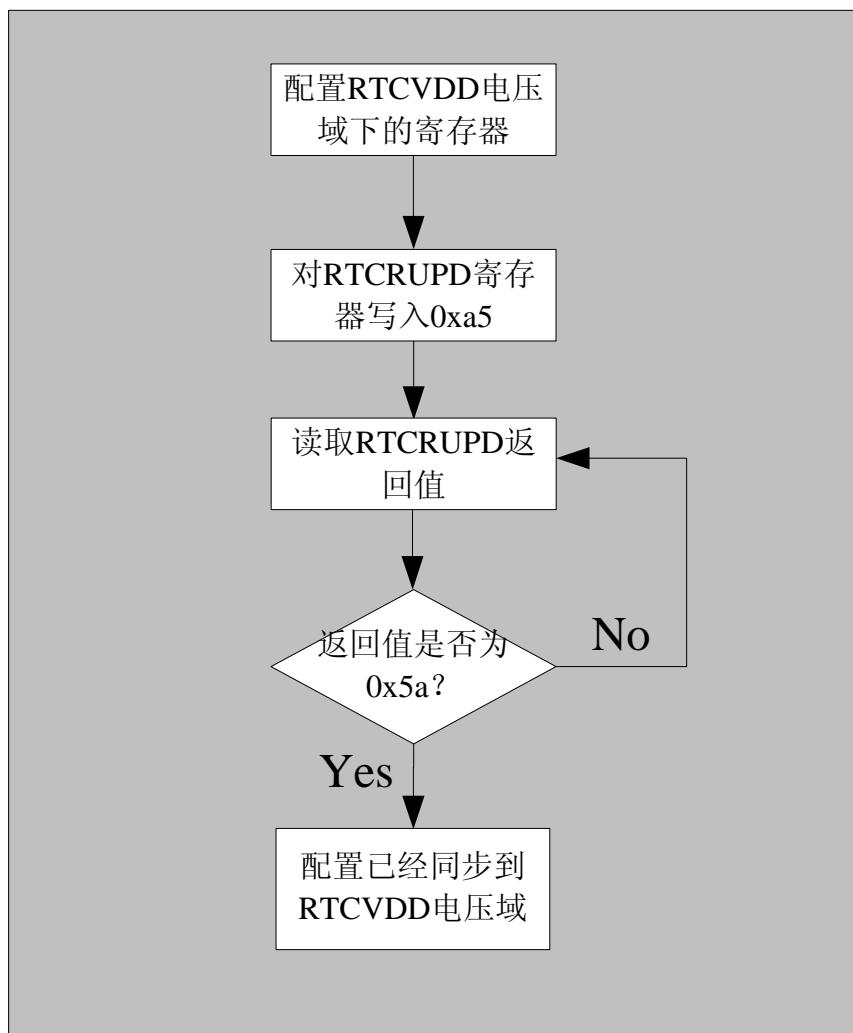
### 7.3.4.2 BROM 引导过程说明



**BROM**

Figure 7-6 系统引导过程流程图

1. HOSC\_CTL、LOSC\_CTL、CALENDAR\_CLK\_CTL、RTC\_CTL0、RTCRDM0~RTCRDM22 上述的寄存器都是设置在 RTCVDD 电压域的，如果软件有操作上述寄存器中的控制配置 bit，请注意操作后，对 RTCRUPD 进行写密码保护操作以同步寄存器；否则，软件设置不起作用！配置 RTCVDD 电压域寄存器流程如下所示：



## 配置RTCVDD电压域寄存器流程

Figure 7-7 配置 RTCVDD 电压域寄存器流程

- RTC\_CTL0 的 bit1 和 bit2、RTC 时间寄存器（时分秒、年月日）是只读位和 pending 位，是随时能够实现从 RTCVDD 电压域到 VDD 电压域同步的，所以软件可以正常使用这些寄存器标志位；

### 7.3.5 CMU\_analog Register List

Table CMU Controller Registers

| Index | Mnemonic    | Description             | BANK |
|-------|-------------|-------------------------|------|
| 0xb3  | MCU_PLL_CTL | MCUPLL control register | 0x01 |
| 0xb6  | HOSC_CTL    | HOSC control register   | 0x01 |
| 0xb7  | LOSC_CTL    | LOSC control register   | 0x01 |

|      |                   |   |      |
|------|-------------------|---|------|
| 0xb8 | HCL_CLK_CTL       | HCL clock control register                                    | 0x01 |
| 0xb5 | HCL_INTERVAL_CTL  | HCL calibration interval control register                     | 0x01 |
| 0xb9 | AUDIO_PLL_CTL     | AUDIO PLL control register                                    | 0x01 |
| 0xba | MCU_PLL_SSC_CTL   | MCU PLL spread spectrum control register                      | 0x01 |
| 0xbb | MCU_PLL_SSC_FSTEP | MCU PLL spread spectrum frequency step register HOSC_OSC_TIME | 0x01 |
| 0xbc | MCU_PLL_DEBUG_CTL | MCU PLL debug control register                                | 0x01 |

### 7.3.1 Register Description

#### 7.3.1.1 MCU\_PLL\_CTL

MCUPLL control register

SFR: 0xb3, SFR BANK:0x01 (VDD domain)

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7      | HOSC_EN    | High Frequency crystal Oscillator Enable:<br>0: disable<br>1: enable  | R/W | 1     |
| 6      | MCUPLL_PMD | MCUPLL phase match detect:<br>0: MCU PLL phase not match<br>1: MCU PLL phase match  | R   | X     |
| 5      | MCUPLL_EN  | MCUPLL Enable:<br>0: Disable<br>1: Enable   | R/W | 0     |
| 4:0    | MCUPLL_SEL | MCUPLL Frequency Select:<br>Formula:4M*SMCU<br>Range:12 ~ 124M<br>Value must be bigger than 3<br>0-2: reserved<br>3: 3*4M=12M<br>4: 4*4M=16M<br>.....<br>30: 30*4M=120M<br>31: 31*4M=124M | R/W | 0x06  |

#### 7.3.1.2 HOSC\_CTL

HOSC Control Register

SFR: 0xb6, SFR BANK:0x01 (RTCVDD domain)

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |              |  |     |     |
|-----|--------------|--|-----|-----|
| 7   | -            | Reserved for analog future use   | R/W | 0   |
| 6:4 | HOSC_CAP_SEL | HOSC PAD CAP SELECTION:<br>000: 0p (need off-chip cap)<br>001: 13p<br>010: 14p<br>011: 15p ***<br>100: 16p<br>101: 17p<br>110: 18p<br>111: 19p | R/W | 011 |
| 3:2 | HOSC_GMC     | High Frequency crystal Oscillator GMMIN<br>select bits:<br>00: 3.2<br>01: 6.13 ***<br>10: 8.0<br>11: 10.1                                      | R/W | 01  |
| 1:0 | -            | Reserved for analog future use   | R/W | 0   |

### 7.3.1.3 LOSC\_CTL

LOSC control register

SFR: 0xb7, SFR BANK:0x01 (RTCVDD domin)

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:3    | -        | Reserved for analog future use  | R/W | 0     |
| 2:1    | LOSC_GMC | Low Frequency crystal Oscillator GMMIN select<br>bits:<br>00: level 1 (low)<br>01: level 2 ***<br>10: level 3<br>11: level 4 (high) | R/W | 11    |
| 0      | LOSC_EN  | Low Frequency crystal Oscillator Enable:<br>0: disable<br>1: enable   | R/W | 1     |

### 7.3.1.4 HCL\_CLK\_CTL

HCL clock control register

SFR: 0xb8, SFR BANK:0x01 (RTCVDD domain)

| Bit(s) | Name            | Description  | R/W | Reset |
|--------|-----------------|--|-----|-------|
| 7:6    | HOSC_OSC_TIME   | HOSC calibrate LOSC circuit HOSC oscillator time:<br>00: 125ms<br>01: 250ms<br>10: 500ms***<br>11: 1000ms  | R/W | 10    |
| 5:4    | CAL_DELAY_TIME  | HOSC calibrate LOSC circuit wait for HOSC oscillator time:<br>00: 128 int_losc cycle***<br>01: 256 int_losc cycle<br>10: 512 int_losc cycle<br>11: 1024 int_losc cycle     | R/W | 00    |
| 3:2    | HCL_PRECISION   | HCL Precision select:<br>00: HCL CNT remain, 1 RC precision***<br>01: HCL CNT remain, 1 RC precision<br>10: HCL CNT+1, 1 RC precision<br>11: HCL CNT fix, 1/2 RC precision | R/W | 00    |
| 1      | HOSC_4HZ_DIV_EN | HOSC 4Hz divide circuit enable:<br>0: disable***<br>1: enable  | R/W | 0     |
| 0      | HOSC_CAL_EN     | HOSC calibrate LOSC circuit enable:<br>0: disable<br>1: enable***  | R/W | 1     |

### 7.3.1.5 HCL\_INTERVAL\_CTL

HCL calibration interval control register

SFR: 0xb5, SFR BANK:0x01 (RTCVDD domain)

| Bit(s) | Name         | Description  | R/W | Reset |
|--------|--------------|--|-----|-------|
| 7:3    | -            | Reserved, be read as zero  | R   | 0     |
| 2:0    | HCL_INTERVAL | HCL calibration interval control:<br>000: 4s<br>001: 8s<br>010: 16s<br>011: 32s<br>100: 64s*** | R/W | 100   |

|  |  |                                     |  |  |
|--|--|-------------------------------------|--|--|
|  |  | 101: 128s<br>110: 256s<br>111: 512s |  |  |
|--|--|-------------------------------------|--|--|

### 7.3.1.6 AUDIO\_PLL\_CTL

AUDIO PLL control register

SFR: 0xb9, SFR BANK:0x01 (VDD domain)

| Bit(s) | Name               | Description  | R/W | Reset |
|--------|--------------------|--|-----|-------|
| 7      | AUDIO_PLL_PMD      | AUDIOPLL phase match detect:<br>0: AUDIO PLL phase not match<br>1: AUDIO PLL phase match | R   | X     |
| 6      | AUDIOPLL_BYPASS_EN | AUDIO PLL BYPASS Enable:<br>0: disable<br>1: enable                                      | R/W | 0     |
| 5:4    | IBIAS_AUDIOPLL     | AUDUO PLL BIAS Current Select<br>BIAS current = 2Ua + 1* IBIAS_AUDIOPLL                  | R/W | 01    |
| 3:1    | SRSEL[2:0]         | AUDIO PLL Clock Select <sup>(1)</sup>  | R/W | 110   |
| 0      | AUDIO_PLL_EN       | AUDIO PLL Enable:<br>0: Disable<br>1: Enable   | R/W | 0     |

NOTE:

2025、AUDIO PLL clock select is list below:

|     | Fs    | Pre_div | Loop_div | Fout(MHz)       |
|-----|-------|---------|----------|-----------------|
| 000 | 44.1k | 49      | 46       | 22.53 (-94Hz)   |
| 001 |       | 50      | 47       | 22.56 (-37Hz)   |
| 010 |       | 34      | 32       | 22.588 (17Hz)   |
| 011 |       | 35      | 33       | 22.628 (96Hz)   |
| 100 | 48k   | 46      | 47       | 24.522 (-105Hz) |
| 101 |       | 43      | 44       | 24.558 (-34Hz)  |
| 110 |       | 41      | 42       | 24.585 (18Hz)   |
| 111 |       | 38      | 39       | 24.63 (108Hz)   |

### 7.3.1.7 MCU\_PLL\_SSC\_CTL

MCU PLL spread spectrum control register

SFR: 0xba, SFR BANK:0x01(VDD domain)

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|--------|------|-------------|-----|-------|

|     |           |  |     |     |
|-----|-----------|--|-----|-----|
| 7   | SSC_DIV   | Set ssc block clock division<br>0: /1<br>1: /2   | R/W | 0   |
| 6:4 | HNUM      | Set the number of steps in a half of ssc cycle<br>Number = 16 * (HNUM)+1                 | R/W | 110 |
| 3   | SDM_ORDER | Set sigma delta modulator order in MCU PLL<br>0: first order<br>1: second order          | R/W | 1   |
| 2:1 | PSTEP     | Select MCU PLL vco phase step for FN divider<br>00: 1/4<br>01: 2/4<br>10: 3/4<br>11: 4/4 | R/W | 00  |
| 0   | SSC_EN    | MCU PLL spread spectrum enable control:<br>0: disable<br>1: enable                       | R/W | 0   |

### 7.3.1.8 MCU\_PLL\_SSC\_FSTEP

MCU PLL spread spectrum frequency step register.

SFR: 0xbb, SFR BANK:0x01(VDD domain)

| Bit(s) | Name         | Description   | R/W | Reset   |
|--------|--------------|---|-----|---------|
| 7      | -            | Reserved for analog future use  | R/W | 0       |
| 6      | FCODE_EXT_EN | 0: tri-wave fcode generator enable<br>1: fcode fixed by FSTEP         | R/W | 0       |
| 5:0    | FSTEP        | Set step value for ssc tri-angular waveform.<br>Accumulator is 10bits | R/W | 00_1000 |

### 7.3.1.9 MCU\_PLL\_DEBUG\_CTL

MCU PLL debug control register

SFR: 0xbc, SFR BANK:0x01(VDD domain)

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:5    | CMU_DEBUG | CMU debug signal select:<br>000: disable<br>001: HOSC<br>010: LOSC<br>011: MCUPLL/4<br>100: AUDIOPLL<br>101: ADC_CLK | R/W | 000   |

|     |                  |   |     |    |
|-----|------------------|---|-----|----|
|     |                  | 110: DAC_CLK<br>111: Reserved                                       |     |    |
| 4   | MCUPLL_BYPASS_EN | MCUPLL BYPASS Enable:<br>0: disable<br>1: enable                    | R/W | 0  |
| 3:2 | IBIAS_MCUPLL     | MCU PLL BIAS Current Select<br>BIAS current = 2Ua + 1* IBIAS_MCUPLL | R/W | 01 |
| 1   | HOSC_OK          | HOSC Detect bit:<br>0: HOSC don't oscillate<br>1: HOSC oscillate ok | R   | 0  |
| 0   | LOSC_OK          | LOSC Detect bit:<br>0: LOSC don't oscillate<br>1: LOSC oscillate ok | R   | 0  |

### 7.3.1 TESTMODE

#### 7.3.1.1 BLOCK DIAGRAM

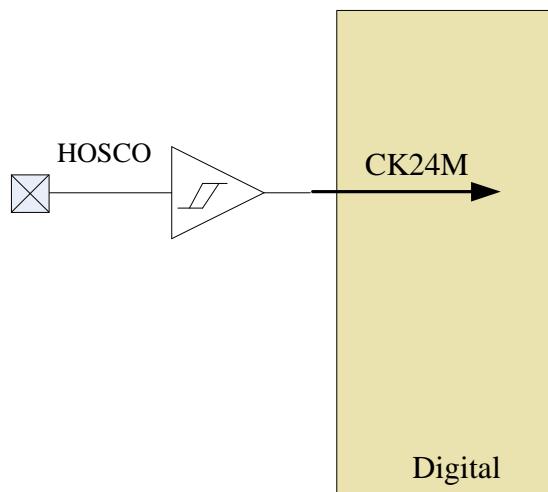


Figure 7-8 Tester's clock block Diagram

#### 7.3.1.2 SIGNAL DESCRIPTION

### 7.3.1.3 TEST METHOD

TEST 时候，从 HOSCO 外灌 24M Clock，系统选择使用 HOSC。

#### 7.3.1.1 Debug Signal

使能 CMU Analog back door 时，可以选择 GPIO\_D0 输出 CMU Analog debug 信号，具体的 debug 信号由 MCU\_PLL\_DEBUG\_CTL[7:5]选择，选择关系见下表所示：

| GPIO 名称 | MCU_PLL_DEBUG_CTL[7:5] | 输出信号     |
|---------|------------------------|----------|
| GPIO_D0 | 000                    | Disable  |
|         | 001                    | HOSC     |
|         | 010                    | ELOSC    |
|         | 011                    | McuPLL/4 |
|         | 100                    | AudioPLL |
|         | 101                    | ADC_CLK  |
|         | 110                    | DAC_CLK  |
|         | 111                    | Reserved |

因为设计的缘故，GPIO\_D0 在切换为 LOSC 时，只能够输出 ELOSC 时钟；

如果需要输出 ADC\_CLK/DAC\_CLK，需要在 cmu digital 中将 ADC/DAC 模块唤醒；

如果需要输出 MCUPLL/4 分频时钟，那么不能够使能 CMU 的 backdoor 信号(GPIO 模块的 DBGSEL bit7 控制)，否则 GPIO\_D0 输出 MCUPLL/16 分频时钟；

## 7.4 CMU Digital (彭洪、刘惠民、蔡瑞仁)

| 日期         | 版本     | 描述   | 修订人 |
|------------|--------|--|-----|
| 2012-07-24 | V1. 00 | initial  | 彭洪  |
| 2012-08-09 | V1. 01 | 1、修改了 FIR_MPX_RAM 的输入信号描述<br>2、增加了 SPDIF 时钟的分频电路，因此修改了配图和输入信号列表<br>3、在 AECLKCTL 寄存器的 bit7:5 增加了控制 SPDIF 分频的寄存器位<br>4、增加了 DAC/IISTX 时钟分频的描述，特别注意无缝切换的描述<br>5、为了选择到 DAC 的时钟，在 FMRDSCLKCTL 寄存器中增加 DACIISTXCLKSRCSEL bit<br>6、修改了 DAC 的时钟配图，有两点改动：1 是 DAC 的时钟选择，2 是 DAC 的时钟 gating<br>7、修改 SPDIF 的时钟分频寄存器<br>8、因为 SPDIF 时钟分频的变化修改了 ADC 的配图 | 彭洪  |

|            |       |   |     |
|------------|-------|---|-----|
|            |       | <p>9、修改了 FMc1k 的时钟描述寄存器<br/>         10、修改了 FM 的时钟配图以及后面的文字描述<br/>         11、修改了 DAC 时钟描述配图中 DAC/IIS TX CH1 FIFO 的时钟选择位<br/>         12、将原来 GPIO 中的 PWMDUTY 寄存器移到 CMU 模块中<br/>         13、在寄存器列表中增加了 PWMDUTY 寄存器描述，并且将 PWMCLKSEL 从 PWMCLKCTL 寄存器挪动到 PWMDUTY 寄存器中，PWMCLKCTL 的分频比有变化，默认值也有变化<br/>         14、修改了 PWM 的寄存器配图描述<br/>         15、修改了 PCMRAM 的时钟描述，增加 URAMc1k 的选择<br/>         16、修改了 PCMRAM 的时钟配图描述<br/>         17、修改了 ADC and SPDIF rx controller 配图中的 SPDIF 的时钟分频描述</p>  |     |
| 2012-09-12 | V1.02 | <p>1、修改 DAC 模块的配图，在选择 write DAC CH0 FIFO write clock 时的 FMRDS_CLK1 对应的编码应该是 10 而不是 00<br/>         2、修改 FMRDSCLKCTL，将 bit 1 DACIISTXCLKSRCSEL 搬到 bit 4;<br/>         3、修改 FMRDSCLKCTL，拓展 FMRDSCLKSEL 为 2bits，可以选择 FMRDS_DPLL_CLK0、HOSC、38M_FROM_DEBUG_B6;<br/>         4、修改 FMRDSCLKCTL，将 bit 3:2 修改为 FMRX_ADC_CLK_SEL，可以选择 FMRDS_DPLL_CLK1、FMRDS_CLK1、19M_FROM_DEBUG_B7;<br/>         5、根据前三点，修改 FMRDS 的时钟配图;<br/>         6、根据 FMRDS 时钟配图，修改 MEMCLKSELCTL0、MEMCLKSELCTL1，将 FIR_CS_RAM、FIR_AA_RAM、FIR_MPX_RAM_LM、FIR_RDS_RAM 的 FMRDS_CLK0 修改为 FMRDS_CLK0_GT;<br/>         7、根据 FMRDS 时钟配图，修改 MEMCLKSELCTL1，将 PCMRAM 的 FMRDS_CLK0 修改为 FMRX_ADC_CLK_GT;<br/>         8、根据 FMRDS 时钟配图，修改 MEMCLKSELCTL0，将 MURAN1 的 FMRDS_CLK0 修改为 FMRX_ADC_CLK<br/>         9、将 DAC/IIS-TX 中的 FMRDS_CLK1 选择修改为 FMRDS_CLK1_GT，此时 FMRX 模块写 DAC FIFO 的时钟变更为 gating 后的时钟;<br/>         10、修改 Clock Requirements 中的 FMRDS 时钟说明，增加 IIS、LCD/LED 时钟需求的说明</p> | 刘惠民 |

|            |        |   |     |
|------------|--------|---|-----|
|            |        | 11、修改 IR 时钟配图，将 IR 内部的 1KHz 测试时钟源头由 LOSC 修改为 200KHz, 这是因为 GL5115 ELOSC 是不封装出来的;   |     |
| 2012-10-20 | V2. 00 | <p>21、删除 FMRDS 模块的说明以及配置，将 FMRDS 时钟控制位 reserved;</p> <p>22、修改 FIR_CS_RAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替;</p> <p>23、修改 FIR_CS_RAM 的时钟配图以及说明;</p> <p>24、修改 FIR_AA_RAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替;</p> <p>25、修改 FIR_AA_RAM 的时钟配图以及说明;</p> <p>26、修改 FIR_MPX_RAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替;</p> <p>27、修改 FIR_MPX_RAM 的时钟配图以及说明;</p> <p>28、修改 FIR_RDS_RAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替;</p> <p>29、修改 FIR_RDS_RAM 的时钟配图以及说明;</p> <p>30、修改 MURAM1 的时钟选择，FMRDS 的时钟选择用 reserved 代替;</p> <p>31、修改 MURAM1 的时钟配图以及说明;</p> <p>32、修改 PCMRAM 的时钟选择，FMRDS 的时钟选择用 reserved 代替;</p> <p>33、修改 PCMRAM 的时钟配图以及说明;</p> <p>34、修改 DAC_IIS_TX 的时钟配图，删除当中 FMRDS_19M_GT 的时钟选择;</p> <p>35、删除 IIC 的时钟选择以及配置说明，将 IIC 时钟控制位 reserved;</p> <p>36、删除 SPIBOOT 的时钟选择以及配置说明，将 SPIBOOT 时钟控制位 reserved;</p> <p>37、修改 IR 模块时钟框图，删除 IR_TEST_1KHz 信号，因为 IR_ANALOG 电路已经删除;</p> <p>38、修改 CLKENCTL2，增加 TK 模块的时钟使能;</p> <p>39、增加 TK 模块的时钟框图和说明;</p> <p>40、根据 1 到 19 点的修改，更新 Block Diagram 框图;</p> <p>41、根据 1 到 19 点的修改，更新 clock requirements 表格;</p> | 刘惠民 |
| 2013-01-25 | V2. 03 | <p>1、对 CPU 时钟框图的 LOSC 源头选择进行说明;</p> <p>2、对 RTC 时钟框图中的 LOSC 源头选择进行说明;</p> <p>3、对 DAC/IIS-TX 时钟框图中的 LOSC 源头选择</p>   | 刘惠民 |

|  |   |  |
|--|---|--|
|  | <p>进行说明；</p> <p>4、对 FMCLK 时钟框图中的 LOSC 源头选择进行说明；</p> <p>5、增加 PWM 的时钟源无缝切换描述；</p> <p>6、对 LED&amp;SEG LCD 时钟框图中的 LOSC 源头选择进行说明；</p> <p>7、增加 ASRC 的时钟源无缝切换描述；</p> <p>8、将 CLKENCTL0 的 bit2 由只读位修改为可读写位；</p> <p>9、将 CLKENCTL1 的 bit4 和 bit1 由只读位修改为可读写位</p> <p>10、根据设计预期，修改 DAC/IIS-TX 的时钟框图，对写 DAC/IIS TX ch0 FIFO 的时钟增加 ADC_CLK2 的选项；</p> <p>11、根据设计预期，修改 ADC 时钟框图，将写 ADC/IIS RX FIFO 的时钟源，由 ADCIFclk 修改到 ADC_CLK2；</p> <p>12、删除 IIS-RX 中信号列表中的 ADCIFclk 输出项，同时将 IIS_rx_clock 修改为输出项；这是因为 IIS-RX 的输出，只有 IIS_rx_clock 这一项；</p> |  |
|--|---|--|

### 7.4.1 Features

The CMU Controller of GL5115 has following features:

- (1) The CMU (Clock Management Unit) can select MCUPLL, LOSC, HOSC or DC as the clock of each peripheral.
- (2) The clock sources of the memory blocks can be selected by CMU.

### 7.4.2 Function Description

The CMU (Clock Management Unit) can select MCUPLL and HOSC as the clock of each peripheral and select different clock sources to each memory block.

If the peripheral clock is disable, the selk for SFR access of this peripheral is also disable, except for some USB register which is needed to access in some power save mode.

### 7.4.3 Module Description

#### 7.4.3.1 Block Diagram

All modules except the MCU core are driven to DC after power on reset. The MCU core and all the RAM/ROM blocks are driven by the HOSC at this time.

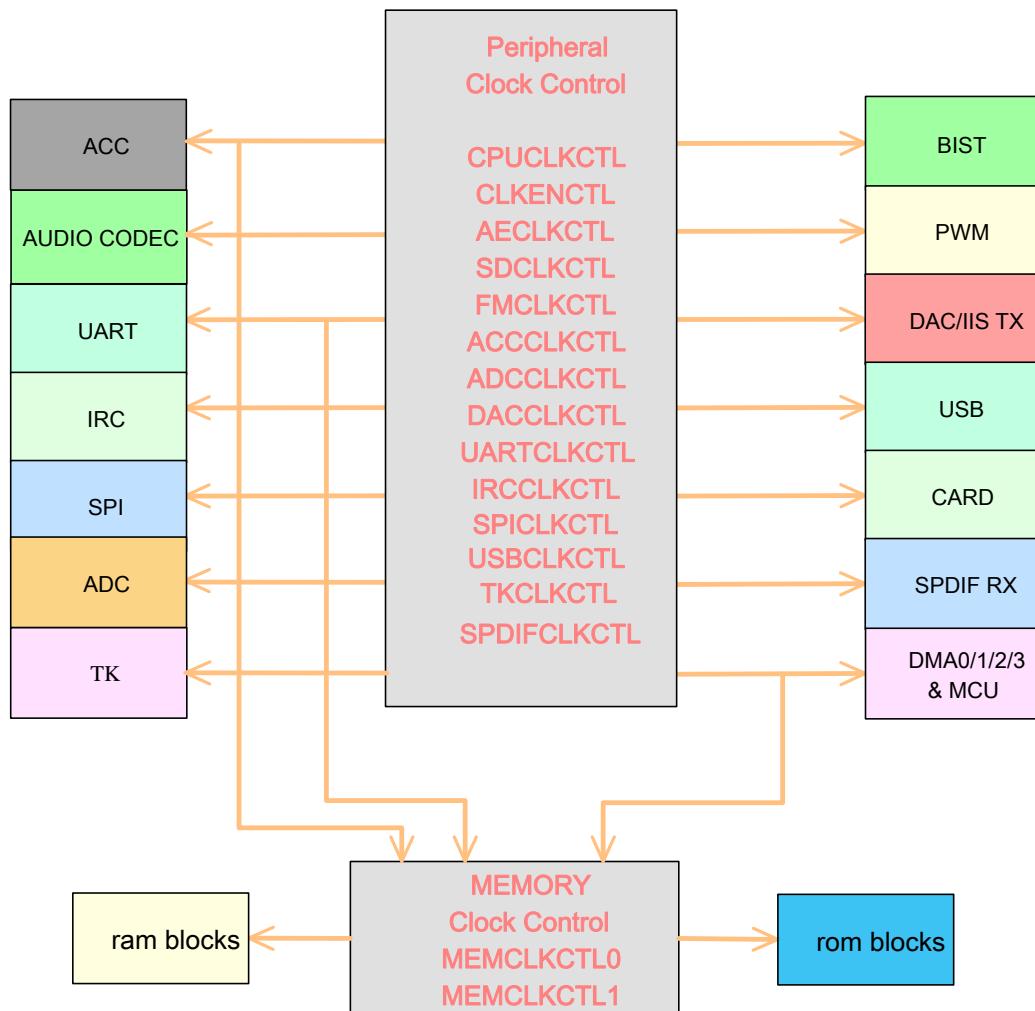


Figure 7-9 Block Diagram of CMU

Note: \* The reset bit of memory controller and RTC are not listed in the RMU register.

\*\* The reset wire of SPI BOOT controller is connected to the wire of MCU reset. It can be reset while the power on reset, watch dog reset or reset pin of MCU is set low.

### 7.4.3.2 Clock Requirements

| Module     | Clock source          | Min clock | Max clock | Typical Clock |
|------------|-----------------------|-----------|-----------|---------------|
| CPU&DMA    | CPUclk                | 32768Hz   | 50MHz     |               |
| ROM0,1     | CPUclk                | 32768Hz   | 100MHz    |               |
| DRAM251    | CPUclk                | 32768Hz   | 100MHz    |               |
| PRAM251    | CPUclk                | 32768Hz   | 100MHz    |               |
| MURAM1     | CPUclk                | 32768Hz   | 100MHz    |               |
|            | Aeclk                 | 4 MHz     | 24MHz     |               |
| FIR_CS_RAM | CPUclk                | 32768Hz   | 100MHz    |               |
|            | ASRCclk (for DAC CH1) | 24MHz     | 50MHz     |               |

|                 |                         |                 |                   |         |
|-----------------|-------------------------|-----------------|-------------------|---------|
|                 | AUIP                    | 24 MHz          | 24 MHz            |         |
| FIR_AA_RAM      | CPUclk                  | 32768Hz         | 100MHz            |         |
|                 | ASRCclk (for ADC)       | 24MHz           | 50MHz             |         |
|                 | Aeclk                   | 4 MHz           | 24 MHz            |         |
| FIR_MPX_RAM     | CPU/DMA                 | 32768Hz         | 100MHz            |         |
|                 | AUIP                    | 24 MHz          | 24 MHz            |         |
| FIR_RDS_RAM     | CPU/DMA                 | 32768Hz         | 100MHz            |         |
|                 | ASRCclk (for DAC CH0)   | 24MHz           | 50MHz             |         |
|                 | USBctl_URAM_CLK         | 60MHz           | 60MHz             |         |
| PCMRAM          | CPUclk                  | 32768Hz         | 100MHz            |         |
|                 | USBclk                  | 60MHz           | 60MHz             |         |
|                 | Aeclk                   | 4 MHz           | 24MHz             |         |
| URAM            | USBctl_URAM_CLK         | 60MHz           | 60MHz             |         |
|                 | CPUclk                  | 32768Hz         | 100MHz            |         |
| Audio codec     | MCUPLL or HOSC          | 4 MHz           | 24MHz             |         |
| SD card         | MCUPLL or HOSC          | 50 kHz          | 50MHz             |         |
| UART            | HOSC                    | 24 MHz          | 24 MHz            | 24 MHz  |
| SPI             | MCUPLL, HOSC or CoreClk |                 | 60MHz (pad)       |         |
| USB             | USBPLL                  |                 | 120MHz            | 120MHz  |
| CTC             | HOSC                    | 24 MHz          | 24 MHz            | 24 MHz  |
| RTC             | LOSC                    | 32768Hz         | 32768Hz           | 32768Hz |
| DAC             | audiopl                 | 2.048/1.8816MHz | 24.576/22.5792MHz |         |
| ADC             | audiopl                 | 2.048/1.8816MHz | 24.576/22.5792MHz |         |
| IR              | HOSC                    |                 | 200kHz            |         |
| INTC            | CoreClk                 | 8192Hz          | 50MHz (CPUclk)    |         |
| Peripheral regs | CoreClk                 | 8192Hz          | 50MHz (CPUclk)    |         |
| PWM             | HOSC                    | 24 MHz          | 24 MHz            | 24 MHz  |
| ACC             | CoreClk                 |                 | 50MHz             | 24 MHz  |
| FM              | MCUPLL                  |                 | 13MHz/7.6M        |         |
| IIS             | audiopl                 | 2.048/1.8816MHz | 24.576/22.5792MHz |         |
| LCD/ LED        | HOSC                    |                 | 24MHz             |         |
| TK              | HOSC                    |                 | 24MHz             |         |

## 7.4.4 Signal List

### 7.4.4.1 CPU clock

The clocks to drive CPU are listed below:

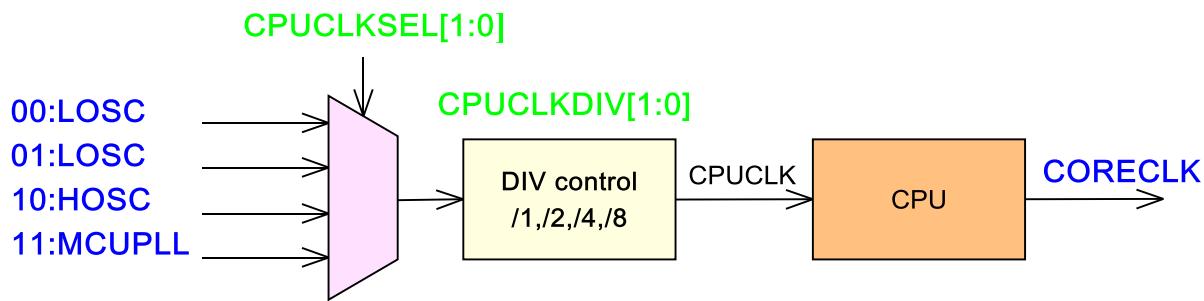


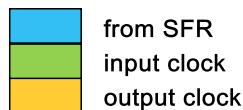
Figure 7-10 CPU clock control

Note:

- (1) CPUCLK and DMACLK must be synchronous
- (2) The default value : CPUCLKSEL[1:0]==2'b10 (HOSC is selected), CPUCLKDIV[1:0] == 2'b00
- (3) The internal or external L OSC is selected by L OSC\_sel bit of RTC\_CTL1 (bit5).
- (4) Seamless clock switching is supported.

| CPU clock |       |     |              |                               |
|-----------|-------|-----|--------------|-------------------------------|
| Signal    | Width | I/O | Clock Domain | Description                   |
| CPUCLKSEL | 2     | I   | CPUCLK       | Bit 1:0 of CPUCLKCTL register |
| CPUCLKDIV | 2     | I   | CPUCLK       | Bit 2:3 of CPUCLKCTL register |
| LOSC      | 1     | I   | -            | 32768 Hz                      |
| HOSC      | 1     | I   | -            | 24MHz                         |
| MCUPLL    | 1     | I   | -            | 4MHz~124MHz                   |
| CPUCLK    | 1     | O   | -            | 32768Hz~50MHz                 |

signal type legend:



#### 7.4.4.2 DMA01234clock

The clock to drive DMA0123 controller is listed below:

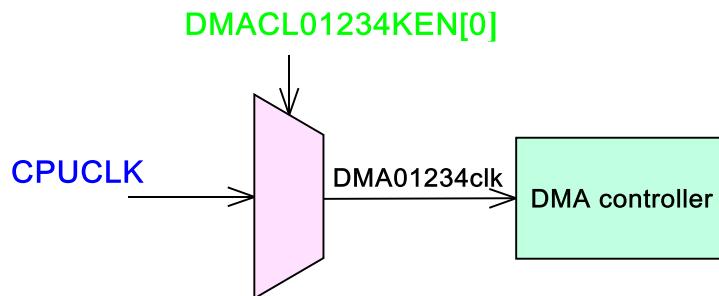


Figure 7-11 DMA01234 clock control

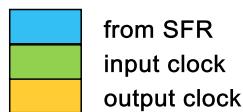
Note:

- (1) Memory access clock(including ROM and RAM access clock) and DMA clock must be synchronous

- (2) The default value is: DMA01234CLKEN[0]==1'b0 (DMA clock is disable).

| DMA01234 clock |       |     |              |  |
|----------------|-------|-----|--------------|--|
| Signal         | Width | I/O | Clock Domain | Description                            |
| DMA01234CLKEN  | 1     | I   | CPUCLK       | Bit 5 of CLKENCTL2 register            |
| CPUCLK         | 1     | I   | -            | 32768Hz~50MHz<br>SFR of DMA controller |
| DMA01234CLK    | 1     | O   | -            | 32768Hz~50MHz                          |

signal type legend:



#### 7.4.4.3 AUIP clock

The clock to drive Audio Code is listed below:

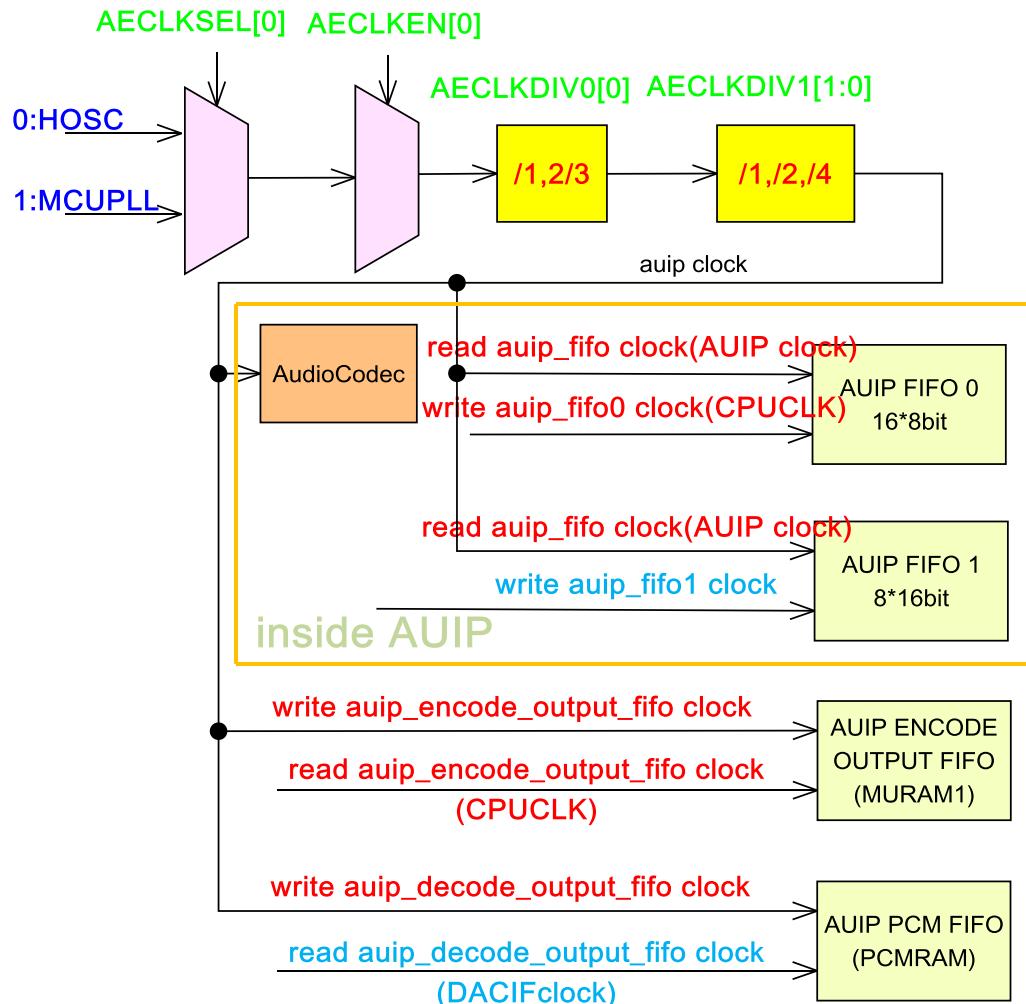


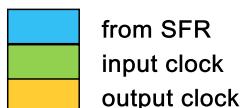
Figure 7-12 Audio Codec clock control

Note:

- (1) Seamless clock switching is supported.
- (2) The default value is: AECLKSEL[0]==1'b0 (HOSC is selected), VECLKEN[0]==1'b0 (audio codec clock is disable). AECLKDIV0[0] = 1'b0 (divider is /1). AECLKDIV1[1:0] = 2'b00 (divider is /1).
- (3) Write\_aupip\_fifo0\_clock is CPUCLK(accessed by CPU and DMA controller)
- (4) Write\_aupip\_fifo1\_clock is selected by WAVEncCTL.InputSelect[1:0] or WAVCTL.InputSelect[1:0]. [See the section below.](#)

| AUIP clock                              |       |     |              |  |
|---|-------|-----|--------------|--|
| Signal                                  | Width | I/O | Clock Domain | Description  |
| AECLKEN                                 | 1     | I   | CPUCLK       | Bit 7 of CLKENCTL1 register  |
| AECLKSEL                                | 1     | I   | CPUCLK       | Bit 0 of AECLKCTL register   |
| AECLKDIV1                               | 2     | I   | CPUCLK       | Bit 2:1 of AECLKCTL register   |
| AECLKDIV0                               | 1     | I   | CPUCLK       | Bit 3 of AECLKCTL register   |
| HOSC                                    | 1     | I   | -            | 24MHz  |
| MCUPLL                                  | 1     | I   | -            | 4MHz~124MHz  |
| CPUCLK                                  | 1     | I   | -            | (1) Write_aupip_fifo0_clock<br>Aupip_fifo0 is accessed by CPU/DMA.<br>(2) read aupip_encode_output_fifo_clock<br>Aupip encode output FIFO is accessed by CPU/DMA<br>(3) SFR of AUIP controller |
| <a href="#">Write_aupip_fifo1_clock</a> | 1     | I   | -            | Aupip_fifo1 is accessed by <a href="#">IISRX/SPDIFRX/ADC</a> .   |
| <a href="#">DACIFclock</a>              | 1     | I   | -            | DAC/IIS-TX controller  |
| AUIPclk                                 | 1     | O   | -            | 4Hz~24MHz  |

signal type legend:



#### 7.4.4.4 SD Card controller clock

The clock to drive SD card is listed below:

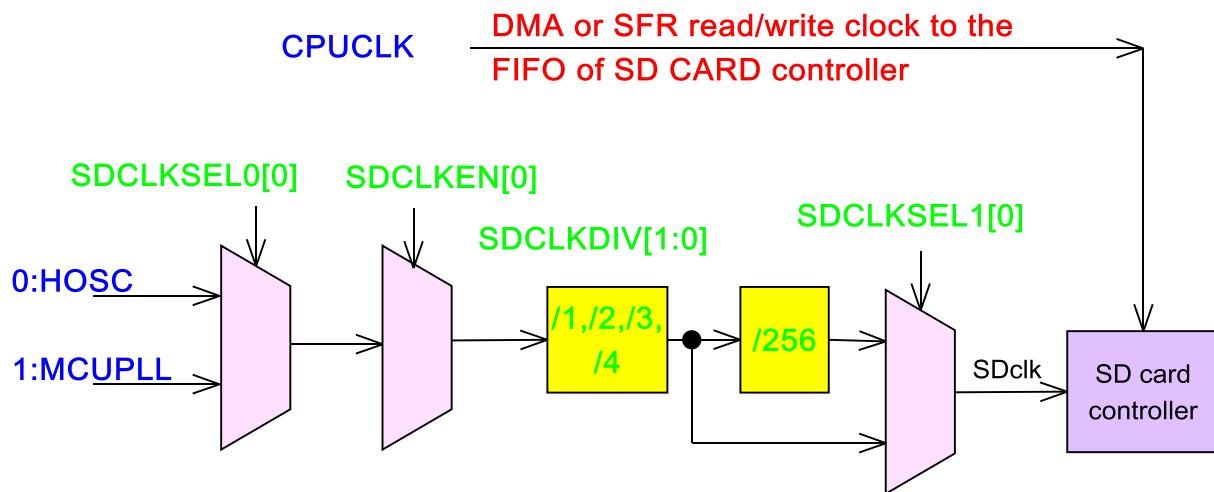


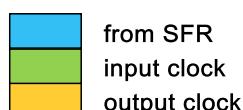
Figure 7-13 SD card controller clock control

Note:

- (1) Seamless clock switching is supported.
- (2) The default value is: SDCLKSEL0[0]==1'b0 (HOSC is selected), SDCLKEN[0]==1'b0 (SD card controller clock is disable). SDCLKDIV[1:0] = 2'b11 (SD card controller clock divider is 1), SDCLKSEL1[0]==1'b0 (divide 256).
- (3) Delay chain should be added to improve compatibility.
- (4) The control register is in the SD card controller registers page.

| SD card controller clock |       |     |              |   |
|--------------------------|-------|-----|--------------|---|
| Signal                   | Width | I/O | Clock Domain | Description   |
| SDCLKEN                  | 1     | I   | CPUCLK       | Bit 3 of CLKENCTL0 register   |
| SDCLKDIV                 | 2     | I   | CPUCLK       | Bit 3:2 of SDCLKCTL register  |
| SDCLKSEL1                | 1     | I   | CPUCLK       | Bit 1 of SDCLKCTL register  |
| SDCLKSEL0                | 1     | I   | CPUCLK       | Bit 0 of SDCLKCTL register  |
| HOSC                     | 1     | I   | -            | 24MHz   |
| MCUPLL                   | 1     | I   | -            | 4MHz~124MHz   |
| CPUCLK                   | 1     | I   | -            | (1) SD_FIFO which is accessed by CPU/DMA.<br>(2) SFR of SD controller |
| SDclk                    | 1     | O   | -            | 4Hz~24MHz   |

signal type legend:



#### 7.4.4.5 TK controller clock

The clock to drive TK controller is listed below:

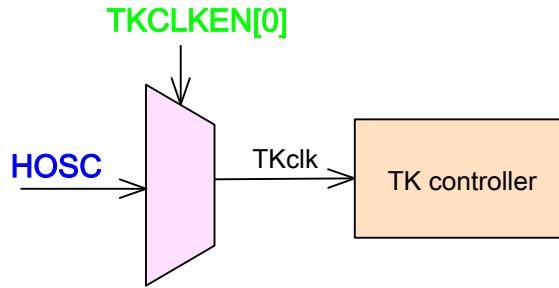


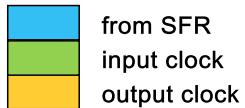
Figure 7-14 TK controller clock control

Note:

- (1) The default value is: TKCLKEN[0]==1'b0 (TK controller clock is disable).

| TK controller clock |       |     |              |                             |
|---------------------|-------|-----|--------------|-----------------------------|
| Signal              | Width | I/O | Clock Domain | Description                 |
| TKCLKEN             | 1     | I   | CPUCLK       | Bit 7 of CLKENCTL2 register |
| HOSC                | 1     | I   | -            | 24MHz                       |
| CPUCLK              | 1     | I   | -            | SFR of IIC controller       |
| TKclk               | 1     | O   | -            | 24MHz                       |

signal type legend:



#### 7.4.4.6 UART controller clock

The clock to drive UART controller is listed below:

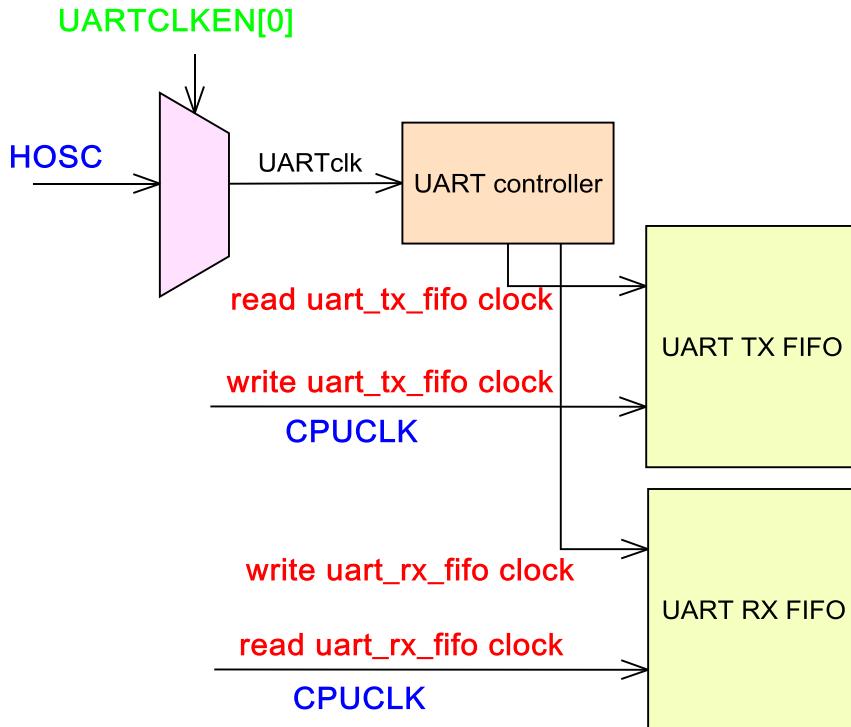


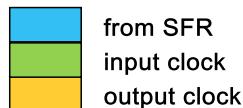
Figure 7-15 UART controller clock control

Note:

- (1) The default value is: `UARTCLKEN[0]==1'b0` (UART controller clock is disable).

| UART controller clock |       |     |              |  |
|-----------------------|-------|-----|--------------|--|
| Signal                | Width | I/O | Clock Domain | Description  |
| UARTCLKEN             | 1     | I   | CPUCLK       | Bit 0 of CLKENCTL1 register  |
| HOSC                  | 1     | I   | -            | 24MHz  |
| CPUCLK                | 1     | I   | -            | (1)UART_tx_FIFO is write by CPU/DMA.<br>(2)UART_rx_FIFO is read by CPU/DMA.<br>(3)SFR of UART controller |
| UARTclk               | 1     | O   | -            | 24MHz  |

signal type legend:



#### 7.4.4.7 SPI controller clock

The clock to drive SPI controller is listed below:

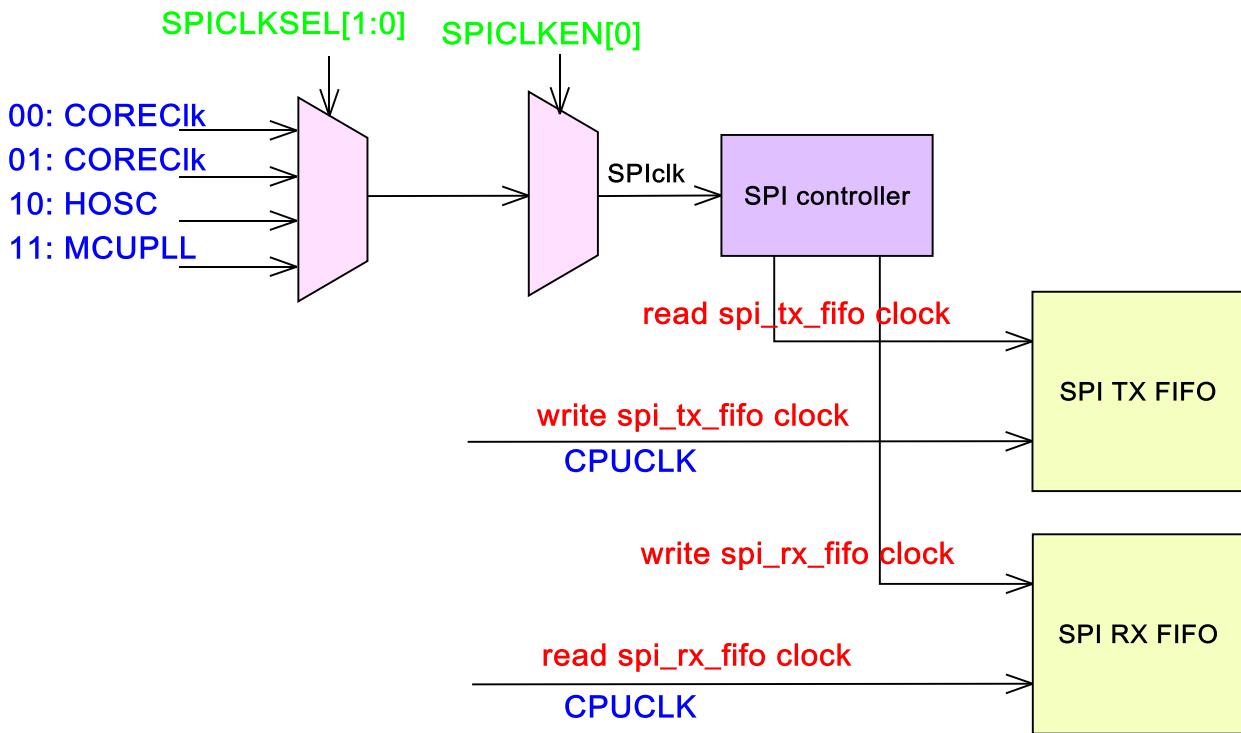


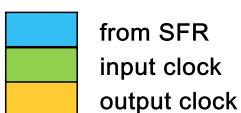
Figure 7-16 SPI controller clock control

Note:

- (1) Seamless clock switching is supported.
- (2) The default value is: SPICLKSEL[0]==1'b0 (HOSC is selected), SPICLKEN[0]==1'b0 (SPI controller clock is disable). SPICLKDIV[1:0]== 2'b11 (SPI controller clock divider is 8).
- (3) Delay chain should be added to improve compatibility.

| SPI clock |       |     |              |  |
|-----------|-------|-----|--------------|--|
| Signal    | Width | I/O | Clock Domain | Description  |
| SPICLKEN  | 1     | I   | CPUCLK       | Bit 3 of CLKENCTL1 register  |
| SPICLKSEL | 2     | I   | CPUCLK       | Bit 1:0 of SPICLKCTL register  |
| COREclk   | 1     | I   | -            | 32768~60MHz  |
| HOSC      | 1     | I   | -            | 24MHz  |
| MCUPLL    | 1     | I   | -            | 4MHz~124MHz  |
| CPUCLK    | 1     | I   | -            | (1) SPITX FIFO is write by CPU/DMA<br>(2) SPIRX FIFO is read by CPU/DMA<br>(3) SFR of SPI controller |
| SPIclk    | 1     | O   | -            | 32768~60MHz  |

signal type legend:



#### 7.4.4.8 USB controller clock

The clock to drive USB controller is listed below:

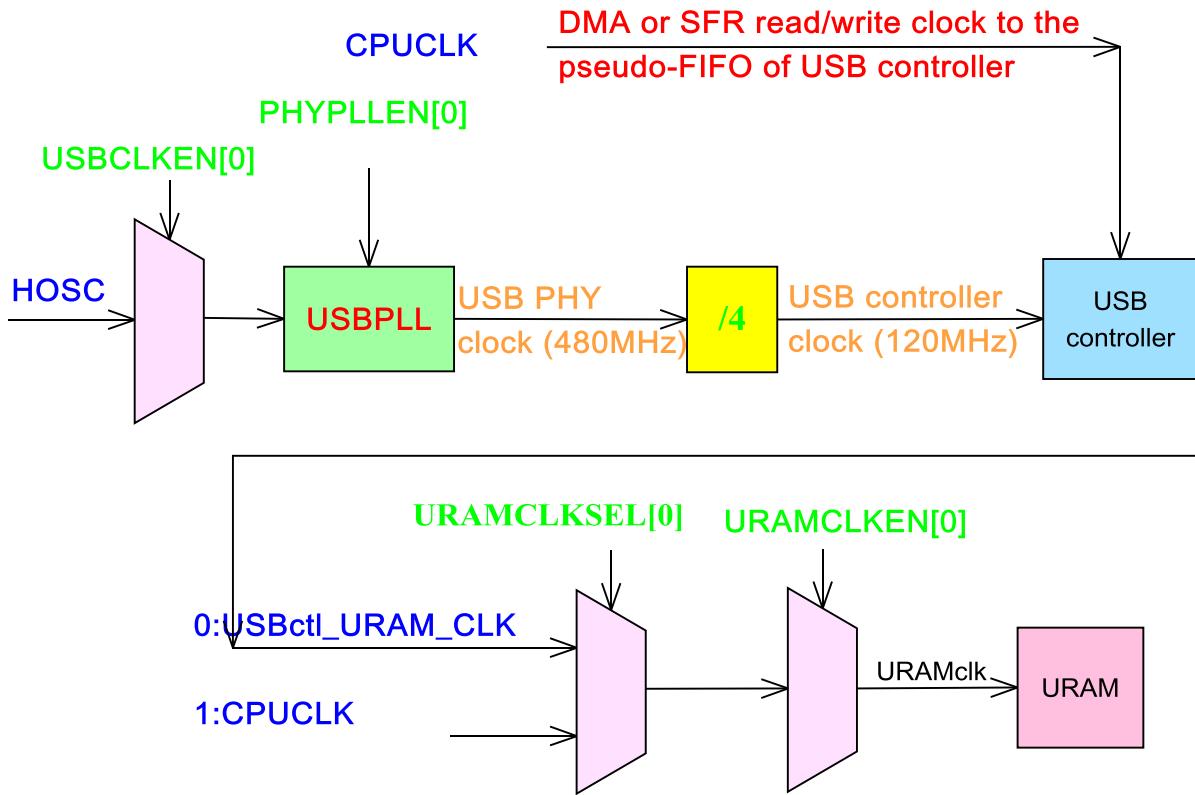


Figure 7-17 USB controller clock control

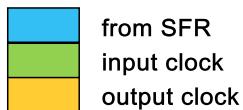
Note:

- (1) The default value is:  $\text{USBCLKEN}[0]==1'b0$  (HOSC to USBPLL is disable).
- (2) The default value is:  $\text{PHYPLLEN}[0]==1'0$  (USBPLL is disable). This bit is in USB controller register group.

USB 模块在进入 standby 情况下，有部分寄存器仍需可读，以便能够正确判断 USB 插拔线。USB 模块内部分成两类寄存器，在关闭 USB 模块 CLK 时候，需要保证有一部分寄存器的可读写。在 USB 的模块中有描述。

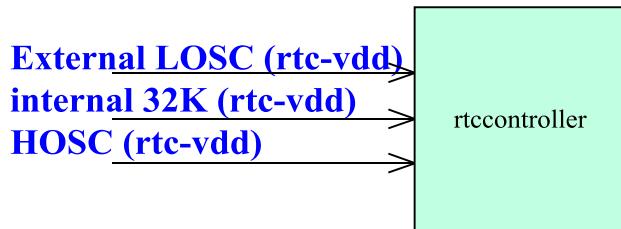
| USB controller clock |       |     |              |  |
|----------------------|-------|-----|--------------|--|
| Signal               | Width | I/O | Clock Domain | Description  |
| USBCLKEN             | 1     | I   | CPUCLK       | Bit 6 of CLKENCTL1 register                              |
| PHYPLLEN             | 1     | I   | CPUCLK       | Bit 7 of USB_PHYCTRL register                            |
| HOSC                 | 1     | I   | -            | 24MHz  |
| CPUCLK               | 1     | I   | -            | (1) SFR of USB controller<br>(2) DMA read/write USB FIFO |
| USB_controller_clk   | 1     | O   | -            | 120MHz(USB controller digital part)                      |
| USBctl_RAM_clk       | 1     | O   | -            | 60MHz(URAM,PCMRAM,FIR_RDS_RAM)                           |

signal type legend:



#### 7.4.4.9 RTC controller clock

The clock to drive RTC controller is listed below:



2026、 The internal or external LOSC is selected by LOSC\_sel bit of RTC\_CTL1 (bit5).

Figure7-18 RTC clock control

#### 7.4.4.10 DAC/IIS-TX controller clock

The clock to drive DAC/IIS-TX controller is listed below:

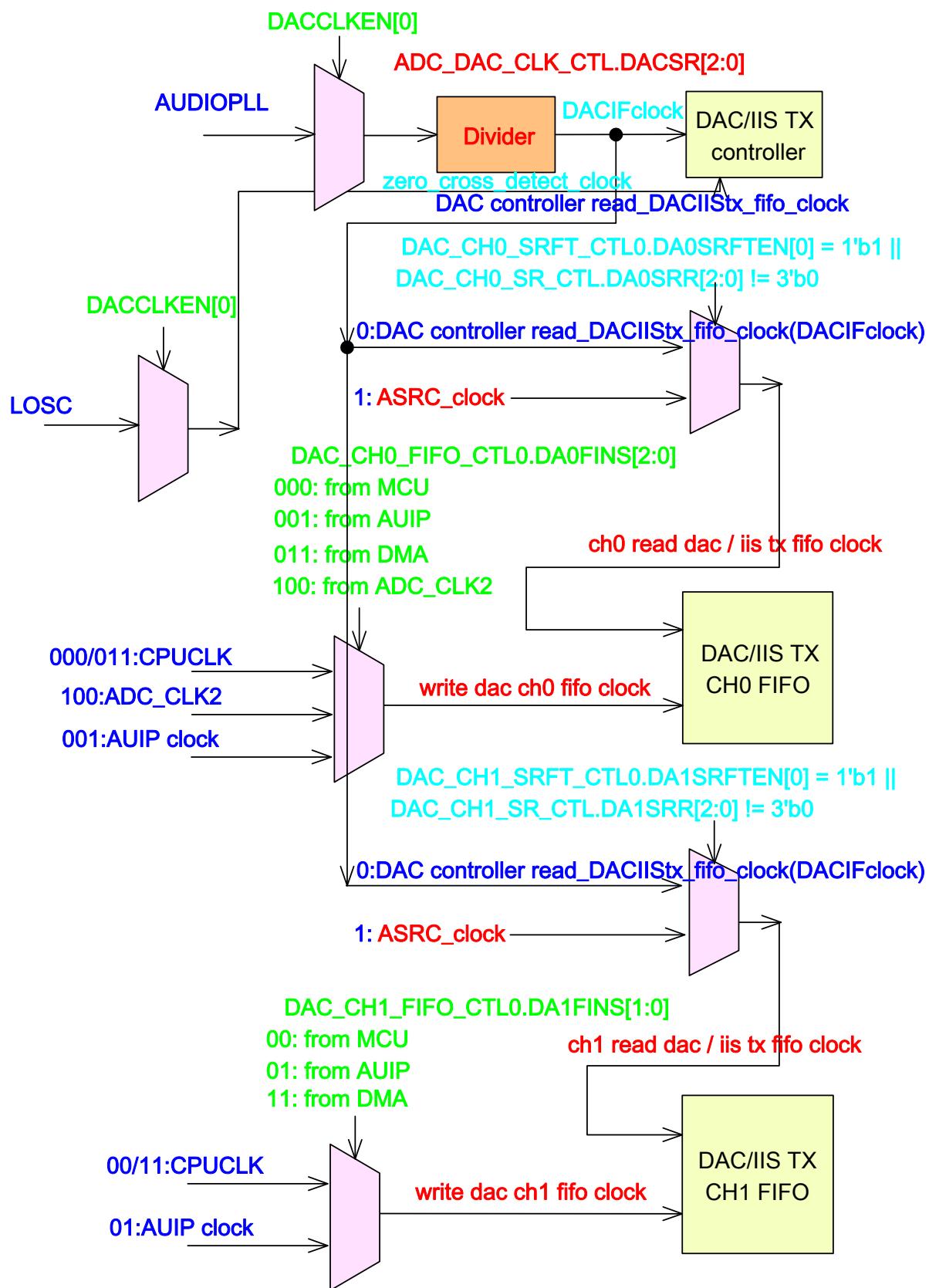


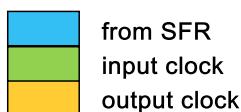
Figure7-19 DAC/IIS-TX controller clock control

Note:

- (1) The DACCLK[2:0] field is in the ADC\_DAC\_CLK\_CTL register.
- (2) The DA0FINS[2:0] field is in the DAC\_CH0FIFO\_CTL0 register. No seamless clock switching is needed.
- (3) The DA1FINS[1:0] field is in the DAC\_CH1FIFO\_CTL0 register. No seamless clock switching is needed.
- (4) If DAC\_CH0\_FIFOCTL0.DA0FINS[2:0] == 3'b100, the frequency of CPUCLK must greater than DACIFclock.
- (5) The internal or external L OSC is selected by LOSC\_sel bit of RTC\_CTL1 (bit5).
- (6) Seamless clock switching is not supported except for DAC clock dividend switching controlled by the register field ADC\_DAC\_CLK\_CTL.DACSR[2:0].

| DAC/IIS TX controller clock |       |     |              |  |
|-----------------------------|-------|-----|--------------|--|
| Signal                      | Width | I/O | Clock Domain | Description  |
| DACCLKEN                    | 1     | I   | CPUCLK       | Bit 5 of CLKENCTL0 register                                      |
| DACSR                       | 3     | I   | CPUCLK       | Bit 2:0 of ADC_DAC_CLK_CTL register                              |
| DA0FINS                     | 3     | I   | CPUCLK       | Bit 3:1 of DAC_CH0_FIFO_CTL0 register                            |
| DA1FINS                     | 2     | I   | CPUCLK       | Bit 2:1 of DAC_CH1_FIFO_CTL1 register                            |
| DA0SRFTEN                   | 1     | I   | CPUCLK       | Bit 0 of DAC_CH0_SRFT_CTL0 register                              |
| DA0SRR                      | 3     | I   | CPUCLK       | Bit 2:0 of DAC_CH0_SR_CTL register                               |
| DA1SRFTEN                   | 1     | I   | CPUCLK       | Bit 0 of DAC_CH1_SRFT_CTL0 register                              |
| DA1SRR                      | 3     | I   | CPUCLK       | Bit 2:0 of DAC_CH1_SR_CTL register                               |
| AUDIOPLL                    | 1     | I   | -            | 24.576/22.0592MHz  |
| CPUCLK                      | 1     | I   | -            | (1) SFR of DAC/IISTX controller<br>(2) DMA/CPU write DAC FIFO0/1 |
| ASRC_clock                  | 1     | I   | -            | ASRC controller  |
| AUIP clk                    | 1     | I   | -            | Auiip clock  |
| ADC_CLK2                    | 1     | I   | -            | ADC analog clock   |
| LOSC                        | 1     | I   | -            | For zero cross detect  |
| DACIFclock                  | 1     | O   | -            | DAC/IISTX controller   |
| write dac ch0 fifo clock    | 1     | O   | -            | DAC/IISTX FIFO0  |
| write dac ch1 fifo clock    | 1     | O   | -            | DAC/IISTX FIFO1  |
| Ch0 read DAC/ IIS tx clock  | 1     | O   | -            | DAC/IISTX FIFO0  |
| Ch1 read DAC/ IIS tx clock  | 1     | O   | -            | DAC/IISTX FIFO1  |
| Zero_cross_detect_clock     | 1     | O   | -            | DAC/IISTX controller   |

signal type legend:



### 7.4.4.11 ADC and SPDIF rx controller clock

The clock to drive ADC and SPDIF rx controller is listed below:

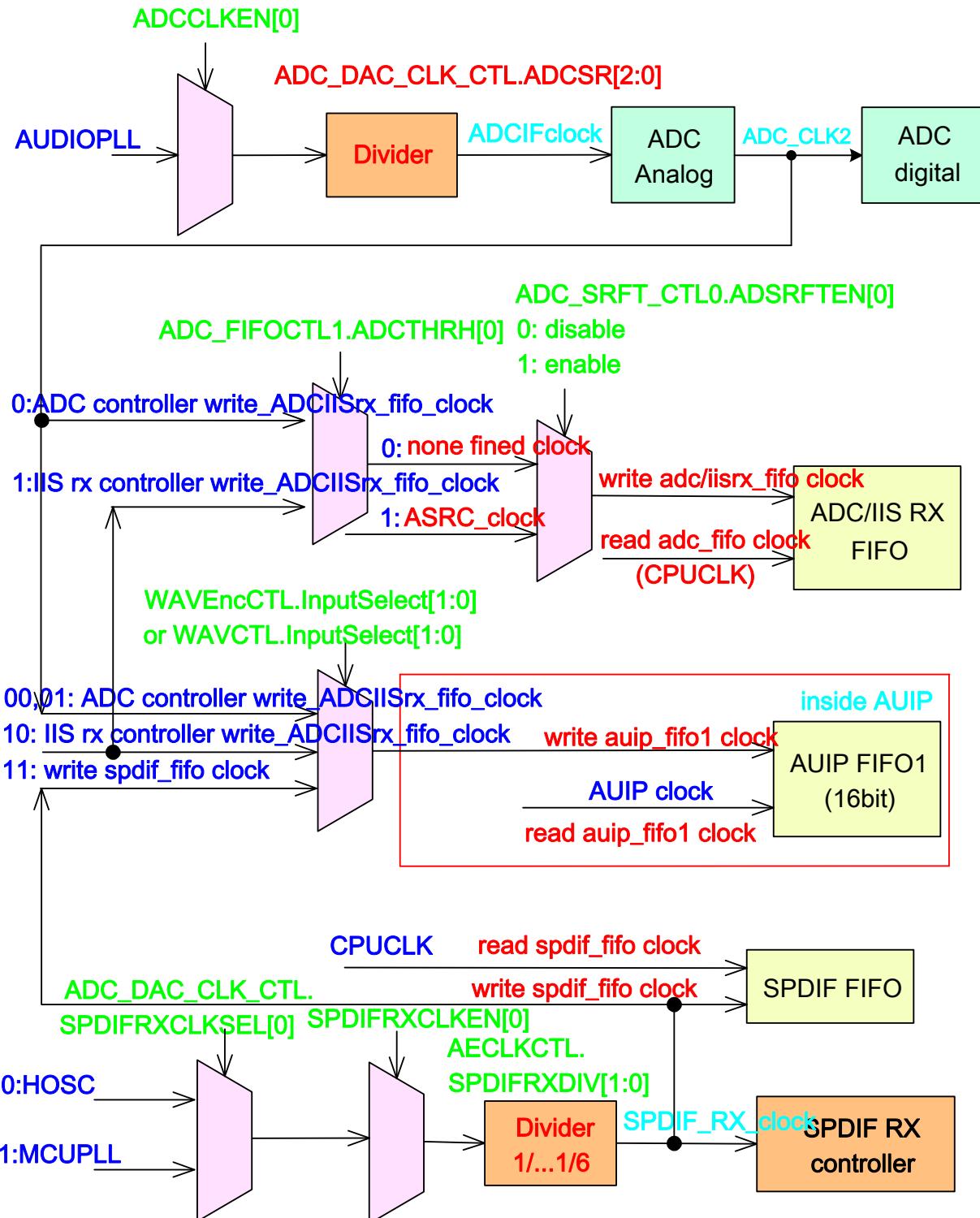


Figure 7-20 ADC and SPDIF rx controller clock control

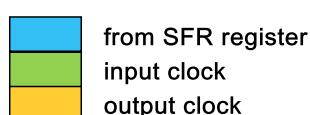
Note:

- (1) The **ADCSR[2:0]** field is in the **ADC\_DAC\_CLK\_CTL** register.

- (2) The bit 5:4 (ADC0FSS[1:0]) of ADC\_FIFOCTL1 register is used to control the source of ADC-FIFO-READ clock. The reset value of ADC0FSS[1:0] is 2'b0.
- (3) The bit 3 (ADC1FSS[0]) of ADC\_CTL1 register is used to control the source of ADC-FIFO-READ clock. The reset value of ADC1FSS [0] is 1'b0.
- (4) ASRC write\_ADCIISrx\_FIFO\_clock is generated by the [ASRC module](#).
- (5) Seamless clock switching of ADCIFCLK is supported.
- (6) Seamless clock switching of none\_defined\_clock, write adc/iisrx\_fifo\_clock and write\_aup\_fifo1\_clock is not supported.
- (7) Seamless clock switching of SPDIF-RX controller is supported
- (8) The bit 3 of [ADC\\_DAC\\_CLK\\_CTL](#) register is SPDIFRXCLKSEL bit.
- (9) The frequency of SPDIF rx module clock must be greater than 49.152MHz to ensure receive audio samples @ 192ksps.
- (10) [The description of IIS\\_rx\\_controller\\_write\\_ADC/IISrx\\_fifo\\_clock is in the following section.](#)

| ADC and SPDIF RX controller clock                          |          |          |               |   |
|--|----------|----------|---------------|---|
| Signal   | Width    | I/O      | Clock Domain  | Description   |
| ADCCLKEN   | 1        | I        | CPUCLK        | Bit 4 of CLKENCTL0 register   |
| ADCSR  | 3        | I        | CPUCLK        | Bit 6:4 of ADC_DAC_CLK_CTL register   |
| ADCTHRH  | 1        | I        | CPUCLK        | Bit 0 of ADC_FIFOCTL1 register  |
| ADSRFTEN   | 1        | I        | CPUCLK        | Bit 0 of ADC_SRFT_CTL0 register   |
| InputSelect  | 2        | I        | CPUCLK        | Bit 2:1 of WAVCTL and WAVEncCTL register  |
| SPDIFRXCLKSEL  | 1        | I        | CPUCLK        | Bit 3 of ADC_DAC_CLK_CTL register   |
| SPDIFRXCLKEN   | 1        | I        | CPUCLK        | Bit 6 of CLKENCTL0 register   |
| <b>SPDIFRXDIV</b>  | <b>2</b> | <b>I</b> | <b>CPUCLK</b> | <b>Bit 7:6 of AECLKCTL register</b>   |
| AUDIOPLL   | 1        | I        | -             | 24.576/22.0592MHz   |
| HOSC   | 1        | I        | -             | 24MHz   |
| MCUPLL   | 1        | I        | -             | 24~60MHz  |
| CPUCLK   | 1        | I        | -             | (1) SFR of ADC and SPDIFRX controller<br>(2) DMA/CPU read ADC/IISRX FIFO<br>(3) DMA/CPU read SPDIF FIFO |
| ASRC_clock   | 1        | I        | -             | ASRC controller   |
| <a href="#">IISrx_controller_write_ADCIISrx_fifo_clock</a> | 1        | I        | -             | One of the clock source of write AUIP FIFO1   |
| ADCIFclock   | 1        | O        | -             | ADC/IISRX controller  |
| ADC_CLK2   | 1        | O        | -             | ADC controller  |
| write adc/iisrx_fifo clock                                 | 1        | O        | -             | ADC/IISRX FIFO  |
| write aup_fifo1 clock                                      | 1        | O        | -             | AUIP FIFO1(8*16bit)   |
| SPDIF_RX_clock   | 1        | O        | -             | SPDIFRX controller  |

signal type legend:



#### 7.4.4.12 IIS-RX controller clock

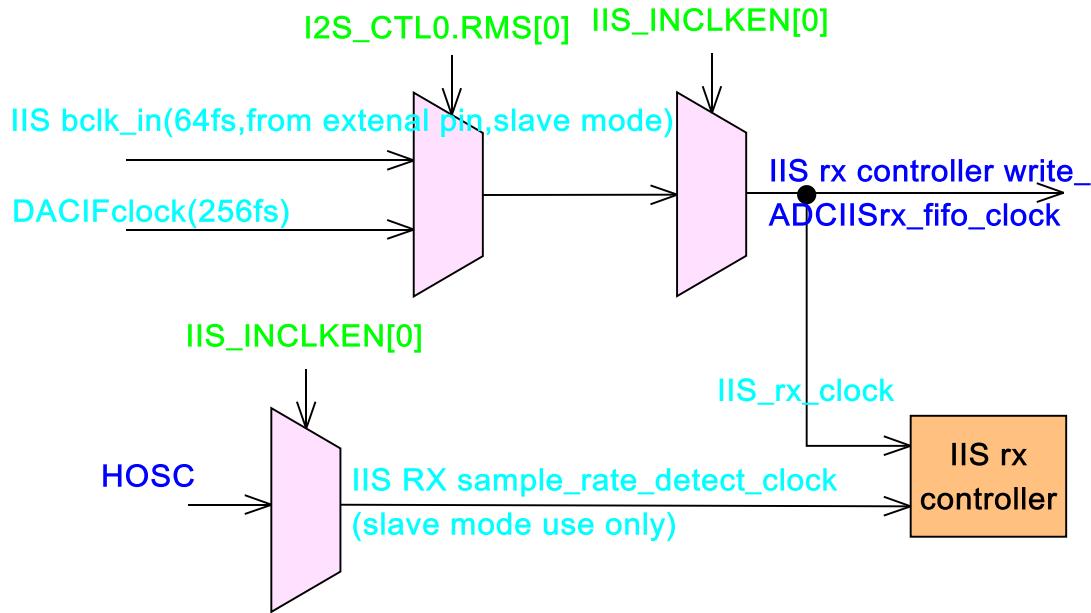
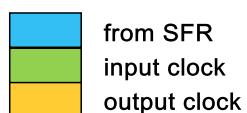


Figure 7-21 IIS-RX controller clock control

- (1) Seamless clock switching is not supported.
- (2) [IIS-RX controller use DACIFclock](#)

| IIS RX controller clock   |       |     |              |  |
|---|-------|-----|--------------|--|
| Signal  | Width | I/O | Clock Domain | Description  |
| IIS_INCLKEN   | 1     | I   | CPUCLK       | Bit 1 of CLKENCTL0 register  |
| RMS   | 1     | I   | CPUCLK       | Bit 3 of IIS_CTL0 register (fix to 0)  |
| IIS_bclk_in   | 1     | I   | -            | External clock input of IISrx controller(64Fs)                                 |
| DACIF_clock   | 1     | I   | -            | DAC controller clock(256Fs)  |
| HOSC  | 1     | I   |              | 24MHz  |
| CPUCLK  | 1     | I   | -            | (1) SFR of IIS RX controller<br>(2) DMA/CPU read ADC/IISRX FIFO                |
| <u><a href="#">IISrx_controller_write_ADCIISrx_fifo_clock</a></u> | 1     | O   | -            | (1) One of the clock source of write AUIP FIFO1<br>(2) IIS rx controller clock |

signal type legend:



#### 7.4.4.13 IR controller clock

The clock to drive IR controller is listed below:

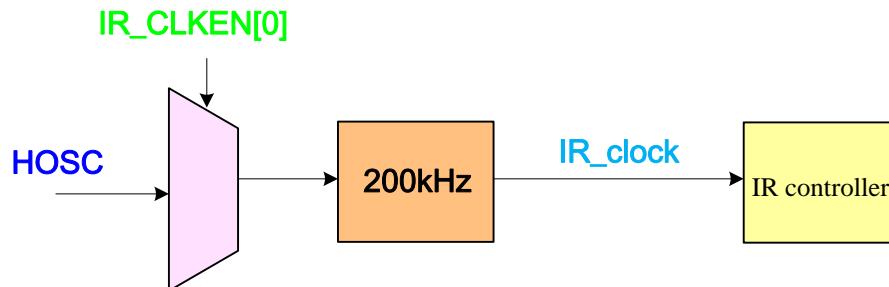
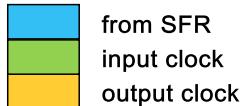


Figure 7-22 IR controller clock control

| IR controller clock |       |     |              |                             |
|---------------------|-------|-----|--------------|-----------------------------|
| Signal              | Width | I/O | Clock Domain | Description                 |
| IR_CLKEN            | 1     | I   | CPUCLK       | Bit 5 of CLKENCTL1 register |
| HOSC                | 1     | I   |              | 24MHz                       |
| CPUCLK              | 1     | I   | -            | SFR of IR controller        |
| IR_clock            | 1     | O   | -            | IR controller               |

signal type legend:



#### 7.4.4.14 ACC controller clock

The clock to drive file system accelerator controller is listed below

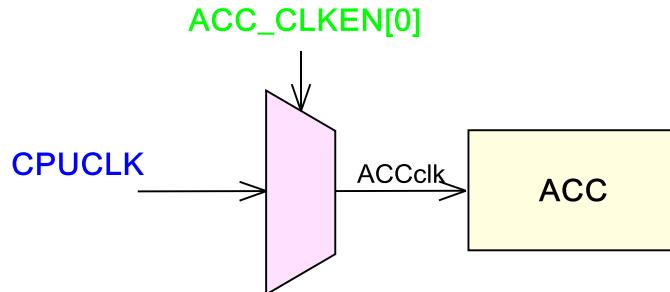
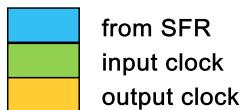


Figure 7-23 file system accelerator controller clock control

ACC controller clock

| Signal    | Width | I/O | Clock Domain | Description                 |
|-----------|-------|-----|--------------|-----------------------------|
| ACC_CLKEN | 1     | I   | CPUCLK       | Bit 3 of CLKENCTL1 register |
| CPUCLK    | 1     | I   | -            | SFR of IR controller        |
| ACCclk    | 1     | O   | -            | ACC controller              |

signal type legend:



#### 7.4.4.15 FM clock

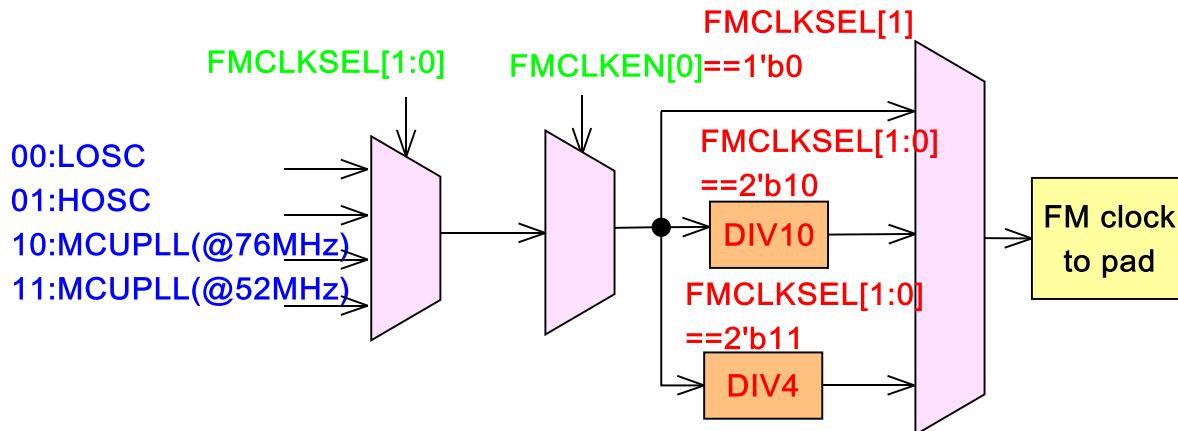


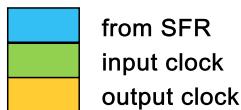
Figure 7-24 FM clock output control

Note:

- (1) The default value of FMCLKSEL[1:0] field is 2'b00, and the default value of FMCLKEN[0] field is 1'b0.
- (2) The clock divider is 10 if FMCLKSEL[1] = 1'b1.
- (3) The internal or external L OSC is selected by L OSC\_sel bit of RTC\_CTL1 (bit5).
- (4) Seamless clock switching is not supported.

| FM controller clock |       |     |              |                              |
|---------------------|-------|-----|--------------|------------------------------|
| Signal              | Width | I/O | Clock Domain | Description                  |
| FMCLKSEL            | 2     | I   | CPUCLK       | Bit 1:0 of FMCLKCTL register |
| FMCLKEN             | 1     | I   | CPUCLK       | Bit 0 of CLKENCTL0 register  |
| CPUCLK              | 1     | I   | -            | SFR of FM controller         |
| LOSC                | 1     | I   | -            | 32768Hz                      |
| HOSC                | 1     | I   | -            | 24MHz                        |
| MCUPLL              | 1     | I   | -            | 24~60MHz                     |
| FMclk               | 1     | O   | -            | FM clock to pad              |

signal type legend:



#### 7.4.4.16 PWM controller clock

The clock to drive PWM controller is listed below:

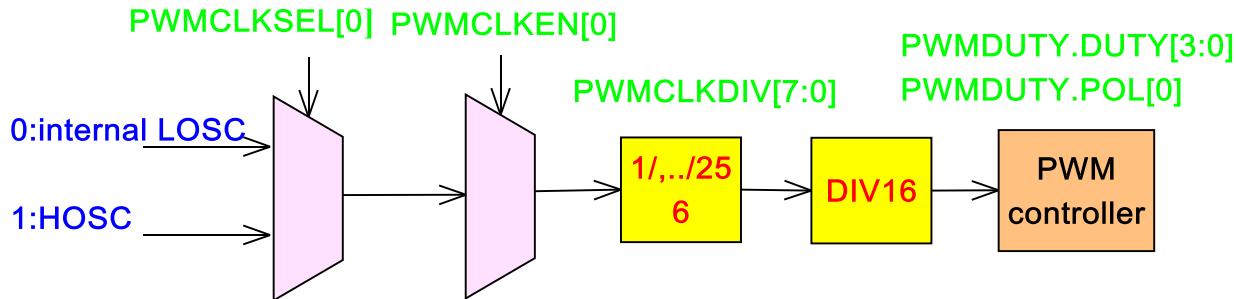


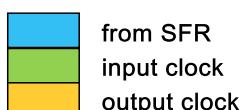
Figure 7-25 PWM clock control

Note:

- (1) Seamless clock switching is supported.
- (2) The default value is: PWMCLKEN[0]==1'b0 (PWMCLK is disable).
- (3) The default value is: PWMCLKSEL[0]==1'b0 (internal LOSC is selected).
- (4) The default value is: PWMCLKDIV[7:0]==8'hff (divider is 256).

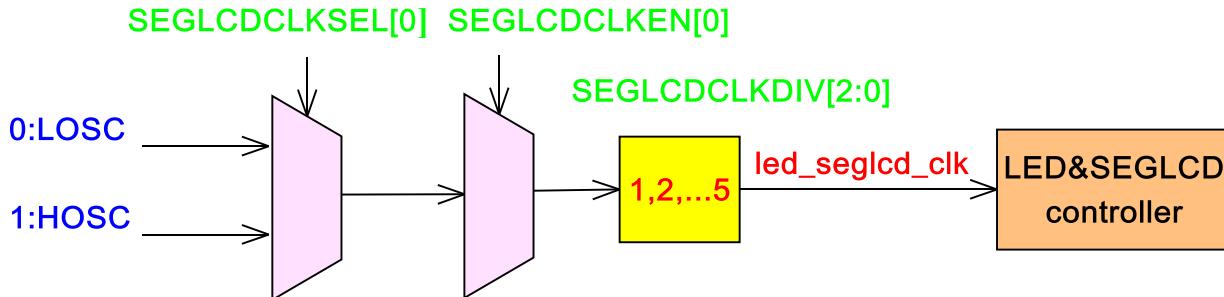
| PWM controller clock |       |     |              |                               |
|----------------------|-------|-----|--------------|-------------------------------|
| Signal               | Width | I/O | Clock Domain | Description                   |
| PWMCLKSEL            | 1     | I   | CPUCLK       | Bit 5 of PWMDUTY register     |
| PWMCLKDIV            | 8     | I   | CPUCLK       | Bit 7:0 of PWMCLKCTL register |
| POL                  | 1     | I   | CPUCLK       | Bit 4 of PWMDUTY register     |
| DUTY                 | 1     | I   | CPUCLK       | Bit 3:0 of PWMDUTY register   |
| PWMCLKEN             | 1     | I   | CPUCLK       | Bit 7 of CLKENCTL1 register   |
| CPUCLK               | 1     | I   | -            | SFR of PWM controller         |
| Internal LOSC        | 1     | I   | -            | 32768Hz                       |
| HOSC                 | 1     | I   | -            | 24MHz                         |
| PWMclk               | 1     | O   | -            | PWM clock                     |

signal type legend:



#### 7.4.4.17 LED & SEG LCD clock

The clock to drive LED & SEG LCD controller is listed below:

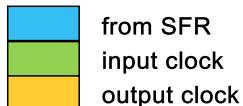


- (1) The internal or external LOSC is selected by LOSC\_sel bit of RTC\_CTL1 (bit5).
- (2) Seamless clock switching is not supported.

Figure 7-26 LED & SEG LCD clock control

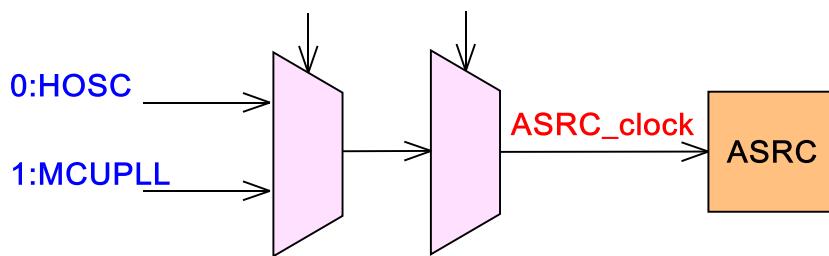
| LED & SEG LCD controller clock |       |     |              |                                      |
|--------------------------------|-------|-----|--------------|--------------------------------------|
| Signal                         | Width | I/O | Clock Domain | Description                          |
| SEGLCDCLKSEL                   | 1     | I   | CPUCLK       | Bit 0 of LED_SEGLCDCLKCTL register   |
| SEGLCDCLKDIV                   | 3     | I   | CPUCLK       | Bit 3:1 of LED_SEGLCDCLKCTL register |
| SEGLCDCLKEN                    | 1     | I   | CPUCLK       | Bit 4 of CLKENCTL2 register          |
| CPUCLK                         | 1     | I   | -            | SFR of LED & SEG LCD controller      |
| LOSC                           | 1     | I   | -            | 32768Hz                              |
| HOSC                           | 1     | I   | -            | 24MHz                                |
| Led_seglcd_clk                 | 1     | O   | -            | Led and seglcd clock                 |

signal type legend:



#### 7.4.4.18 ASRC clock

The clock to drive ASRC controller is listed below:

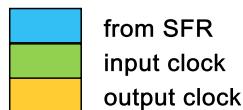
**ASRCCLKSEL[0] ASRCCLKEN[0]**


- (1) Seamless clock switching is supported.

Figure 7-27 ASRC clock control

| ASRC controller clock |       |     |              |                                   |
|-----------------------|-------|-----|--------------|-----------------------------------|
| Signal                | Width | I/O | Clock Domain | Description                       |
| ASRCCLKSEL            | 1     | I   | CPUCLK       | Bit 7 of ADC_DAC_CLK_CTL register |
| ASRCCLKEN             | 1     | I   | CPUCLK       | Bit 6 of CLKENCTL2 register       |
| CPUCLK                | 1     | I   | -            | SFR of ASRC controller            |
| HOSC                  | 1     | I   | -            | 24MHz                             |
| MCUPLL                | 1     | I   | -            | 24~60MHz                          |
| ASRC_clock            | 1     | O   | -            | ASRC controller                   |

signal type legend:



#### 7.4.4.19 ROM0/1 clock

The memory clock control of ROM0 and ROM1 are listed below:

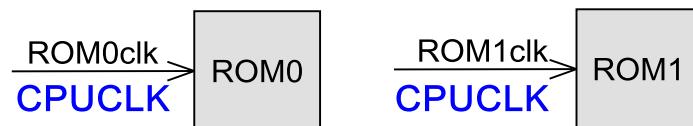


Figure 7-28 ROM0~1 clock control

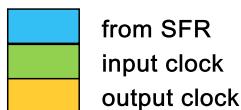
Note:

- (1) Memory access clock(including ROM and RAM access clock) and DMA clock must be synchronous

| ROM0/1 clock |       |     |              |             |
|--------------|-------|-----|--------------|-------------|
| Signal       | Width | I/O | Clock Domain | Description |

|            |   |   |   |        |
|------------|---|---|---|--------|
| ROM0_clock | 1 | O | - | CPUCLK |
| ROM1_clock | 1 | O | - | CPUCLK |

signal type legend:



#### 7.4.4.20 DRAM251/PRAM251 clock

The memory clock control of DRAM251 and PRAM251 listed below:

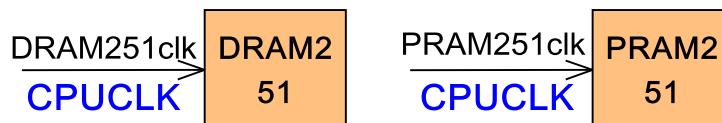


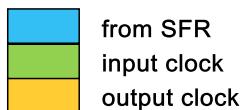
Figure 7-29 DRAM251 & PRAM251 clock control

Note:

- (1) Memory access clock(including ROM and RAM access clock) and DMA clock must be synchronous.

| DRAM251/PRAM251 clock |       |     |              |             |
|-----------------------|-------|-----|--------------|-------------|
| Signal                | Width | I/O | Clock Domain | Description |
| DRAM251_clock         | 1     | O   | -            | CPUCLK      |
| PRAM251_clock         | 1     | O   | -            | CPUCLK      |

signal type legend:



#### 7.4.4.21 URAM clock

The memory clock control of URAM is listed below:

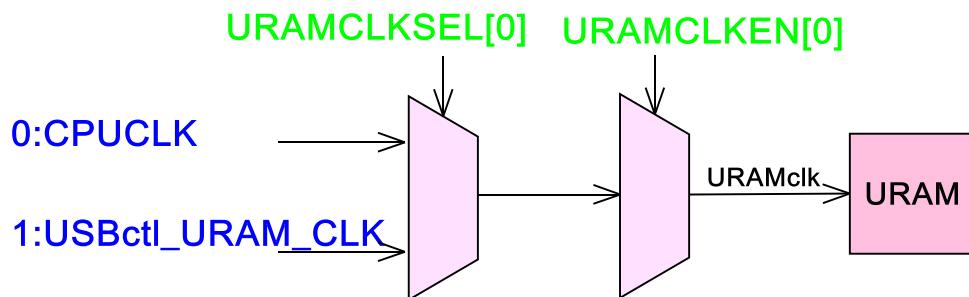


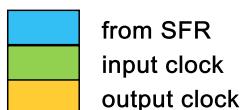
Figure 7-30 URAM clock control

Note:

- (1) The default value is: URAMxCLKSEL[0]==1'b1 (CPUclk is selected). URAMCLKEN[0]==1'b1 (URAM clock is enable).
- (2) USBctl\_URAM\_clk and USB controller clock must be synchronous.
- (3) Seamless clock switching is not supported.

| URAM clock      |       |     |              |                                 |
|-----------------|-------|-----|--------------|---------------------------------|
| Signal          | Width | I/O | Clock Domain | Description                     |
| URAMCLKSEL      | 1     | I   | CPUCLK       | Bit 7 of MEMCLKSELCTL1 register |
| URAMCLKEN       | 1     | I   | CPUCLK       | Bit 7 of MEMCLKENCTL0 register  |
| CPUCLK          | 1     | I   | -            | SFR clock                       |
| USBctl_URAM_clk | 1     | I   | -            | 60MHz                           |
| URAM_clock      | 1     | O   | -            | URAM                            |

signal type legend:



#### 7.4.4.22 FIR\_CS\_RAM clock

The memory clock control of FIR\_CS\_RAM is listed below:

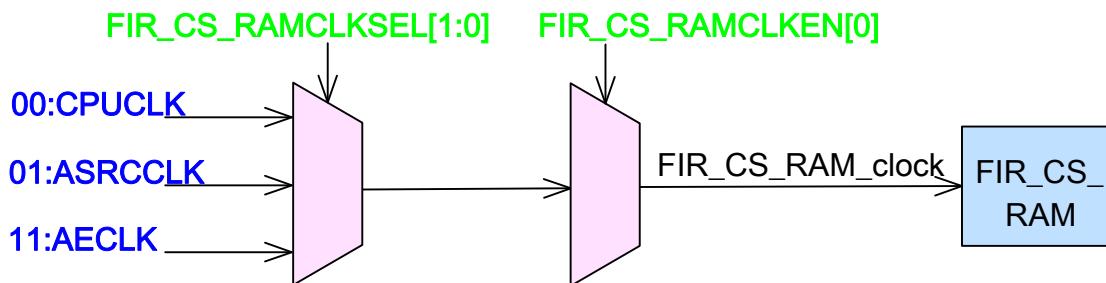


Figure 7-31 FIR\_CS\_RAM clock control

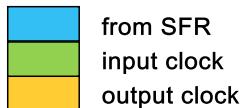
Note:

- (1) Seamless clock switching is not supported.
- (2) The default value is: FIR\_CS\_RAMCLKSEL[1:0]==2'b0 (CPU clock is selected). FIR\_CS\_RAMCLKEN[0]==1'b1 (FIR\_CS\_RAM clock is enable).
- (3) Memory access clock(including ROM and RAM access clock) and DMA clock must be synchronous.

| FIR_CS_RAM clock |       |     |              |                                   |
|------------------|-------|-----|--------------|-----------------------------------|
| Signal           | Width | I/O | Clock Domain | Description                       |
| FIR_CS_RAMCLKSEL | 2     | I   | CPUCLK       | Bit 5:4 of MEMCLKSELCTL0 register |

|                  |   |   |        |   |
|------------------|---|---|--------|---|
| FIR_CS_RAMCLKEN  | 1 | I | CPUCLK | Bit 5 of MEMCLKENCTL0 register                      |
| CPUCLK           | 1 | I | -      | (1) SFR clock<br>(2) one clock source of FIR_CS_RAM |
| ASRCCLK          | 1 | I | -      | 24~60MHz  |
| AECLK            | 1 | I | -      | 4~24MHz   |
| FIR_CS_RAM_clock | 1 | O | -      | FIR_CS_RAM  |

signal type legend:



#### 7.4.4.23 FIR\_AA\_RAM clock

The memory clock control of FIR\_AA\_RAM is listed below:

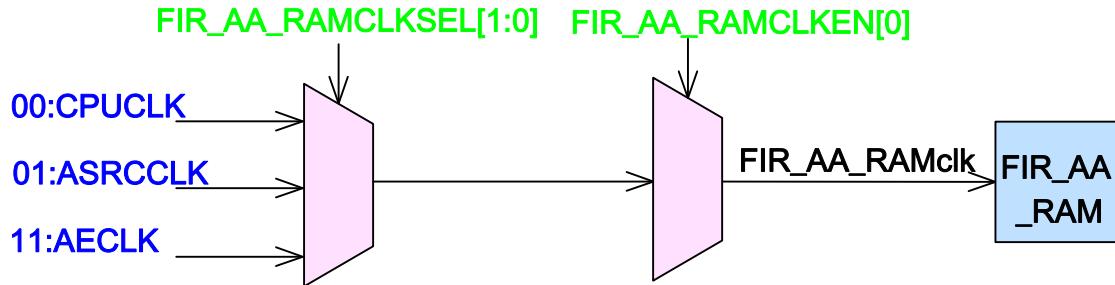


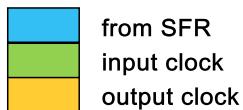
Figure 7-32 FIR\_AA\_RAM clock control

Note:

- (1) Seamless clock switching is not supported.
- (2) The default value is:  $\text{FIR\_AA\_RAMCLKSEL}[0]==2'b00$  (CPU clock is selected).  $\text{FIR\_AA\_RAMCLKEN}[0]==1'b1$  (FIR\_AA\_RAM clock is enable).

| FIR_AA_RAM clock |       |     |              |   |
|------------------|-------|-----|--------------|---|
| Signal           | Width | I/O | Clock Domain | Description   |
| FIR_AA_RAMCLKSEL | 2     | I   | CPUCLK       | Bit 3:2 of MEMCLKSELCTL0 register                   |
| FIR_AA_RAMCLKEN  | 1     | I   | CPUCLK       | Bit 4 of MEMCLKENCTL0 register                      |
| CPUCLK           | 1     | I   | -            | (1) SFR clock<br>(2) one clock source of FIR_AA_RAM |
| ASRCCLK          | 1     | I   | -            | 24~60MHz  |
| AECLK            | 1     | I   | -            | 4~24MHz   |
| FIR_AA_RAM_clock | 1     | O   | -            | FIR_AA_RAM  |

signal type legend:



#### 7.4.4.24 FIR\_MPX\_RAM clock

The memory clock control of FIR\_MPX\_RAM is listed below:

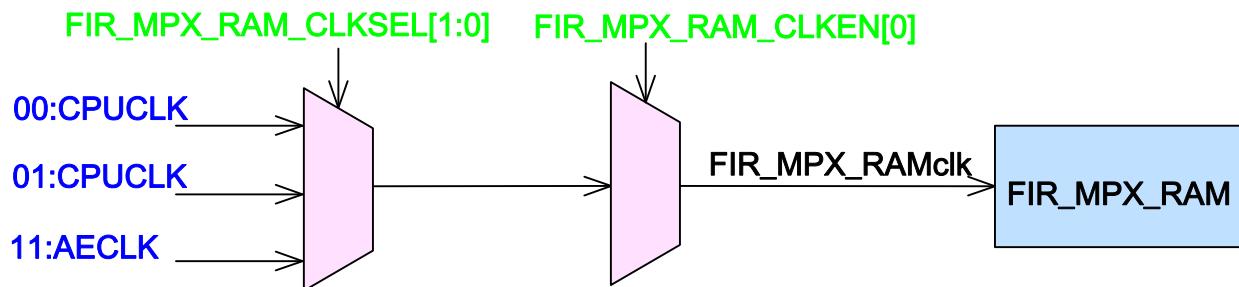


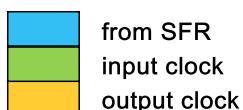
Figure 7-33 FIR\_MPX\_RAM\_LM clock control

Note:

- (1) Seamless clock switching is not supported.
- (2) The default value is:  $\text{FIR\_MPX\_RAMCLKSEL}[0]==2'b00$  (CPU clock is selected).  $\text{FIR\_MPX\_RAMCLKEN}[0]==1'b1$  (FIR\_MPX\_RAM clock is enable).

| FIR_MPX_RAM clock              |       |     |              |  |
|--------------------------------|-------|-----|--------------|--|
| Signal                         | Width | I/O | Clock Domain | Description  |
| <code>FIR_MPX_RAMCLKSEL</code> | 2     | I   | CPUCLK       | Bit 1:0 of MEMCLKSELCTL0 register                    |
| <code>FIR_MPX_RAMCLKEN</code>  | 1     | I   | CPUCLK       | Bit 3 of MEMCLKENCTL0 register                       |
| <code>CPUCLK</code>            | 1     | I   | -            | (1) SFR clock<br>(2) one clock source of FIR_MPX_RAM |
| <code>AECLK</code>             | 1     | I   | -            | 4~24MHz  |
| <code>FIR_MPX_RAM_clock</code> | 1     | O   | -            | FIR_MPX_RAM  |

signal type legend:



#### 7.4.4.25 FIR\_RDS\_RAM clock

The memory clock control of FIR\_RDS\_RAM is listed below:

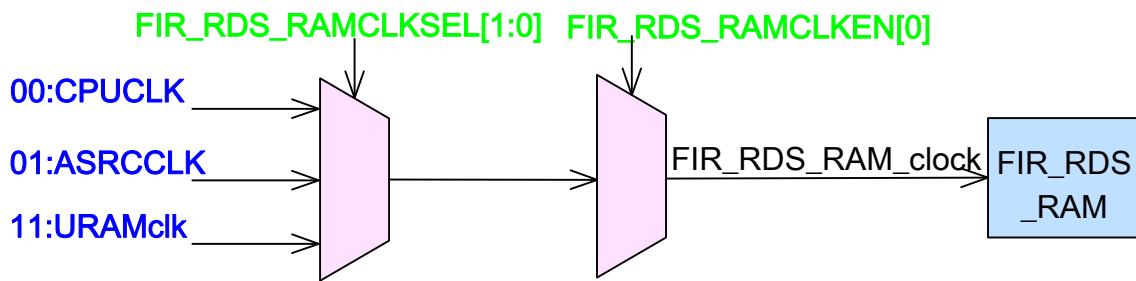


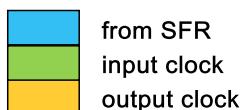
Figure 7-34 FIR\_RDS\_RAM clock control

Note:

- (1) Seamless clock switching is not supported.
- (2) The default value is: FIR\_RDS\_RAMCLKSEL[1:0]==2'b00 (CPU clock is selected). FIR\_RDS\_RAMCLKEN[0]==1'b1 (FIR\_RDS\_RAM clock is enable).
- (3) Memory access clock (including ROM and RAM access clock) and DMA clock must be synchronous.

| FIR_RDS_RAM clock |       |     |              |  |
|-------------------|-------|-----|--------------|--|
| Signal            | Width | I/O | Clock Domain | Description  |
| FIR_RDS_RAMCLKSEL | 2     | I   | CPUCLK       | Bit 3:2 of MEMCLKSELCTL1 register                    |
| FIR_RDS_RAMCLKEN  | 1     | I   | CPUCLK       | Bit 1 of MEMCLKENCTL0 register                       |
| CPUCLK            | 1     | I   | -            | (1) SFR clock<br>(2) one clock source of FIR_RDS_RAM |
| ASRCCLK           | 1     | I   | -            | 24~60MHz   |
| FIR_RDS_RAM_clock | 1     | O   | -            | FIR_RDS_RAM  |

signal type legend:



#### 7.4.4.26 PCMRAM clock

The memory clock control of PCMRAM is listed below:

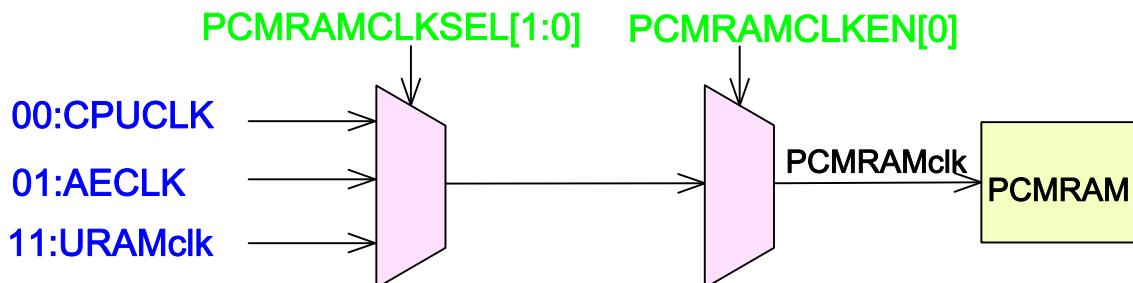


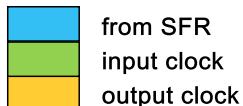
Figure 7-35 PCMRAM clock control

Note:

- (1) Seamless clock switching is not supported.
- (2) The default value is: PCMRAMCLKSEL[1:0]==2'b0 (CPU clock is selected).  
PCMRAMCLKEN[0]==1'b1 (PCMRAM clock is enable).

| PCMRAM clock |       |     |              |   |
|--------------|-------|-----|--------------|---|
| Signal       | Width | I/O | Clock Domain | Description                                     |
| PCMRAMCLKSEL | 2     | I   | CPUCLK       | Bit 1:0 of MEMCLKSELCTL1 register               |
| PCMRAMCLKEN  | 1     | I   | CPUCLK       | Bit 0 of MEMCLKENCTL0 register                  |
| CPUCLK       | 1     | I   | -            | (1) SFR clock<br>(2) one clock source of PCMRAM |
| AECLK        | 1     | I   | -            | 4~24MHz   |
| PCMRAMclk    | 1     | O   | -            | PCMRAM  |

signal type legend:



#### 7.4.4.27 MURAM1 clock

The memory clock control of MURAM1 is listed below:

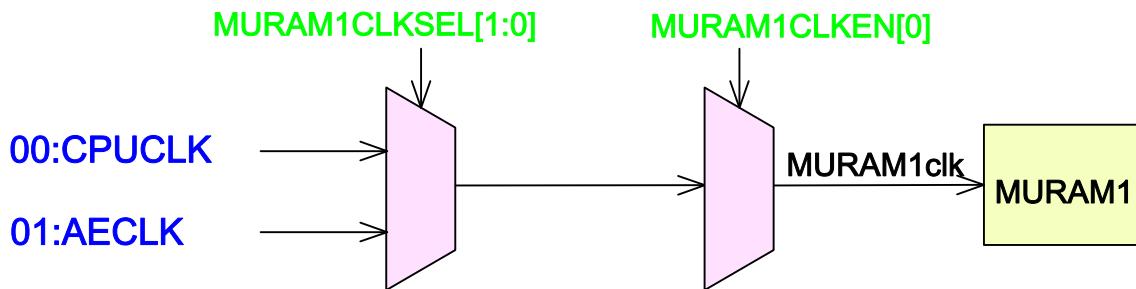


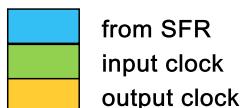
Figure 7-36 MURAM1 clock control

- (1) Seamless clock switching is not supported.
- (2) The default value is: MURAM1CLKSEL[1:0]==2'b0 (CPUCLK is selected).

| MURAM1 clock |       |     |              |   |
|--------------|-------|-----|--------------|---|
| Signal       | Width | I/O | Clock Domain | Description                                     |
| MURAM1CLKSEL | 2     | I   | CPUCLK       | Bit 7:6 of MEMCLKSELCTL0 register               |
| MURAM1CLKEN  | 1     | I   | CPUCLK       | Bit 6 of MEMCLKENCTL0 register                  |
| CPUCLK       | 1     | I   | -            | (1) SFR clock<br>(2) one clock source of MURAM1 |

|           |   |   |   |         |
|-----------|---|---|---|---------|
| AECLK     | 1 | I | - | 4~24MHz |
| MURAM1clk | 1 | O | - | MURAM1  |

signal type legend:



### 7.4.5 CMU\_Digital\_Part Register List

The register of clock management module include 2 parts of register with different functions:

2027、 Peripheral Clock Control Register group:

These registers includes DMA, audio codec, video codec, USB controller, LCD controller are driven by LOSC or HOSC or MCUPLL as peripheral clock of each module.

2028、 Memory Clock Control Register group:

These registers includes MEMCLKCTL0~ MEMCLKCTL1 which select peripheral clock, CPU clock as clock source of each memory block.

| Index | Mnemonic         | Description                               | BANK |
|-------|------------------|---|------|
| 0xa9  | CLKENCTL0        | Clock Enable Control Register 0           | 0x01 |
| 0xaa  | CLKENCTL1        | Clock Enable Control Register 1           | 0x01 |
| 0xab  | CLKENCTL2        | Clock Enable Control Register 2           | 0x01 |
| 0xac  | SDCLKCTL         | SD Card Control Register                  | 0x01 |
| 0xc2  | CPUCLKCTL        | CPU Clock Control Register                | 0x01 |
| 0xc3  | PWMCLKCTL        | PWM Clock Control Register                | 0x01 |
| 0xaf  | PWMDUTY          | PWM Duty Control Register                 | 0x01 |
| 0xc4  | AECLKCTL         | Audio Codec Clock Control Register        | 0x01 |
| 0xc7  | Reserved         | Reserved Register                         | 0x01 |
| 0xc8  | FMCLKCTL         | FM Clock Control Register                 | 0x01 |
| 0xc9  | LED_SEGLCDCLKCTL | LED & SEG LCD clock Control Register      | 0x01 |
| 0xca  | SPICLKCTL        | SPI controller Clock Control Register     | 0x01 |
| 0xbd  | ADC_DAC_CLK_CTL  | ADC & DAC Clock Control Register          | 0x01 |
| 0xad  | MEMCLKSELCTL0    | Memory Clock Selection Control Register 0 | 0x01 |
| 0xae  | MEMCLKSELCTL1    | Memory Clock Selection Control Register 1 | 0x01 |
| 0xb4  | MEMCLKENCTL0     | Memory Clock Enable Control Register 0    | 0x01 |

### 7.4.6 Register Description

#### 7.4.6.1 CLKENCTL0

**CLKENCTL0 (Clock Enable Control Register 0, SFR: 0xa9, SFR bank 0x01)**

| Bit Number | Bit Mnemonic     | Function   | Access | Reset |
|------------|------------------|--|--------|-------|
| 7          | AECLKEN          | Audio codec clock enable bit:<br>0: disable;<br>1: enable;         | R/W    | 0     |
| 6          | SPDIFRXCLKE<br>N | SPDIF RX controller clock enable bit:<br>0: disable;<br>1: enable; | R/W    | 0     |
| 5          | DACCLKEN         | DAC controller clock enable bit:<br>0: disable;<br>1: enable;      | R/W    | 0     |
| 4          | ADCCLKEN         | ADC controller clock enable bit:<br>0: disable;<br>1: enable;      | R/W    | 0     |
| 3          | SDCLKEN          | SD card controller clock enable bit:<br>0: disable;<br>1: enable;  | R/W    | 0     |
| 2          | Reserved         | Reserved for digital future use.                                   | R/W    | 0     |
| 1          | IIS_INCLKEN      | IISIN special clock enable bit:<br>0: disable;<br>1: enable;       | R/W    | 0     |
| 0          | FMCLKEN          | FM clock enable bit:<br>0: disable;<br>1: enable;                  | R/W    | 0     |

#### 7.4.6.2 CLKENCTL1

**CLKENCTL1 (Clock Enable Control Register 1, SFR: 0xaa, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | PWMCLKEN     | PWM clock enable bit:<br>0: disable;<br>1: enable;            | R/W    | 0     |
| 6          | USBCLKEN     | USB controller clock enable bit:<br>0: disable;<br>1: enable; | R/W    | 0     |
| 5          | IR_CLKEN     | IR clock enable bit:<br>0: disable;<br>1: enable;             | R/W    | 0     |
| 4          | Reserved     | Reserved for digital future use.                              | R/W    | 0     |
| 3          | SPICLKEN     | SPI controller clock enable bit:<br>0: disable;<br>1: enable; | R/W    | 0     |

|   |           |  |     |   |
|---|-----------|--|-----|---|
| 2 | Reserved  | Be read as zero.   | -   | - |
| 1 | Reserved  | Reserved for digital future use.                               | R/W | 0 |
| 0 | UARTCLKEN | UART controller clock enable bit:<br>0: disable;<br>1: enable; | R/W | 0 |

#### 7.4.6.3 CLKENCTL2

**CLKENCTL2 (Clock Enable Control Register 3, SFR: 0xab, SFR bank 0x01)**

| Bit Number | Bit Mnemonic      | Function   | Access | Reset |
|------------|-------------------|--|--------|-------|
| 7          | TKCLKEN           | Touch Key clock enable bit:<br>0: disable<br>1: enable       | R/W    | 0     |
| 6          | ASRCCLKEN         | ASRC clock enable bit:<br>0: disable;<br>1: enable;          | R/W    | 0     |
| 5          | DMA01234CLK<br>EN | DMA01234 clock enable bit:<br>0: disable;<br>1: enable;      | R/W    | 0     |
| 4          | SEGLCDCLKE<br>N   | LCD & SEG LCD clock enable bit:<br>0: disable;<br>1: enable; | R/W    | 0     |
| 3          | ACC_CLKEN         | ACC clock enable bit:<br>0: disable;<br>1: enable;           | R/W    | 0     |
| 2:0        | Reserved          | Be read as 3 zeros.  | -      | -     |

Note: The register bits FM\_CLKEN and FMCLKSEL are in the bits 2:0 of MFPSEL2 (SFR bank: 0x6 , address = 0xcc).

#### 7.4.6.4 SDCLKCTL

**SDCLKCTL (SD Card Clock Control Register, SFR: 0xac, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | Reserved     | Be read as 4 zeros   | -      | -     |
| 3:2        | SDCLKDIV     | SD clock Divisor<br>00: Divisor is 1<br>01: Divisor is 2<br>10: Divisor is 3<br>11: Divisor is 4 | R/W    | 0     |
| 1          | SDCLKSEL1    | SD card controller clock select bit 1:   | R/W    | 0     |

|   |           |  |     |   |
|---|-----------|--|-----|---|
|   |           | 0: low frequency selected (divisor is 256)<br>1: high frequency selected (divisor is1) |     |   |
| 0 | SDCLKSEL0 | SD card controller clock select bit 0:<br>0: HOSC;<br>1: MCUPLL                        | R/W | 0 |

### 7.4.6.5 CPUCLKCTL

**CPUCLKCTL (CPU Clock Control Register, SFR: 0xc2, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | Reserved     | Be read as 27-zeros   | -      | -     |
| 3:2        | CPUCLKDIV    | CPU clock Divisor<br>00: Divisor is 1<br>01: Divisor is 2<br>10: Divisor is 4<br>11: Divisor is 8 | R/W    | 0     |
| 0          | CPUCLKSEL    | CPU clock selection:<br>00: LOSC;<br>01: LOSC;<br>10: HOSC;<br>11: MCUPLL;                        | R/W    | 2'b10 |

Note:

- (1) CPUCLK and SCLK must be synchronous.
- (2) HOSC is generated by LOSC or driven by HOSC
- (3) The default value : CPUCLKSEL[1:0]==2'b10, CPUCLKDIV[1:0] == 2'b00
- (4) Seamless clock switching is supported.

### 7.4.6.6 PWMCLKCTL

**PWMCLKCTL (PWM Controller Clock Control Register, SFR: 0xc3, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | PWMCLKDIV    | PWM clock Divisor<br>8'h00: Divisor is 1<br>8'h00: Divisor is 2<br>8'h01: Divisor is 3<br>. .<br>8'hff: Divisor is 256 | R/W    | 8'hff |

Note:

- (1) Seamless clock switching is supported.

- (2) The default value is: PWMCLKSEL [0]==1'b0 (internal LOSC is selected) , PWMCLKDIV[7:0]==8'hff.

### 7.4.6.7 PWMDUTY

**PWMDUTY (PWM Duty Control Register, SFR: 0xaf, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset       |
|------------|--------------|--|--------|-------------|
| 7:6        | Reserved     | Be read as 2 zeros.  | -      | -           |
| 5          | PWMCLSEL     | PWM controller clock select bit:<br>0: internal LOSC;<br>1: HOSC;  | R/W    | 0           |
| 4          | POL          | Active Polarity Select.<br>0:The PWM is High level active<br>1:The PWM is Low level active   | R/W    | 0           |
| 3:0        | DUTY         | Active Duty Occupancy.<br>0000: 0/16<br>0001: 1/16<br>0010: 2/16<br>0011: 3/16<br>0100: 4/16<br>0101: 5/16<br>0110: 6/16<br>0111: 7/16<br>1000: 8/16<br>1001: 9/16<br>1010: 10/16<br>1011: 11/16<br>1100: 12/16<br>1101: 13/16<br>1110: 14/16<br>1111: 15/16 | R/W    | 4'b<br>1000 |

### 7.4.6.8 AECLKCTL

**AECLKCTL (Audio Codec Clock Control Register, SFR: 0xc4, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:6        | SPDIFRXDIV   | SPDIF-RX controller clock divisor :<br>00 : Divisor is 1;<br>01 : Divisor is 2;<br>10 : Divisor is 4;<br>11 : Divisor is 6 | R/W    | 0     |

|     |           |  |     |    |
|-----|-----------|--|-----|----|
| 5:4 | Reserved  | Be read as 2 zeros   | -   | -  |
| 3   | AECLKDIV0 | Audio Codec clock Divisor 0:<br>0: Divisor is 1<br>1: Divisor is 2/3   | R/W | 0  |
| 2:1 | AECLKDIV1 | Audio Codec clock Divisor 1:<br>00: Divisor is 1;<br>01: Divisor is 2;<br>10: Divisor is 4;<br>11: Divisor is 8; | R/W | 00 |
| 0   | AECLKSEL  | Audio codec clock select bit:<br>0: HOSC;<br>1: MCUPLL;  | R/W | 0  |

Note:

- (1) Seamless clock switching of AE clock is supported.
- (2) Seamless clock switching of ASRC clock is supported.
- (3) The default value is: AECLKSEL[0]==1'b0 (HOSC is selected), VECLKEN[0]==1'b0 (audio codec clock is disable). AECLKDIV0[0] = 1'b0 (divider is /1). AECLKDIV1[1:0] = 2'b00 (divider is /1).

#### 7.4.6.9 Reserved

**Reserved (Reserved Register, SFR: 0xc7, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function         | Access | Reset |
|------------|--------------|------------------|--------|-------|
| 7:0        | Reserved     | Be read as zeros | -      | -     |

#### 7.4.6.10 FMCLKCTL

**FMCLKCTL (FM Clock Control Register, SFR: 0xc8, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros   | -      | -     |
| 1:0        | FMCLKSEL     | FM clock select bit:<br>00: LOSC;<br>01: HOSC<br>10: MCUPLL(@76MHz) div 10;<br>11: MCUPLL(@52MHz) div 4; | R/W    | 00    |

Note:

- (1) The default value of FMCLKSEL[1:0] field is 2'b00, and the default value of FMCLKEN[0] field is 1'b0.
- (2) The clock divider is 15 if FMCLKSEL[1] = 1'b1.
- (3) The internal or external LOSC is selected by LFPLL\_SEL bit of LFPLL\_CTL register.
- (4) Seamless clock switching is not supported.

### 7.4.6.11 LED\_SEGLCDCLKCTL

**LED\_SEGLCDCLKCTL (LED & SEG LCD clock Control Register, SFR: 0xc9, SFR bank 0x01)**

| Bit Number | Bit Mnemonic     | Function  | Access | Reset |
|------------|------------------|---|--------|-------|
| 7:4        | Reserved         | Be read as 5-zeros  | -      | -     |
| 3:1        | SEGLCDCLKD<br>IV | LED & SEG LCD clock select bit:<br>000: Divisor is 1;<br>001: Divisor is 2;<br>010: Divisor is 3;<br>011: Divisor is 4;<br>100: Divisor is 5;<br>Others: reserved | R/W    | 00    |
| 0          | SEGLCDCLKS<br>EL | LED & SEG LCD clock select bit:<br>0: LOSC;<br>1: HOSC  | R/W    | 0     |

### 7.4.6.12 SPICLKCTL

**SPICLKCTL (SPI Controller Clock Control Register, SFR: 0xca, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:2        | Reserved     | Be read as 6 zeros.   | -      | -     |
| 1:0        | SPICLKSEL    | SPI Controller Clock Select<br>00: COREClk<br>01: COREClk<br>10: HOSC<br>11: MCUPLL | R/W    | 00    |

Note:

- (1) The default value is: SPICLKEN[0]==1'b0 (SPICLK is disable).
- (2) The default value is: SPICLKSEL[1:0]==2'd0 (COREClk is selected).

### 7.4.6.13 ADC\_DAC\_CLK\_CTL

**ADC\_DAC\_CLK\_CTL (ADC & DAC Clock Control Register, SFR: 0xbd, SFR bank 0x01)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset  |
|------------|--------------|--|--------|--------|
| 7          | ASRCCLKSEL   | ASRC clock select bit:<br>0: HOSC;<br>1: MCUPLL; | R/W    | 0      |
| 6:4        | ADCSR        | ADC Clock Divisor, output frequency is           | R/W    | 3'b001 |

|     |                   |  |     |        |
|-----|-------------------|--|-----|--------|
|     |                   | Fs*256 <sup>(1)</sup>  |     |        |
| 3   | SPDIFRXCLKS<br>EL | SPDIF RX controller clock select bit:<br>0: HOSC;<br>1: MCUPLL | R/W | 0      |
| 2:0 | DACSR             | DAC Clock Divisor <sup>(2)</sup>                               | R/W | 3'b000 |

Note:

- (1) The ADC sample rate (ADCSR field) is listed below:

| DIV | PLL CLK  |            |         |
|-----|----------|------------|---------|
|     | 48K_Fout | 44.1K_Fout |         |
| 000 | 1        | 96k        | -       |
| 001 | 2        | 48k        | 44.1k   |
| 010 | 3        | 32k        | -       |
| 011 | 4        | 24k        | 22.05k  |
| 100 | 6        | 16k        | -       |
| 101 | 8        | 12k        | 11.025k |
| 110 | 12       | 8k         | -       |
| 111 | -        | -          | -       |

- (2) The DAC sample rate (DACSR field) is listed below:

| DIV | 48K_Fout |       | 44.1K_Fout |         |         |
|-----|----------|-------|------------|---------|---------|
|     | 512fs    | 256fs | 512fs      | 256fs   |         |
| 000 | 1        | 48k   | 96k        | 44.1k   | -       |
| 001 | 1.5      | 32k   | -          | -       | -       |
| 010 | 2        | 24k   | 48k        | 22.05k  | 44.1k   |
| 011 | 3        | 16k   | 32k        | -       | -       |
| 100 | 4        | 12k   | 24k        | 11.025k | 22.05k  |
| 101 | 6        | 8k    | 16k        | -       | -       |
| 110 | 8        | -     | 12k        | -       | 11.025k |
| 111 | 12       | -     | 8k         | -       | -       |

#### 7.4.6.14 MEMCLKENCTL0

**MEMCLKENCTL0 (Memory Clock Enable Control Register 0, SFR: 0xb4, SFR bank 0x01)**

| Bit Number | Bit Mnemonic        | Function  | Access | Reset |
|------------|---------------------|---|--------|-------|
| 7          | URAMClkEN           | URAM clock enable bit:<br>0: disable;<br>1: enable; | R/W    | 1     |
| 6          | MURAM1ClkEn         | MURAM1 Clock Enable<br>0: disable<br>1: enable      | R/W    | 1     |
| 5          | FIR_CS_RAMClk<br>En | FIR_CS_RAM Clock Enable<br>0: disable<br>1: enable  | R/W    | 1     |

|   |                  |   |     |   |
|---|------------------|---|-----|---|
| 4 | FIR_AA_RAMClkEn  | FIR_AA_RAM Clock Enable<br>0: disable<br>1: enable  | R/W | 1 |
| 3 | FIR_MPX_RAMClkEn | FIR_MPX_RAM Clock Enable<br>0: disable<br>1: enable | R/W | 1 |
| 2 | Reserved         | Be read as zero.                                    | -   | - |
| 1 | FIR_RDS_RAMClkEn | FIR_RDS_RAM Clock Enable<br>0: disable<br>1: enable | R/W | 1 |
| 0 | PCMRAMClkEn      | PCMRAM Clock Enable<br>0: disable<br>1: enable      | R/W | 1 |

Note:

- (1) Memory access clock(including ROM and RAM access clock) and DMA clock must be synchronous

#### 7.4.6.15 MEMCLKSELCTL0

**MEMCLKSELCTL0 (Memory Clock Selection Control Register 0, SFR: 0xad, SFR bank 0x01)**

| Bit Number | Bit Mnemonic      | Function  | Access | Reset |
|------------|-------------------|---|--------|-------|
| 7:6        | MURAM1CLKSEL      | MURAM1 clock selection bit:<br>00: CPU clock;<br>01: AE clock<br><b>10: Reserved</b><br><b>11: Reserved</b> | R/W    | 0     |
| 5:4        | FIR_CS_RAMCLKSEL  | FIR_CS_RAM clock selection bit:<br>00:CPUCLK<br>01:ASRCCLK<br><b>10: Reserved</b><br>11: AE clock           | R/W    | 0     |
| 3:2        | FIR_AA_RAMCLKSEL  | FIR_AA_RAM clock selection bit:<br>00:CPUCLK<br>01:ASRCCLK<br><b>10: Reserved</b><br>11:AECLK               | R/W    | 0     |
| 1:0        | FIR_MPX_RAMCLKSEL | FIR_MPX_RAM_LM clock selection bit:<br>00:CPUCLK<br>01:CPUCLK<br><b>10: Reserved</b><br>11:AECLK            | R/W    | 0     |

### 7.4.6.16 MEMCLKSELCTL1

**MEMCLKSELCTL1 (Memory Clock Selection Control Register 1, SFR: 0xae, SFR bank 0x01)**

| Bit Number | Bit Mnemonic       | Function   | Access | Reset |
|------------|--------------------|--|--------|-------|
| 7          | URAMCLKSEL         | URAM clock selection bit:<br>0: CPU clock;<br>1: USBctl_URAM_clk;                            | R/W    | 0     |
| 6:4        | Reserved           | Be read as 3 zeros.  | -      | -     |
| 3:2        | FIR_RDS_RAM CLKSEL | FIR_RDS_RAM clock selection bit:<br>00:CPUCLK<br>01:ASRCCLK<br>10: Reserved<br>11: URAMclk   | R/W    | 0     |
| 1:0        | PCMRAMCLK SEL      | PCMRAM clock selection bit:<br>00: CPU clock;<br>01: AE clock<br>10: Reserved<br>11: URAMclk | R/W    | 0     |

## 7.5 RTC (冯崧祥、黄俏)

| 日期         | 版本     | 描述   | 修订人 |
|------------|--------|--|-----|
| 2012-07-24 | V1. 00 | initial  | 冯崧祥 |
| 2012-09-12 | V1. 02 | 1. 修正 RTC_CTL0 和 RTC_CTL1 的 reserved 位为可使用<br>2. 修正 RTCTimeD 和 RTCTimeMon 的默认值为 01<br>3. 修改 RTCTimeMint 寄存器名为 RTCTimeMin<br>4. 增加 1 路 CTC，及三个相关寄存器 | 冯崧祥 |
| 2012-10-20 | V2. 00 | 1. 将 RTCTimeMon 修改为 bit3:0   | 冯崧祥 |
| 2012-11-07 | V2. 01 | 修正 RTCTimeY 的有效位为 bit[6:0]   | 冯崧祥 |
| 2012-12-12 | V2. 02 | 1. 将 RTC debug 信号 GPIOD7 修改为高较低的高频使能信号。<br>2. 补充完善 operation manual 章节<br>3. 在 RTC_CTL0 增加高频 4HZ 分频选择 bit。<br>4. 修改高频 4HZ 相关时钟图，operation manul。   | 冯崧祥 |
| 2013-01-09 | V2. 03 | 1. 将 operation manual 里高频 4HZ 切回高较低等待时间修改为 800ms。  | 冯崧祥 |

|            |        |                             |     |
|------------|--------|-----------------------------|-----|
| 2013-07-05 | V2. 04 | 在 operation manul 里增加上电使用说明 | 冯崧祥 |
|------------|--------|-----------------------------|-----|

## 7.5.1 Features

- Individual power supply :RTCVDD
- Built-in a 32k oscillator
- Internal or external oscillator optional
- Calendar with a alarm IRQ which can wake up the system
- 2Hz IRQ
- A Timer with IRQ
- A watch dog which can be configured optional as IRQ or Reset
- A CTC with IRQ
- An adjust circuit

## 7.5.2 Function Description

The RTC module has 5 separately units. Show in The following illustration (Figure 1). Some units are supplied by RTCVDD, others are supplied by VDD.

### Calendar with alarm IRQ:

This unit is supplied by RTCVDD. When cal\_en enable, the calendar start to count, the minimum unit is second.

The alarm is a compare which alarm daily. When the alm\_en is enable and “alarm time = calendar time”, an alm\_irq is sent out. One the way sends to CPU as interrupt signal, the other way sends to PMU as waking up signal.

The cal\_en bit must be disabled when The RTC Time register being written. And all RTC Time register must be written before cal\_en is enabled ,when set the time, Or some error will occur.

### 2hz:

2hz IRQ (2hz\_ip) will be set every 0.5 second if the 2hz\_en is enable . It will be cleared by writing “1” to 2hz\_ip.

### Timer:

Timer is a down counter with LOSC as clock. When the timer\_en enable and the counter is overflow, a IRQ (timer\_ip) will send out. The IRQ can be cleared by writing 1 to timer\_ip. The timer will reload the value and restart the counter when a counting cycle is finished.

### Watch dog:

When wd\_en=0, the unit do not work.

An internal reset (WDRST-) or an IRQ is generated when wd\_en =1 and the WD timer overflows. The WD timer overflows’ time is set by clk\_sel.

A reset or IRQ generated is selected by mode\_sel. 0: An IRQ is generated. 1: An internal reset is generated to force the system into reset status and then reboot.

Write 1 to CLR, will clear the WD timer. The CLR will be cleared automatically after the WD timer cleared

### CTC:

The CTC is a timer like timer. But its clock is HOSC.

### Wakeup timer

It will send a adjust signal to adjust circuit when overflows. The timer can be config by clk\_sel.

#### Cal\_clk division:

It generates the 4hz clock from the LOSC clock. The divisor is from adjust circuit and it reset value is 8194.

### 7.5.3 Module Description

#### 7.5.3.1 Block Diagram

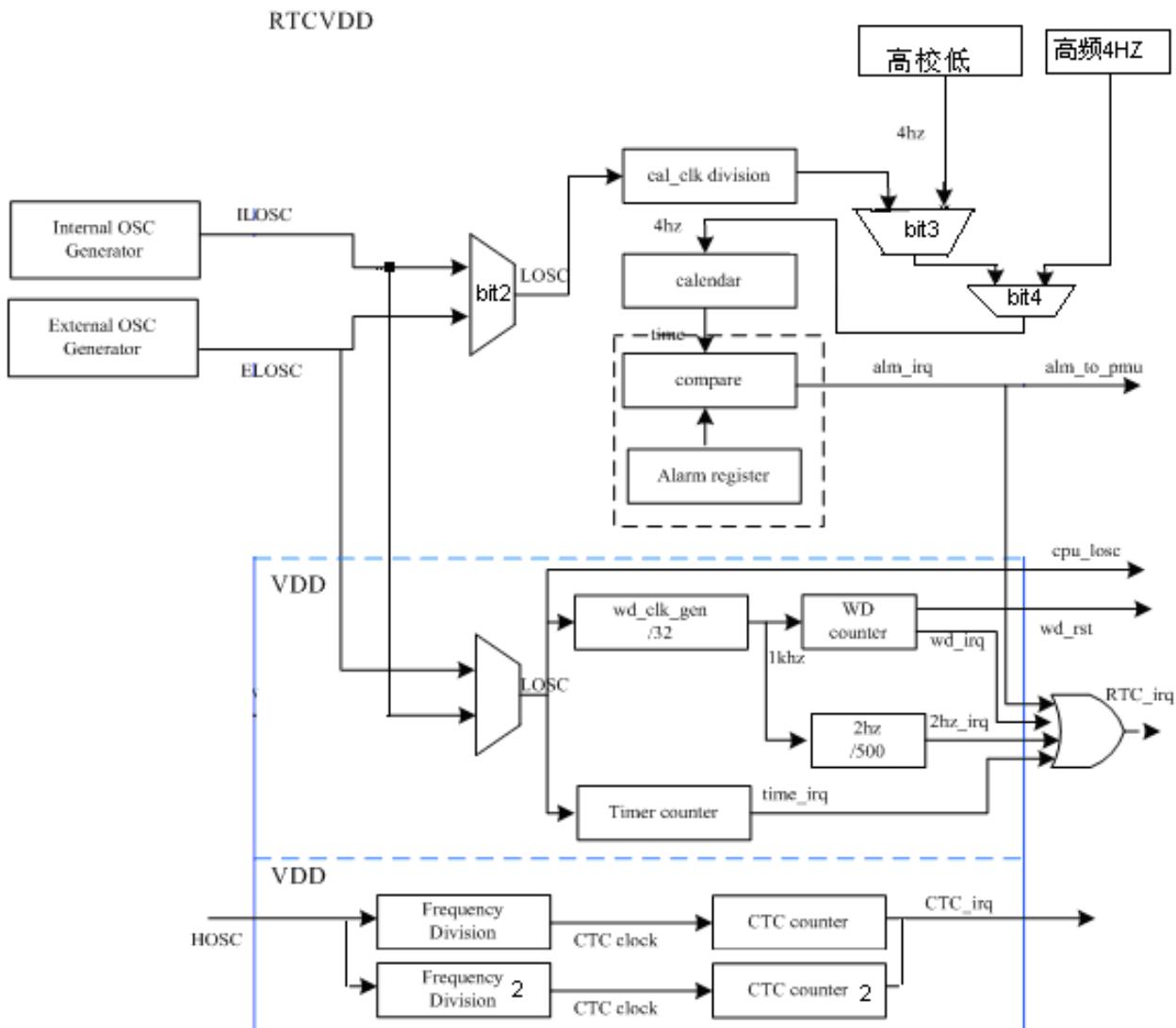


Figure 7-37 Block Diagram of RTC Controller

### 7.5.3.2 Signal List

| SIGNAL Name  | I/O Type | POWER  | Short Description  |
|--------------|----------|--------|--|
| scanmode     | INPUT    | VDD    | scan mode  |
| scanclk      | INPUT    | VDD    | scan clock   |
| hosc         | INPUT    | RTCVDD | High frequency crystal OSC input   |
| elosc        | INPUT    | RTCVDD | External low frequency crystal OSC input. Frequency: 32.768k                             |
| iłosc        | INPUT    | RTCVDD | Internal low frequency crystal OSC input. Frequency: 12k-40k.                            |
| coreclk      | INPUT    | VDD    | SFR clock  |
| pmu_losc     | INPUT    | VDD    | PMU clock. Frequency: 25k-40k.   |
| sfraddr      | INPUT    | VDD    | SFR address  |
| rtc_sfr_sel  | INPUT    | VDD    | /SFR access RTC register addr match  |
| sfrdatain    | INPUT    | VDD    | SFR data input   |
| sfrdataout   | INPUT    | VDD    | SFR data output  |
| sfroe        | INPUT    | VDD    | SFR read control. Active high.   |
| Sfrwe        | INPUT    | VDD    | SFR write enable control. Active high.   |
| Rtcvddok     | INPUT    | RTCVDD | RTCVDD POWER OK  |
| rst_n        | INPUT    | VDD    | System reset signal. Use for resetting the VDD registers (except wd_flag).               |
| Alm_to_pmu   | output   | RTCVDD | Alarm signal to PMU, level trigger   |
| wd_RST_n     | output   | VDD    | Watch dog reset signal to system   |
| ctc_irq      | output   | VDD    | CTC IRQ signal to INTC   |
| rtc_irq      | output   | VDD    | RTC IRQ signal to INTC   |
| powerok      | input    | VDD    | VCC OK signal. When the signal is invalid, RTVDD register is disconnected from VDD BUS . |
| łosc_mcu     | output   | VDD    | CPU'S LOSC   |
| łosc_for_pmu | output   | RTCVDD | łosc for pmu analog use  |
| hosc_cap_sel | I/O      | RTCVDD | HOSC PAD CAP SELECTION   |
| hosc_gmc     | I/O      | RTCVDD | High Frequency crystal Oscillator GMMIN select bits                                      |
| łosc_gmc     | I/O      | RTCVDD | Low Frequency crystal Oscillator GMMIN select bits                                       |
| łosc_en      | I/O      | RTCVDD | Low Frequency crystal Oscillator Enable  |

|                |     |        |   |
|----------------|-----|--------|---|
| cal_delay_time | I/O | RTCVDD | HOSC calibrate LOSC circuit wait for HOSC oscillator time |
| cal_interval   | I/O | RTCVDD | HOSC calibrate LOSC interval                              |
| cal_clk_sel    | I/O | RTCVDD | calendar clock select(2hz)                                |
| Hosc_cal_en    | I/O | RTCVDD | HOSC calibrate LOSC circuit enable                        |

Table 2-1 RTC Signal List

## 7.5.4 Operation Manual

### 7.5.4.1 更新 RTCVDD 寄存器函数

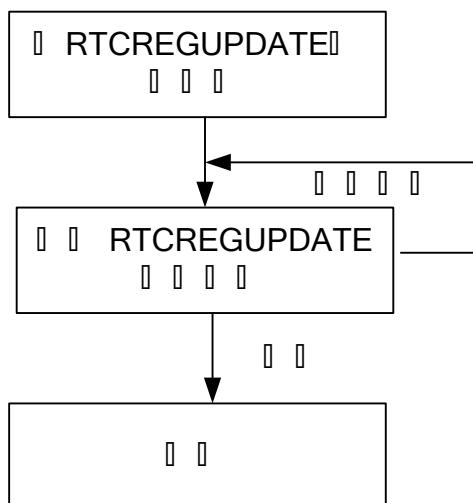


Figure 7-38 更新 RCVDD 寄存器函数

### 7.5.4.2 万年历时间设置

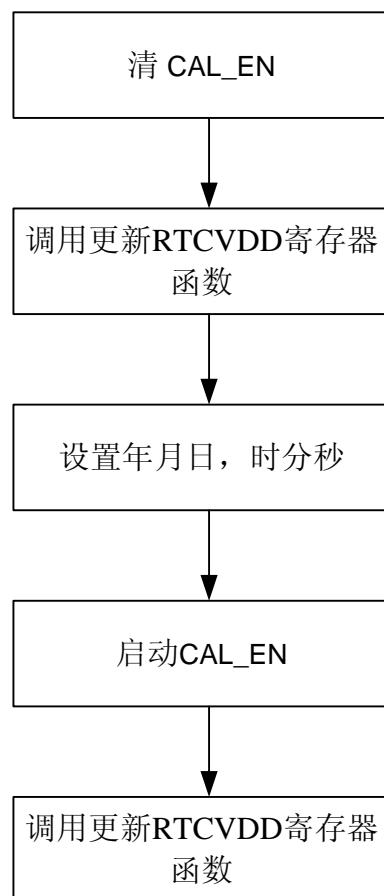


Figure 7-39 时间设置

### 7.5.4.3 ALARM 设置

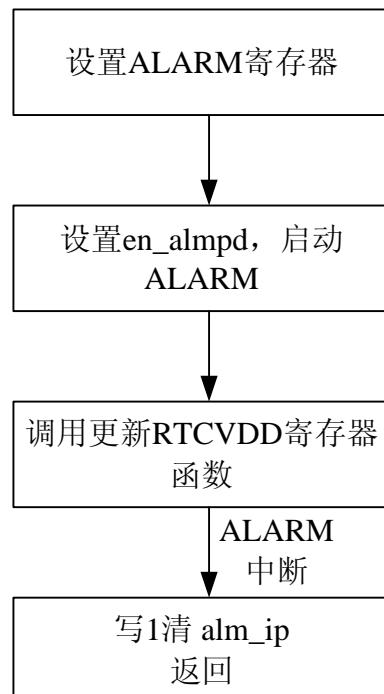


Figure 7-40 ALARM 设置

#### 7.5.4.4 Watch dog 设置

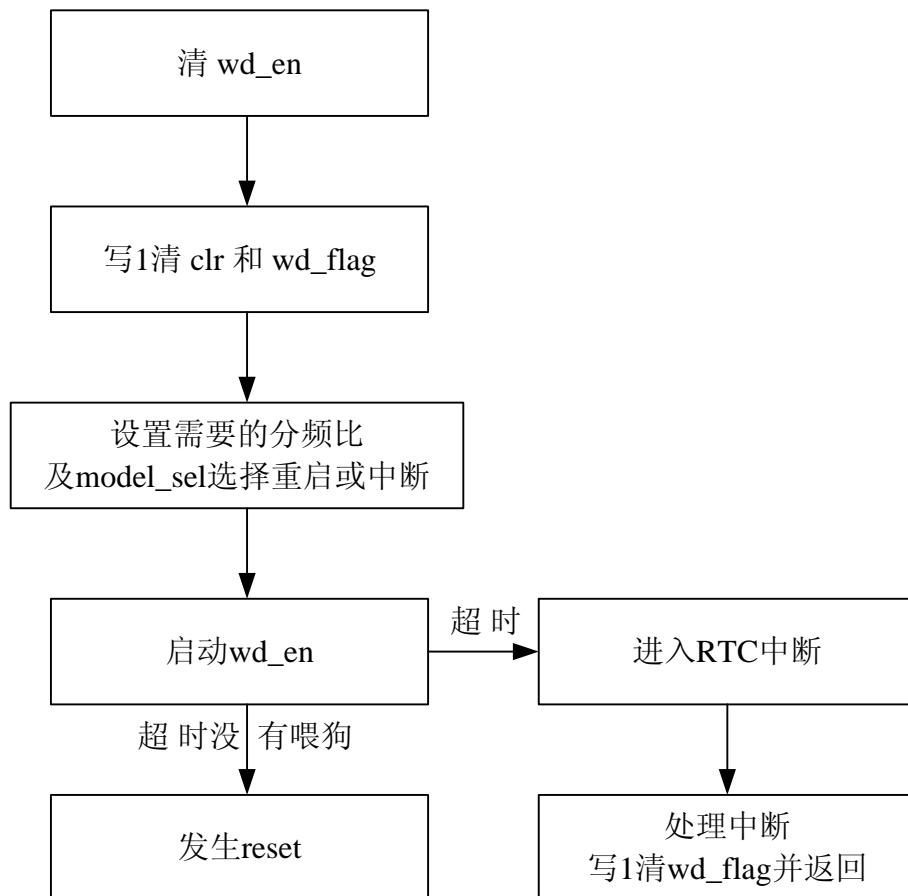


Figure 7-41 watch dog 设置流程

#### 7.5.4.5 Timer & CTC 设置

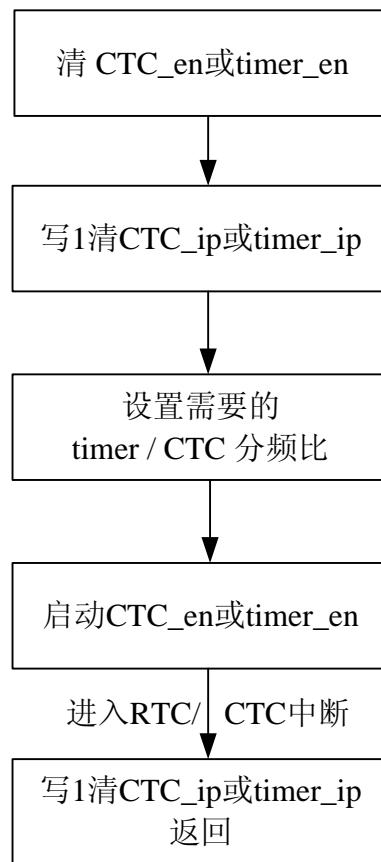
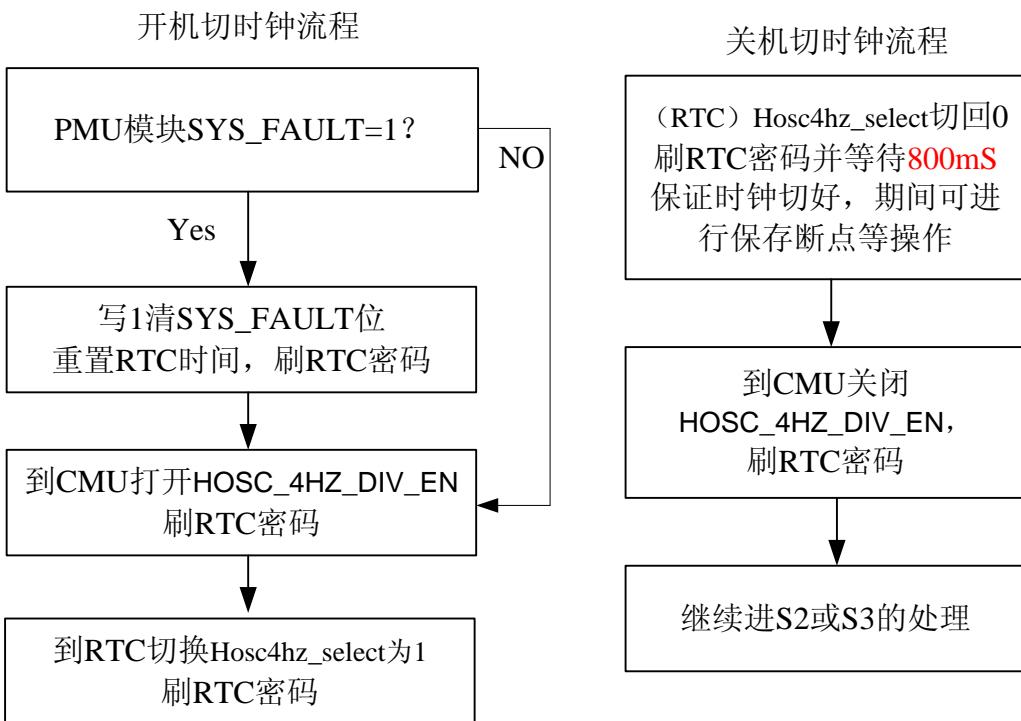


Figure 7-42 Timer&CTC 设置

#### 7.5.4.6 RTC 日历时钟说明

1. RTC\_CTL0 是 RTCVDD 域寄存器，控制 RTC 域低频时钟，包括 alarm 和万年历，PMU 用低频也会随之切换。
2. RTC\_CTL1 是 VDD 域寄存器，控制 VDD 域低频时钟，包括 timer, WD, 2HZ 等，MCU 用低频也会随之切换。
3. 日历时钟由 RTC\_CTL0 bit 4:2 这三位决定，上电默认选择高较低，软件 SDK 应在开机时，将时钟切到 HOSC4HZ，关机时切回高较低，进 S2, S3 都要切回高较低。流程如下：



注：系统上电后需等待至少 3 个低频周期再操作 PMU 或 RTC 的寄存器，digital 需要这么长的时间对这两个模块的寄存器进行同步，2~3 个周期后开放读写。

### 7.5.5 RTC Register List

| Index | Mnemonic   | Description                  | BANK |
|-------|------------|------------------------------|------|
| 0xa9  | RTC_CTL0   | RTC Control 0 Register       | 0x0c |
| 0xaa  | RTC_CTL1   | RTC Control 1 register       | 0x0c |
| 0xab  | RTCTimeS   | RTC Time Second Register     | 0x0c |
| 0xac  | RTCTimeMin | RTC Time Minute Register     | 0x0c |
| 0xad  | RTCTimeH   | RTC Time Hour Register       | 0x0c |
| 0xae  | RTCTimeD   | RTC Time Day Register        | 0x0c |
| 0xaf  | RTCTimeMon | RTC Time Month Register      | 0x0c |
| 0xb0  | RTCTimeY   | RTC Time Year Register       | 0x0c |
| 0xb1  | RTCALMS    | RTC Alarm Second Register    | 0x0c |
| 0xb2  | RTCALMM    | RTC Alarm Minute Register    | 0x0c |
| 0xb3  | RTCALMH    | RTC Alarm Hour Register      | 0x0c |
| 0xb4  | RTCRUPD    | RTC Register update Register | 0x0c |
| 0xb5  | TimerLB    | Timer low Byte               | 0x0c |
| 0xb6  | TimerMB    | Timer middle Byte            | 0x0c |
| 0xb7  | TimerHB    | Timer high Byte              | 0x0c |
| 0xb8  | WDCTL      | watch dog control register   | 0x0c |
| 0xb9  | CTCCTL     | CTC control register         | 0x0c |
| 0xba  | CTCCNTL    | CTC counter low register     | 0x0c |
| 0xbb  | CTCCNTH    | CTC counter high register    | 0x0c |

|      |            |                            |      |
|------|------------|----------------------------|------|
| 0xbc | RTC RD M0  | RTC Random access Register | 0x0c |
| 0xbd | RTC RD M1  | RTC Random access Register | 0x0c |
| 0xbf | RTC RD M2  | RTC Random access Register | 0x0c |
| 0xc1 | RTC RD M3  | RTC Random access Register | 0x0c |
| 0xc2 | RTC RD M4  | RTC Random access Register | 0x0c |
| 0xc3 | RTC RD M5  | RTC Random access Register | 0x0c |
| 0xc4 | RTC RD M6  | RTC Random access Register | 0x0c |
| 0xc5 | RTC RD M7  | RTC Random access Register | 0x0c |
| 0xc6 | RTC RD M8  | RTC Random access Register | 0x0c |
| 0xc7 | RTC RD M9  | RTC Random access Register | 0x0c |
| 0xc8 | RTC RD M10 | RTC Random access Register | 0x0c |
| 0xc9 | RTC RD M11 | RTC Random access Register | 0x0c |
| 0xca | RTC RD M12 | RTC Random access Register | 0x0c |
| 0xcb | RTC RD M13 | RTC Random access Register | 0x0c |
| 0xcc | RTC RD M14 | RTC Random access Register | 0x0c |
| 0xcd | RTC RD M15 | RTC Random access Register | 0x0c |
| 0xce | RTC RD M16 | RTC Random access Register | 0x0c |
| 0xcf | RTC RD M17 | RTC Random access Register | 0x0c |
| 0xd2 | RTC RD M18 | RTC Random access Register | 0x0c |
| 0xd3 | RTC RD M19 | RTC Random access Register | 0x0c |
| 0xd4 | RTC RD M20 | RTC Random access Register | 0x0c |
| 0xd5 | RTC RD M21 | RTC Random access Register | 0x0c |
| 0xd6 | RTC RD M22 | RTC Random access Register | 0x0c |
| 0xd7 | CTC CTL2   | CTC2 control register      | 0x0c |
| 0xd8 | CTC CNTL2  | CTC2 counter low register  | 0x0c |
| 0xd9 | CTC CNTH2  | CTC3 counter high register | 0x0c |

Table 2-2 RTC Register List

## NOTE:

The following Register marked by RTCVDD, means “The register’s power is supplied by RTCVDD. And the register is reset by RTCVDDOK.”

And that marked by VDD, means “The register’s power is supplied by VDD. And the register is reset by VDDOK and RST\_N.”

## 7.5.6 Register Description

### 7.5.6.1 RTC\_CTL0

RTC Control register

Offset = 0xa9

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|--------|------|-------------|-----|-------|

|   |                   |  |     |   |
|---|-------------------|--|-----|---|
| 7 | cal_en            | Calendar enable<br>0:disable<br>1:enable   | R/W | 0 |
| 6 | en_almpd          | Alarm pending enable.<br>0: disable. The alarm pending bit is disabled.<br>The pending bit is not set when alarm is occurred<br>1: enable. | R/W | 0 |
| 5 | test_en           | Test enable<br>0: disable<br>1: enable. The RTC's LOSC is changed to HOSC  | R/W | 0 |
| 4 | Hosc4hz_select    | Calendar clock select (优先级最高)<br>0: 由bit3决定<br>1: select 高频_division   | R/W | 0 |
| 3 | Cal4hz_clk_select | Calendar clock select (优先级其次)<br>0:select 高校低_low_division<br>1:select cal_clk_division (由bit2决定)<br>Not: bit1的是备用选择外部低频时使用, 正常都使用高校低来做时钟源 | R/W | 0 |
| 2 | cal_clk_select    | Calendar clock select (优先级最低)<br>0:select ILOSC<br>1:select ELOSC  | R/W | 0 |
| 1 | Leap_year         | RTC Leap Year bit<br>1: leap year<br>0: not leap year  | R   | 1 |
| 0 | alm_ip            | Alarm IRQ pending bit.<br>Writing 1 to this bit will clear it.   | R/W | 0 |

NOTE: The cal\_en bit must be disabled when The RTC Time register being written.

And all RTC Time register must be written before cal\_en is enabled when set the time, Or error will occur.

### 7.5.6.2 RTC\_CTL1

RTC Control register

Offset = 0xaa

| Bit(s) | Name     | Description                                    | R/W | Reset |
|--------|----------|--|-----|-------|
| 7      | 2hz_en   | 2hz IRQ enable<br>0:disable<br>1:enable        | R/W | 0     |
| 6      | timer_en | RTC Timer enable<br>0:disable<br>1:enable      | R/W | 0     |
| 5      | LOSC_sel | Cmu, watch dog, 2hz,timer 's clock select bit: | R/W | 0     |

|     |          |   |     |     |
|-----|----------|---|-----|-----|
|     |          | 0: ILOSC<br>1: ELOSC  |     |     |
| 4:2 | RESERVED | Reserved for future use                                       | R/W | 101 |
| 1   | 2hz_ip   | 2hz IRQ pending bit<br>Writing 1 to this bit will clear it.   | R/W | 0   |
| 0   | timer_ip | Timer IRQ pending bit<br>Writing 1 to this bit will clear it. | R/W | 0   |

### 7.5.6.3 RTCTimeS

RTC Time Second Register

Offset = 0xab

| Bit(s) | Name     | Description               | R/W | Reset |
|--------|----------|---------------------------|-----|-------|
| 7:6    | RESERVED | Reserved                  | R   | 0     |
| 5:0    | Time_sec | Calendar Time Second[5:0] | R/W | 0     |

### 7.5.6.4 RTCTimeMin

RTC Time Minute Register

Offset = 0xac

| Bit(s) | Name     | Description                | R/W | Reset |
|--------|----------|----------------------------|-----|-------|
| 7:6    | RESERVED | Reserved                   | R   | 0     |
| 5:0    | Time_min | Calendar Time Minute [5:0] | R/W | 0     |

### 7.5.6.5 RTCTimeH

RTC Time Hour Register

Offset = 0xad

| Bit(s) | Name      | Description              | R/W | Reset |
|--------|-----------|--------------------------|-----|-------|
| 7:5    | RESERVED  | Reserved                 | R   | 0     |
| 4:0    | Time_hour | Calendar Time Hour [4:0] | R/W | 0     |

### 7.5.6.6 RTCTimeD

RTC Time Hour Register

Offset = 0xae

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |          |                         |     |       |
|-----|----------|-------------------------|-----|-------|
| 7:5 | RESERVED | Reserved                | R   | 0     |
| 4:0 | Time_day | Calendar Time Day [4:0] | R/W | 00001 |

### 7.5.6.7 RTCTimeMon

RTC Time Month Register

Offset = 0xaf

| Bit(s) | Name     | Description             | R/W | Reset |
|--------|----------|-------------------------|-----|-------|
| 7:4    | RESERVED | Reserved                | R   | 0     |
| 3:0    | Time_mon | Calendar Time Mon [3:0] | R/W | 0001  |

### 7.5.6.8 RTCTimeY

RTC Time Year Register

Offset = 0xb0

| Bit(s) | Name      | Description              | R/W | Reset |
|--------|-----------|--------------------------|-----|-------|
| 7      | RESERVED  | Reserved                 | R   | 0     |
| 6:0    | Time_year | Calendar Time Year [6:0] | R/W | 0     |

### 7.5.6.9 RTCALMS

RTC Alarm Second Register

Offset = 0xb1

| Bit(s) | Name     | Description        | R/W | Reset |
|--------|----------|--------------------|-----|-------|
| 7:6    | RESERVED | Reserved           | R   | 0     |
| 5:0    | Alm_sec  | Alarm Second [5:0] | R/W | 0     |

### 7.5.6.10 RTCALMM

RTC Alarm Minute Register

Offset = 0xb2

| Bit(s) | Name     | Description        | R/W | Reset |
|--------|----------|--------------------|-----|-------|
| 7:6    | RESERVED | Reserved           | R   | 0     |
| 5:0    | Alm_min  | Alarm Minute [5:0] | R/W | 0     |

### 7.5.6.11 RTCALMH

RTC Alarm Hour Register

Offset = 0xb3

| Bit(s) | Name     | Description      | R/W | Reset |
|--------|----------|------------------|-----|-------|
| 7:5    | RESERVED | Reserved         | R   | 0     |
| 4:0    | Alm_hour | Alarm Hour [4:0] | R/W | 0     |

### 7. 5. 6. 12 RTCRUPD

RTC Register update control Register

Offset = 0xb4

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7:0    | update | <p>The RTCVDD register update control Register.</p> <p>When writing the RTC registers (except RTCREGUPDATE register or bit “alm_ip”), the RTC registers’ values are not update immediately. The value is written to backup registers(in VDD) first.</p> <p>Just when writing RTCREGUPDATE register “A5H”, the RTCVDD registers’ values are update with the backup registers’ value.</p> <p>RTCREGUPDATE register is automatically reset as “5AH” after the RTCVDD register is update.</p> <p>NOTE: Do not write RTCVDD registers when this register value is “A5H”</p> <p>NOTE: When writing the bit “alm_ip”, it will take effect immediately. Do not need writing this register.</p> | R/W | 0x5a  |

### 7. 5. 6. 13 TimerLB

Timer Low Byte

Offset = 0xb5

| Bit(s) | Name    | Description   | R/W | Reset |
|--------|---------|---|-----|-------|
| 7:0    | timerlb | <p>Low byte of timer Register</p> <p>Timer is a down counter with LOSC as clock.</p> <p>When the Counter Overflow, timer_ip will occur.</p> <p>Timer_ip = [1/(Time bit[23:0]+1)] *FLOSC</p> | R/W | X     |

### 7.5.6.14 TimerMB

Timer Middle Byte

Offset = 0xb6

| Bit(s) | Name    | Description                          | R/W | Reset |
|--------|---------|--------------------------------------|-----|-------|
| 7:0    | timerhb | Middle byte of LOSC Divider Register | R/W | X     |

### 7.5.6.15 TimerHB

Timer High Byte

Offset = 0xb7

| Bit(s) | Name    | Description                 | R/W | Reset |
|--------|---------|-----------------------------|-----|-------|
| 7:0    | timerhb | High byte of timer Register | R/W | X     |

### 7.5.6.16 WDCTL

Watch Dog Control Register

Offset = 0xb8

| Bit(s) | Name     | Description  | R/W | Reset |
|--------|----------|--|-----|-------|
| 7      | wd_en    | Watch Dog timer enable, when WD timer is enabled and the WD timer overflows, either an internal reset (WDRST-) is generated to force the system into reset status and then reboot, or a IRQ is sent to CPU.  | R/W | 0     |
| 6:4    | clk_sel  | Watch Dog timer clock select,<br>WDCKS Clock Selected Watch Dog Length<br>The watch dog's overflow value is 180.<br>000 1khz 176 ms<br>2029、 512hz 352 ms<br>010 256hz 703ms<br>011 128hz 1.4 s<br>100 64hz 2.8s<br>101 32hz 5.6 s<br>110 16hz 11.2s<br>111 8hz 22.5 s | R/W | 010   |
| 3      | clr      | Clear bit, write 1 to clear WD timer, cleared automatically  | W   | 0     |
| 2      | mode_sel | Watchdog IRQ or Reset- Select.   | R/W | 0     |

|   |          |  |     |   |
|---|----------|--|-----|---|
|   |          | 0: sent reset when Dog timer overflows<br>1:sent IRQ when Dog timer overflows  |     |   |
| 1 | wd_flag  | Watch dog overflow flag<br>1:means WD reset or irq ever occurred. Writing 1 to this bit clears it.<br>0:not occurred<br>This bit is reset by powerok signal. | R/W | 0 |
| 0 | Reserved | Reserved   | R   | 0 |

### 7.5.6.17 CTCCTL

CTC Control Register

Offset = 0xb9

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7      | ctc_en    | CTC enable<br>0: Disable, 1: Enable.   | R/W | 0     |
| 6:3    | pre_scale | Pre-scale register.<br>0000: the CTC clock is /1 of the HOSC.<br>0001: /2<br>0010: /4<br>0011: /8<br>0100: /16<br>0101: /32<br>0110: /64<br>0111: /128<br>1000: /256<br>1001: /512<br>Others are reserve | R/W | 0     |
| 2:1    | Reserved  | Reserved   | R   | 0     |
| 0      | ctc_ip    | CTC pending bit<br>Writing 1 to this bit will clear it.  | R/W | 0     |

### 7.5.6.18 CTCCNTL

CTC Counter Low Register

Offset = 0xba

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:0    | Counterl | CTC Counter[7:0]<br>CTC is a down counter using CTC clock. When | R/W | 0     |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | the counter is overflow , a CTC IRQ is sent to CPU.<br>Overflow time as enable CTC is:<br>(Counter[15:0]+1)/( CTC clock) |  |  |
|--|--|--|--|--|

### 7. 5. 6. 19 CTCCNTH

CTC Counter High Register

Offset = 0xbb

| Bit(s) | Name     | Description       | R/W | Reset |
|--------|----------|-------------------|-----|-------|
| 7:0    | counterh | CTC Counter[15:8] | R/W | 0     |

### 7. 5. 6. 20 RTCRDM0

RTC Random Access Register 0

Offset = 0xbc

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7. 5. 6. 21 RTCRDM1

RTC Random Access Register 1

Offset = 0xbd

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7. 5. 6. 22 RTCRDM2

RTC Random Access Register 2

Offset = 0xbf

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.23 RTCRDM3

RTC Random Access Register 3

Offset = 0xc1

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.24 RTCRDM4

RTC Random Access Register 4

Offset = 0xc2

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.25 RTCRDM5

RTC Random Access Register 5

Offset = 0xc3

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.26 RTCRDM6

RTC Random Access Register 6

Offset = 0xc4

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.27 RTCRDM7

RTC Random Access Register 7

Offset = 0xc5

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.28 RTCRDM8

RTC Random Access Register 8

Offset = 0xc6

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.29 RTCRDM9

RTC Random Access Register 9

Offset = 0xc7

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.30 RTCRDM10

RTC Random Access Register 10

Offset = 0xc8

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.31 RTCRDM11

RTC Random Access Register 11

Offset = 0xc9

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.32 RTCRDM12

RTC Random Access Register 12

Offset = 0xca

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.33 RTCRDM13

RTC Random Access Register 13

Offset = 0xcb

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.34 RTCRDM14

RTC Random Access Register 14

Offset = 0xcc

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.35 RTCRDM15

RTC Random Access Register 15

Offset = 0xcd

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.36 RTCRDM16

RTC Random Access Register 16

Offset = 0xce

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.37 RTCRDM17

RTC Random Access Register 17

Offset = 0xcf

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.38 RTCRDM18

RTC Random Access Register 18

Offset = 0xd2

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.39 RTCRDM19

RTC Random Access Register 19

Offset = 0xd3

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.40 RTCRDM20

RTC Random Access Register 20

Offset = 0xd4

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.41 RTCRDM21

RTC Random Access Register 21

Offset = 0xd5

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7.5.6.42 RTCRDM22

RTC Random Access Register 22

Offset = 0xd6

| Bit(s) | Name   | Description                               | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | Random | These bits can be accessed by CPU freely. | R/W | 0     |

### 7. 5. 6. 43 CTCCTL2

CTC2 Control Register

Offset = 0xd7

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7      | ctc_en2    | CTC2 enable<br>0: Disable, 1: Enable.   | R/W | 0     |
| 6:3    | pre_scale2 | Pre-scale register.<br><br>0000: the CTC2 clock is /1 of the HOSC.<br>0001: /2<br>0010: /4<br>0011: /8<br>0100: /16<br>0101: /32<br>0110: /64<br>0111: /128<br>1000: /256<br>1001: /512<br>Others are reserve | R/W | 0     |
| 2:1    | Reserved   | Reserved  | R   | 0     |
| 0      | ctc_ip2    | CTC2 pending bit<br><br>Writing 1 to this bit will clear it.  | R/W | 0     |

### 7. 5. 6. 44 CTCCNTL2

CTC2 Counter Low Register

Offset = 0xd8

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | Counterl2 | CTC2 Counter[7:0]<br><br>CTC is a down counter using CTC clock. When the counter is overflow , a CTC IRQ is sent to CPU.<br><br>Overflow time as enable CTC is:<br>(Counter[15:0]+1)/( CTC clock) | R/W | 0     |

### 7. 5. 6. 45 CTCCNTH2

CTC2 Counter High Register

Offset = 0xd9

| Bit(s) | Name      | Description        | R/W | Reset |
|--------|-----------|--------------------|-----|-------|
| 7:0    | Counterh2 | CTC2 Counter[15:8] | R/W | 0     |

注：两个 CTC 相互独立，有各自的 pending bit 使用同一中断。

### 7.5.7 DEBUG SIGNAL

When “DBGSEL=0dh” and “DBGDOE=0ffh”, the output of GPIOD[7:0] are:

{hosc\_open (高较低的高频使能) , alm\_wk\_dbgo, rtc\_irq, ctc\_irq, losc\_test, wd\_clk, hz2\_clk, hz4\_clk\_dbgo};

## 7.6 Interrupt Controller (黃少彬、黃俏)

| 日期         | 版本     | 描述  | 修订人 |
|------------|--------|---|-----|
| 2012-07-24 | V1. 00 | initial   | 黃少彬 |
| 2012-08-07 | V1. 01 | 更改 register list 的描述  | 黃少彬 |
| 2012-10-20 | V2. 00 | 1. 寄存器 AIE bit4 EAUDIO 描述修改, AUDIO 中断由 DAC_IRQ、ADC_IRQ、SPDIF_IRQ 三个中断组成;<br>2. 删除 IIC 中断, 更替为 TK 中断 | 黃少彬 |

### 7.6.1 Features

The Interrupt Controller of GL5115 has following features:

- (3) The GL5115 has the similar interrupt sources as the Intel 8Xc251 relative to the Flip80251 specification.
- (4) The interrupt sources are handled the same as on the original 8Xc251, however the Flip80251 has a shorter interrupt latency period, and can distinguish shorter external interrupt pulses.
- (5) Fourteen of interrupts can be enabled or disable by the system designer
- (6) 4 level of interrupt priority

### 7.6.2 Function Description

The Flip80251, like other control-oriented computer architectures, employs a program interrupt method. This operation branches to a subroutine and performs some service in response to the interrupt. When the subroutine completes, execution resumes at the point where the interrupt occurred. Interrupts may occur as a result of internal Flip80251 activity (e.g., timer overflow) or at the initiation of electrical signals external to the microcontroller (e.g., serial port communication). In all cases, interrupt operation is programmed by the system designer, who determines priority of interrupt service relative to normal code execution and other interrupt service routines. Fourteen of the seventeen interrupts are enabled or disabled by the system designer and may be manipulated dynamically.

A typical interrupt event chain occurs as follows. An internal or external device initiates an interrupt-request signal. This signal, connected to an input pin and periodically sampled by the Flip80251 (see figure 2), latches the event into a flag buffer. The priority of the flag (see figure 3 and 4, Interrupt System Special Function Registers) is compared to the priority of other interrupts by the interrupt handler. A high priority causes the handler to set an interrupt flag. This signals the instruction execution unit to execute a context switch. This context switch breaks the current flow of instruction sequences. The execution unit completes the current instruction prior to a save of the program counter (PC) and reloads the PC with the start address of a software service routine. The software service routine executes assigned tasks and as a final activity performs a RETI (return from interrupt) instruction. This instruction signals completion of the interrupt, resets the interrupt-in-progress priority, and reloads the program counter. Program operation then continues from the original point of interruption.

## 7.6.3 Module Description

### 7.6.3.1 Block Diagram

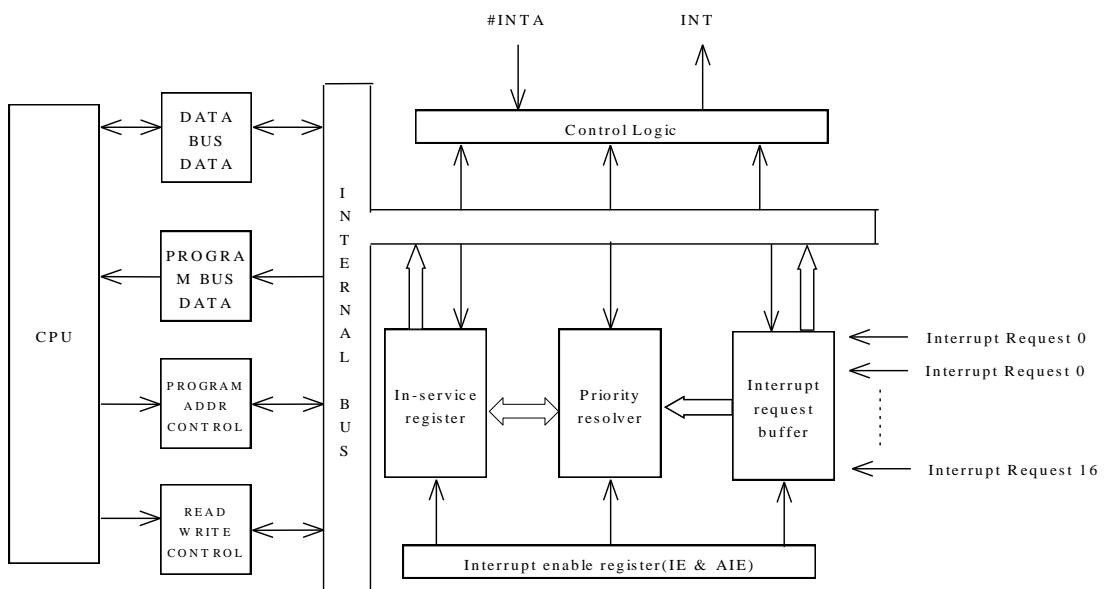


Figure 7-43 Block Diagram of Interrupt Controller

The mcu handles interrupts in four main phases:

- (1) Receive the interrupt request. Software or hardware requests a suspension of the current program sequence.
- (2) Acknowledge the interrupt request. The CPU must acknowledge the request. If the interrupt is maskable, certain conditions must be met for acknowledgment. For nonmaskable interrupts, acknowledgment is immediate.
- (3) Prepare for the interrupt service routine. The main tasks performed by the CPU are:
  - ① Complete execution of the current instruction and flush from the pipeline any instructions that have not reached the decode phase.
  - ② Automatically store certain register values to the data stack and the system stack.
  - ③ Fetch the interrupt vector that you store at a preset vector address. The interrupt vector points to the interrupt

service routine.

2030、 Execute the interrupt service routine. The CPU executes the ISR that you have written. The ISR is concluded with a return-from-interrupt instruction, which automatically restores the register values that were automatically saved.

### 7.6.3.2 Signal List

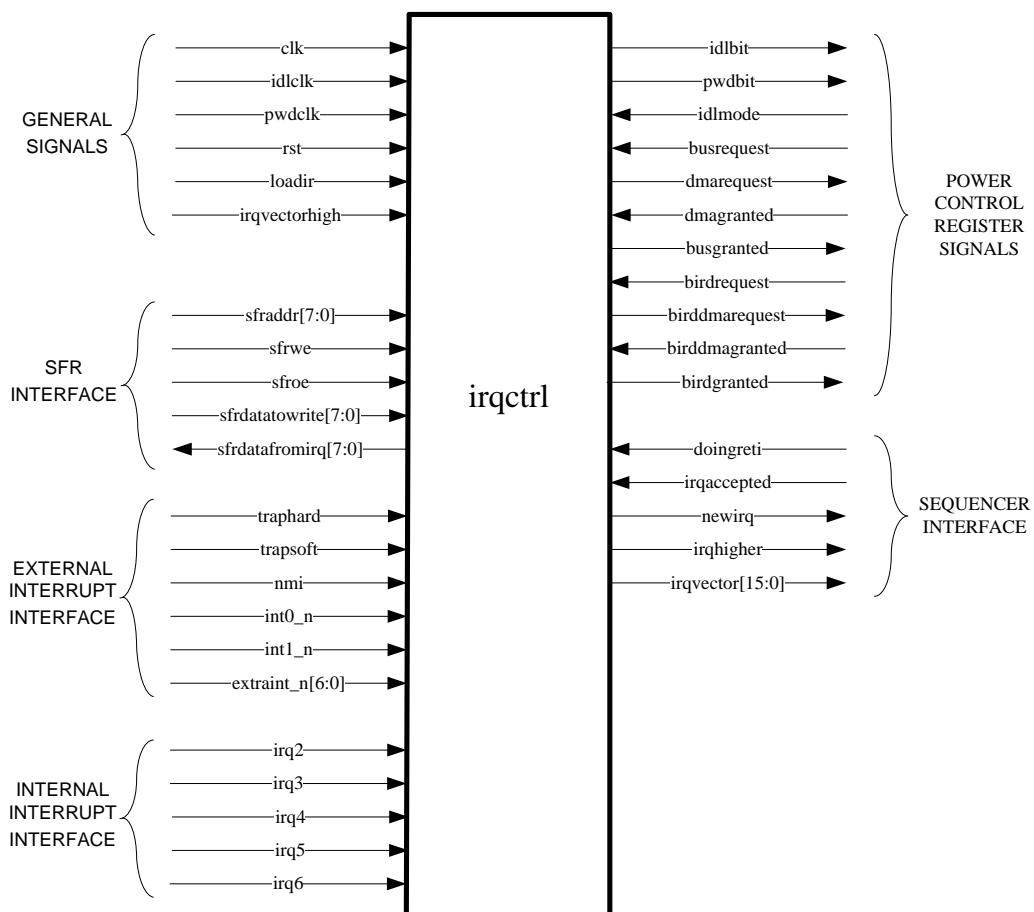


Figure 7-44 internal and external signal interrupt interface to the Interrupt Controller  
 the following signal are related with the interrupt source:

| EXTERNAL INTERRUPT INTERFACE |   |   |                       |
|------------------------------|---|---|-----------------------|
| traphard                     | 1 | I | hardware breakpoint   |
| trapsoft                     | 1 | I | software trap request |
| nmi                          | 1 | I | nmi input             |
| int0_n                       | 1 | I | int0_n input          |
| int1_n                       | 1 | I | int1_n input*         |
| extaint_n[6:0]               | 7 | I | extaint_n input       |
| INTERNAL INTERRUPT INTERFACE |   |   |                       |
| Irq2                         | 1 | I | CTC irq               |
| Irq3                         | 1 | I | RTC irq               |

|      |   |   |              |
|------|---|---|--------------|
| Irq4 | 1 | I | UART irq     |
| Irq5 | 1 | I | TK / IRC irq |
| Irq6 | 1 | I | SPI/SDC irq  |

Table 1 Interrupt Source

Notes:

\* Not used in normal mode

The clock to drive the interrupt controller is identical to the clock that drives the MCU core.

### 7.6.3.3 Intel 80251 to Flip80251 Migration

The original timer0 irq, timer1 irq, serial irq, timer2 irq, pca irq in the Intel 80251 and Flip80251 are no longer used. So the left interrupt source are distributed to the new IRQ.

GL5115 in Intel80251/Dolphin80251 in the so the current distribution of the above are new IRQ interrupt sources :

| <b>EXTERNAL INTERRUPT INTERFACE</b> |                        |                                 |                  |                          |
|-------------------------------------|------------------------|---------------------------------|------------------|--------------------------|
| IRQ Name                            | Intel/Dolphin8025<br>1 | Modified 80251 core<br>(GL5115) | Atj203x (GL5101) | Interrupt Vector Address |
| traphard                            | TRAP                   | TRAP                            | --               | IVT+7b*                  |
| trapsoft                            | TRAP                   | TRAP                            | --               | IVT+7b*                  |
| nmi                                 | nmi                    | nmi                             | --               | IVT+3b*                  |
| int0_n                              | int0_n                 | int0_n                          | Reg 27h 的 bit3   | IVT+03*                  |
| int1_n                              | int1_n                 | int1_n                          | Reg 27h 的 bit3   | IVT+13*                  |
| extraint_n[0]                       | --                     | USB irq                         | Reg 27h 的 bit2   | IVT+43*                  |
| extraint_n[1]                       | --                     | DMA2 irq                        | Reg 27h 的 bit4   | IVT+4b*                  |
| extraint_n[2]                       | --                     | DMA1 irq                        | Reg 27h 的 bit4   | IVT+53*                  |
| extraint_n[3]                       | --                     | DMA3 irq                        | Reg 27h 的 bit3   | IVT+5b*                  |
| extraint_n[4]                       | --                     | Audio and LRADC irq             | Reg 27h 的 bit7   | IVT+63*                  |
| extraint_n[5]                       | --                     | audio/video irq                 | Reg 27h 的 bit1   | IVT+6b*                  |
| extraint_n[6]                       | --                     | DMA0 irq                        | Reg 27h 的 bit4   | IVT+73*                  |
| <b>INTERNAL INTERRUPT INTERFACE</b> |                        |                                 |                  |                          |
| IRQ Name                            | Intel/Dolphin8025<br>1 | Modified 80251 core<br>(GL5115) | Atj203x (GL5101) | Interrupt Vector Address |
| Irq2                                | timer0 irq             | CTC irq                         | Reg 27h 的 bit4   | IVT+0b*                  |
| Irq3                                | timer1 irq             | RTC irq                         | Reg 27h 的 bit5   | IVT+1b*                  |
| Irq4                                | serial irq             | UART/IRC irq                    | Reg 27h 的 bit1   | IVT+23*                  |
| Irq5                                | timer2 irq             | TK / FM irq                     | Reg 27h 的 bit3   | IVT+2b*                  |
| Irq6                                | pca irq                | SPI/SDC irq                     | Reg 27h 的 bit3   | IVT+33*                  |

Table 2 Migration of Interrupt Source from other MCU to the Actions' 80251

\*The start address of interrupt table can be mapped to ff:0000h or ff:8000h by setting the bit2 of PCON register.

### 7.6.3.4 Interrupt Vector

The clock to drive the interrupt controller is identical to the clock that drives the MCU core.

| Interrupt number | Interrupt vector address | Description                            |
|------------------|--------------------------|--|
| -                | Ffc000h                  | reset                                  |
| 0                | IVT+03                   | int0_n interrupt                       |
| 1                | IVT+0b                   | CTC interrupt                          |
| 2                | IVT+13                   | Int1_n /DMA4 interrupt                 |
| 3                | IVT+1b                   | RTC interrupt                          |
| 4                | IVT+23                   | UART/ IRC interrupt                    |
| 5                | IVT+2b                   | TK interrupt                           |
| 6                | IVT+33                   | SPI/ SDC interrupt                     |
| 7                | IVT+3b                   | NMI interrupt(not used in normal mode) |
| 8                | IVT+43                   | USB interrupt                          |
| 9                | IVT+4b                   | DMA2 interrupt                         |
| 10               | IVT+53                   | DMA1 interrupt                         |
| 11               | IVT+5b                   | DMA3 interrupt                         |
| 12               | IVT+63                   | Audio interrupt                        |
| 13               | IVT+6b                   | audio/video codec interrupt            |
| 14               | IVT+73                   | DMA0 interrupt                         |
| 15               | IVT+7b                   | TRAP                                   |

Table 3 Interrupt Vector of GL5115

IVT = Off0000h, Offc000h

### 7.6.3.5 Interrupt Priorities

Each of the fourteen Actions' 80251 interrupt sources may be individually programmed to one of four priority levels. This is accomplished with the IPH0.x/IPL0.x bit pairs in the interrupt priority high (IPH0) and interrupt priority low (IPL0) registers. Specify the priority level as shown in table below using IPH0.x as the MSB and IPL0.x as the LSB.

| IPH0.x | (MSB) | IPL0.x             |
|--------|-------|--------------------|
| 0      | 0     | 0 Lowest Priority  |
| 0      | 1     | 1                  |
| 1      | 0     | 2                  |
| 1      | 1     | 3 Highest Priority |

Table 4 Level of Priority

A low-priority interrupt is always interrupted by a higher priority interrupt but not by another interrupt of equal or lower priority. The highest priority interrupt is not interrupted by any other interrupt source. Higher priority interrupts are serviced before lower priority interrupts. The response to simultaneous occurrence of equal priority interrupts (i.e., sampled within the same four state interrupt cycle) is determined by a hardware priority-within-level resolver, see below.

| Priority   | Interrupt number | Interrupt Name                         |
|------------|------------------|--|
| 0          | 16               | reset                                  |
| 1          | 15               | TRAP                                   |
| 2          | 7                | NMI interrupt(not used in normal mode) |
| 3          | 0                | int0_n                                 |
| 4          | 1                | CTC interrupt                          |
| 5          | 2                | Int1_n /DMA4 interrupt                 |
| 6          | 3                | RTC interrupt                          |
| 7          | 4                | UART/ IRC interrupt                    |
| 8          | 5                | TK interrupt                           |
| 9          | 6                | SPI/SDC interrupt interrupt            |
| 10         | 8                | USB interrupt                          |
| 11         | 9                | DMA2 interrupt                         |
| 12         | 10               | DMA1 interrupt                         |
| 13         | 11               | DMA3 interrupt                         |
| 14         | 12               | Audio interrupt                        |
| 15         | 13               | audio/video codec interrupt            |
| 16 highest | 14               | DMA0 interrupt                         |

Table 5 Interrupt Priority

Notes: The Actions' 80251 Interrupt Priority table differs from MCS® 51 microcontrollers. Other MCS 251 microcontrollers may have unique interrupt priority within level tables.

### 7.6.3.6 Interrupt processing

Interrupt processing is a dynamic operation that begins when a source requests an interrupt and lasts until the execution of the first instruction in the interrupt service routine. Response time is the amount of time between the interrupt request and the resulting break in the current instruction stream. Latency is the amount of time between the interrupt request and the execution of the first instruction in the interrupt service routine. These periods are dynamic due to the presence of both fixed-time sequences and several variable conditions. These conditions contribute to total elapsed time.

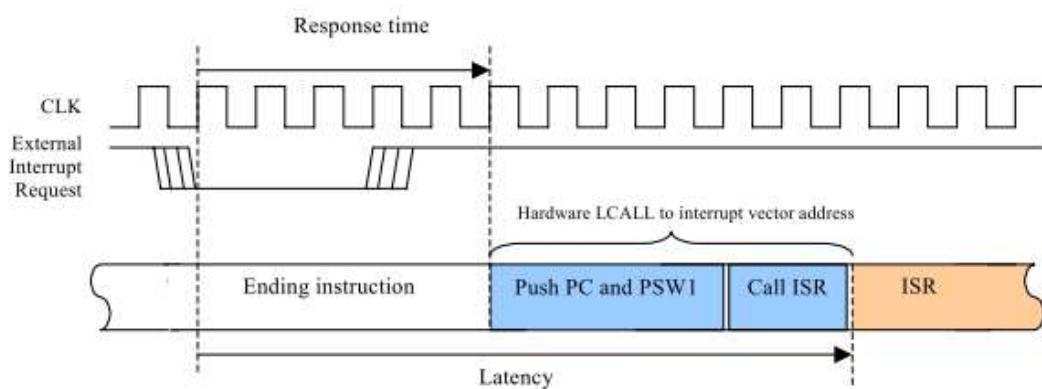


Figure 7-45 the Interrupt Process

Both response time and latency begin with the request. The subsequent minimum fixed sequence comprises the interrupt sample, poll, and request operations. The variables consist of (but are not limited to): specific

instructions in use at request time, internal versus external interrupt source requests, internal versus external program operation, stack location, presence of wait states, page mode operation, and branch pointer length.

NOTE:

In the following discussion, external interrupt request pins are assumed to be inactive for at least four state times prior to assertion. In this chapter all external hardware signals maintain some setup period (i.e., less than one state time). Signals must meet VIH and VIL specifications prior to any state time under discussion. This setup state time is not included in examples or calculations for either response or latency.

The PC (taken 3 bytes) and PSW1 will be push to the stack while intermode bit is set. The low 2 bytes' PC value will be push to the stack while intermode bit is cleared.

The relationship between the interrupt control register to the controller are shown below:

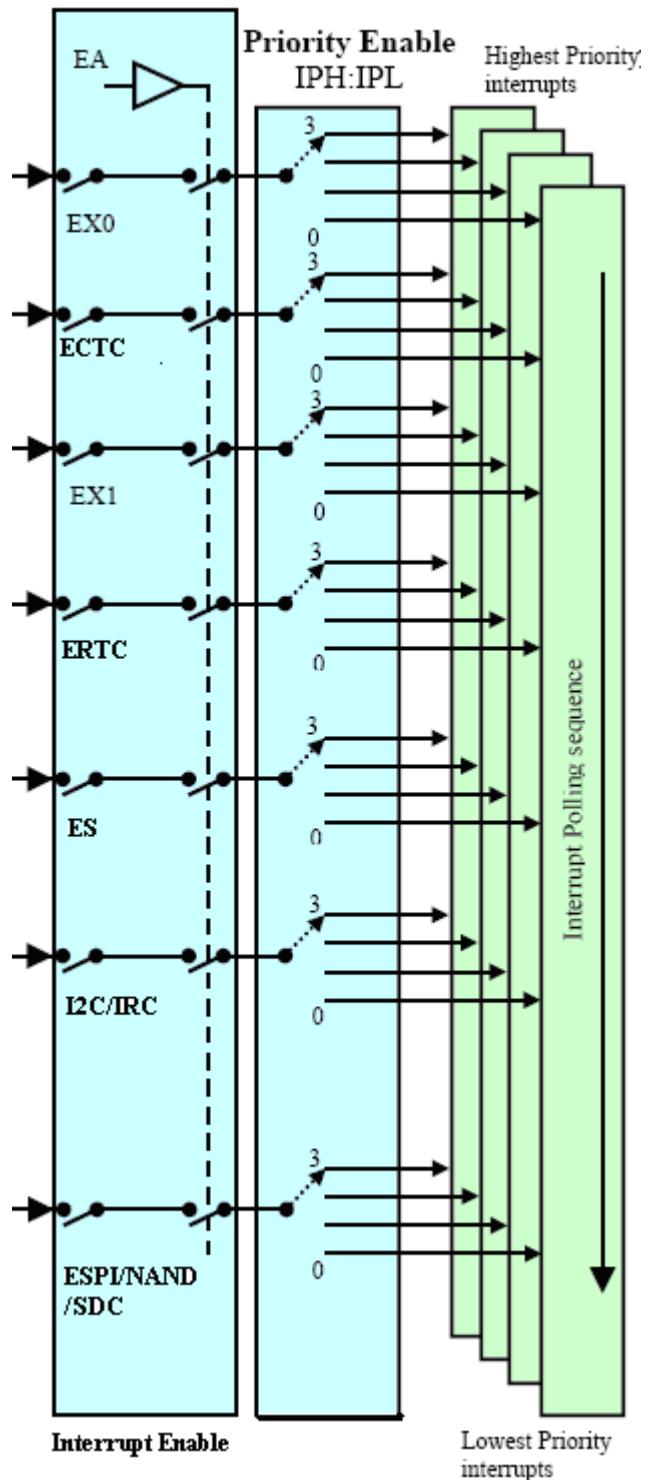


Figure 7-46 Interrupt control system(part1)

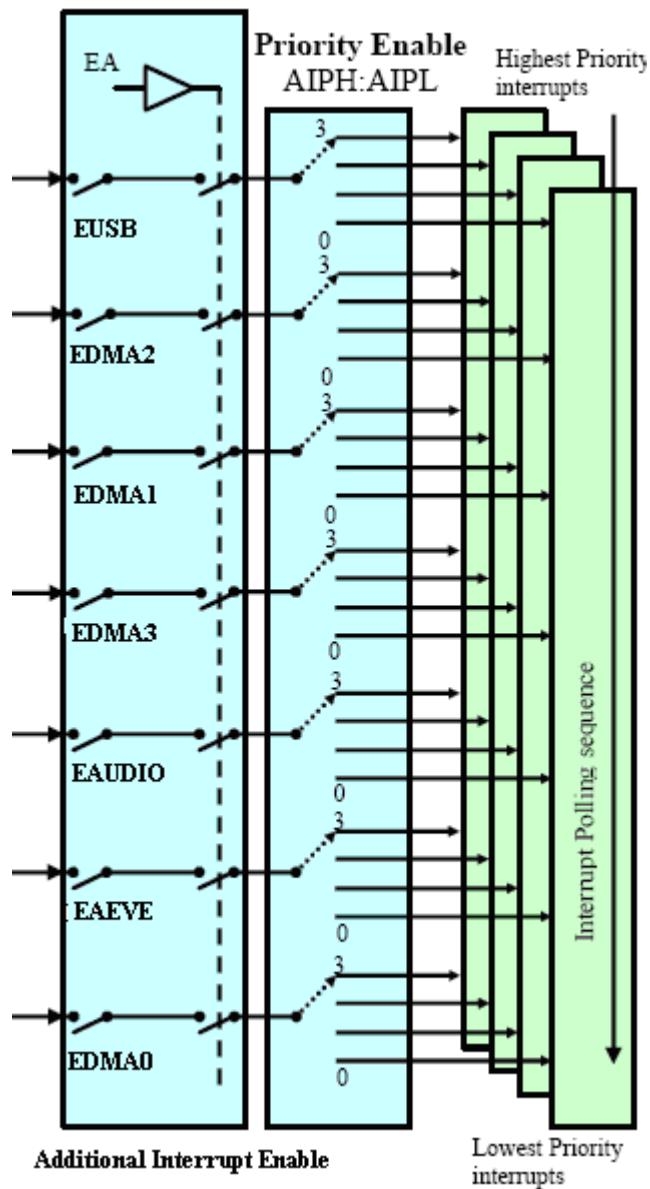


Figure 7-47 Interrupt control system (part2)

### 7.6.3.7 Interrupt Controller Clock

There are 3 running modes for the 80251, the normal mode, idle mode and power down mode. The Core clock , Idle clock and power down clock are controlled by the running modes of the 80251. The Core clock is active only in normal mode, otherwise it is in DC state. The Idle clock is active in normal mode and idle mode, otherwise it is in DC state. The power down clock is active in three state which is always the same as MCU clock.

Interrupt Controller uses Core Clock , IDLE clock and power down clock. Parts of clocks of the controller are closed for power-saving consideration. The clocks of the controller are showed below.

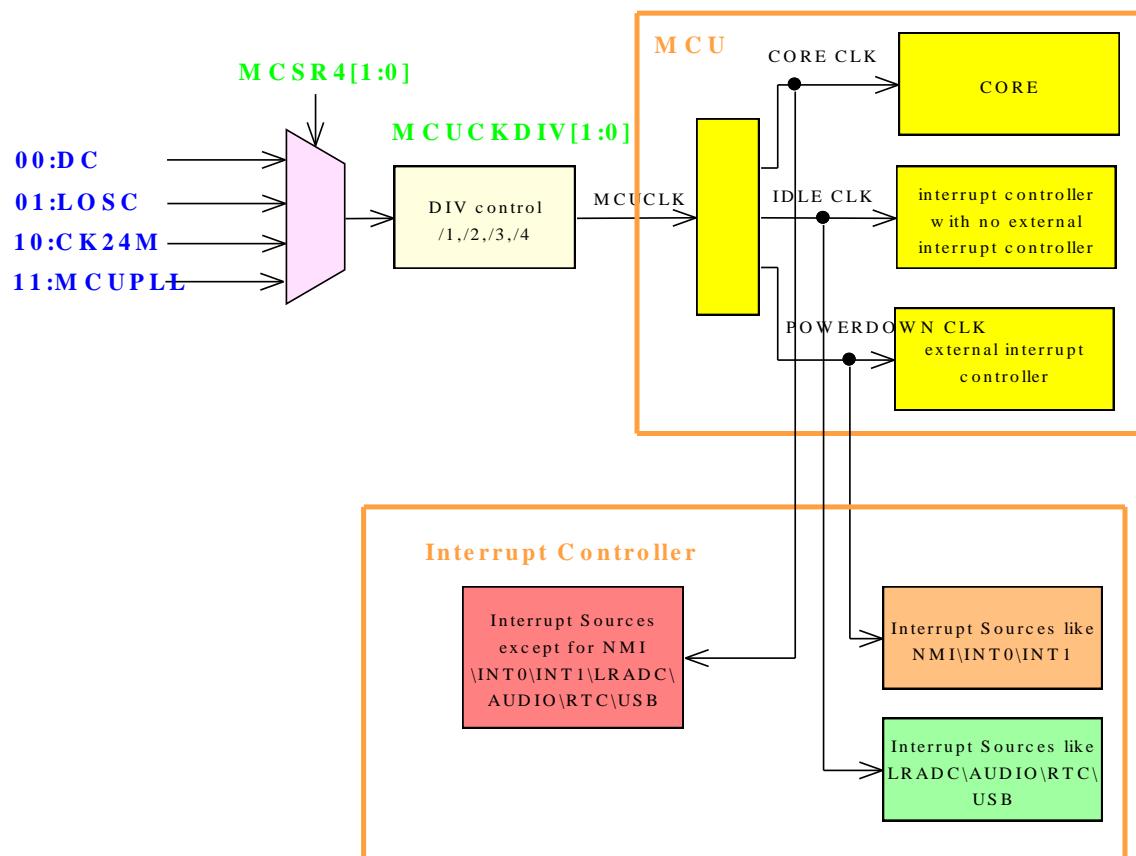


Figure 7-48 Interrupt controller clock

#### 7.6.4 Interrupt Control Register List

| Index       | Mnemonic                        | Description                                 | BANK     |
|-------------|---------------------------------|---|----------|
| <b>0Xa8</b> | <a href="#"><u>IE0</u></a>      | <b>Interrupt Enable register 0</b>          | All bank |
| <b>0Xe8</b> | <a href="#"><u>AIE</u></a>      | Additional Interrupt Enable register        | All bank |
| <b>0Xb7</b> | <a href="#"><u>IPH0</u></a>     | <b>Interrupt Priority High register 0</b>   | 0x00     |
| <b>0Xb8</b> | <a href="#"><u>IPL0</u></a>     | Interrupt Priority Low register 0           | 0x00     |
| <b>0Xf7</b> | <a href="#"><u>AIPH</u></a>     | Additional Interrupt Priority High register | 0x00     |
| <b>0Xf8</b> | <a href="#"><u>AIPL</u></a>     | Additional Interrupt Priority Low register  | 0x00     |
| <b>0x88</b> | <a href="#"><u>IF0</u></a>      | <b>Interrupt Flag register 0</b>            | All bank |
| <b>0Xc0</b> | <a href="#"><u>AIF</u></a>      | Additional interrupt flag register          | All bank |
| <b>0XD8</b> | <a href="#"><u>EXTINT</u></a>   | External Interrupt Control                  | 0x06     |
| <b>0Xf5</b> | <a href="#"><u>IFDebug0</u></a> | Interrupt Flag Debug register 0             | 0x01     |
| <b>0Xf6</b> | <a href="#"><u>IFDebug1</u></a> | Interrupt Flag Debug register 0             | 0x01     |

Interrupt Control Register

## 7.6.5 Register Description

### 7.6.5.1 IE0

**IE0 (Interrupt Enable Register, SFR Address 0Xa8, all bank)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | EA           | Global Interrupt Enable<br>0: disable all interrupts, except the TRAP and NMI interrupts, which are always enabled.<br>1: enable all interrupt that are individually enabled in IE0. | R/W    | 0     |
| 6          | ESPI_SDC     | SPI/SDC Interrupt Enable<br>0: disable SPI/SDC Interrupt<br>1: enable SPI/SDC Interrupt.   | R/W    | 0     |
| 5          | ETK          | TK Interrupt Enable<br>0: disable TK decoder Interrupt<br>1: enable TK decoder Interrupt.  | R/W    | 0     |
| 4          | UART_IRC     | UART/IRC Interrupt Enable<br>0: disable UART/IRC Interrupt<br>1: enable UART/IRC Interrupt.  | R/W    | 0     |
| 3          | ERTC         | ERTC Interrupt Enable*<br>0: disable ERTC Interrupt<br>1: Set to enable ERTC Interrupt.  | R/W    | 0     |
| 2          | EX1_DMA4     | External Interrupt 1/ DMA4 enable* **<br>0: disable External Interrupt 1/ DMA4.<br>1: enable External Interrupt 1/ DMA4.   | R/W    | 0     |
| 1          | ECTC         | ECTC Interrupt Enable<br>0: disable ECTC Interrupt<br>1: enable ECTC Interrupt.  | R/W    | 0     |
| 0          | EX0          | External Interrupt 0 enable* **<br>0: disable External Interrupt 0.<br>1: enable External Interrupt 0.   | R/W    | 0     |

### 7.6.5.2 AIE

**AIE(Additional Interrupt Enable register, SFR Address 0Xe8, all bank)**

| Bit Number | Bit Mnemonic | Function | Access | Reset |
|------------|--------------|----------|--------|-------|
| 7          | -            | -        | -      | -     |

|   |         |   |     |   |
|---|---------|---|-----|---|
| 6 | EDMA0   | DMA0 Interrupt Enable<br>0: disable DMA0 interrupt<br>1: enable DMA0 interrupt.   | R/W | 0 |
| 5 | E_CODEC | Audio/video codec Interrupt Enable<br>0 : disable audio/video codec Interrupt<br>1 : enable audio/video codec Interrupt.  | R/W | 0 |
| 4 | EAUDIO  | Audio Interrupt Enable*:3 IRQs are multiplexed together:<br>(1) SPDIF IRQ<br>(2) Audio ADC IRQ<br>(3) Audio DAC IRQ<br>0: disable the Interrupt<br>1: enable audio Interrupt. | R/W | 0 |
| 3 | EDMA3   | DMA3 Interrupt Enable<br>0: disable DMA3 interrupt<br>1: enable DMA3 interrupt.   | R/W | 0 |
| 2 | EDMA1   | DMA1 Interrupt Enable<br>0: disable DMA1 interrupt<br>1: enable DMA1 interrupt.   | R/W | 0 |
| 1 | EDMA2   | DMA2 Interrupt Enable<br>0: disable DMA2 interrupt<br>1: enable DMA2 interrupt.   | R/W | 0 |
| 0 | EUSB    | USB interrupt enable*<br>0: disable USB interrupt.<br>1: enable USB interrupt.  | R/W | 0 |

**Note:**

\* These interrupt pending can force MCU exit IDLE mode.

\*\* The external interrupt (EX0 and EX1 ) used to exit powerdown mode must be configured as level sensitive and must be assigned the highest priority. In addition, the duration of the interrupt must be of sufficient length to allow the oscillator to stabilize.

\*\*\* Only Reset , EX0 and EX1 pending can force MCU exit POWDOWN mode.

**Special attention:**

External interrupts INT0# and INT1# (INTx#) pins may each be programmed to be low level-triggered or edge-triggered, dependent upon bits IT0 and IT1 in the EXTINT register . If Itx = 0, INTx# is triggered by a detected low at the pin. If Itx = 1, INTx# is negative-edge triggered. External interrupts are enabled with bits EX0 and EX1 (Exx) in the IE0 register. Events on the external interrupt pins set the interrupt request flags Iex in EXTINT. These request bits are cleared by hardware vectors to service routines only if the interrupt is negative-edge triggered. If the interrupt is level-triggered, the interrupt service routine must clear the request bit. External hardware must de-assert INTx# before the service routine completes, or an additional interrupt is requested. External interrupt pins must be deasserted for at least four state times prior to a request.

External interrupt pins must be de-asserted for at least two clock cycles prior to a request. External interrupt inputs are sampled at each clock cycle. A level-triggered interrupt pin held low or high for any

two clock cycles time period guarantees detection. Edge-triggered external interrupts must hold the request pin low for at least two clock cycles. This ensures edge recognition and sets interrupt request bit Iex\_. The CPU clears Iex\_ automatically during service routine fetch cycles for edge-triggered interrupts.

**External interrupt inputs int0\_n and int1\_n provide both the capability to exit from Power-down mode on low-level signal while the interrupt priority bits of int0\_n and int1\_n are set to '1'.**

### 7.6.5.3 IPH0

**IPH0(Interrupt Priority High register 0, SFR Address 0xb7, bank 0x00)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7          | Reserved        | The value read from this bit is 0                              | -      | 0     |
| 6          | IPHSPI_SDC      | SPI/SDC interrupt priority level most significant bit          | R/W    | 0     |
| 5          | IPHTK           | TK Interrupt priority level most significant bit               | R/W    | 0     |
| 4          | IPHUART_I<br>RC | UART/IRC interrupt Priority level most significant bit         | R/W    | 0     |
| 3          | IPHRRTC         | RTC interrupt priority level most significant bit              | R/W    | 0     |
| 2          | IPHX1_DMA<br>4  | External interrupt 1/ DMA4 priority level most significant bit | R/W    | 0     |
| 1          | IPHCTC          | CTC interrupt priority level most significant bit              | R/W    | 0     |
| 0          | IPHX0           | External interrupt 0 priority level most significant bit       | R/W    | 0     |

### 7.6.5.4 IPL0

**IPL0(Interrupt Priority Low register 0, SFR Address 0xb8, bank 0x00)**

| Bit Number | Bit Mnemonic   | Function  | Access | Reset |
|------------|----------------|---|--------|-------|
| 7          | Reserved       | The value read from this bit is 0                               | -      | 0     |
| 6          | IPLSPI_SDC     | SPI/SDC interrupt priority level least significant bit          | R/W    | 0     |
| 5          | IPLTK_IRC      | TK/IRC Interrupt priority level least significant bit           | R/W    | 0     |
| 4          | IPLS           | UART interrupt Priority level least significant bit             | R/W    | 0     |
| 3          | IPLRTC         | RTC interrupt priority level least significant bit              | R/W    | 0     |
| 2          | IPLX1_DMA<br>4 | External interrupt 1/ DMA4 priority level least significant bit | R/W    | 0     |
| 1          | IPLCTC         | CTC interrupt priority level least significant bit              | R/W    | 0     |
| 0          | IPLX0          | External interrupt 0 priority level least                       | R/W    | 0     |

|  |                 |  |  |
|--|-----------------|--|--|
|  | significant bit |  |  |
|--|-----------------|--|--|

### 7.6.5.5 AIPH

**AIPH(Additional Interrupt Priority High register, SFR Address 0xf7, bank 0x00)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | -            | -   | R      | 0     |
| 6          | IPHDMA0      | DMA0 interrupt priority level most significant bit              | R/W    | 0     |
| 5          | IPH_CODEC    | AUDIO/VIDEO CODEC interrupt Priority level most significant bit | R/W    | 0     |
| 4          | IPHAUDIO     | Audio and LRADC interrupt priority level most significant bit   | R/W    | 0     |
| 3          | IPHDMA3      | DMA3 interrupt priority level most significant bit              | R/W    | 0     |
| 2          | IPHDMA1      | DMA1 interrupt priority level most significant bit              | R/W    | 0     |
| 1          | IPHDMA2      | DMA2 interrupt priority level most significant bit              | R/W    | 0     |
| 0          | IPHUSB       | USB interrupt priority level most significant bit               | R/W    | 0     |

### 7.6.5.6 AIPL

**AIPL(Additional Interrupt Priority Low register, SFR Address 0xf8, bank 0x00)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | -            | -  | R      | 0     |
| 6          | IPLDMA0      | DMA0 interrupt priority level least significant bit              | R/W    | 0     |
| 5          | IPL_CODEC    | AUDIO/VIDEO CODEC interrupt Priority level least significant bit | R/W    | 0     |
| 4          | IPLAUDIO     | Audio and LRADC interrupt priority level least significant bit   | R/W    | 0     |
| 3          | IPLDMA3      | DMA3 interrupt priority level least significant bit              | R/W    | 0     |
| 2          | IPLDMA1      | DMA1 interrupt priority level least significant bit              | R/W    | 0     |
| 1          | IPLDMA2      | DMA2 interrupt priority level least significant bit              | R/W    | 0     |
| 0          | IPLUSB       | USB interrupt priority level least significant bit               | R/W    | 0     |

### 7.6.5.7 IF0

**IF0(Interrupt Flag register 0, SFR Address 0x88, all bank)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | Reserved     | The value read from this bit is 0  | -      | 0     |
| 6          | FSPI_SDC     | This bit will automatically cleared only when All SPI/SDC interrupt pending bits are cleared, otherwise unchanged.                               | R      | 0     |
| 5          | FTK          | This bit will automatically cleared only when All TK interrupt pending bits are cleared, otherwise unchanged                                     | R      | 0     |
| 4          | UART_IRC     | This bit will automatically cleared only when All UART/IRC interrupt pending bits are cleared, otherwise unchanged                               | R      | 0     |
| 3          | FRTC         | This bit will automatically cleared only when all RTC interrupt pending bits are cleared, otherwise unchanged                                    | R      | 0     |
| 2          | FX1_DMA4     | This bit will automatically cleared only when all the pending bit of IE1_ (in 06 BANK, at sfr address 0xd8)/DMA4 is cleared, otherwise unchanged | R      | 0     |
| 1          | FCTC         | This bit will automatically cleared only when all CTC interrupt pending bits are cleared, otherwise unchanged                                    | R      | 0     |
| 0          | FX0          | This bit will automatically cleared only when all the pending bit of IE0_ (in 06 BANK, at sfr address 0xd8) is cleared, otherwise unchanged      | R      | 0     |

### 7.6.5.8 AIF

**AIF(Additional interrupt flag register, SFR Address 0xc0, all bank)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | -            | -  | R      | 0     |
| 6          | FDMA0        | This bit will automatically cleared only when all DMA0 interrupt pending bits are cleared, otherwise unchanged   | R      | 0     |
| 5          | FCODEC       | This bit will automatically cleared only when all CODEC interrupt pending bits are cleared, otherwise unchanged  | R      | 0     |
| 4          | FAUDIO       | This bit will automatically cleared only when all FAUDIO interrupt pending bits are cleared, otherwise unchanged | R      | 0     |

|   |       |  |   |   |
|---|-------|--|---|---|
|   |       | audio interrupt pending bits are cleared, otherwise unchanged  |   |   |
| 3 | FDMA3 | This bit will automatically cleared only when all DMA3 interrupt pending bits are cleared, otherwise unchanged | R | 0 |
| 2 | FDMA1 | This bit will automatically cleared only when all DMA1 interrupt pending bits are cleared, otherwise unchanged | R | 0 |
| 1 | FDMA2 | This bit will automatically cleared only when all DMA2 interrupt pending bits are cleared, otherwise unchanged | R | 0 |
| 0 | FUSB  | This bit will automatically cleared only when all USB interrupt pending bits are cleared, otherwise unchanged  | R | 0 |

### 7.6.5.9 EXTINT

**EXTINT(External Interrupt Control, SFR Address 0xd8, bank 0x06)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | Reserved     | Be read as '0000'  | --     | --    |
| 3          | IE1_         | External interrupt 1 edge flag. Hardware controlled<br><br>Set when external interrupt 1 is detected.<br>Cleared when interrupt is processed.<br>Write '1' will clear this bit.        | R/W    | 0     |
| 2          | IT1          | External interrupt 1 signal type control bit.<br><br>Set to specify External interrupt 1 as falling edge triggered.<br>Cleared to specify External interrupt 1 as low level triggered. | R/W    | 0     |
| 1          | IE0_         | External interrupt 0 edge flag. Hardware controlled<br><br>Set when external interrupt 0 is detected.<br>Cleared when interrupt is processed<br>Write '1' will clear this bit.         | R/W    | 0     |
| 0          | IT0          | External interrupt 0 signal type control bit.<br><br>Set to specify External interrupt 0 as falling edge triggered.<br>Cleared to specify External interrupt 0 as low level triggered. | R/W    | 0     |

### 7.6.5.10 IFDebug0

**IFDebug0(Interrupt Flag Debug register 0, SFR Address 0xf5, bank 0x01)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | IFDebugEn    | Interrupt Flag Debug Enable bit.<br>0: disable interrupt flag debug<br>1: enable interrupt flag debug | R/W    | 0     |
| 6          | FSPI_SDC     | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set  | R/W    | 0     |
| 5          | FTK          | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set  | R/W    | 0     |
| 4          | FUART_IRC    | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set  | R/W    | 0     |
| 3          | FRTC         | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set  | R/W    | 0     |
| 2          | FDMA4        | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set  | R/W    | 0     |
| 1          | FCTC         | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set  | R/W    | 0     |
| 0          | FX0          | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set  | R/W    | 0     |

### 7.6.5.11 IFDebug1

**IFDebug1(Interrupt Flag Debug register 1, SFR Address 0xf6, bank 0x01)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | NMI          | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set | R/W    | 0     |
| 6          | FDMA0        | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set | R/W    | 0     |
| 5          | FCODEC       | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set | R/W    | 0     |
| 4          | FAUDIO       | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set | R/W    | 0     |
| 3          | FDMA3        | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set | R/W    | 0     |
| 2          | FDMA1        | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set | R/W    | 0     |
| 1          | FDMA2        | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set | R/W    | 0     |

|   |      |  |     |   |
|---|------|--|-----|---|
| 0 | FUSB | 0: Interrupt Pending is cleared<br>1: Interrupt Pending is set | R/W | 0 |
|---|------|--|-----|---|

## 8 Co-Processor

### 8.1 DMA controller (黄少彬、黄俏)

| 日期         | 版本     | 描述  | 修订人 |
|------------|--------|---|-----|
| 2012-07-24 | V1. 00 | initial   | 黄少彬 |
| 2012-12-12 | V2. 02 | 1、完善 operation manual 章节<br>2、删除 DMAXSADDR1 和 DMAXDADDR1 寄存器及 Symbols and abbreviations 中相关描述 | 黄少彬 |

#### 8.1.1 Features

DMA controller can transfer data from RAM to RAM or from some peripheral to RAM or from RAM to some peripheral when the CPU is working.

The DMA controller has features as:

- (1) Operation that is independent of the CPU:
- (2) five channels
- (3) Event triggers

DMA transfers in each channel can be triggered on the occurrence of selected events. Such as:

Audio Codec Input DRQ

Audio Codec Output DRQ

SPI TX DRQ

SPI RX DRQ

UART TX DRQ

UART RX DRQ

USB DRQ

ADC DRQ

SD/MMC DRQ

DAC DRQ

SPDIF DRQ

2031、 interrupt

Each channel can send an interrupt to the CPU on completion of certain operational events. DMA0/1/2/3/4 has ten kind of interrupt flag, such as:

6 : DMA4HFIP

8 : DMA3HFIP

7 : DMA2HFIP

6 : DMA1HFIP  
5 : DMA0HFIP  
4 : DMA1TCIP  
3 : DMA3TCIP  
2 : DMA2TCIP  
1 : DMA1TCIP  
0 : DMA0TCIP

2032、 burst mode

burst8 mode is supported only.DMA controller check DRQ to make sure that whether burst8 should be 1.

## 8.1.2 Memory and Peripheral Access Description

### 8.1.2.1 Access memory

GL5115 can access Memory as Figure 8-4

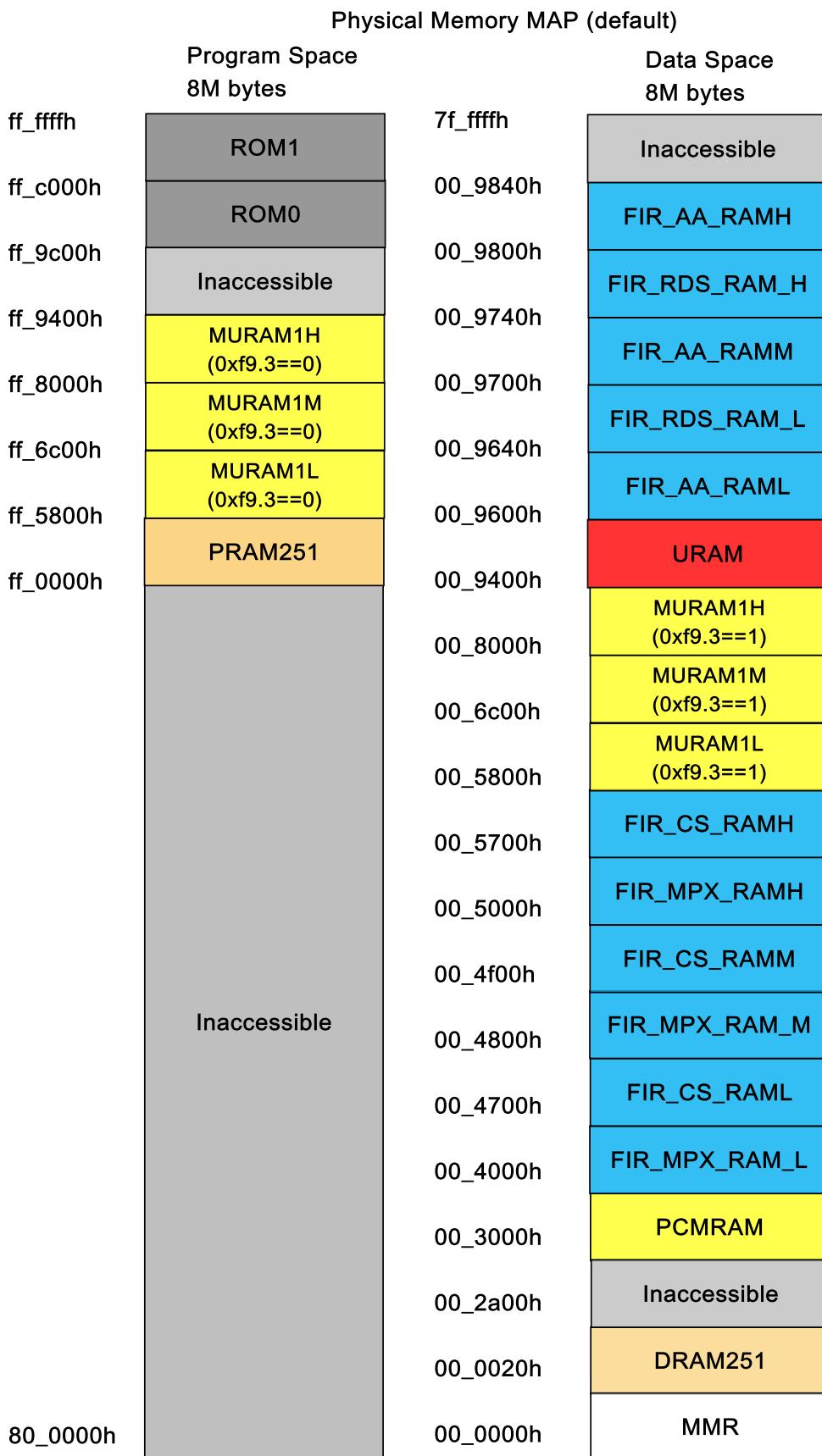


Figure 8-1 memory

DMA0/1/2/3/4有以下访问规则:

- (1) 配置DMA的起始地址均为虚拟地址。通过Memory Controller中的PMMT模块映射到来访问内部RAM均按照RAM的物理地址访问，并且只使用source/destination地址的低18bit。
- (2) 已经被分配给其它硬件模块的RAM（通过CMU中的MEMCLKCTL1寄存器判断时钟源是否为CPU clock），或者对应RAM的时钟根本没有打开（通过CMU中的MEMCLKCTL0寄存器判断时钟源是否disable），或者源/目的端口都是IO，以上4种错误情况都会导致DMA传输出错。
- (3) DMA访问外设的时候，只需要把外设的DRQ源头设置正确即可。比如把数据从RAM搬移到AudioIP的audio input FIFO, SRCTYPE 设置成memory, DSTTYPE 设置成AUIP FIFO就行了。
- (4) 如果DMA访问外设的FIFO，再通过CPU访问该外设挂在SFR总线上FIFO是无效的（即通过DMA控制器的传输控制位和DRQ的选择判断是通过DMA还是通过CPU访问对应外设的FIFO）。

### 8.1.2.2 Access Peripheral FIFO

下面是 DMA 能访问到的外设，以及外设支持的 FIFO 宽度：

| FIFO 类型                | FIFO 宽度 | FIFO 深度 |
|------------------------|---------|---------|
| Audio Codec Input FIFO | 8       | 16      |
| AUIP ENCOUNT           | 8       | 16      |
| SPI TX FIFO            | 8       | 16      |
| SPI RX FIFO            | 8       | 16      |
| UART TX FIFO           | 8       | 16      |
| UART RX FIFO           | 8       | 16      |
| USB FIFO               | 32      | 8       |
| ADC FIFO               | 24      | 16      |
| SD/MMC FIFO            | 32      | 8       |
| DAC FIFO               | 24      | 16      |
| SPDIF FIFO             | 24      | 16      |

GL5115 的外设 FIFO 都没有 DRQ 预判功能，DMA 写 FIFO 时每次当 FIFO 还能接收。

- (1) 在非 UDISK 模式下，USB/ CARD 的 FIFO 只能一个 cycle 传递 1 个字节，连续传输 8 次检查一次 DRQ。在 UDISK 模式下 USB 与 CARD 能一次传 4 个字节，连续传输 8 个 word 检查一次 DRQ。
- (2) 通过计算 ADC/DAC/ SPDIF FIFO 的次数计算传输次数的，每次读写一次 FIFO（即读写一个 sample）算传输一次，连续读写 8 次 DAC/ADC/ SPDIF FIFO 检查一次 DRQ。
- (3) 其他无特殊说明，每次传输都是 8bit 传递，数据在 memory 中都是 8bit 对齐存放，每次传输从 memory 中读/写一个 1 字节，每连续 8 次访问 8bit FIFO 后才检测 1 次 DRQ。

### 8.1.3 DMA examples

数据在 memory 中的存放方式并向外设 FIFO 通过 DMA 传输，有以下几种情况：

#### 8.1.3.1 Audio applications

DMA-DAC 和 DMA-ADC 传输是通过记录读写外设 FIFO 的次数计算传输次数的，DMAFrameLen 等于实际的传输次数，对于音频应用中，传输的次数一定要是偶数次（避免左右声道交换），也就是说音频应用中 DMAFrameLen 中的值一定要是偶数。

DMA 从 memory 中获取数据并向 DAC FIFO 传输数据与 DMA 从 ADC FIFO 获取数据并写入 memory 的过程类似，因此在这里只提供写 DAC FIFO 的例子。

(1) 24bit 双声道音频数据交替存放在 memory 向 DAC FIFO 写数

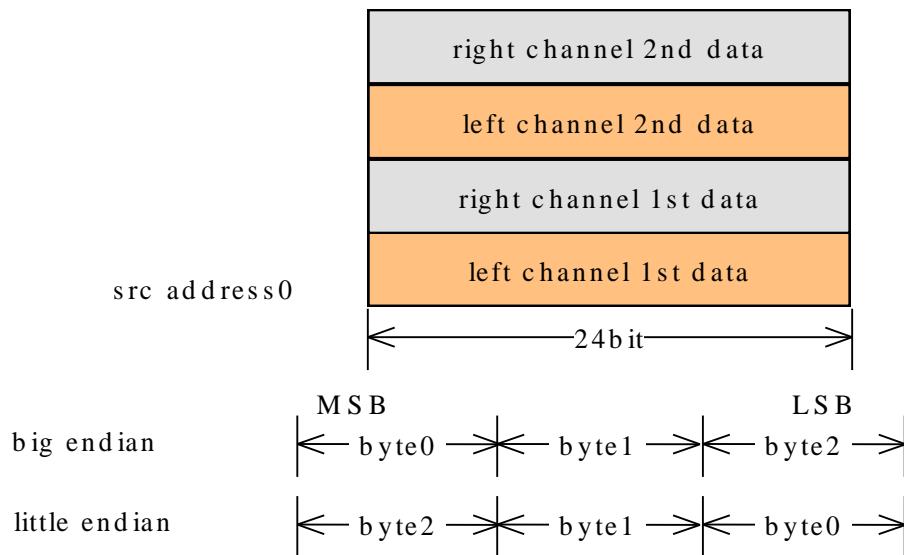


Figure 8-2 24bit 双声道音频数据在 memory 中的交替存放方式

数据相对于存放的起始点来说是先放左声通道再放右声通道数据。传输时先传左声道的第一个数据，再传右声道的第一个数据，依次类推。每个音频数据存放的位置只需要字节对齐，大模式或小模式存放。每个声道数据对应占用的字节位置为byte2~byte0(大模式时是LSB to MSB，小模式是MSB to LSB)。每1笔传输将一个声道的数据读出再写入24bit DAC FIFO，对应的frame counter加1。此时需要将DMAxCTL寄存器DMAxCTL的AUDIOTYPE寄存器设置成0(交替存放模式)，DMAxCTL的DATAWIDTH寄存器设置成 $2^b10$ (24bit精度模式)。DMAxCTL的SRCTYPE寄存器设置 $4'b0000$ (memory)，DMAxCTL的DSTTYPE寄存器设置成 $4'b0101$ (DAC FIFO)。还需要填DMAxSADDR0，写入基地址。

(2) 24bit 双声道音频数据分块存放在 memory 向 DAC FIFO 写数

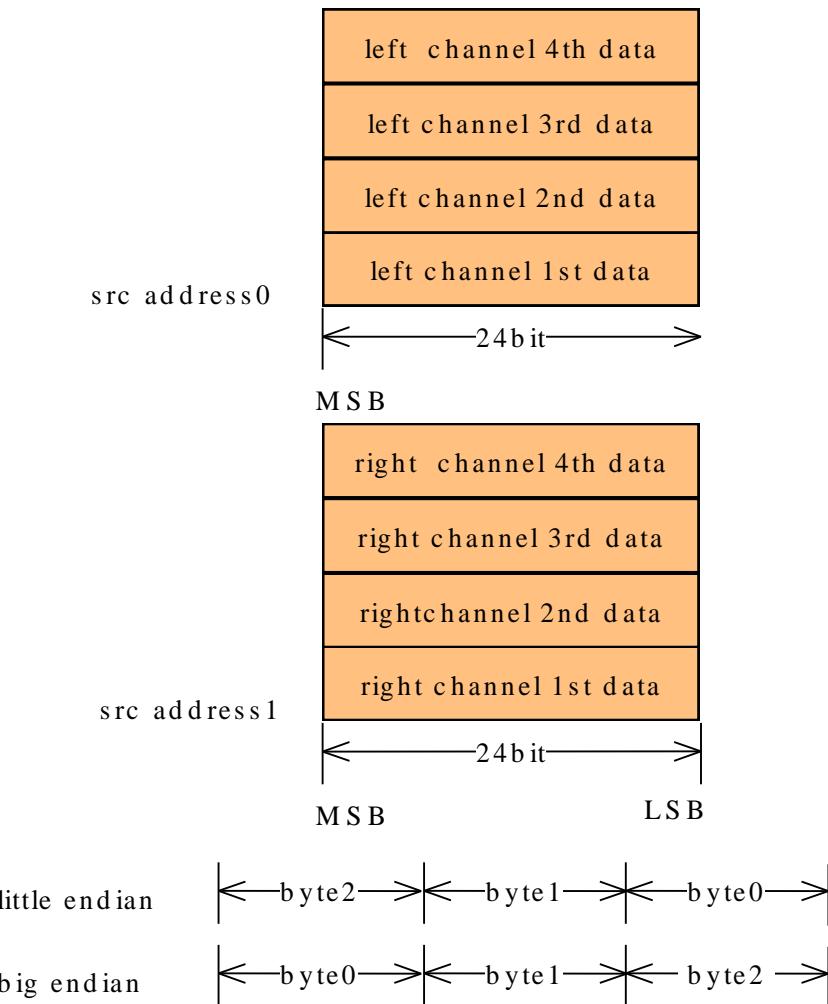


Figure 8-3 24bit 双声道音频数据在 memory 中的分块存放方式

数据相对于存放的起始点来说是先放左声通道再放右声道数据。传输时先传左声道的第一个数据，再传右声道的第一个数据，依次类推。每个声道数据对应占用的字节位置为byte2~byte0(大模式时是LSB to MSB，小模式是MSB to LSB)。每1笔传输将一个声道的数据读出再写入24bit DAC FIFO，对应的frame counter加1。此时需要将DMAxCTL寄存器的DMAxCTL的AUDIOTYPE寄存器设置成1 (分块存放模式)，DMAxCTL的DATAWIDTH寄存器设置成2'b10 (24bit精度模式)。DMAxCTL的SRCTYPE寄存器设置4'b0000 (memory)，DMAxCTL的DSTTYPE寄存器设置成4'b0101 (DAC FIFO)。还需要填DMAxSADDR0，写入左声道数据的基地址。填DMAxSADDR1，写入右声道数据的基地址。对于单声道传输可以将DMAxSADDR0的值设置成与DMAxSADDR1值相同

(3) 16bit 双声道音频数据交替存放在 memory 向 DAC FIFO 写数

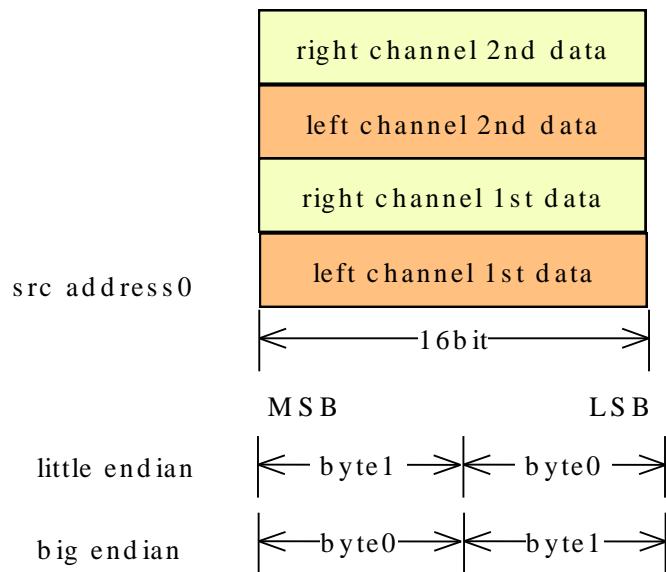


Figure 8-4 16bit 双声道音频数据在 memory 中的交替存放方式

数据相对于存放的起始点来说是先放左声通道再放右声通道数据。传输时先传左声道的第一个数据，再传右声道的第一个数据，依次类推。每个音频数据存放的位置一定是word对齐。每1笔传输将一个声道的数据读出再写入24bit DAC FIFO，对应的frame counter加1。写入DAC FIFO时需要将音频数据的高字节对齐，低8bit补0。此时需要将DMAxCTL寄存器的DMAxCTL的AUDIOTYPE寄存器设置成0（交替存放模式），DMAxCTL的DATAWIDTH寄存器设置成2'b01(16bit精度模式)。DMAxCTL的SRCTYPE寄存器设置4'b0000 (memory)，DMAxCTL的DSTTYPE寄存器设置成4'b0101 (DAC FIFO)。还需要填DMAxSADDR0，写入基地址。

#### (4) 16bit 双声道音频数据分块存放在 memory 向 DAC FIFO 写数

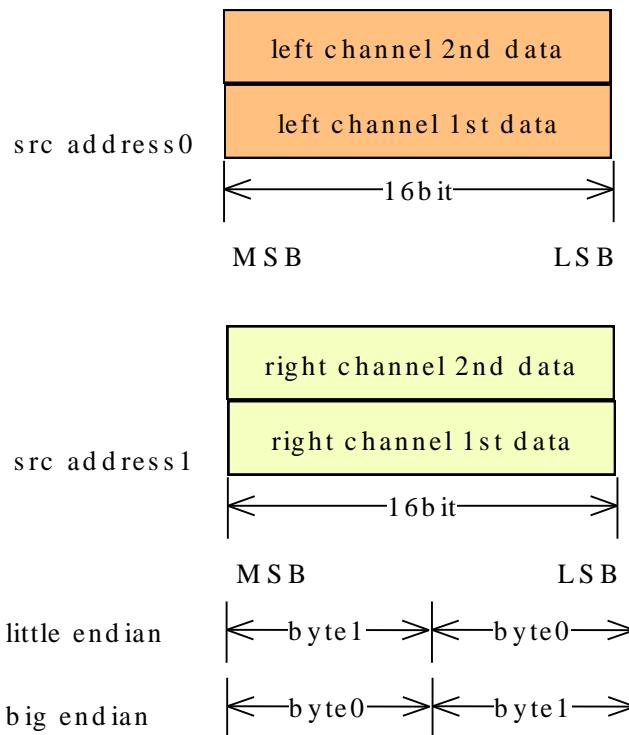


Figure 8-5 16bit 双声道音频数据在 memory 中的分块存放方式

数据相对于存放的起始点来说是先放左声通道再放右声通道数据。传输时先传左声道的第一个数据，再传右

声道的第一个数据，依次类推。每个音频数据存放的位置一定是16bit对齐。每1笔传输将一个声道的数据读出再写入24bit DAC FIFO，对应的frame counter加1。写入DAC FIFO时需要将音频数据的高字节对齐，低8bit补0。此时需要将DMAxCTL寄存器的DMAxCTL的AUDIOTYPE寄存器设置成1（分块存放模式），DMAxCTL的DATAWIDTH寄存器设置成2'b01(16bit精度模式)。DMAxCTL的SRCTYPE寄存器设置4'b0000 (memory)，DMAxCTL的DSTTYPE寄存器设置成4'b0101 (DAC FIFO)。还需要填DMAxSADDR0，写入左声道数据的基址。填DMAxSADDR1，写入右声道数据的基址。对于单声道传输可以将DMAxSADDR0的值设置成与DMAxSADDR1值相同。

(5) 8bit 双声道音频数据交替存放在 memory 向 DAC FIFO 写数

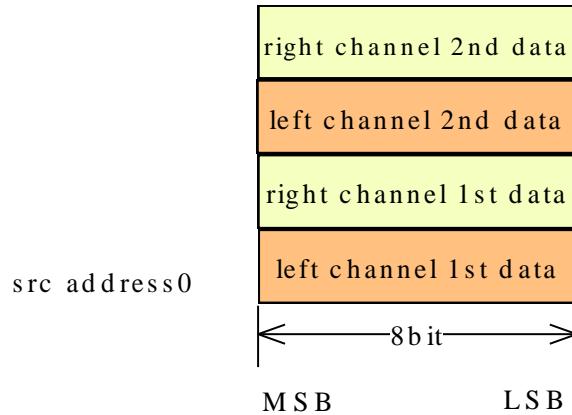


Figure 8-6 8bit 双声道音频数据在 memory 中的交替存放方式

数据相对于存放的起始点来说是先放左声通道再放右声通道数据(注意不是32bit、16bit对齐而是8bit对齐的)。传输时先传左声道的第一个数据，再传右声道的第一个数据，依次类推。每个音频数据存放的位置一定是byte对齐。每1笔传输将一个声道的数据读出再写入24bit DAC FIFO，对应的frame counter加1。写入DAC FIFO时需要将音频数据的高字节对齐，低16bit补0。此时需要将DMAxCTL寄存器的DMAxCTL的AUDIOTYPE寄存器设置成0 (交替存放模式)，DMAxCTL的DATAWIDTH寄存器设置成2'b00 (8bit精度模式)。DMAxCTL的SRCTYPE寄存器设置4'b0000 (memory)，DMAxCTL的DSTTYPE寄存器设置成4'b0101 (DAC FIFO)。还需要填DMAxSADDR0，写入基地址。

(6) 8bit 双声道音频数据分块存放在 memory 向 DAC FIFO 写数

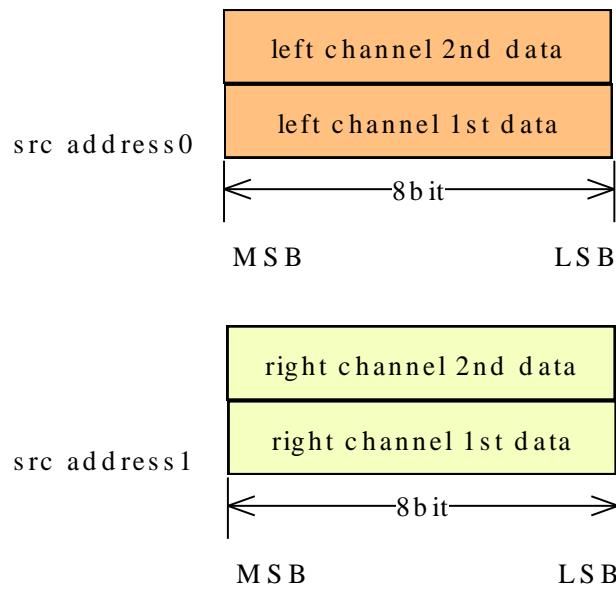


Figure 8-7 8bit 双声道音频数据在 memory 中的分块存放方式

数据相对于存放的起始点来说是先放左声通道再放右声通道数据(注意不是32bit、16bit对齐而是8bit对齐的)。

传输时先传左声道的第一个数据，再传右声道的第一个数据，依次类推。每个音频数据存放的位置一定是word对齐。每1笔传输将一个声道的数据读出再写入24bit DAC FIFO，对应的frame counter加1。写入DAC FIFO时需要将音频数据的高字节对齐，低16bit补0。此时需要将DMAxCTL寄存器的DMAxCTL的AUDIOTYPE寄存器设置成1（分块存放模式），DMAxCTL的DATAWIDTH寄存器设置成2'b00（8bit精度模式）。DMAxCTL的DSTTYPE寄存器设置成4'b0101（DAC FIFO）。还需要填DMAxSADDR0，写入左声道数据的基地址。填DMAxSADDR1，写入右声道数据的基地址。对于单声道传输可以将DMAxSADDR0的值设置成与DMAxSADDR1值相同。

### 8.1.3.2 UDISK applications

在 UDISK 模式下，USB 的 FIFO 可以与 CARD 控制器的 FIFO 直传输数据。位宽为 32 bit。传输长度在 UDISK-CARD 模式下必须是 512 字节的整数倍。

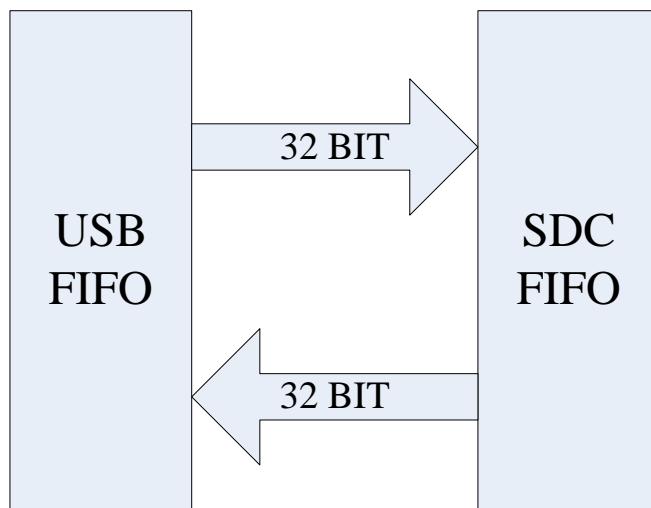


Figure 8-8 UDISK 模式下的传输

遇到以下情况，DMA 控制器的处理方式：

- (1) USB 中途传输错误，则 DMA 控制器依然接收 CARD/NAND 的数据，并将 USB 不能接收的数据 skip 掉
- (2) CARD/NAND 中途传输错误，则 DMA 控制器依然接收 USB 的数据。
- (3) 在非 UDISK 模式下，CARD 控制器的 FIFO 位宽变成 8bit

### 8.1.4 Priorities of DMA0/1/2/3/4

DMA0/1/2/3/4 共用一组写总线一组读总线。因此控制 5 个 DMA 不能同时传输数据，在同一时刻只有一个通道有效。DMA0/1/2/3/4 使用相同的中断源、中断使能控制位，可以作为一个整体来看待。

#### 8.1.4.1 Priority between DMA0/1/2/3/4 and CPU

在 GL5115 中与 DMA0/1/2/3/4 可能存在总线冲突的只有 CPU，这是因为 CPU 有可能与 DMA 抢同一块 memory。DMA 访问 CPU 的优先级始终高于 CPU 的数据访问和程序访问。当 DMA0/1/2/3/4 和 CPU 同时访问一块 memory 的时候，memory controller 将 CPU 的数据或程序总线 stall，直到 DMA0/1/2/3/4 不与

CPU 抢同一块 memory 为止。详细信息可以在 memory\_controller\_specification 中的 memory access rule 那节中看到。应用中不推荐 DMA 和 CPU 同时访问一块 RAM，这样会降低性能。

| Priority0<br>(最高) | Priority1 | Priority2 | Priority3 | Priority4 | Priority5 |
|-------------------|-----------|-----------|-----------|-----------|-----------|
| DMA0              | DMA1      | DMA2      | DMA3      | DMA4      | CPU       |

## 8.1.5 Symbols and abbreviations

### (1) block

这是DMA传输的最小单位，只要传输类型确定，每个block传输方式就确定。比如memory到memory传输中读1次源地址的数据，再写1次目的地址的数据算1个block传输，每次传4个字节。另外从memory到DAC FIFO的传输，写一次DAC FIFO算完成一次传输，每次从memory读取的字节数从1~3字节。

### (2) frame length

有两种定义方式：

- ① 从memory到DAC FIFO的传输，从ADC FIFO到memory的传输是按照写FIFO或读FIFO的次数来算，也就是说frame\_length = 传输的block数。
- ② 从memory到其它外设FIFO以及其它外设FIFO到memory，以及memory到memory的传输都是按照传输的字节数来计算的，也就是说frame\_length = 传输的字节数 = (传输的block数-1) \* 每block传输的字节数 + 最后一个block传输的字节数

### (3) data\_width

每个block传输的数据位宽，目前支持8bit, 16bit, 24bit和32bit位宽。不同位宽的传输对DMA的源/目的地址对齐有要求。比如8bit要求按照字节对齐，16bit要求按照双字节对齐，24/32bit要求按照4字节对齐。如果data\_width小于外设FIFO的宽度，数据一律按照高位传输，低位补0。

### (4) src\_addr

源地址，DMA具有3个源地址寄存器。

当从memory读取数据时，支持虚拟地址，DMA0SADDR0L/ DMA0SADDR0M/ DMA0SADDR0H三个寄存器有效。当DMA0SADDR0H为0x00或者0x7f（0xff）时，则可以认为进行实际物理地址传输

### (5) dst\_addr

目的地址，DMA具有3个目的地址寄存器。

当往memory写数据时，支持虚拟地址，DMA0DADDR0L/ DMA0DADDR0M/ DMA0DADDR0H三个寄存器有效。当DMA0DADDR1L为0x00或者0x7f（0xff0时，则可以认为进行实际物理地址传输

### (6) burst

从memory到peripheral FIFO或从peripheral FIFO到memory传输固定为burst 8，即每次burst开始前检测DRQ信号是否有效，如果有效则传输8次block。

## 8.1.6 DMA0/1/2/3/4 state machine

其状态转换图如下：

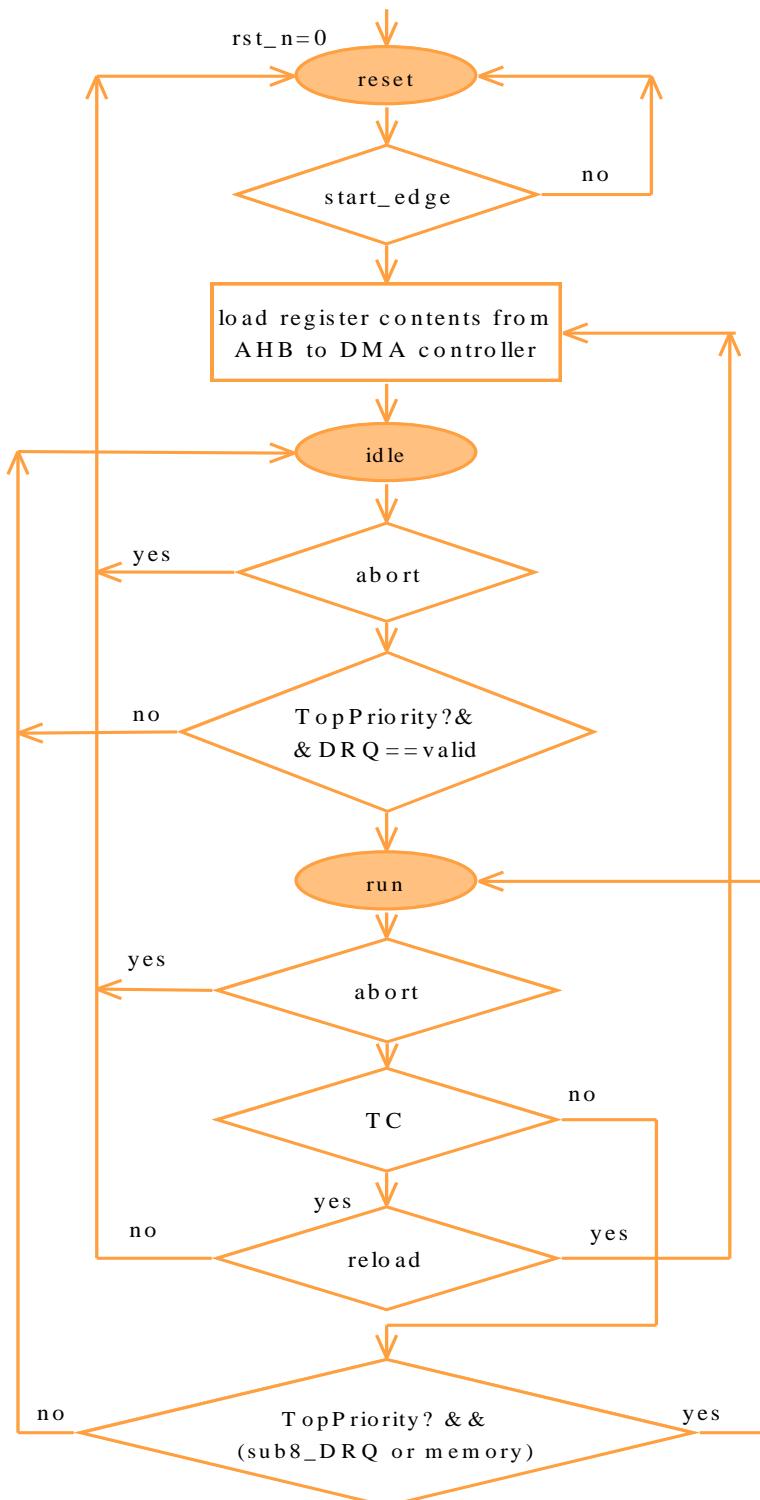


Figure 8-9 DMA0/1/2/3/4 transfer state machine

注:

- (1) 当传输是从 Memory 到 Memory 时，没有用到 DRQ，数据连续不停地传输（若干个 burst8 连续传）
- (2) 当传输是从 Memory 到外设或外设到 memory 时，每一笔 burst8 检测一次 DRQ 信号。决定下一笔 burst8 是否紧接着上一笔 burst8 传输。

## 8.1.7 DMA transfer type

### 8.1.7.1 Memory to Memory

(1) DMA 实现 memory 间的数据传输:

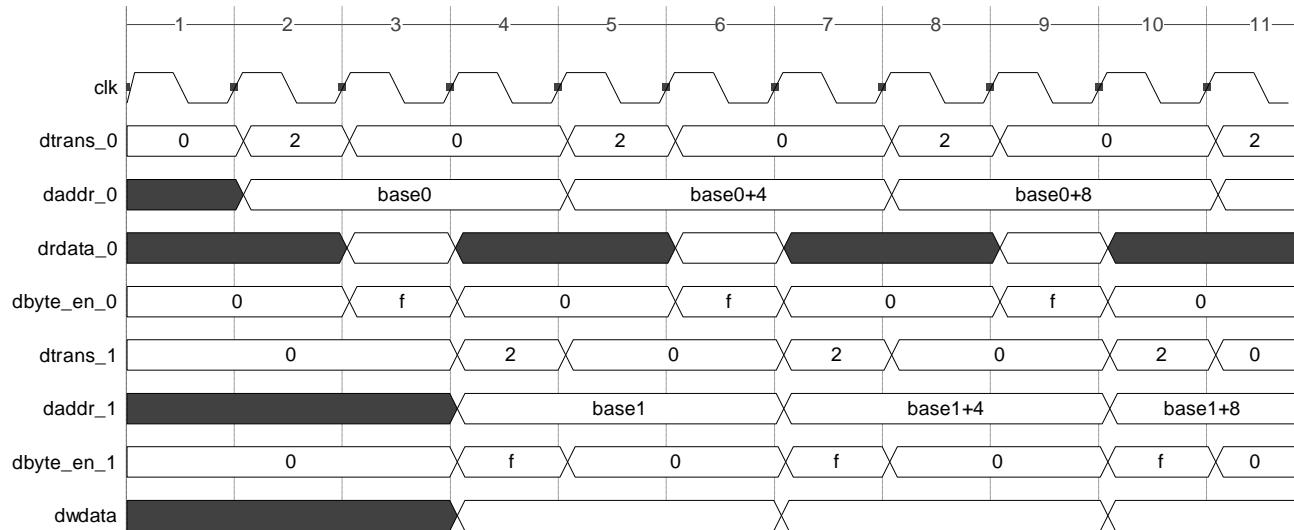


Figure 8-10 DMA access same memory block

### 8.1.7.2 Memory to Peripheral or Peripheral to Memory

有以下特点:

- (1) 支持的传输位宽为 8bit, 16bit, 24bit 和 32bit
- (2) 每次 burst8 检测 1 次 DRQ 信号, 每次 burst8 都有 1 次改变 DMA 优先级的机会。
- (3) 所有外设只支撑 Burst 8 传输 (这里的 Burst 是指读写外设 FIFO 的次数)。
- (4) dburst 用于告诉外设当前 busrt 传输的个数, 0 表示 burst1, 7 表示 burst8。例如该值为 4, 则表示下一次 burst 传输 5 个数据。

下面描述几种常见的 memory 与外设 FIFO 的传输情况:

(1) DMA 写 peripheral FIFO:

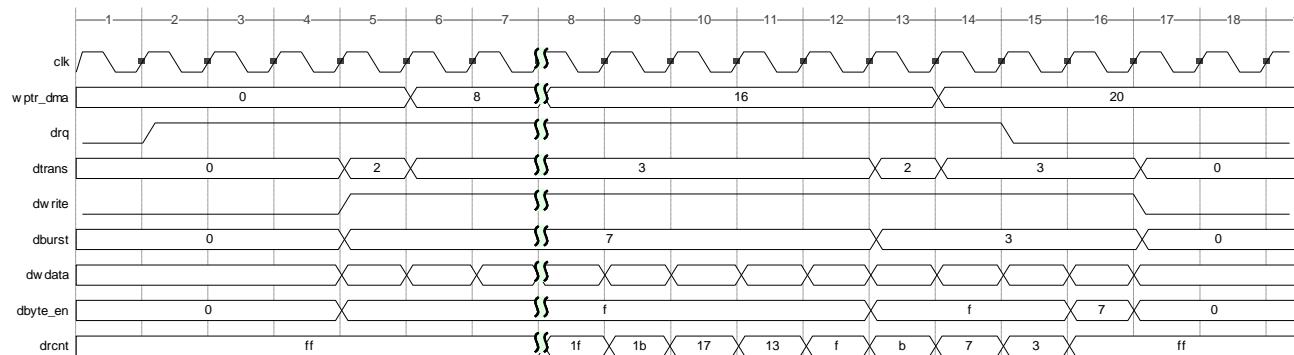


Figure 8-11 DMA read from memory then write peripheral FIFO. (Burst==8)

(2) DMA 读 peripheral FIFO:

通常 dburst 的值为 7, 表示 1 次 burst 传输 8 笔数据

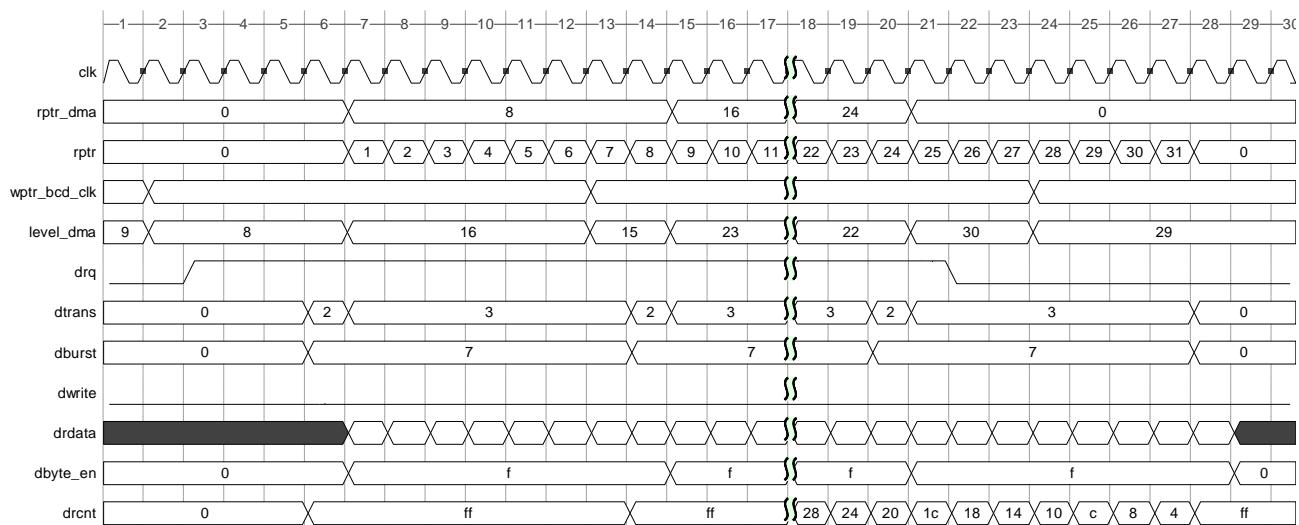


Figure 8-12 DMA read from peripheral FIFO then write memory. (Burst==8)

在传输快要结束的时候 dburst 的值可能不为 7，如下图：

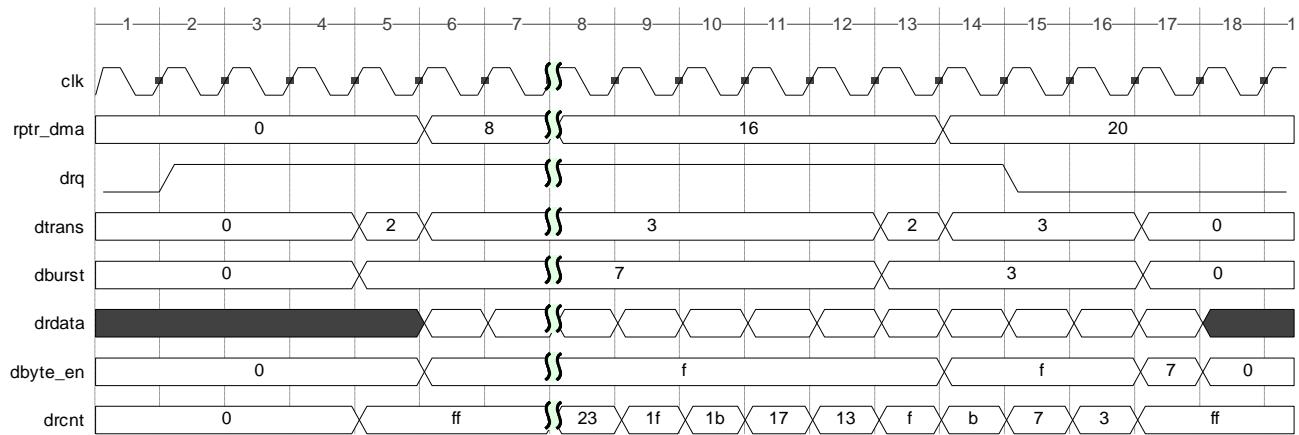


Figure 8-13 DMA read from peripheral FIFO then write memory. (Burst==4 for the last time)

注意此图中的最后 1 笔数据传输 dbyte\_en 不为 0xf，而是 0x7，即表示最后传输的数据不是 4 字节。

### 8.1.8 DMA0/1/2/3/4 access channel list

| DST<br>SRC | SEQ<br>MEM | AUIP<br>FIFO | SPI<br>TX<br>FIFO | UART<br>TX<br>FIFO | USB<br>FIFO | DAC<br>FIFO<br>0 | DAC<br>FIFO<br>1 | SD<br>FIFO | ACC<br>FIFO |
|------------|------------|--------------|-------------------|--------------------|-------------|------------------|------------------|------------|-------------|
| SEQ_MEM    | Y          | Y            | Y                 | Y                  | Y           | Y                | Y                | Y          | Y           |
| SPIRX FIFO | Y          |              |                   |                    |             |                  |                  |            |             |
| URRX FIFO  | Y          |              |                   |                    |             |                  |                  |            |             |
| USB FIFO   | Y          |              |                   |                    |             |                  |                  | Y          |             |
| ADC FIFO   | Y          |              |                   |                    |             |                  |                  |            |             |
| SD FIFO    | Y          |              |                   |                    | Y           |                  |                  |            |             |
| REV_MEM    | Y          | Y            |                   |                    |             |                  |                  |            |             |
| AUIP       | Y          |              |                   |                    |             |                  |                  |            |             |

|                 |   |  |  |  |  |  |  |  |  |
|-----------------|---|--|--|--|--|--|--|--|--|
| ENCOUT          |   |  |  |  |  |  |  |  |  |
| SPDIFRX<br>FIFO | Y |  |  |  |  |  |  |  |  |

Table 1 DMA0/1/2/3/4 access channel list

## 8.1.9 Operation Manual

### 8.1.9.1 SRC and DST of DMA

- Type OF SRC and DST: FIFO or Memory
- If the SRC or DST selects FIFO, the SADDR or DADDR registers are invalid.
- If the SRC or DST selects memory, the SADDR or DADDR registers are valid. And the address is 24 bits, also support virtual address.
- Memory transfers of DMA support sequential transfers and reverse transfers.

### 8.1.9.2 Length of DMA

- Memory to DAC FIFO or ADC FIFO to memory: frame\_length =the number of reading or writing FIFO
- Other: Every time, DMA can transfer 64KB data at the most. And if the DMAFrameLen register is configured to zero, it means that will transfer 64KB one time.

### 8.1.9.3 Reload

If DMA have been set to reload mode, it will transfer again when the last transfers is completed, using the configuration setting in last time.

### 8.1.9.4 Abort

If a transfer is uncompleted and you want to stop it, you can clear the DMASTART bit (DMACTL0\_bit0) to stop it.

### 8.1.9.5 Width

See section DMA transfer type.

### 8.1.9.6 Priorities

- You should admeasure the DMA channel suitably in software.

- DMA of high priorities should be admeasured to the most important transfers to make sure that the transfers are not be intermitted.
- The priorities of DMA: DMA0>DMA1>DMA2>DMA3>DMA4

### 8.1.10 DMA Register List

| Index | Mnemonic                      | Description                                     | BANK                     |
|-------|-------------------------------|---|--------------------------|
| 0x91  | <a href="#">DMA0IP</a>        | DMA0 interrupt pending register                 | 0x0e                     |
| 0x92  | <a href="#">DMA0IE</a>        | DMA0 interrupt enable register                  | 0x0e                     |
| 0x93  | <a href="#">DMA01234DBG</a>   | DMA0/1/2/3/4 debug register                     | 0x0e/0x0f/0x10/0x11/0x12 |
| 0x94  | <a href="#">DMA0CTL0</a>      | DMA0 control register 0                         | 0x0e                     |
| 0x95  | <a href="#">DMA0CTL1</a>      | DMA0 control register 1                         | 0x0e                     |
| 0x97  | <a href="#">DMA0SADDR0H</a>   | DMA0 source address register 0 high byte        | 0x0e                     |
| 0x98  | <a href="#">DMA0SADDR0L</a>   | DMA0 source address register 0 low byte         | 0x0e                     |
| 0x9b  | <a href="#">DMA0SADDR0M</a>   | DMA0 source address register 0 middle byte      | 0x0e                     |
| 0x9d  | <a href="#">DMA0DADDR0H</a>   | DMA0 destination address register 0 high byte   | 0x0e                     |
| 0x9e  | <a href="#">DMA0DADDR0L</a>   | DMA0 destination address register 0 low byte    | 0x0e                     |
| 0x9f  | <a href="#">DMA0DADDR0M</a>   | DMA0 destination address register 0 middle byte | 0x0e                     |
| 0xa4  | <a href="#">DMA0FrameLenH</a> | DMA0 frame length register high byte            | 0x0e                     |
| 0xa5  | <a href="#">DMA0FrameLenL</a> | DMA0 frame length register low byte             | 0x0e                     |
| 0x91  | <a href="#">DMA1IP</a>        | DMA1 interrupt pending register                 | 0x0f                     |
| 0x92  | <a href="#">DMA1IE</a>        | DMA1 interrupt enable register                  | 0x0f                     |
| 0x94  | <a href="#">DMA1CTL0</a>      | DMA1 control register 0                         | 0x0f                     |
| 0x95  | <a href="#">DMA1CTL1</a>      | DMA1 control register 1                         | 0x0f                     |
| 0x97  | <a href="#">DMA1SADDR0H</a>   | DMA1 source address register 0 high byte        | 0x0f                     |
| 0x98  | <a href="#">DMA1SADDR0L</a>   | DMA1 source address register 0 low byte         | 0x0f                     |
| 0x9b  | <a href="#">DMA1SADDR0M</a>   | DMA1 source address register 0 middle byte      | 0x0f                     |
| 0x9d  | <a href="#">DMA1DADDR0H</a>   | DMA1 destination address register 0 high byte   | 0x0f                     |
| 0x9e  | <a href="#">DMA1DADDR0L</a>   | DMA1 destination address register 0 low byte    | 0x0f                     |
| 0x9f  | <a href="#">DMA1DADDR0M</a>   | DMA1 destination address register 0 middle byte | 0x0f                     |
| 0xa4  | <a href="#">DMA1FrameLenH</a> | DMA1 frame length register high byte            | 0x0f                     |
| 0xa5  | <a href="#">DMA1FrameLenL</a> | DMA1 frame length register low byte             | 0x0f                     |
| 0x91  | <a href="#">DMA2IP</a>        | DMA2 interrupt pending register                 | 0x10                     |
| 0x92  | <a href="#">DMA2IE</a>        | DMA2 interrupt enable register                  | 0x10                     |
| 0x94  | <a href="#">DMA2CTL0</a>      | DMA2 control register 0                         | 0x10                     |
| 0x95  | <a href="#">DMA2CTL1</a>      | DMA2 control register 1                         | 0x10                     |
| 0x97  | <a href="#">DMA2SADDR0H</a>   | DMA2 source address register 0 high byte        | 0x10                     |
| 0x98  | <a href="#">DMA2SADDR0L</a>   | DMA2 source address register 0 low byte         | 0x10                     |
| 0x9b  | <a href="#">DMA2SADDR0M</a>   | DMA2 source address register 0 middle byte      | 0x10                     |
| 0x9d  | <a href="#">DMA2DADDR0H</a>   | DMA2 destination address register 0 high byte   | 0x10                     |
| 0x9e  | <a href="#">DMA2DADDR0L</a>   | DMA2 destination address register 0 low byte    | 0x10                     |

|      |                               |   |      |
|------|-------------------------------|---|------|
| 0x9f | <a href="#">DMA2DADDR0M</a>   | DMA2 destination address register 0 middle byte | 0x10 |
| 0xa4 | <a href="#">DMA2FrameLenH</a> | DMA2 frame length register high byte            | 0x10 |
| 0xa5 | <a href="#">DMA2FrameLenL</a> | DMA2 frame length register low byte             | 0x10 |
| 0x91 | <a href="#">DMA3IP</a>        | DMA3 interrupt pending register                 | 0x11 |
| 0x92 | <a href="#">DMA3IE</a>        | DMA3 interrupt enable register                  | 0x11 |
| 0x94 | <a href="#">DMA3CTL0</a>      | DMA3 control register 0                         | 0x11 |
| 0x95 | <a href="#">DMA3CTL1</a>      | DMA3 control register 1                         | 0x11 |
| 0x97 | <a href="#">DMA3SADDR0H</a>   | DMA3 source address register 0 high byte        | 0x11 |
| 0x98 | <a href="#">DMA3SADDR0L</a>   | DMA3 source address register 0 low byte         | 0x11 |
| 0x9b | <a href="#">DMA3SADDR0M</a>   | DMA3 source address register 0 middle byte      | 0x11 |
| 0x9d | <a href="#">DMA3DADDR0H</a>   | DMA3 destination address register 0 high byte   | 0x11 |
| 0x9e | <a href="#">DMA3DADDR0L</a>   | DMA3 destination address register 0 low byte    | 0x11 |
| 0x9f | <a href="#">DMA3DADDR0M</a>   | DMA0 destination address register 0 middle byte | 0x11 |
| 0xa4 | <a href="#">DMA3FrameLenH</a> | DMA3 frame length register high byte            | 0x11 |
| 0xa5 | <a href="#">DMA3FrameLenL</a> | DMA3 frame length register low byte             | 0x11 |
| 0x91 | <a href="#">DMA4IP</a>        | DMA4 interrupt pending register                 | 0x12 |
| 0x92 | <a href="#">DMA4IE</a>        | DMA4 interrupt enable register                  | 0x12 |
| 0x94 | <a href="#">DMA4CTL0</a>      | DMA4 control register 0                         | 0x12 |
| 0x95 | <a href="#">DMA4CTL1</a>      | DMA4 control register 1                         | 0x12 |
| 0x97 | <a href="#">DMA4SADDR0H</a>   | DMA4 source address register 0 high byte        | 0x12 |
| 0x98 | <a href="#">DMA4SADDR0L</a>   | DMA4 source address register 0 low byte         | 0x12 |
| 0x9b | <a href="#">DMA4SADDR0M</a>   | DMA4 source address register 0 middle byte      | 0x12 |
| 0x9d | <a href="#">DMA4DADDR0H</a>   | DMA4 destination address register 0 high byte   | 0x12 |
| 0x9e | <a href="#">DMA4DADDR0L</a>   | DMA4 destination address register 0 low byte    | 0x12 |
| 0x9f | <a href="#">DMA4DADDR0M</a>   | DMA4 destination address register 0 middle byte | 0x12 |
| 0xa4 | <a href="#">DMA4FrameLenH</a> | DMA4 frame length register high byte            | 0x12 |
| 0xa5 | <a href="#">DMA4FrameLenL</a> | DMA4 frame length register low byte             | 0x12 |

## 8.1.11 DMA0/1/2/3/4 Register Description

### 8.1.11.1 DMA0IP

**DMA0IP (DMA0 Interrupt Pending Register, SFR address 0x91, bank:0xe)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros   | -      | -     |
| 1          | DMA0HFIP     | DMA0 Half Transfer IRQ Pending, writing 1 to this bit will clear it. (1)       | R/W    | 0     |
| 0          | DMA0TCIP     | DMA0 Transfer Complete Interrupt Pending, writing 1 to this bit will clear it. | R/W    | 0     |

2033、 The DMAxHFIRQ can be set by DMA controller while more than half number of pixels has

been send to LCD FIFO for DMA transfer of memory to LCD; the DMAxHFIRQ can also be set while more than half bytes of data been transferred by DMA controller.

### 8.1.11.2 DMA0IE

**DMA0IE (DMA0 Interrupt Enable Register, SFR address 0x92, bank:0xe)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros  | -      | -     |
| 1          | DMA0HFIE     | DMA0 Half Transfer Complete IRQ enable bit<br>0: Disable Half Transfer Complete interrupt;<br>1: Enable Half Transfer Complete interrupt. | R/W    | 0     |
| 0          | DMA0TCIE     | DMA0 Transfer Complete Interrupt Enable bit:<br>0: disable DMA0 Transfer Complete interrupt<br>1: enable DMA0 Transfer Complete interrupt | R/W    | 0     |

### 8.1.11.3 DMA01234DBG

**DMA01234DBG (DMA0/1/2/3/4 Debug Register, SFR address 0x93, bank:0xe, 0xf, 0x10,0x11)**

| Bit Number | Bit Mnemonic    | Function           | Access | Reset |
|------------|-----------------|--------------------|--------|-------|
| 7:5        | Reserved        | Be read as 5-zeros | -      | -     |
| 4:0        | DMA01234D<br>BG | DMA debug signal   | R/W    | 0     |

### 8.1.11.4 DMA0CTL0

**DMA0CTL0 (DMA0 control Register 0, SFR address 0x94, bank:0xe)**

| Bit Number | Bit Mnemonic  | Function  | Access | Reset |
|------------|---------------|---|--------|-------|
| 7          | ENDIAN        | The endian type of audio data in memory:<br>0: small endian<br>1: big endian  | R/W    | 0     |
| 6          | Reserved      | Be read as zeros  | -      | -     |
| 5:4        | DATAWIDT<br>H | The data width to write DAC FIFO or read from ADC/SPDIF FIFO:<br>00: 8bit<br>01: 16bit<br>10: 24bit<br>11: reserved | R/W    | 00    |
| 3:2        | Reserved      | Be read as 2-zeros  | -      | -     |

|   |               |   |     |   |
|---|---------------|---|-----|---|
| 1 | RELOAD        | Reload the DMA controller registers and start DMA transfer after current DMA transfer is complete:<br>0: disable reload mode<br>1: enable reload mode   | R/W | 0 |
| 0 | DMA0STAR<br>T | DMA0 start bit:<br><br>A low-to-high conversion of this bit will trigger loading source address, destination address, destination step size, source step size, transfer type, burst_length, DRQ_type, data width to the DMA1 controller. This bit will be automatically cleared by the DMA1 controller if the DMA1 transfer is complete or DMA1 transfer error occurs.<br><br>Write '0' to this bit can abort DMA transfer. | R/W | 0 |

### 8.1.11.5 DMA0CTL1

**DMA0CTL1 (DMA0 control Register 1, SFR address 0x95, bank:0xe)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | DSTTYPE      | Destination type:<br><br>4'b0000: memory<br>4'b0001: Audio Codec Input FIFO<br>4'b0010: SPI TX FIFO<br>4'b0011: UART TX FIFO<br>4'b0100: USB FIFO<br>4'b0101: DAC FIFO0<br>4'b0110: SD/MMC FIFO<br>4'b0111: DAC FIFO1<br>4'b1000: reserved<br>4'b1001: ACC FIFO<br>4'b1010: memory<br>Others: skip all data | R/W    | 0000  |
| 3:0        | SRCTYPE      | Source type:<br><br>4'b0000: sequential access memory<br>4'b0001: AUIP ENC OUT FIFO<br>4'b0010: SPI RX FIFO<br>4'b0011: UART RX FIFO<br>4'b0100: USB FIFO<br>4'b0101: ADC FIFO<br>4'b0110: SD/MMC FIFO<br>4'b0111: SPDIFRX FIFO   | R/W    | 0000  |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | 4'b1000: reserved<br>4'b1001: reserved<br>4'b1010: reverse access memory<br>4'b1011: reserved<br>Others: skip all data |  |  |
|--|--|--|--|--|

### 8.1.11.6 DMA0SADDR0H

**DMA0SADDR0H (DMA0 Source Address Register 0 high byte, SFR address 0x97, bank:0xe)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA0SADD R0H | The bit 23:16 of the source address 0 of DMA0 transfer. | R/W    | 0     |

### 8.1.11.7 DMA0SADDR0M

**DMA0SADDR0M (DMA0 Source Address Register 0 middle byte, SFR address 0x9b, bank:0xe)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | DMA0SADD R0M | The bit 15:8 of the source address 0 of DMA0 transfer. | R/W    | 0     |

### 8.1.11.8 DMA0SADDR0L

**DMA0SADDR0L (DMA0 Source Address Register 0 low byte, SFR address 0x98, bank:0xe)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA0SADD R0L | The bit 7:0 of the source address 0 of DMA0 transfer. | R/W    | 0     |

### 8.1.11.9 DMA0DADDR0H

**DMA0DADDR0H (DMA0 Destination Address Register 0 high byte, SFR address 0x9d, bank:0xe)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | DMA0DADD R0H | The bit 23:16 of the destination address 0 of DMA0 transfer. | R/W    | 0     |

### 8.1.11.10 DMA0DADDR0M

**DMA0DADDR0M(DMA0 Destination Address Register 0 middle byte, SFR address 0x9f, bank:0xe)**

| Bit Number | Bit Mnemonic    | Function  | Access | Reset |
|------------|-----------------|---|--------|-------|
| 7:0        | DMA0DADD<br>R0H | The bit 15:8 of the destination address 0 of DMA0 transfer. | R/W    | 0     |

### 8.1.11.11 DMA0DADDR0L

**DMA0DADDR0L (DMA0 Destination Address Register 0 low byte, SFR address 0x9e, bank:0xe)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7:0        | DMA0DADD<br>R0L | The bit 7:0 of the destination address 0 of DMA0 transfer. | R/W    | 0     |

### 8.1.11.12 DMA0FrameLenH

**DMA0FrameLenH (DMA0 Frame Length Register 1 high byte, SFR address 0xa4, bank:0xe)**

| Bit Number | Bit Mnemonic      | Function  | Access | Reset |
|------------|-------------------|---|--------|-------|
| 7:0        | DMA0Frame<br>LenH | The bit 15:8 of frame length of DMA0 transfer.<br>The “DMA0FrameLen” blocks of data will be transmitted if a high to low transition of DMA0START bit is occur.<br>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA writes FIFO.<br>If SRCTYPE is ADCFIFO the value of DM0FrameLen is equal to the times that DMA reads FIFO.<br>If other DSTTYPE or SRCTYPE the value of DM0FrameLen is equal to the number of bytes transferred by DMA.<br>Write 0 to DMA0FrameLenH and DMA0frameLenL register will force the DMA0 controller to transfer 65536 bytes /samples/pixels. | R/W    | 0     |

### 8.1.11.13 DMA0FrameLenL

**DMA0FrameLenL (DMA0 Frame Length Register 1 low byte, SFR address 0xa5, bank:0xe)**

| Bit Number | Bit Mnemonic  | Function  | Access | Reset |
|------------|---------------|---|--------|-------|
| 7:0        | DMA0FrameLenL | <p>The bit 7:0 of frame length of DMA0 transfer.</p> <p>The “DMA0FrameLen” blocks of data will be transmitted if a high to low transition of DMA0START bit is occur.</p> <p>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA writes FIFO.</p> <p>If SRCTYPE is ADCFIFO the value of DM0FrameLen is equal to the times that DMA reads FIFO.</p> <p>If other DSTTYPE or SRCTYPE the value of DM0FrameLen is equal to the number of bytes transferred by DMA.</p> <p>Write 0 to DMA0FrameLenH and DMA0frameLenL register will force the DMA0 controller to transfer 65536 bytes /samples/pixels.</p> | R/W    | 0     |

### 8.1.11.14 DMA1IP

**DMA1IP (DMA1 Interrupt Pending Register, SFR address 0x91, bank:0xf)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros   | -      | -     |
| 1          | DMA1HFIP     | DMA1 Half Transfer IRQ Pending, writing 1 to this bit will clear it. (1)       | R/W    | 0     |
| 0          | DMA1TCIP     | DMA1 Transfer Complete Interrupt Pending, writing 1 to this bit will clear it. | R/W    | 0     |

2034、 The DMAxHFIRQ can be set by DMA controller while more than half number of pixels has been send to LCD FIFO for DMA transfer of memory to LCD; the DMAxHFIRQ can also be set while more than half bytes of data been transferred by DMA controller.

### 8.1.11.15 DMA1IE

**DMA1IE (DMA1 Interrupt Enable Register, SFR address 0x92, bank:0xf)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros  | -      | -     |
| 1          | DMA1HFIE     | DMA1 Half Transfer Complete IRQ enable bit<br>0: Disable Half Transfer Complete interrupt;<br>1: Enable Half Transfer Complete interrupt. | R/W    | 0     |

|   |          |   |     |   |
|---|----------|---|-----|---|
| 0 | DMA1TCIE | DMA1 Transfer Complete Interrupt Enable bit:<br>0: disable DMA1 Transfer Complete interrupt<br>1: enable DMA1 Transfer Complete interrupt | R/W | 0 |
|---|----------|---|-----|---|

### 8.1.11.16 DMA1CTL0

**DMA1CTL0 (DMA1 control Register 0, SFR address 0x94, bank:0xf)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | ENDIAN       | The endian type of audio data in memory:<br>0: small endian<br>1: big endian  | R/W    | 0     |
| 6          | Reserved     | Be read as zeros  | -      | -     |
| 5:4        | DATAWIDTH    | The data width to write DAC FIFO or read from ADC/SPDIF FIFO:<br>00: 8bit<br>01: 16bit<br>10: 24bit<br>11: reserved   | R/W    | 00    |
| 3:2        | Reserved     | Be read as 2-zeros  | -      | -     |
| 1          | RELOAD       | Reload the DMA controller registers and start DMA transfer after current DMA transfer is complete:<br>0: disable reload mode<br>1: enable reload mode   | R/W    | 0     |
| 0          | DMA1STAR     | DMA1 start bit:<br>A low-to-high conversion of this bit will trigger loading source address, destination address, destination step size, source step size, transfer type, burst_length, DRQ_type, data width to the DMA1 controller. This bit will be automatically cleared by the DMA1 controller if the DMA1 transfer is complete or DMA1 transfer error occurs.<br>Write '0' to this bit can abort DMA transfer. | R/W    | 0     |

### 8.1.11.17 DMA1CTL1

**DMA1CTL1 (DMA1 control Register 1, SFR address 0x95, bank:0xf)**

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:4        | DSTTYPE      | Destination type: | R/W    | 0000  |

|     |         |   |     |      |
|-----|---------|---|-----|------|
|     |         | 4'b0000: memory<br>4'b0001: Audio Codec Input FIFO<br>4'b0010: SPI TX FIFO<br>4'b0011: UART TX FIFO<br>4'b0100: USB FIFO<br>4'b0101: DAC FIFO0<br>4'b0110: SD/MMC FIFO<br>4'b0111: DAC FIFO1<br>4'b1000: reserved<br>4'b1001: ACC FIFO<br>4'b1010: memory<br>Others: skip all data  |     |      |
| 3:0 | SRCTYPE | Source type:<br>4'b0000: sequential access memory<br>4'b0001: AUIP ENC OUT FIFO<br>4'b0010: SPI RX FIFO<br>4'b0011: UART RX FIFO<br>4'b0100: USB FIFO<br>4'b0101: ADC FIFO<br>4'b0110: SD/MMC FIFO<br>4'b0111: SPDIFRX FIFO<br>4'b1000: reserved<br>4'b1001: reserved<br>4'b1010: reverse access memory<br>4'b1011: reserved<br>Others: skip all data | R/W | 0000 |

### 8.1.11.18 DMA1SADDR0H

**DMA1SADDR0H (DMA1 Source Address Register 0 high byte, SFR address 0x97, bank:0xf)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA1SADD R0H | The bit 23:16 of the source address 0 of DMA1 transfer. | R/W    | 0     |

### 8.1.11.19 DMA1SADDR0M

**DMA1SADDR0M (DMA1 Source Address Register 0 middle byte, SFR address 0x9b, bank:0xf)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | DMA1SADD R0H | The bit 15:8 of the source address 0 of DMA1 transfer. | R/W    | 0     |

### 8.1.11.20 DMA1SADDR0L

**DMA1SADDR0L (DMA1 Source Address Register 0 low byte, SFR address 0x98, bank:0xf)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA1SADD R0L | The bit 7:0 of the source address 0 of DMA1 transfer. | R/W    | 0     |

### 8.1.11.21 DMA1DADDR0H

**DMA1DADDR0H (DMA1 Destination Address Register 0 high byte, SFR address 0x9d, bank:0xf)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | DMA1DADD R0H | The bit 23:16 of the destination address 0 of DMA1 transfer. | R/W    | 0     |

### 8.1.11.22 DMA1DADDR0M

**DMA1DADDR0M (DMA1 Destination Address Register 0 middle byte, SFR address 0x9f, bank:0xf)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA1DADD R0H | The bit 15:8 of the destination address 0 of DMA1 transfer. | R/W    | 0     |

### 8.1.11.23 DMA1DADDR0L

**DMA1DADDR0L (DMA1 Destination Address Register 0 low byte, SFR address 0x9e, bank:0xf)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | DMA1DADD R0L | The bit 7:0 of the destination address 0 of DMA1 transfer. | R/W    | 0     |

### 8.1.11.24 DMA1FrameLenH

**DMA1FrameLenH (DMA1 Frame Length Register 1 high byte, SFR address 0xa4, bank:0xf)**

| Bit Number | Bit Mnemonic   | Function  | Access | Reset |
|------------|----------------|---|--------|-------|
| 7:0        | DMA1Frame LenH | The bit 15:8 of frame length of DMA1 transfer.<br>The “DMA1FrameLen” blocks of data will be | R/W    | 0     |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | <p>transmitted if a high to low transition of DMA1START bit is occur.</p> <p>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA writes FIFO.</p> <p>If SRCTYPE is ADCFIFO the value of DM0FrameLen is equal to the times that DMA reads FIFO.</p> <p>If other DSTTYPE or SRCTYPE the value of DM0FrameLen is equal to the number of bytes transferred by DMA.</p> <p>Write 0 to DMA1FrameLenH and DMA1frameLenL register will force the DMA1 controller to transfer 65536 bytes /samples/pixels.</p> |  |  |
|--|--|--|--|--|

### 8.1.11.25 DMA1FrameLenL

**DMA1FrameLenH (DMA1 Frame Length Register 1 low byte, SFR address 0xa5, bank:0xf)**

| Bit Number | Bit Mnemonic  | Function  | Access | Reset |
|------------|---------------|---|--------|-------|
| 7:0        | DMA1FrameLenL | <p>The bit 7:0 of frame length of DMA1 transfer.</p> <p>The “DMA1FrameLen” blocks of data will be transmitted if a high to low transition of DMA1START bit is occur.</p> <p>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA writes FIFO.</p> <p>If SRCTYPE is ADCFIFO the value of DM0FrameLen is equal to the times that DMA reads FIFO.</p> <p>If other DSTTYPE or SRCTYPE the value of DM0FrameLen is equal to the number of bytes transferred by DMA.</p> <p>Write 0 to DMA1FrameLenH and DMA1frameLenL register will force the DMA1 controller to transfer 65536 bytes /samples/pixels.</p> | R/W    | 0     |

### 8.1.11.26 DMA2IP

**DMA2IP (DMA2 Interrupt Pending Register, SFR address 0x91, bank:0x10)**

| Bit Number | Bit Mnemonic | Function           | Access | Reset |
|------------|--------------|--------------------|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros | -      | -     |

|   |          |  |     |   |
|---|----------|--|-----|---|
| 1 | DMA2HFIP | DMA2 Half Transfer IRQ Pending, writing 1 to this bit will clear it. (1)       | R/W | 0 |
| 0 | DMA2TCIP | DMA2 Transfer Complete Interrupt Pending, writing 1 to this bit will clear it. | R/W | 0 |

2035、 The DMAxHFIRQ can be set by DMA controller while more than half number of pixels has been send to LCD FIFO for DMA transfer of memory to LCD; the DMAxHFIRQ can also be set while more than half bytes of data been transferred by DMA controller.

### 8.1.11.27 DMA2IE

**DMA2IE (DMA2 Interrupt Enable Register, SFR address 0x92, bank:0x10)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros  | -      | -     |
| 1          | DMA2HFIE     | DMA2 Half Transfer Complete IRQ enable bit<br>0: Disable Half Transfer Complete interrupt;<br>1: Enable Half Transfer Complete interrupt. | R/W    | 0     |
| 0          | DMA2TCIE     | DMA2 Transfer Complete Interrupt Enable bit:<br>0: disable DMA2 Transfer Complete interrupt<br>1: enable DMA2 Transfer Complete interrupt | R/W    | 0     |

### 8.1.11.28 DMA2CTL0

**DMA2CTL0 (DMA2 control Register 0, SFR address 0x94, bank:0x10)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | ENDIAN       | The endian type of audio data in memory:<br>0: small endian<br>1: big endian  | R/W    | 0     |
| 6          | Reserved     | Be read as zeros  | -      | -     |
| 5:4        | DATAWIDT H   | The data width to write DAC FIFO or read from ADC/SPDIF FIFO:<br>00: 8bit<br>01: 16bit<br>10: 24bit<br>11: reserved                                   | R/W    | 00    |
| 3:2        | Reserved     | Be read as 2-zeros  | -      | -     |
| 1          | RELOAD       | Reload the DMA controller registers and start DMA transfer after current DMA transfer is complete:<br>0: disable reload mode<br>1: enable reload mode | R/W    | 0     |

|   |               |  |     |   |
|---|---------------|--|-----|---|
| 0 | DMA2STAR<br>T | DMA2 start bit:<br><br>A low-to-high conversion of this bit will trigger loading source address, destination address, destination step size, source step size, transfer type, burst_length, DRQ_type, data width to the DMA2 controller. This bit will be automatically cleared by the DMA2 controller if the DMA2 transfer is complete or DMA2 transfer error occurs.<br><br>Write '0' to this bit can abort DMA stransfer. | R/W | 0 |
|---|---------------|--|-----|---|

### 8.1.11.29 DMA2CTL1

**DMA2CTL1 (DMA2 control Register 1, SFR address 0x95, bank:0x10)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | DSTTYPE      | Destination type:<br><br>4'b0000: memory<br>4'b0001: Audio Codec Input FIFO<br>4'b0010: SPI TX FIFO<br>4'b0011: UART TX FIFO<br>4'b0100: USB FIFO<br>4'b0101: DAC FIFO0<br>4'b0110: SD/MMC FIFO<br>4'b0111: DAC FIFO1<br>4'b1000: reserved<br>4'b1001: ACC FIFO<br>4'b1010: memory<br>Others: skip all data   | R/W    | 0000  |
| 3:0        | SRCTYPE      | Source type:<br><br>4'b0000: sequential access memory<br>4'b0001: AUIP ENC OUT FIFO<br>4'b0010: SPI RX FIFO<br>4'b0011: UART RX FIFO<br>4'b0100: USB FIFO<br>4'b0101: ADC FIFO<br>4'b0110: SD/MMC FIFO<br>4'b0111: SPDIFRX FIFO<br>4'b1000: reserved<br>4'b1001: reserved<br>4'b1010: reverse access memory<br>4'b1011: reserved<br>Others: skip all data | R/W    | 0000  |

### 8.1.11.30 DMA2SADDR0H

**DMA2SADDR0H (DMA2 Source Address Register 0 high byte, SFR address 0x97, bank:0x10)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA2SADD R0H | The bit 23:16 of the source address 0 of DMA2 transfer. | R/W    | 0     |

### 8.1.11.31 DMA2SADDR0M

**DMA2SADDR0M (DMA2 Source Address Register 0 middle byte, SFR address 0x9b, bank:0x10)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | DMA2SADD R0H | The bit 15:8 of the source address 0 of DMA2 transfer. | R/W    | 0     |

### 8.1.11.32 DMA2SADDR0L

**DMA2SADDR0L (DMA2 Source Address Register 0 low byte, SFR address 0x98, bank:0x10)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA2SADD R0L | The bit 7:0 of the source address 0 of DMA2 transfer. | R/W    | 0     |

### 8.1.11.33 DMA2DADDR0H

**DMA2DADDR0H (DMA2 Destination Address Register 0 high byte, SFR address 0x9d, bank:0x10)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | DMA2DADD R0H | The bit 23:16 of the destination address 0 of DMA2 transfer. | R/W    | 0     |

### 8.1.11.34 DMA2DADDR0M

**DMA2DADDR0M (DMA2 Destination Address Register 0 middle byte, SFR address 0x9f, bank:0x10)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA2DADD R0H | The bit 15:8 of the destination address 0 of DMA2 transfer. | R/W    | 0     |

### 8.1.11.35 DMA2DADDR0L

**DMA2DADDR0L (DMA2 Destination Address Register 0 low byte, SFR address 0x9e, bank:0x10)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7:0        | DMA2DADD<br>R0L | The bit 7:0 of the destination address 0 of DMA2 transfer. | R/W    | 0     |

### 8.1.11.36 DMA2FrameLenH

**DMA2FrameLenH (DMA2 Frame Length Register 1 high byte, SFR address 0xa4, bank:0x10)**

| Bit Number | Bit Mnemonic      | Function   | Access | Reset |
|------------|-------------------|--|--------|-------|
| 7:0        | DMA2Frame<br>LenH | <p>The bit 15:8 of frame length of DMA2 transfer.</p> <p>The “DMA2FrameLen” blocks of data will be transmitted if a high to low transition of DMA2START bit is occur.</p> <p>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA writes FIFO.</p> <p>If SRCTYPE is ADCFIFO the value of DM0FrameLen is equal to the times that DMA reads FIFO.</p> <p>If other DSTTYPE or SRCTYPE the value of DM0FrameLen is equal to the number of bytes transferred by DMA.</p> <p>Write 0 to DMA2FrameLenH and DMA2frameLenL register will force the DMA2 controller to transfer 65536 bytes /samples/pixels.</p> | R/W    | 0     |

### 8.1.11.37 DMA2FrameLenL

**DMA2FrameLenH (DMA2 Frame Length Register 1 low byte, SFR address 0xa5, bank:0x10)**

| Bit Number | Bit Mnemonic      | Function   | Access | Reset |
|------------|-------------------|--|--------|-------|
| 7:0        | DMA2Frame<br>LenL | <p>The bit 7:0 of frame length of DMA2 transfer.</p> <p>The “DMA2FrameLen” blocks of data will be transmitted if a high to low transition of DMA2START bit is occur.</p> <p>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA</p> | R/W    | 0     |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | writes FIFO.<br>If SRCTYPE is ADCFIFO the value of DM0FrameLen is equal to the times that DMA reads FIFO.<br>If other DSTTYPE or SRCTYPE the value of DM0FrameLen is equal to the number of bytes transferred by DMA.<br>Write 0 to DMA2FrameLenH and DMA2frameLenL register will force the DMA2 controller to transfer 65536 bytes /samples/pixels. |  |  |
|--|--|--|--|--|

### 8.1.11.38 DMA3IP

**DMA3IP (DMA3 Interrupt Pending Register, SFR address 0x91, bank:0x11)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros   | -      | -     |
| 1          | DMA3HFIP     | DMA3 Half Transfer IRQ Pending, writing 1 to this bit will clear it. (1)       | R/W    | 0     |
| 0          | DMA3TCIP     | DMA3 Transfer Complete Interrupt Pending, writing 1 to this bit will clear it. | R/W    | 0     |

2036、 The DMAxHFIRQ can be set by DMA controller while more than half number of pixels has been send to LCD FIFO for DMA transfer of memory to LCD; the DMAxHFIRQ can also be set while more than half bytes of data been transferred by DMA controller.

### 8.1.11.39 DMA3IE

**DMA3IE (DMA3 Interrupt Enable Register, SFR address 0x92, bank:0x11)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros  | -      | -     |
| 1          | DMA3HFIE     | DMA3 Half Transfer Complete IRQ enable bit<br>0: Disable Half Transfer Complete interrupt;<br>1: Enable Half Transfer Complete interrupt. | R/W    | 0     |
| 0          | DMA3TCIE     | DMA3 Transfer Complete Interrupt Enable bit:<br>0: disable DMA2 Transfer Complete interrupt<br>1: enable DMA2 Transfer Complete interrupt | R/W    | 0     |

### 8.1.11.40 DMA3CTL0

**DMA3CTL0 (DMA3 control Register 0, SFR address 0x94, bank:0x11)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | ENDIAN       | The endian type of audio data in memory:<br>0: small endian<br>1: big endian  | R/W    | 0     |
| 6          | Reserved     | Be read as zeros  | -      | -     |
| 5:4        | DATAWIDTH    | The data width to write DAC FIFO or read from ADC/SPDIF FIFO:<br>00: 8bit<br>01: 16bit<br>10: 24bit<br>11: reserved   | R/W    | 00    |
| 3:2        | Reserved     | Be read as 2-zeros  | -      | -     |
| 1          | RELOAD       | Reload the DMA controller registers and start DMA transfer after current DMA transfer is complete:<br>0: disable reload mode<br>1: enable reload mode   | R/W    | 0     |
| 0          | DMA3STAR     | DMA3 start bit:<br>A low-to-high conversion of this bit will trigger loading source address, destination address, destination step size, source step size, transfer type, burst_length, DRQ_type, data width to the DMA3 controller. This bit will be automatically cleared by the DMA3 controller if the DMA3 transfer is complete or DMA3 transfer error occurs.<br>Write '0' to this bit can abort DMA transfer. | R/W    | 0     |

### 8.1.11.41 DMA3CTL1

**DMA3CTL1 (DMA3 control Register 1, SFR address 0x95, bank:0x11)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | DSTTYPE      | Destination type:<br>4'b0000: memory<br>4'b0001: Audio Codec Input FIFO<br>4'b0010: SPI TX FIFO<br>4'b0011: UART TX FIFO<br>4'b0100: USB FIFO<br>4'b0101: DAC FIFO0<br>4'b0110: SD/MMC FIFO<br>4'b0111: DAC FIFO1 | R/W    | 0000  |

|     |         |   |     |      |
|-----|---------|---|-----|------|
|     |         | 4'b1000: reserved<br>4'b1001: ACC FIFO<br>4'b1010: memory<br>Others: skip all data  |     |      |
| 3:0 | SRCTYPE | Source type:<br>4'b0000: sequential access memory<br>4'b0001: AUIP ENC OUT FIFO<br>4'b0010: SPI RX FIFO<br>4'b0011: UART RX FIFO<br>4'b0100: USB FIFO<br>4'b0101: ADC FIFO<br>4'b0110: SD/MMC FIFO<br>4'b0111: SPDIFRX FIFO<br>4'b1000: reserved<br>4'b1001: reserved<br>4'b1010: reverse access memory<br>4'b1011: reserved<br>Others: skip all data | R/W | 0000 |

### 8.1.11.42 DMA3SADDR0H

DMA3SADDR0H (DMA3 Source Address Register 0 high byte, SFR address 0x97, bank:0x11)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA3SADD R0H | The bit 23:16 of the source address 0 of DMA3 transfer. | R/W    | 0     |

### 8.1.11.43 DMA3SADDR0M

DMA3SADDR0M (DMA3 Source Address Register 0 middle byte, SFR address 0x9b, bank:0x11)

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | DMA3SADD R0H | The bit 15:8 of the source address 0 of DMA3 transfer. | R/W    | 0     |

### 8.1.11.44 DMA3SADDR0L

DMA3SADDR0L (DMA3 Source Address Register 0 low byte, SFR address 0x98, bank:0x11)

| Bit Number | Bit Mnemonic | Function                                    | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | DMA3SADD     | The bit 7:0 of the source address 0 of DMA3 | R/W    | 0     |

|     |           |  |  |
|-----|-----------|--|--|
| ROL | transfer. |  |  |
|-----|-----------|--|--|

### 8.1.11.45 DMA3DADDR0H

**DMA3DADDR0H (DMA3 Destination Address Register 0 high byte, SFR address 0x9d, bank:0x11)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7:0        | DMA3DADD<br>R0H | The bit 23:16 of the destination address 0 of DMA3 transfer. | R/W    | 0     |

### 8.1.11.46 DMA3DADDR0M

**DMA3DADDR0M (DMA3 Destination Address Register 0 middle byte, SFR address 0x9f, bank:0x11)**

| Bit Number | Bit Mnemonic    | Function  | Access | Reset |
|------------|-----------------|---|--------|-------|
| 7:0        | DMA3DADD<br>R0H | The bit 15:8 of the destination address 0 of DMA3 transfer. | R/W    | 0     |

### 8.1.11.47 DMA3DADDR0L

**DMA3DADDR0L (DMA3 Destination Address Register 0 low byte, SFR address 0x9e, bank:0x11)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7:0        | DMA3DADD<br>R0L | The bit 7:0 of the destination address 0 of DMA3 transfer. | R/W    | 0     |

### 8.1.11.48 DMA3FrameLenH

**DMA3FrameLenH (DMA3 Frame Length Register 1 high byte, SFR address 0xa4, bank:0x11)**

| Bit Number | Bit Mnemonic      | Function   | Access | Reset |
|------------|-------------------|--|--------|-------|
| 7:0        | DMA3Frame<br>LenH | The bit 15:8 of frame length of DMA3 transfer.<br>The “DMA3FrameLen” blocks of data will be transmitted if a high to low transition of DMA3START bit is occur.<br>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA writes FIFO.<br>If SRCTYPE is ADCFIFO the value of DM0FrameLen is equal to the times that DMA | R/W    | 0     |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | <p>reads FIFO.</p> <p>If other DSTTYPE or SRCTYPE the value of DM0FrameLen is equal to the number of bytes transferred by DMA.</p> <p>Write 0 to DMA3FrameLenH and DMA3frameLenL register will force the DMA3 controller to transfer 65536 bytes /samples/pixels.</p> |  |  |
|--|--|---|--|--|

### 8.1.11.49 DMA3FrameLenL

**DMA3FrameLenH (DMA3 Frame Length Register 1 low byte, SFR address 0xa5, bank:0x11)**

| Bit Number | Bit Mnemonic  | Function  | Access | Reset |
|------------|---------------|---|--------|-------|
| 7:0        | DMA3FrameLenL | <p>The bit 7:0 of frame length of DMA3 transfer.</p> <p>The “DMA3FrameLen” blocks of data will be transmitted if a high to low transition of DMA3START bit is occur.</p> <p>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA writes FIFO.</p> <p>If SRCTYPE is ADCFIFO the value of DM3FrameLen is equal to the times that DMA reads FIFO.</p> <p>If other DSTTYPE or SRCTYPE the value of DM3FrameLen is equal to the number of bytes transferred by DMA.</p> <p>Write 0 to DMA3FrameLenH and DMA3frameLenL register will force the DMA3 controller to transfer 65536 bytes /samples/pixels.</p> | R/W    | 0     |

### 8.1.11.50 DMA4IP

**DMA4IP (DMA4 Interrupt Pending Register, SFR address 0x91, bank:0x12)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros   | -      | -     |
| 1          | DMA4HFIP     | DMA4 Half Transfer IRQ Pending, writing 1 to this bit will clear it. (1)       | R/W    | 0     |
| 0          | DMA4TCIP     | DMA4 Transfer Complete Interrupt Pending, writing 1 to this bit will clear it. | R/W    | 0     |

2037、 The DMAxHFIRQ can be set by DMA controller while more than half number of pixels has been send to LCD FIFO for DMA transfer of memory to LCD; the DMAxHFIRQ can also be set

while more than half bytes of data been transferred by DMA controller.

### 8.1.11.51 DMA4IE

**DMA4IE (DMA4 Interrupt Enable Register, SFR address 0x92, bank:0x12)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:2        | Reserved     | Be read as 6-zeros  | -      | -     |
| 1          | DMA4HFIE     | DMA4 Half Transfer Complete IRQ enable bit<br>0: Disable Half Transfer Complete interrupt;<br>1: Enable Half Transfer Complete interrupt. | R/W    | 0     |
| 0          | DMA4TCIE     | DMA4 Transfer Complete Interrupt Enable bit:<br>0: disable DMA2 Transfer Complete interrupt<br>1: enable DMA2 Transfer Complete interrupt | R/W    | 0     |

### 8.1.11.52 DMA4CTL0

**DMA4CTL0 (DMA4 control Register 0, SFR address 0x94, bank:0x12)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | ENDIAN       | The endian type of audio data in memory:<br>0: small endian<br>1: big endian  | R/W    | 0     |
| 6          | Reserved     | Be read as zeros  | -      | -     |
| 5:4        | DATAWIDT H   | The data width to write DAC FIFO or read from ADC/SPDIF FIFO:<br>00: 8bit<br>01: 16bit<br>10: 24bit<br>11: reserved   | R/W    | 00    |
| 3:2        | Reserved     | Be read as 2-zeros  | -      | -     |
| 1          | RELOAD       | Reload the DMA controller registers and start DMA transfer after current DMA transfer is complete:<br>0: disable reload mode<br>1: enable reload mode   | R/W    | 0     |
| 0          | DMA3STAR T   | DMA4 start bit:<br>A low-to-high conversion of this bit will trigger loading source address, destination address, destination step size, source step size, transfer type, burst_length, DRQ_type, data width to the DMA3 controller. This bit will be automatically | R/W    | 0     |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | cleared by the DMA3 controller if the DMA3 transfer is complete or DMA3 transfer error occurs.<br>Write '0' to this bit can abort DMA transfer. |  |  |
|--|--|---|--|--|

### 8.1.11.53 DMA4CTL1

**DMA4CTL1 (DMA4 control Register 1, SFR address 0x95, bank:0x12)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | DSTTYPE      | Destination type:<br>4'b0000: memory<br>4'b0001: Audio Codec Input FIFO<br>4'b0010: SPI TX FIFO<br>4'b0011: UART TX FIFO<br>4'b0100: USB FIFO<br>4'b0101: DAC FIFO0<br>4'b0110: SD/MMC FIFO<br>4'b0111: DAC FIFO1<br>4'b1000: reserved<br>4'b1001: ACC FIFO<br>4'b1010: memory<br>Others: skip all data   | R/W    | 0000  |
| 3:0        | SRCTYPE      | Source type:<br>4'b0000: sequential access memory<br>4'b0001: AUIP ENC OUT FIFO<br>4'b0010: SPI RX FIFO<br>4'b0011: UART RX FIFO<br>4'b0100: USB FIFO<br>4'b0101: ADC FIFO<br>4'b0110: SD/MMC FIFO<br>4'b0111: SPDIFRX FIFO<br>4'b1000: reserved<br>4'b1001: reserved<br>4'b1010: reverse access memory<br>4'b1011: reserved<br>Others: skip all data | R/W    | 0000  |

### 8.1.11.54 DMA4SADDR0H

**DMA4SADDR0H (DMA4 Source Address Register 0 high byte, SFR address 0x97, bank:0x12)**

| Bit | Bit | Function | Access | Reset |
|-----|-----|----------|--------|-------|
|-----|-----|----------|--------|-------|

| Number | Mnemonic        |   |     |   |
|--------|-----------------|---|-----|---|
| 7:0    | DMA4SADD<br>R0H | The bit 23:16 of the source address 0 of DMA4 transfer. | R/W | 0 |

### 8.1.11.55 DMA4SADDR0M

**DMA4SADDR0M (DMA4 Source Address Register 0 middle byte, SFR address 0x9b, bank:0x12)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7:0        | DMA4SADD<br>R0H | The bit 15:8 of the source address 0 of DMA4 transfer. | R/W    | 0     |

### 8.1.11.56 DMA4SADDR0L

**DMA4SADDR0L (DMA4 Source Address Register 0 low byte, SFR address 0x98, bank:0x12)**

| Bit Number | Bit Mnemonic    | Function  | Access | Reset |
|------------|-----------------|---|--------|-------|
| 7:0        | DMA4SADD<br>R0L | The bit 7:0 of the source address 0 of DMA4 transfer. | R/W    | 0     |

### 8.1.11.57 DMA4DADDR0H

**DMA4DADDR0H (DMA4 Destination Address Register 0 high byte, SFR address 0x9d, bank:0x12)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7:0        | DMA4DADD<br>R0H | The bit 23:16 of the destination address 0 of DMA4 transfer. | R/W    | 0     |

### 8.1.11.58 DMA4DADDR0M

**DMA4DADDR0M (DMA4 Destination Address Register 0 middle byte, SFR address 0x9f, bank:0x12)**

| Bit Number | Bit Mnemonic    | Function  | Access | Reset |
|------------|-----------------|---|--------|-------|
| 7:0        | DMA4DADD<br>R0H | The bit 15:8 of the destination address 0 of DMA4 transfer. | R/W    | 0     |

### 8.1.11.59 DMA4DADDR0L

**DMA4DADDR0L (DMA4 Destination Address Register 0 low byte, SFR address 0x9e, bank:0x12)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | DMA4DADD R0L | The bit 7:0 of the destination address 0 of DMA4 transfer. | R/W    | 0     |

### 8.1.11.60 DMA4FrameLenH

**DMA4FrameLenH (DMA4 Frame Length Register 1 high byte, SFR address 0xa4, bank:0x12)**

| Bit Number | Bit Mnemonic  | Function   | Access | Reset |
|------------|---------------|--|--------|-------|
| 7:0        | DMA4FrameLenH | <p>The bit 15:8 of frame length of DMA4 transfer.</p> <p>The “DMA4FrameLen” blocks of data will be transmitted if a high to low transition of DMA4START bit is occur.</p> <p>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA writes FIFO.</p> <p>If SRCTYPE is ADCFIFO the value of DM0FrameLen is equal to the times that DMA reads FIFO.</p> <p>If other DSTTYPE or SRCTYPE the value of DM0FrameLen is equal to the number of bytes transferred by DMA.</p> <p>Write 0 to DMA4FrameLenH and DMA2frameLenL register will force the DMA4 controller to transfer 65536 bytes /samples/pixels.</p> | R/W    | 0     |

### 8.1.11.61 DMA4FrameLenL

**DMA4FrameLenH (DMA4 Frame Length Register 1 low byte, SFR address 0xa5, bank:0x12)**

| Bit Number | Bit Mnemonic  | Function   | Access | Reset |
|------------|---------------|--|--------|-------|
| 7:0        | DMA4FrameLenL | <p>The bit 7:0 of frame length of DMA2 transfer.</p> <p>The “DMA4FrameLen” blocks of data will be transmitted if a high to low transition of DMA4START bit is occur.</p> <p>If DSTTYPE is DACFIFO or LCDFIFO the value of DM0FrameLen is equal to the times that DMA writes FIFO.</p> <p>If SRCTYPE is ADCFIFO the value of DM4FrameLen is equal to the times that DMA reads FIFO.</p> | R/W    | 0     |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | If other DSTTYPE or SRCTYPE the value of DM4FrameLen is equal to the number of bytes transferred by DMA.<br><br>Write 0 to DMA4FrameLenH and DMA3frameLenL register will force the DMA4 controller to transfer 65536 bytes /samples/pixels. |  |  |
|--|--|---|--|--|

## 8.2 Audio IP (彭洪、鲍超、郑思、唐振中)

| 日期         | 版本     | 描述   | 修订人 |
|------------|--------|--|-----|
| 2012-07-24 | V1. 00 | initial  | 彭洪  |
| 2012-09-12 | V1. 02 | 1、增加 MP3 编码模式<br>2、在寄存器列表中增加 MP3 编码的寄存器<br>3、修改了 fade in 和 fade out 的时长<br>4、删除了 WAVCtl 寄存器中 bit4 的描述<br>5、修改了 WAVCtl 寄存器中 bit5:4 的描述，当值为 2' b11 时，为正常播放<br>6、在 wave decoder 中增加 ChannelSel bit 来选择左右声道<br>7、增加 input FIFO1 的专用 Reset bit，直通模式专用<br>8、删除 EQ 各频带的增益寄存器，将其该到 RAM 中<br>9、删除寄存器列表中的 EQ 寄存器描述 | 彭洪  |
| 2012-10-20 | V2. 00 | 1、修改了 EQ 的 memory 占用空间<br>2、增加了 SRS 的描述<br>3、修改了 MUROM1 的大小需求<br>4、修改了 BEPCTL1 寄存器的 EffectEn bit 描述，<br>5、该 bit 只会控制 SRS 和 karaoke 音效的使能<br>6、修改了原来的 EffectMode 寄存器位，将起变成 EQEn 和 EffectMode 位<br>7、修改了寄存器列表中的 SRS 寄存器<br>8、修改了 Efuse Option 的描述<br>9、修改信号列表中关于 SRS 使能控制 bit 0x05a7 的 bit6 的描述            | 彭洪  |
| 2012-11-07 | V2. 01 | 1、SRS 的控制寄存器与 karaoke 的控制寄存器地址相同，因此修改 karaoke 的默认值<br>2、删除 SRSCTL2 寄存器中的 DefinitionREn 和 DefinitionLEn 位   | 彭洪  |
| 2012-12-12 | V2. 02 | 1、修改了 WavHeaderInfo1 寄存器的默认值<br>2、删除 SRSCTL0 的 bit0 SRSEn，将该 bit 修改成 SurroundEn  | 彭洪  |

|            |       |  |    |
|------------|-------|--|----|
|            |       | 3、SRSCTL0 的 bit0 变成 SurroundEn 控制位<br>4、SRSCTL0 的 bit6 变成 SRSTruBassProcessMode 控制位<br>5、由于 SRSCTL0 寄存器的默认值变化，并且 LCHGain 与 SRSCTL0 的寄存器的复用的。因此 LCHGain 的默认值变成 0x71。  |    |
| 2013-01-25 | V2.03 | 1、增加 operation manual 描述<br>2、修改了 MP3CTIMEH 地址，应该是 0xb2 而不是 0xa2（笔误）<br>3、修改了 WMACTIMEH 地址，应该是 0xb2 而不是 0xa2（笔误）<br>4、修改了 WAVCTIMEH 地址，应该是 0xb2 而不是 0xa2（笔误）<br>5、修改 PCM 的 almost empty/almost full 的门限值设定 | 彭洪 |

## 8.2.1 Features

The Audio Codec is designed for low power audio applications. All decoding data paths are optimized to ensure the best performance with the least silicon area. This core has following features:

- (1) Support mpeg1 layer1/2/3 mpeg2 layer3 & mpeg2 2.5 layer 3 decoding
- (2) Support wma decoding.
- (3) Support PCM, IMA-ADPCM, MS-ADPCM, a-law, u-law decoding.
- (4) Support PCM and IMA-ADPCM encoding
- (5) Support mp3 encoding
- (6) 7 or 5 bands equalizer with user-defined 64-step-gain from -12 to 12 Db.
- (7) Fast Forward and Fast Reverse.
- (8) Fade in and Fade Out.
- (9) Digital global gain control
- (10) Energy detection
- (11) Audio enhancement solutions: EQ, SRS TruSurround HD , SRS WOW HD and karaoke.

Supported SRS TruSurround HD features:

- (12) Surround sound virtualization for 1.0 to 6.1
- (13) input channels and 2.0 output channel configuration
- (14) Extremely broad sweet spot for maximum realism and enjoyment
- (15) Sculpted bass to optimize low frequency performance of small drivers and enclosures
- (16) Dialog enhancement to ensure clear and intelligible vocals
- (17) Definition control for maximum high-frequency realism and clarity
- (18) Bass management for subwoofer and center speaker designs

Supported SRS WOW HD features:

- (1) Stereo enhancement which improves dynamic audio performance
- (2) Sculpted bass response for maximum thump on small speakers

- (3) More natural audio with a wider sound field and raised sound image
- (4) High-definition control for realistic clarity
- (5) Center control for dynamic extraction and positioning of dialog

## 8.2.2 Function Description

The Audio Codec has 6 modes:

- (1) MP3 decoding mode
- (2) WMA decoding mode
- (3) PCM, IMA-ADPCM, MS-ADPCM, a-law, u-law decoding mode
- (4) IMA-ADPCM/PCM encoding mode

In decoding mode, audio Codec takes any 8 bits bitstream from the audio file FIFO, automatically detects the header and extracts all header/side information, and then uses the information to decode compressed MP3/WMA /WAV stream. The output from the Audio Codec is 16 bits PCM samples. These PCM samples can be transferred to audio DAC via I2S bus or through a rate converter to a low cost fixed sample rate audio DAC. In encoding mode, audio Codec extracts the raw PCM from the ADC, then compresses it to IMA-ADPCM or MP3 stream and fill the audio encoding buffers.

## 8.2.3 Signal List

This section describes the signal interface of the AUIP of GL5115.

The pin direction key for the signal descriptions is shown in table below.

| Dir | Description   |
|-----|---|
| I   | <b>Input to the AudioIP sampled on the rising edge of the appropriate CLK signal.</b>   |
| O   | <b>Output of the AudioIP, unless otherwise noted, driven at the rising edge of the appropriate clock signal..</b>   |
| A   | <b>Asynchronous inputs that are synchronized by the core.</b>   |
| S   | <b>Static input to the AudioIP. These signals are normally tied to either power or ground and should not change state while AudioIPReset is deasserted.</b> |

Table 1 AudioIP signal type

The AUIP signals are listed in table below. Note that the signals are grouped by logical function, not by expected physical location.

| AUIP of GL5115 |       |     |              |   |  |
|----------------|-------|-----|--------------|---|--|
| reset & clock  |       |     |              |   |  |
| Signal         | Width | I/O | Clock Domain | Description   |  |
| rst_n          | 1     | I   | A            | Connect the MRCR1 register bit 7(AudioCodecReset)   |  |
| ioclk          | 1     | I   | -            | SFR register read/write clock, AUIP fifo0(16*8bit) and AUIP encode output FIFO write clock which is connect to cpuclock |  |
| dclk           | 1     | I   | -            | Auip decoder/encoder clock  |  |

| Write_aupi_fifo1_clock                     | 1     | I   | -             | Aupi_fifo1 is accessed by IISRX/SPDIFRX/ADC.     |
|--|-------|-----|---------------|--|
| Connected to DAC/IIS TX                    |       |     |               |  |
| Signal                                     | Width | I/O | Clock Domain  | Description                                      |
| dac_req                                    | 1     | I   | DAC/IISTx clk | IISTX/ DAC data request                          |
| dac_wr                                     | 1     | O   | dclk          | Write signal to IISTX/DAC fifo                   |
| dac_wr_l_n                                 | 1     | O   | dclk          | Write the left CH audio data to IISTX/DAC fifo.  |
| Dac_wr_r_n                                 | 1     | O   | dclk          | Write the right CH audio data to IISTX/DAC fifo. |
| Dac_wdata[15:0]                            | 1     | O   | dclk          | Audio data.                                      |
| Pcm_half_full                              | 1     | O   | dclk          | PCM Half full                                    |
| pcm_half_empty                             | 1     | O   | dclk          | PCM Half empty                                   |
| Connected to ADC                           |       |     |               |  |
| Signal                                     | Width | I/O | Clock Domain  | Description                                      |
| adc_write_left_ch                          | 1     | I   | ADC clk       | ADC write left channel                           |
| adc_write_right_ch                         | 1     | I   | ADC clk       | ADC write right channel                          |
| adc_data[15:0]                             | 1     | O   | Adc clock     | ADC write aupi data                              |
| Connected to IIS RX                        |       |     |               |  |
| Signal                                     | Width | I/O | Clock Domain  | Description                                      |
| IIS_rx_write_left_ch                       | 1     | I   | IISRX clk     | IISRX write left channel                         |
| IIS_rx_write_right_ch                      | 1     | I   | IISRX clk     | IISRX write right channel                        |
| IIS_data[15:0]                             | 1     | O   | IISRX clk     | IISRX write aupi data                            |
| Connected to SPDIF RX                      |       |     |               |  |
| Signal                                     | Width | I/O | Clock Domain  | Description                                      |
| SPDIF_rx_write_left_ch                     | 1     | I   | SPDIFRXclk    | SPDIFRXwrite left channel                        |
| SPDIF_rx_write_right_ch                    | 1     | I   | SPDIFRXclk    | SPDIFRXwrite right channel                       |
| SPDIF_rx_data[15:0]                        | 1     | O   | SPDIFRXclk    | SPDIFRXwrite aupi data                           |
| Connected to DMA controller (aupi tx fifo) |       |     |               |  |
| Signal                                     | Width | I/O | Clock Domain  | Description                                      |
| drq_tx                                     | 1     | O   | dclk          | Data write request to DMA controller             |
| dmawr                                      | 1     | I   | dmaclk        | dma write strobe, high level is active.          |
| Dmadin[7:0]                                | 1     | I   | dmaclk        | dma data input                                   |
| remain_counter[7:0]                        | 1     | I   | dmaclk        | dma remain counter                               |
| Connected to DMA controller (aupi rx fifo) |       |     |               |  |
| Signal                                     | Width | I/O | Clock Domain  | Description                                      |
| drq_rx                                     | 1     | O   | dclk          | Data read request to DMA controller              |
| dmard                                      | 1     | I   | dmaclk        | dma read strobe, high level is active.           |
| Dmadout[7:0]                               | 1     | I   | dmaclk        | dma data output                                  |

|                     |    |   |        |   |
|---------------------|----|---|--------|---|
| remain_counter[7:0] | 1  | I | dmaclk | dma remain counter                              |
| Static signal       |    |   |        |   |
| SRS_en              | 1  | S | -      | see bit 6 of image register of Efuse1(0x05a7)   |
| Connected to SFR    |    |   |        |   |
| sfraudioipsel       | 1  | I | CPUclk | SFR audio ip select                             |
| sfroe               | 1  | I | CPUclk | SFR output enable                               |
| sfrwe               | 1  | I | CPUclk | SFR write                                       |
| addr_io [31:0]      | 32 | I | CPUclk | SFR address                                     |
| data_io_in [7:0]    | 8  | I | CPUclk | SFR write data                                  |
| data_io_out [7:0]   | 8  | O | CPUclk | SFR read data                                   |
| Connected to memory |    |   |        |   |
| pcmramwr_           | 1  | O | gclk   | Connected to PCMRAM write enable port of AUIP.  |
| Pcmramrd_           | 1  | O | gclk   | Connected to PCMRAM write port of AUIP.         |
| Pcmramsel_          | 1  | O | gclk   | Connected to PCMRAM select port of AUIP.        |
| Pcmramaddr [9:0]    | 10 | O | gclk   | Connected to PCMRAM address port of AUIP.       |
| Pcmramwdata[31:0]   | 32 | O | gclk   | Connected to PCMRAM write data port of AUIP.    |
| Mram1wr_            | 1  | O | gclk   | Connected to MURAM1 write enable port of AUIP.  |
| Mram1rd_            | 1  | O | gclk   | Connected to MURAM1 write port of AUIP.         |
| Mram1hceb           | 1  | O | gclk   | Connected to MURAM1H select port of AUIP.       |
| Mram1mceb           | 1  | O | gclk   | Connected to MURAM1M select port of AUIP.       |
| Mram1lceb           | 1  | O | gclk   | Connected to MURAM1L select port of AUIP.       |
| Mram1addr[12:0]     | 13 | O | gclk   | Connected to MURAM1 address port of AUIP.       |
| Mram1wdata [23:0]   | 24 | O | gclk   | Connected to MURAM1 write data port of AUIP.    |
| Mram1rdata[23:0]    | 24 | I | gclk   | Connected to MURAM1 read data port of AUIP.     |
| Mram2wr_            | 1  | O | gclk   | Connected to MURAM2 write enable port of AUIP.  |
| Mram2rd_            | 1  | O | gclk   | Connected to MURAM2 write port of AUIP.         |
| Mram2sel_           | 1  | O | gclk   | Connected to MURAM2 select port of AUIP.        |
| Mram2addr[9:0]      | 12 | O | gclk   | Connected to MURAM2 address port of AUIP.       |
| Mram2wdata[7:0]     | 8  | O | gclk   | Connected to MURAM2 write data port of AUIP.    |
| Mram2rdata [7:0]    | 8  | I | gclk   | Connected to MURAM2 read data port of AUIP.     |
| Pcmramrwr_          | 1  | O | gclk   | Connected to PCMRAMR write enable port of AUIP. |
| Bist                |    |   |        |   |
| bist_start          | 1  | I | gclk   | ROM1,2,3 bist start signal                      |
| bist_done           | 1  | O | gclk   | ROM1,2,3 bist complete signal                   |
| bist_fail           | 1  | O | gclk   | ROM1,2,3 bist failed signal                     |
| scan                |    |   |        |   |
| scan_mode           | 1  | I | dclk   |   |

Table 2 AudioIP signal list

## 8.2.4 Module Description

### 8.2.4.1 Block Diagram

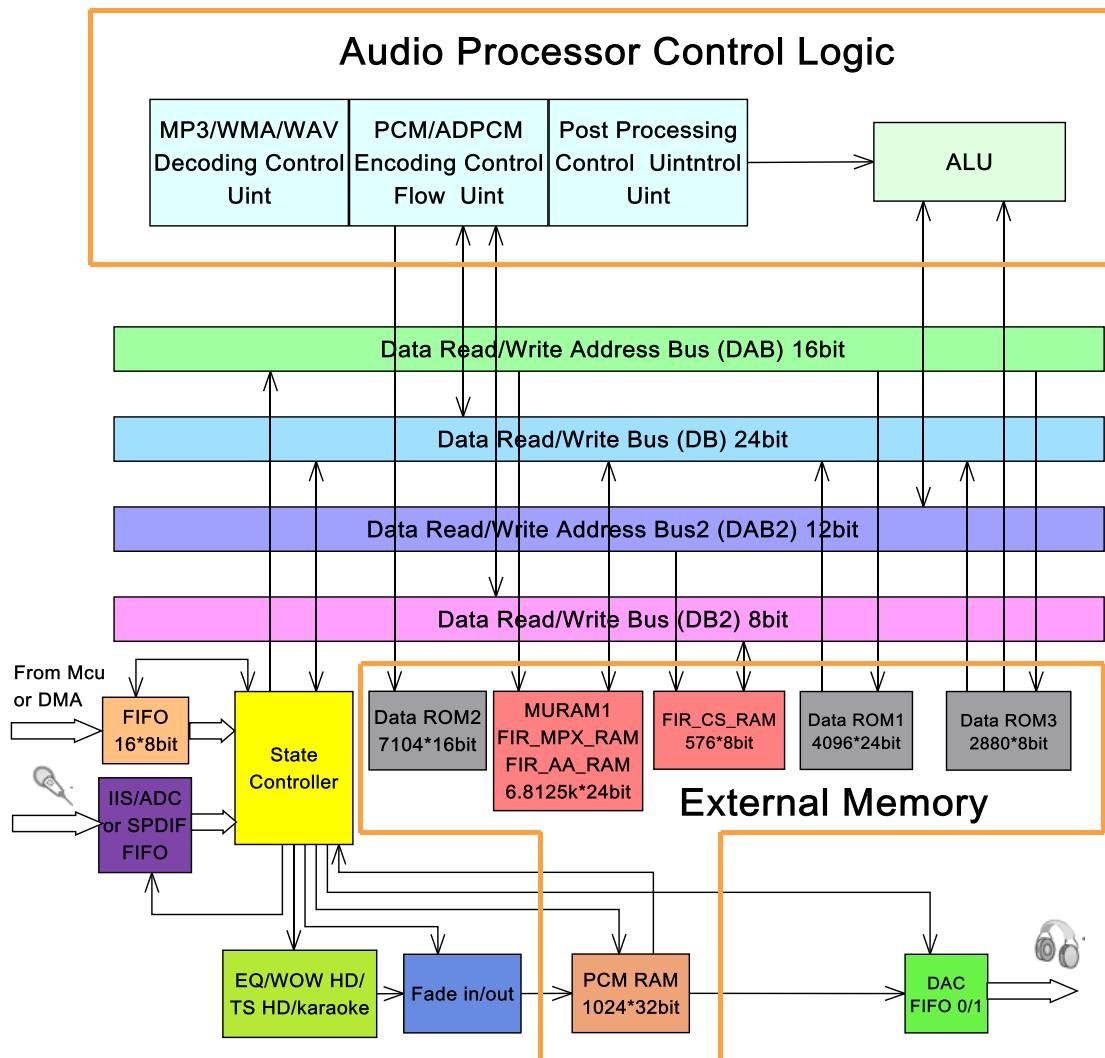


Figure 8-14 block diagram of audio Codec

### 8.2.4.2 Memory Request

The audio code shares the memory with MCU, so the memory should be assigned to the audio code before encoding or decoding started.

| Name        | buswidth | Size  | Address Mapping |
|-------------|----------|-------|-----------------|
| MURAM1      | 24       | 5 k   | 0x0~0x13FF      |
| FIR_MPX_RAM | 24       | 1.75k | 0x0~0x6FF       |
| FIR_AA_RAM  | 24       | 64    | 0x0~0x3F        |
| FIR_CS_RAM  | 8        | 768   | 0x0~0x2FF       |
| PCM RAM     | 32       | 1k    | 0x0~0x3FF       |

|        |    |      |            |
|--------|----|------|------------|
| MUROM1 | 24 | 5k   | 0x0~0x13FF |
| MUROM2 | 16 | 8k   | 0x0~0x1FFF |
| MUROM3 | 8  | 2880 | 0x0~0Xb3F  |

Total RAM size is 35k bytes and ROM size is 25024 bytes.

GL5102 MUROM1 organization: 3424\*24 bit

GL5105 MUROM1 organization: 3958\*24 bit

GL5106 MUROM1 organization: 3424\*24 bit (only needs 3260\*24bit)

GL5109 MUROM1 organization: 3260\*24 bit

### 8.2.4.3 Memory Access Table

Each decoder accesses different hardware resources. The memory requirement is listed below:

| Module mode      | MURAM1 | FIR_MPX_RAM<br>FIR_AA_RAM | FIR_CS_RAM | PCMRAM |
|------------------|--------|---------------------------|------------|--------|
| MP3              | R/W    | R/W                       | R/W        | R/W    |
| WMA/ WAV         | R/W    | R/W                       | R/W        | R/W    |
| WAV/MP3 encoding | R/W    | R/W                       | R/W        | R/W    |

The access sources are listed below:



MCU/DMA0/1/2/3/4



AUIP

### 8.2.4.4 Efuse Option Description

The SRS WOW HD and SRS TruSurround HD effect is activated by the Efuse image register.

| Efuse bit name | Description   | Efuse image regists state                       |
|----------------|---|---|
| SRS_disable    | 0:Both SRS WowHD and TruSurroundHD are disable if the corresponding register bit of ffuse is cleared<br><br>1: :Both SRS WowHD and TruSurroundHD are enable if the corresponding register bit of ffuse is set | Same as the bit 6 of register of Efuse1(0x05a7) |

## 8.2.5 Operation Manual

### 8.2.5.1 Decoding Control Flow Chart

The following steps must be applied for WAV/MP3/WMA stream decoding:

1. The configuration of AUIP and DAC clock:

- (1) Reset AUIP
- (2) Reset DAC and IIS-TX controller
- (3) Enable AUIP clock
- (4) Enable DAC clock
- (5) Enable MURAM1/FIR\_MPX\_RAM/FIR\_AA\_RAM/FIR\_CS\_RAM/PCMRAM clock
- (6) Configure MURAM1/FIR\_MPX\_RAM/FIR\_AA\_RAM/FIR\_CS\_RAM/PCMRAM clock for AUIP access
- (7) Wakeup AUIP

(8) Wakeup DAC and IIS-TX controller

2. Configure DAC and IIS-TX controller

- (1) Wakeup DAC & IIS controller and delay 1ms
- (2) Enable DAC&IIS controller clock
- (3) Configure sample rate fine tune controller if necessary
- (4) Configure MFP of IIS-TX
- (5) Configure sample rat of DAC/IIS-TX controller
- (6) Configure volume
- (7) Reset DAC/IIS-TX FIFO and delay 1ms
- (8) DAC Digital Output Select. 0: to On-Chip sigma-delta DAC 1: to I2S
- (9) DAC Channel 0 FIFO Input Select, select auiip
- (10) Disable IIS-RX controller
- (11) Disable DAC/ADC debug output
- (12) Enable IIS-TX if necessary
- (13) Enable DAC\_IF
- (14) Digital Mixer, DAC Channel 0 Digital Mix to Dac Digital Enable
- (15) Enable DAIFEN

3. Configure AUIP:

- (1) Clear bit 7 of AuCodecCtl register to asynchronous reset AUIP and configure bit 6 to bit 4 to select decoding or encoding mode
  - (2) Reset DAC/IIS-TX FIFO and delay 1ms
  - (3) Abort the transmit of AUIP DMA
  - (4) Enable DAC/IIS-TX FIFO
  - (5) Set bit 7 of AuCodecCtl register to activate AUIP
  - (6) Configure global gain of AUIP
  - (7) Select input source from AUIP input FIFO0
  - (8) Disable AUIP interrupt and disable mute
  - (9) Clear AUIP interrupt pending register
  - (10) Reset AUIP input FIFO0 and configure the FIFO depth at the same time
  - (11) Clear audio effect or configure audio effect
  - (12) Configure energy detection
  - (13) Enable PCM ready interrupt and information ready interrupt
  - (14) Clear Audio Codec Interrupt Pending bits again
  - (15) Enable auiip interrupt
  - (16) Initial frame counter (MP3) / block counter (WAV)/ packet counter (WMA)
  - (17) Initial nCurrentTime register
  - (18) Initial nCurrentEnergy for sentence break detection
-

- (19) Enable nSentenceBreak
- (20) Enable fade in
- (21) Disable whisht & mute
- (22) Configure fade duration
- (23) Set AuDecEncCtl\_NewFile bit to generate newfile signal
- (24) Wait until AuDecEncCtl\_NewFile = 0
- (25) Set AuDecEncCtl\_En bit to enable audio decoding or encoding

#### 4. The control of AUIP:

- (1) Error control

- (2) Fade out control

① The fade out is able to active while BEPCtl4\_FadeInActive is cleared.

② Set BEPCtl1\_FadeOutP bit to clear it.

③ Wait until PCM empty it set to disable the decoder

2038、 Stream read and fill AUIP input FIFO0

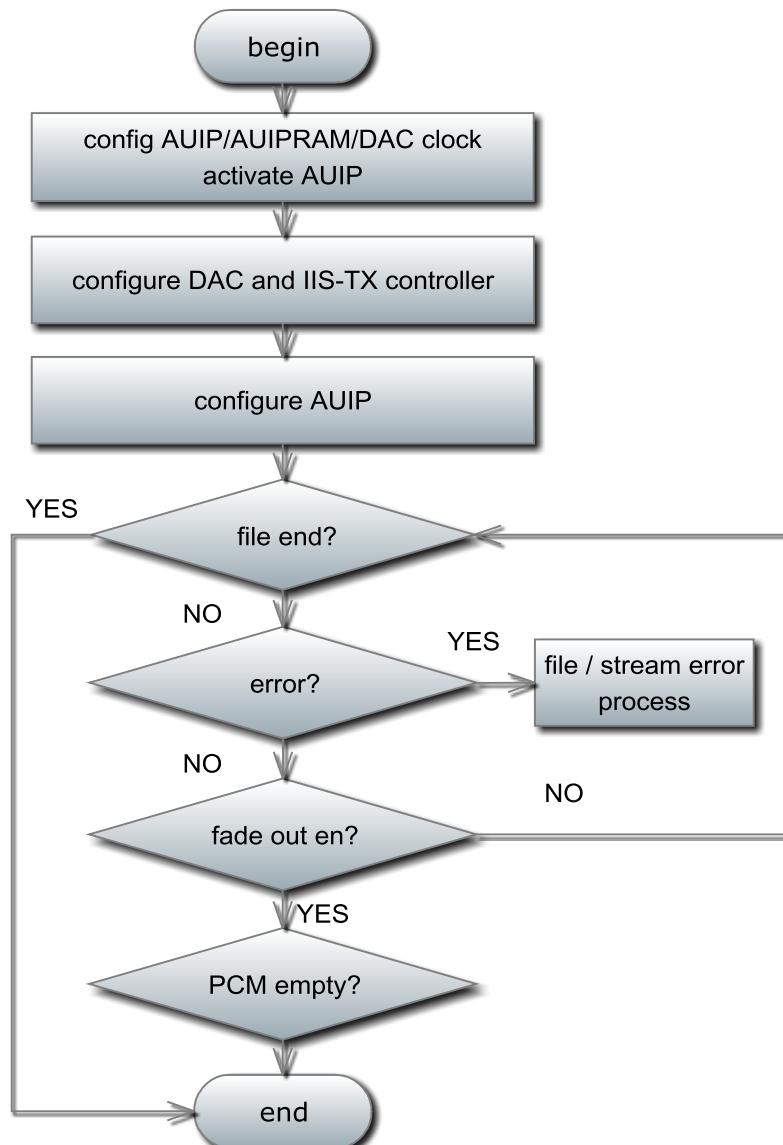


Figure 8-5 decoding control flow chart

### 8.2.5.2 Encoding Control Flow Chart

The following steps must be applied for PCM/IMA-ADPCM/MP2 stream encoding:

1. The configuration of AUIP and ADC clock:

- (1) Reset AUIP
- (2) Reset ADC and IIS-RX controller
- (3) Enable AUIP clock
- (4) Enable ADC clock
- (5) Enable MURAM1 clock
- (6) Configure MURAM1 clock for AUIP access
- (7) Wakeup AUIP

2. Configure ADC and IIS-RX controller

- (1) Wakeup ADC or IIS-RX controller and delay 1ms
- (2) Enable DAC or IIS-RX controller clock
- (3) Configure MFP of IIS-TX
- (4) Configure sample rat of DAC/IIS-TX controller
- (5) Configure volume
- (7) Enable ADOP left ad right channel, enable adc L&R channel
- (8) Set adc channel, both channels are enable

3. Configure AUIP:

- (1) Clear bit 7 of AuCodecCtl register to asynchronous reset AUIP and configure bit 6 to bit 4 to select encoding mode
- (2) Reset DAC/IIS-TX FIFO and delay 1ms
- (3) Abort the transmit of AUIP DMA
- (4) Enable DAC/IIS-TX FIFO
- (5) Set bit 7 of AuCodecCtl register to activate AUIP
- (6) Configure encoding global gain of AUIP
- (7) Select input source from AUIP input FIFO1
- (8) Configure Punctuate Detection
- (9) Set AuDecEncCtl\_NewFile bit to generate newfile signal
- (10) Wait until AuDecEncCtl\_NewFile = 0
- (11) Set AuDecEncCtl\_En bit to enable audio decoding or encoding

4. The control of AUIP:

- (1) Energy calculation control
  - ① Sentence break / mute detection
  - ② Auto disable encoding or hardware whisht

2039、 Read AUIP encoding output FIFO and store to SD CARD

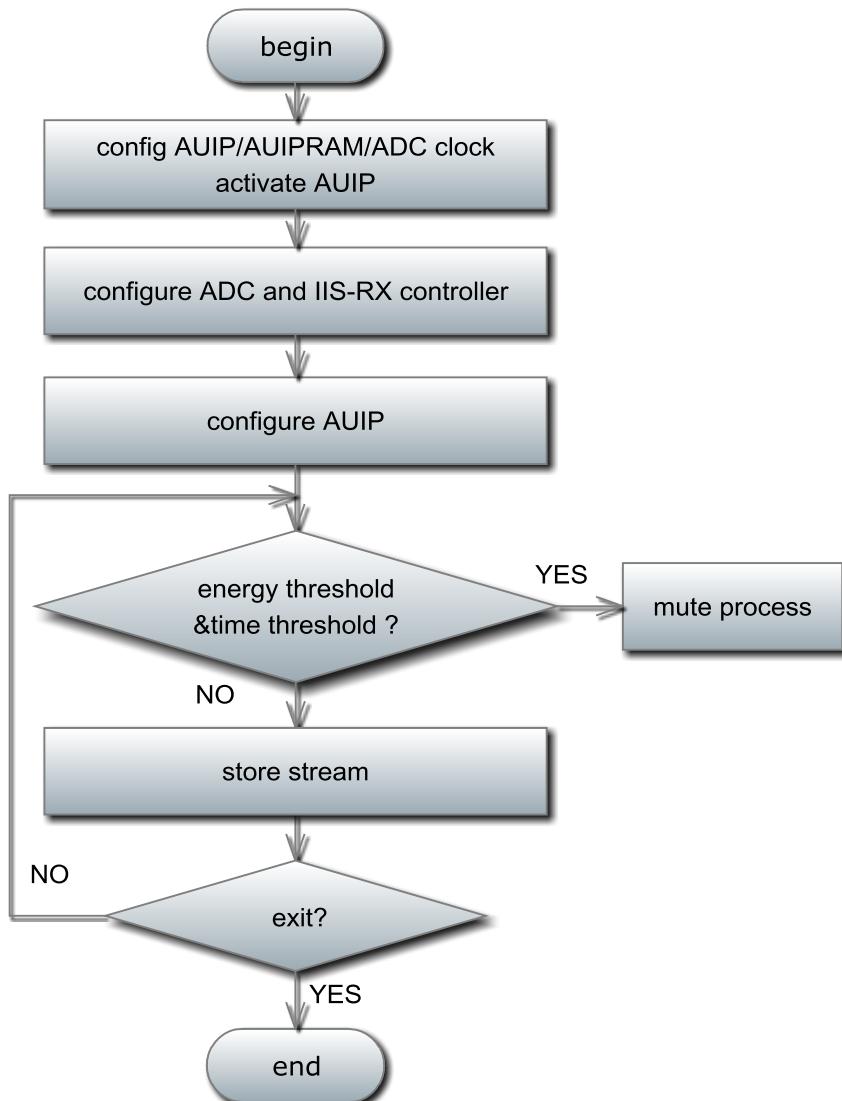


Figure 8-6 encoding control flow chart

### 8.2.5.3 MP3 Break-Point Play Control Flow Chart

The follow steps must be applied for recording a break-point.

1. Stop filling stream to AUIP
2. Waiting AUIP input FIFO0 empty
3. Read the value of MP3FrameNumH/M/L and SynHeaderData1/2 registers for Break-Point Play.

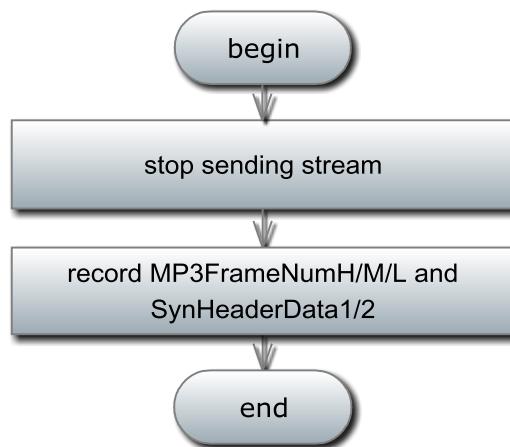


Figure 8-7 record the break-point for MP3 stream

The following steps must be applied for break-point play:

1. Same as the 1<sup>st</sup> step of Decoding Control Flow Chart.
2. Same as the 2<sup>nd</sup> step of Decoding Control Flow Chart.
3. Configure AUIP:
  - (1) Clear bit 7 of AuCodecCtl register to asynchronous reset AUIP and configure bit 6 to bit 4 to select decoding or encoding mode
  - (2) Reset DAC/IIS-TX FIFO and delay 1ms
  - (3) Abort the transmit of AUIP DMA
  - (4) Enable DAC/IIS-TX FIFO
  - (5) Set bit 7 of AuCodecCtl register to activate AUIP
  - (6) Configure global gain of AUIP
  - (7) Select input source from AUIP input FIFO0
  - (8) Disable AUIP interrupt and disable mute
  - (9) Clear AUIP interrupt pending register
  - (10) Reset AUIP input FIFO0 and configure the FIFO depth at the same time
  - (11) Clear audio effect or configure audio effect
  - (12) Configure energy detection
  - (13) Enable PCM ready interrupt and information ready interrupt
  - (14) Clear Audio Codec Interrupt Pending bits again
  - (15) Enable AUIP interrupt
  - (16) Initialize frame counter (MP3) which is also the start frame of the stream.
  - (17) Set MP3 header synchronization parameter and set BKPlayEn bit
  - (18) Initial nCurrentEnergy for sentence break detection
  - (19) Enable nSentenceBreak
  - (20) Enable fade in
  - (21) Disable whisht & mute
  - (22) Configure fade duration
  - (23) Set AuDecEncCtl\_NewFile bit to generate newfile signal
  - (24) Wait until AuDecEncCtl\_NewFile = 0
  - (25) Set AuDecEncCtl\_En bit to enable audio decoding or encoding
  - (26) Disable Halt2IE
  - (27) Clear Halt2IP

(28) Enable Halt2IE

(29) Enable Halt2

#### 4. The control of AUIP:

(1) Error control for header synchronization

(2) Stream read at the start frame of break-point and fill AUIP input FIFO0

(3) Clear both AuCodecDebug2\_Halt2IP and BKPlayEn bit while receive Halt2 interrupt.

(4) Normal play control flow.

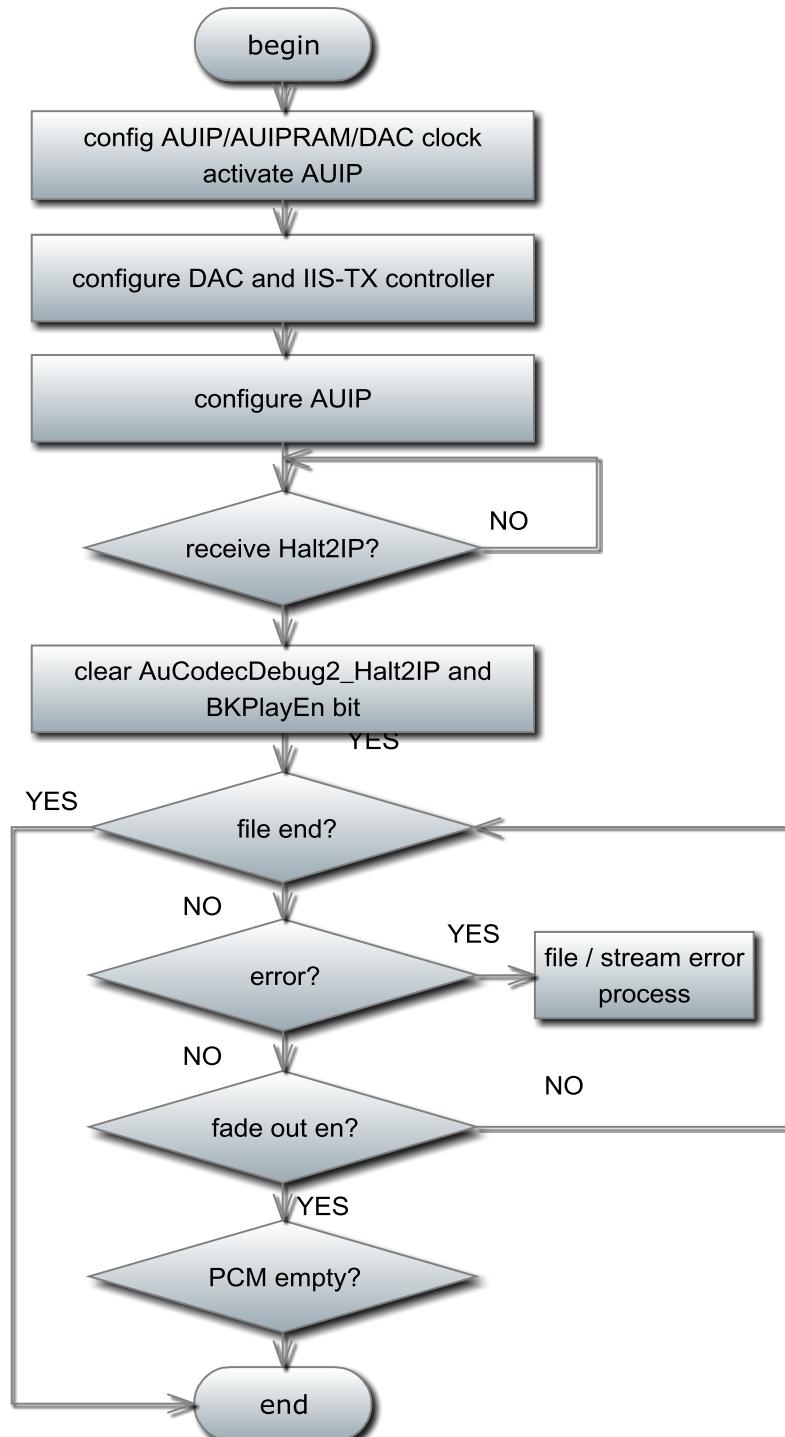


Figure 8-8 MP3 break-point control flow chart

### 8.2.5.4 WMA Break-Point Play Control Flow Chart

The follow steps must be applied for recording a break-point.

1. Stop filling stream to AUIP
2. Waiting AUIP input FIFO0 empty
3. Read the value of WMAPackNumH/M/L and WMAPackSizeH/M/L (the information of packet-size can also be obtain from stream) registers for Break-Point Play.

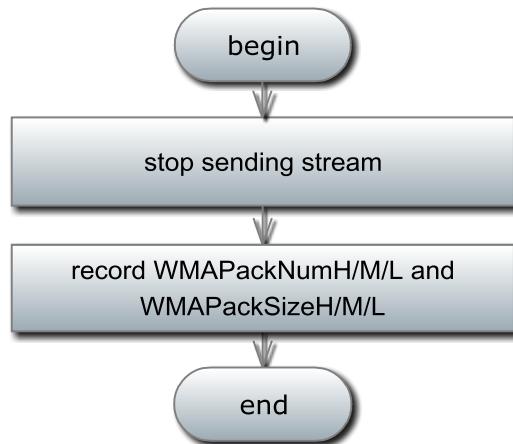


Figure 8-9 record the break-point for WMA stream

The following steps must be applied for break-point play:

1. Same as the 1<sup>st</sup> step of Decoding Control Flow Chart.
2. Same as the 2<sup>nd</sup> step of Decoding Control Flow Chart.
3. Configure AUIP:

- (1)~(25) Same as the 3<sup>rd</sup> step of Decoding Control Flow Chart
- (26) Disable Halt2IE
- (27) Clear Halt2IP
- (28) Enable Halt2IE
- (29) Enable Halt2

4. The control of AUIP:

- (1) Send the PayLoad of the stream. The PayLoad contains the flowing components:
  - ① header object (the packet-size field is at the position of 0x7a bytes from the begin of the stream)
  - ② Header of data object

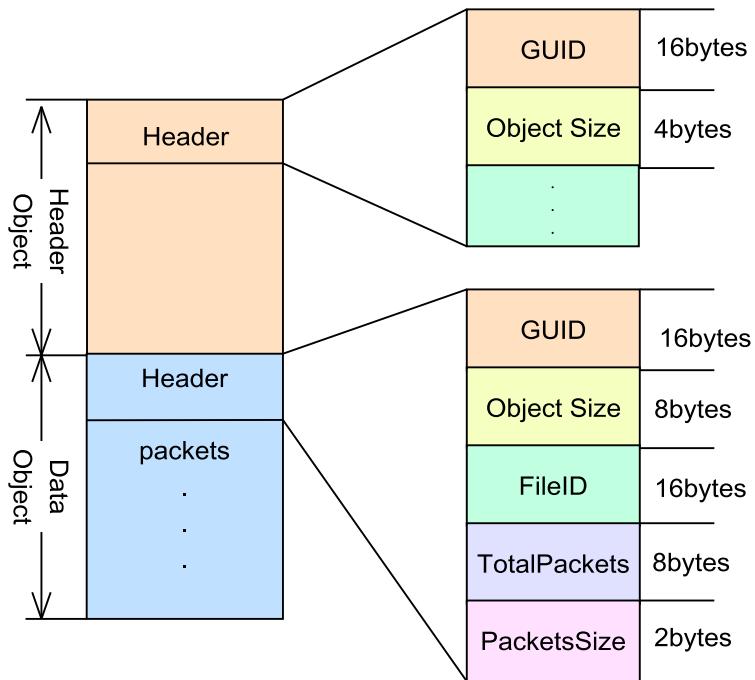


Figure 8-10 the structure of pay-load of a WMA stream

(2) Dead loop if no information ready pending is received.

(3) Fast forward one packet of stream which the packet number is the value is minus 1 of previous break-point packet number which is from WMAPackNumH/M/L registers. The location of the start position from the begin of the stream is calculated by the flowing formulas:

$$\textcircled{1} \quad n\text{FileIndex} = n\text{BreakpointPos} =$$

$(\text{total\_packet\_number}-K\_BREAK\_POINT\_PACKET)*\text{packet\_size}+n\text{Payload};$

\textcircled{2}  $n\text{SectorInternalIndex} = n\text{BreakpointPos}\&(K\_SECTOR\_SIZE-1);$

\textcircled{3}  $n\text{SectorNum} = (n\text{BreakpointPos} - n\text{SectorInternalIndex})/K\_SECTOR\_SIZE;$

2040、 Disable Fast forward and Normal play

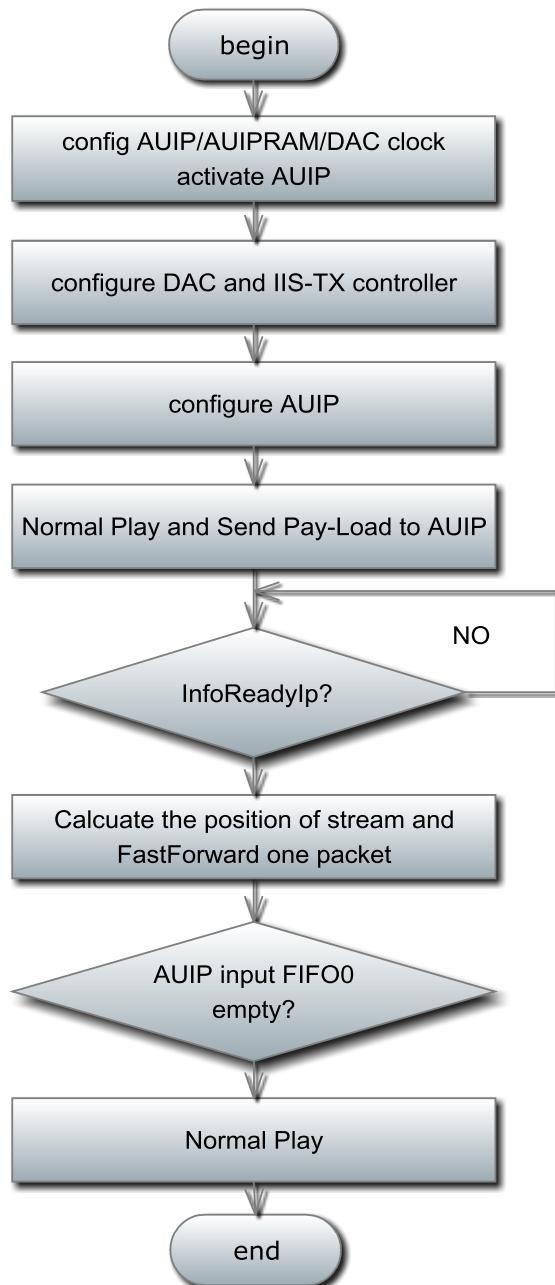


Figure 8-11 WMA break-point control flow chart

### 8.2.5.5 WAV Break-Point Play Control Flow Chart

The follow steps must be applied for recording a break-point.

1. Stop filling stream to AUIP
2. Waiting AUIP input FIFO0 empty
3. Read the value of WAVBlockNumH/M/L registers for Break-Point Play.

Note:

- (1) The block size of the stream is 512 sample pairs (for mono stream the sample\_pairs is 1, for 7.1 channel stream the sample\_pairs is 8) if the stream is PCM, A-Law, U-Law format.
- (2) The block size of the stream is from stream header if the stream is ima-adpcm or ms-adpcm.

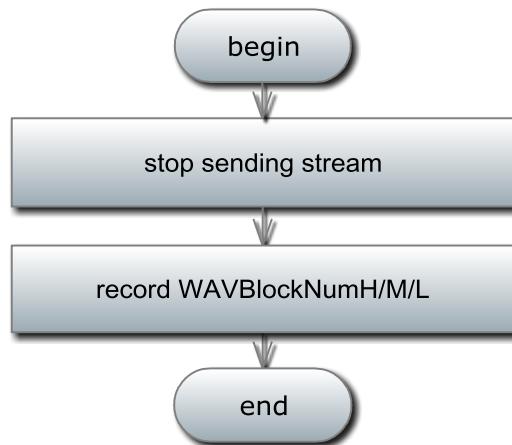


Figure 8-12 record the break-point for WAV stream

The following steps must be applied for break-point play:

1. Same as the 1<sup>st</sup> step of Decoding Control Flow Chart.
2. Same as the 2<sup>nd</sup> step of Decoding Control Flow Chart.
3. Configure AUIP:
  - (1)~(15) Same as the 3<sup>rd</sup> step of MP3 Break-Point Play Control Flow Chart
  - (16) Initialize block number register.
  - (17) nop operation
  - (15)~(29) Same as the 3<sup>rd</sup> step of MP3 Break-Point Play Control Flow Chart
4. The control of AUIP:
  - (1) Send the WAV header to AUIP input FIFO0.
  - (2) Fill the stream at the specified block.

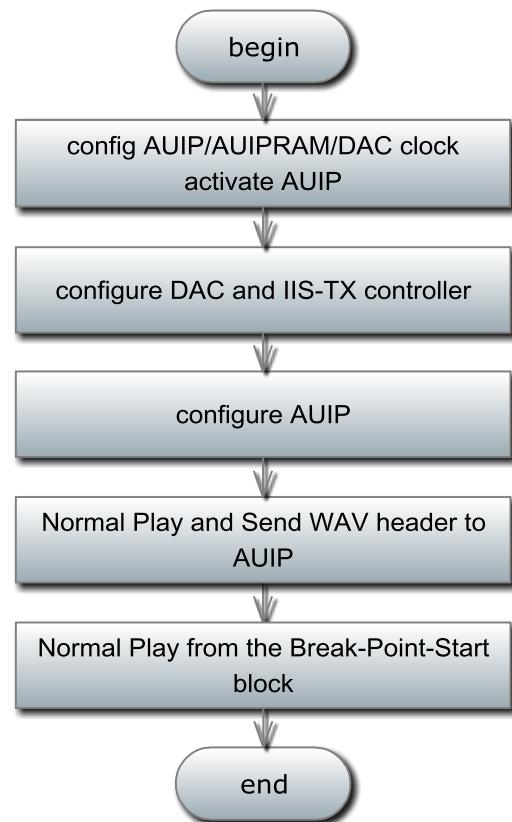


Figure 8-13 WAV break-point control flow chart

### 8.2.5.6 Fast Forward Control Flow Chart

The following steps must be applied for break-point play:

1. Same as the 1<sup>st</sup> step of Decoding Control Flow Chart.
2. Same as the 2<sup>nd</sup> step of Decoding Control Flow Chart.
3. Enable FFEn bit and disable REVEn bit of MP3Ctl or WMACtl register.
4. Send the stream in sequence.

There is no fast forward/fast reverse control bit for WAV decoding, as bit-rate/sample-rate is a constant. Fast Forward or Fast reverse can be performed by software if the stream is WAV format.

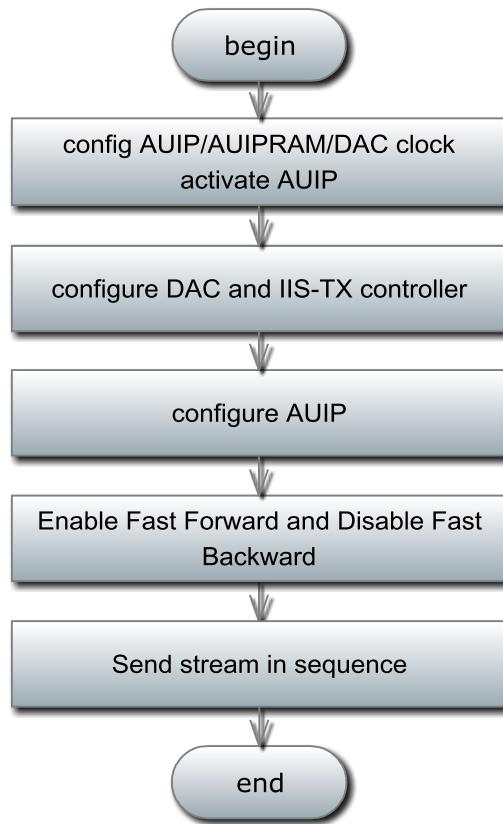


Figure 8-14 MP3/WMA fast forward control flow chart

### 8.2.5.7 MP3 Fast Reverse Control Flow Chart

The following steps must be applied for break-point play:

1. Same as the 1<sup>st</sup> step of Decoding Control Flow Chart.
2. Same as the 2<sup>nd</sup> step of Decoding Control Flow Chart.
3. Disable FFEn bit and Enable REVEn bit of MP3Ctl or WMACtl register.
4. Send the stream in reverse order.

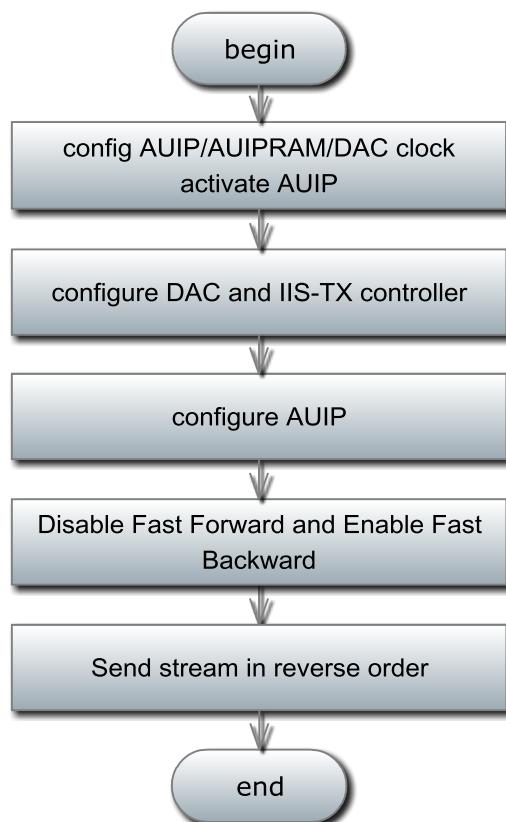


Figure 8-15 MP3 fast reverse control flow chart

### 8.2.5.8 WMA Fast Reverse Control Flow Chart

The following steps must be applied for break-point play:

1. Same as the 1<sup>st</sup> step of Decoding Control Flow Chart.
2. Same as the 2<sup>nd</sup> step of Decoding Control Flow Chart.
3. Enable FFEn bit and Disable REVEn bit of WMACtl register and set REV\_SEL bit.
4. Send perverse packet in reverse order while the stream in a packet is in sequence. The figure below presents the difference of packet order between normal-play/fast-forward and fast-reverse.

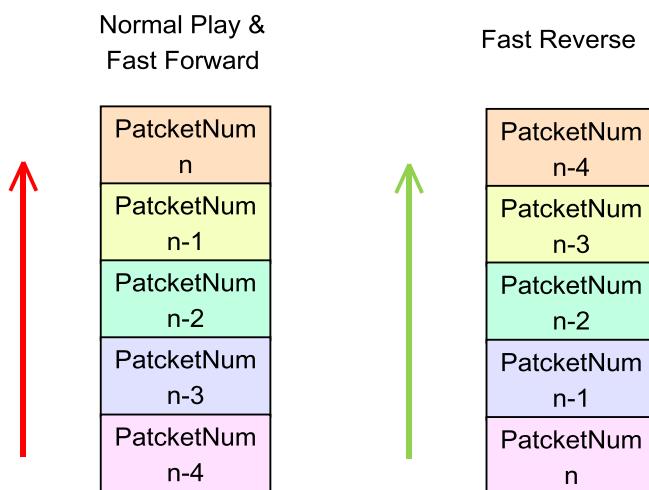


Figure 8-16 Packet order difference between Normal-play/Fast-Forward and Fast Reverse

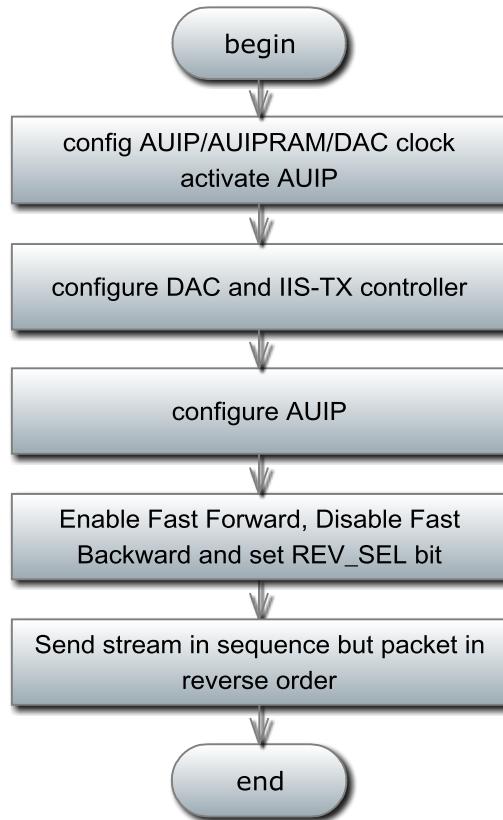


Figure 8-17 WMA fast reverse control flow chart

### 8.2.6 AUIP Register List

| Index | Mnemonic         | Description                            | BANK |
|-------|------------------|--|------|
| 0x90  | AuCodecCtl       | Audio Codec Control Register           | 0x04 |
| 0x91  | AuDebugLength    | Audio Codec Debug Length Register      | 0x04 |
| 0x92  | AuCodecDebug     | Audio Codec Debug Register             | 0x04 |
| 0x93  | AuCodecFIFOCtl   | Audio Codec FIFO control Register      | 0x04 |
| 0x94  | AuCodecFIFOData  | Audio Codec FIFO Data Register         | 0x04 |
| 0x95  | AuCodecDebug2    | Audio Codec Debug Register 2           | 0x04 |
| 0x97  | AuCodecDecStateH | Decoding State Register High byte      | 0x04 |
| 0x98  | AuCodecDecStateL | Decoding State Register Low byte       | 0x04 |
| 0xa9  | MP3IE            | MP3 Decoder Interrupt Enable Register  | 0x04 |
| 0xaa  | MP3IP            | MP3 Decoder Interrupt Pending Register | 0x04 |
| 0xab  | MP3Ctl           | MP3 Decoder Control Register           | 0x04 |
| 0xac  | MP3HeaderInfo    | Header information register            | 0x04 |
| 0xad  | MP3BitRateH      | High byte of bit rate index register   | 0x04 |
| 0xae  | MP3BitRateL      | Low byte of bit rate index register    | 0x04 |
| 0xaf  | MP3TtimeH        | Total time hours register              | 0x04 |
| 0xb0  | MP3TtimeM        | Total time minutes register            | 0x04 |

|      |                 |  |      |
|------|-----------------|--|------|
| 0xb1 | MP3TtimeS       | Total time seconds register                    | 0x04 |
| 0xb2 | MP3CtimeH       | Current time hours register                    | 0x04 |
| 0xb3 | MP3CtimeM       | Current time minutes register                  | 0x04 |
| 0xb4 | MP3CtimeS       | Current time seconds register                  | 0x04 |
| 0xb5 | MP3FrameNumH    | High byte of Frame Number register             | 0x04 |
| 0xb6 | MP3FrameNumM    | Middle byte of Frame Number register           | 0x04 |
| 0xb7 | MP3FrameNumL    | Low byte of Frame Number register              | 0x04 |
| 0xb8 | MP3FileLen3     | Byte 3 of File Length register                 | 0x04 |
| 0xb9 | MP3FileLen2     | Byte 2 of File Length register                 | 0x04 |
| 0xba | MP3FileLen1     | Byte 1 of File Length register                 | 0x04 |
| 0xbb | MP3HeaderSynCtl | MP3 header synchronization control register    | 0x04 |
| 0xbc | SynHeaderData1  | MP3 header synchronization Data register 1     | 0x04 |
| 0xbd | SynHeaderData2  | MP3 header synchronization Data register 2     | 0x04 |
| 0xa9 | WMAIE           | WMA Decoder Interrupt Enable Register          | 0x04 |
| 0xaa | WMAIP           | WMA Decoder Interrupt Pending Register         | 0x04 |
| 0xab | WMACtl          | WMA Decoder Control Register                   | 0x04 |
| 0xac | WMAHeaderInfo   | Header information register                    | 0x04 |
| 0xad | WMABitRateH     | High byte of bit rate index register           | 0x04 |
| 0xae | WMABitRateL     | Low byte of bit rate index register            | 0x04 |
| 0xaf | WMATTimeH       | Total time hours register                      | 0x04 |
| 0xb0 | WMATTimeM       | Total time minutes register                    | 0x04 |
| 0xb1 | WMATTimeS       | Total time seconds register                    | 0x04 |
| 0xb2 | WMACTimeH       | Current time hours register                    | 0x04 |
| 0xb3 | WMACTimeM       | Current time minutes register                  | 0x04 |
| 0xb4 | WMACTimeS       | Current time seconds register                  | 0x04 |
| 0xb5 | WMAPackNumH     | High byte of Packet Number register            | 0x04 |
| 0xb6 | WMAPackNumM     | Middle byte of Packet Number register          | 0x04 |
| 0xb7 | WMAPackNumL     | Low byte of Packet Number register             | 0x04 |
| 0xb8 | WMAPackSizeH    | High byte of Packet Size register              | 0x04 |
| 0xb9 | WMAPackSizeM    | Middle byte of Packet Size register            | 0x04 |
| 0xba | WMAPackSizeL    | Low byte of Packet Size register               | 0x04 |
| 0xbb | VirtualAddrH    | High byte of virtual address of current frame  | 0x04 |
| 0xbc | VirtualAddrL    | Low byte of virtual address of current frame   | 0x04 |
| 0xbd | PhysicalAddrH   | High byte of physical address of current frame | 0x04 |
| 0xbf | PhysicalAddrL   | Low byte of physical address of current frame  | 0x04 |
| 0xc1 | DiscardLen      | Discard length of current subframe             | 0x04 |
| 0xa9 | WAVIE           | WAV Decoder Interrupt Enable Register          | 0x04 |
| 0xaa | WAVIP           | WAV Decoder Interrupt Pending Register         | 0x04 |
| 0xab | WAVCtl          | WAV Decoder Control Register                   | 0x04 |
| 0xac | WAVHeaderInfo1  | Header information register 1                  | 0x04 |
| 0xad | WAVHeaderInfo2  | Header information register 2                  | 0x04 |
| 0xaf | WAVTTTimeH      | Total time hours register                      | 0x04 |
| 0xb0 | WAVTTTimeM      | Total time minutes register                    | 0x04 |

|      |               |   |      |
|------|---------------|---|------|
| 0xb1 | WAVTTimeS     | Total time seconds register                 | 0x04 |
| 0xb2 | WAVCTimeH     | Current time hours register                 | 0x04 |
| 0xb3 | WAVCTimeM     | Current time minutes register               | 0x04 |
| 0xb4 | WAVCTimeS     | Current time seconds register               | 0x04 |
| 0xb5 | WAVBlockNumH  | High byte of Block Number register          | 0x04 |
| 0xb6 | WAVBlockNumM  | Middle byte of Block Number register        | 0x04 |
| 0xb7 | WAVBlockNumL  | Low byte of Block Number register           | 0x04 |
| 0xab | WAVEncCtl     | WAV Encoder Control Register                | 0x04 |
| 0xac | WAVEncInfo    | WAV Encoder Information Register            | 0x04 |
| 0xab | MP3EncCtl     | MP3 Encoder Control Register                | 0x04 |
| 0xac | MP3EncInfo    | MP3 Encoder Information Register            | 0x04 |
| 0xa9 | SoftIE        | Software Decoder Interrupt Enable Register  | 0x04 |
| 0xaa | SoftIP        | Software Decoder Interrupt Pending Register | 0x04 |
| 0xab | SoftCtl       | Software Decoder Control Register           | 0x04 |
| 0xb8 | FrameLen1     | Frame Length Register 1                     | 0x04 |
| 0xb9 | FrameLen0     | Frame Length Register 0                     | 0x04 |
| 0xc2 | BEPCtl1       | Post Processor Control Register 1           | 0x04 |
| 0xc3 | BEPCtl2       | Post Processor Control Register 2           | 0x04 |
| 0xc4 | BEPCtl3       | Post Processor Control Register 3           | 0x04 |
| 0xc5 | BEPCtl4       | Post Processor Control Register 4           | 0x04 |
| 0xc6 | BEPCtl5       | Post Processor Control Register 5           | 0x04 |
| 0xc7 | GlobalGainH   | Global Gain High Byte Register              | 0x04 |
| 0xc8 | GlobalGainM   | Global Gain Middle Byte Register            | 0x04 |
| 0xc9 | GlobalGainL   | Global Gain Low Byte Register               | 0x04 |
| 0xca | CurrentEnergy | Current Energy Register                     | 0x04 |
| 0xcb | KaraokeCtl    | Karaoke Control Register                    | 0x04 |
| 0xcc | LCHGain       | Left channel gain                           | 0x04 |
| 0xcd | RCHGain       | Right channel gain                          | 0x04 |
| 0xce | LPFGain       | Low pass filter gain                        | 0x04 |
| 0xcf | HPFGain       | High pass filter gain                       | 0x04 |
| 0xcc | SRSCtl0       | SRS Control Register 0                      | 0x04 |
| 0xcd | SRSCtl1       | SRS Control Register 1                      | 0x04 |
| 0xce | SRSCtl2       | SRS Control Register 2                      | 0x04 |

## 8.2.7 Codec Top Register Description

### 8.2.7.1 AuCodecCtl

AuCodecCtl (Audio Codec Control Register, SFR Address 0x90, SFR bank 04)

| Bit Number | Bit Mnemonic | Function | Access | Reset |
|------------|--------------|----------|--------|-------|
|            |              |          |        |       |

|     |              |  |     |     |
|-----|--------------|--|-----|-----|
| 7   | Reset        | 0 : Audio Codec reset except AuCodecCtl register ;<br>1: Normal operation.   | R/W | 0   |
| 6:4 | Mode         | Audio Codec Mode<br>000 : MP3, MP2, MP1 decoder ;<br>001: WMA decoder;<br>010: WAV decoder;<br>011: WAV encoder;<br>100: reserved;<br>101: MP3 encoder;<br>110: reserved;<br>111: Post processor | R/W | 000 |
| 3   | EnableAllClk | 0: dynamic clock gating enable<br>1: all clocks are enable   | R/W | 0   |
| 2:1 | Reserved     | Be read as 2 zeros   | -   | -   |
| 0   | ResetFIFO1   | Input FIFO1 State Reset<br>0: Input FIFO1 State Reset;<br>1: Normal operation;   | R/W | 0   |

2041、 The results of mp3/wma decoding each frame is stored in muram1 as follows:

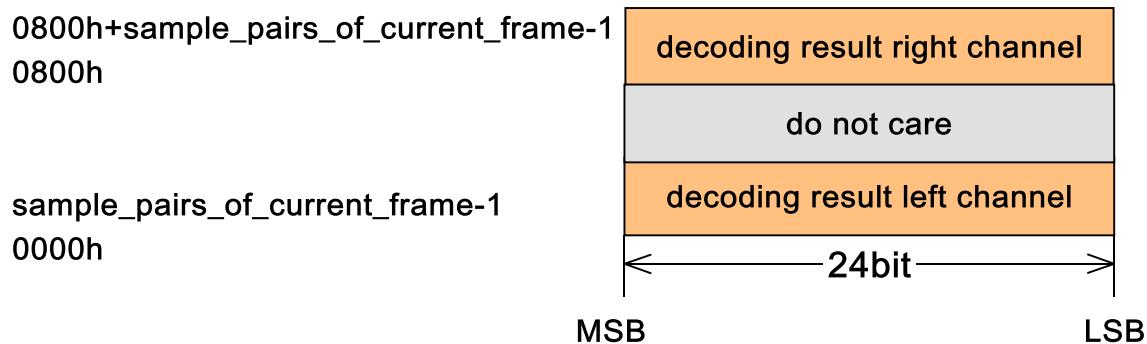


Figure 8-15 the results location of left and right channel of each frame for mp3/wma decoding

2042、 The results of wav decoding each frame are stored in muram1 as follows:

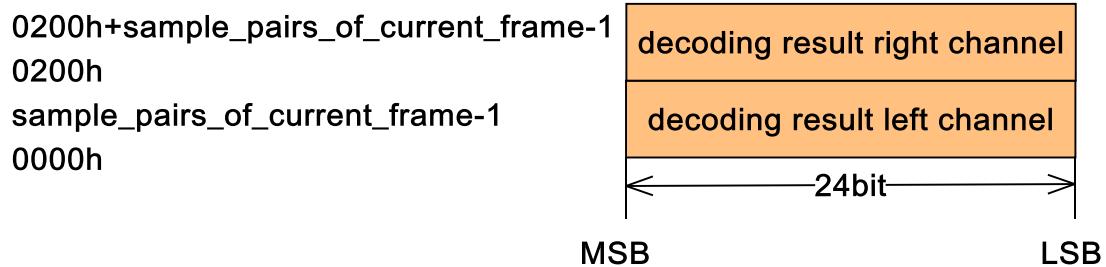


Figure 8-16 the results location of left and right channel of each frame for wav decoding

2043、 the relation between **AuDebugLength** register and sample\_pair\_of\_current\_frame is:  
 $\text{sample\_pairs\_of\_current\_frame} = \text{AuDebugLength}/2$

### 8.2.7.2 AuDebugLength

**AuDebugLength (Audio Codec Debug Length Register, SFR Address 0x91, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | PcmLenL      | Low byte of length of Sub-Frame or Sub-block of the Codec decoded, PCM Length [7:0]. | R      | 00h   |

### 8.2.7.3 AuCodecDebug

**AuCodecDebug (Audio Codec Debug Register, SFR Address 0x92, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | DebugMode    | 0: Normal Mode<br>1: Debug Mode   | R/W    | 0     |
| 6          | DebugState   | Codec will pause decoding the audio stream after one frame is decoded and this bit will be set , then CPU/DMA can move pcm data from address 0 ~ (PcmLength-1) to any other memory. After data is move done, write 1 to this bit to start decoding next frame, and it will keep cleared before complete the decoding of the next frame. | R/W    | 0     |
| 5          | Reserved     | Be read as 1 zero.  | -      | -     |
| 4:0        | PcmLenH      | High byte of length of Sub-Frame or Sub-block of the Codec decoded, PCM Length [12:8]. <sup>(1)(2)(3)(4)(5)(6)</sup>  | R      | 00000 |

Note:

- (1) The maximum frame length of WMA decoding output is 0x800, the bit12 of PcmLen register is to indicate the lower half and higher half of the output frame.
- (2) The maximum sub-block length of WAV decoding output is 0x200, the bit12 and bit11 of PcmLen register is always ‘0’ in this mode.(sub-block is defined by the decoder to use less memory)
- (3) The maximum sub-block length of MP3 decoding output is 0x240(one granule), the bit12 of PcmLen register is always ‘0’ in this mode.
- (4) The encoded IMA-ADPCM block size is 0x800 bytes for dual channel and 0x400 bytes for mono channel.  
The value of PcmLen register is invalid.
- (5) The encoded PCM block size is 0x800 bytes for dual channel and 0x400 bytes for mono channel. The value of PcmLen register is invalid.
- (6) The encoded MP3 frame size is 192 bytes for 64kbps, 384 bytes for 128kbps and 576bytes for 192kbps.  
The value of PcmLen register is invalid.
- (7) This bit is valid only in decoding mode.

### 8.2.7.4 AuCodecFIFOCtl

**AuCodecFIFOCtl (Audio Codec FIFO control Register, SFR Address 0x93, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | Empty        | FIFO Empty Indicator<br>0: The FIFO is not empty;<br>1: The FIFO is empty.  | R      | 1     |
| 6          | AlmostEmpty  | FIFO Almost Empty Indicator<br>0: The FIFO is not almost empty;<br>1: The FIFO is almost empty.<br><br>When the number of remain data in the FIFO is less than DRQlevel pointed to, the bit of AlmostEmpty will be set. When the number of remain data is more than DRQlevel + 1, the bit of AlmostEmpty will be cleared. | R      | 1     |
| 5          | OverFlow     | FIFO Overflow<br>0: FIFO has not been overflowed;<br>1: FIFO has been overflowed.<br><br>Write 1 will cleared this bit.   | R/W    | 0     |
| 4:1        | DRQLevel     | When the number of remain data in the FIFO is less than DRQlevel DRQ will be set; when it is more than 14, DRQ will be cleared. The value of DRQlevel must greater than 1 and less than 7.  | R/W    | 0010  |
| 0          | ResetFIFO    | Input FIFO0 State Reset<br>0: Input FIFO0 State Reset, Including Empty, AlmostEmpty, OverFlow flag;<br>1: Normal operation;   | R/W    | 0     |

Note: The depth of the input FIFO of Audio CODEC is 16.

### 8.2.7.5 AuCodecFIFOData

**AuCodecFIFOData (Audio Codec FIFO Data Register, SFR Address 0x94, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function               | Access | Reset |
|------------|--------------|------------------------|--------|-------|
| 7:0        | FIFOData     | Data written into FIFO | W      | 00h   |

### 8.2.7.6 AuCodecDebug2

**AuCodecDebug2 (Audio Codec Debug Register 2, SFR Address 0x95, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:6        | Reserved     | Be read as “0”   | -      | -     |
| 5          | Halt2IE      | 0: The “enter-HALT2-state” interrupt is disable.<br>1: The “enter-HALT2-state” interrupt is enable.  | R/W    | 0     |
| 4          | Halt2IP      | 0: The interrupt pending is cleared if there is no transition from other state to “HALT2” state.<br>1: The interrupt pending is set if there is transition from other state to “HALT2” state.<br>Writing ‘1’ to this bit will force audio codec to quite “HALT2” state and enter “BACKEND PROCESS” state | R/W    | 0     |
| 3          | PowerSave    | The clock of muarm2 will be set to “DC” while first half of audio frame is in pcmbuf.<br>0: disable power saving mode<br>1: enable power saving mode   | R/W    | 0     |
| 2          | DEBUG_EN     | Audio Codec Debug Enable:<br>0: No Debug<br>1: Debug   | R/W    | 0     |
| 1          | DebugSel     | Audio Codec Debug Select:<br>0: Audio decoder Debug<br>1: Post processor Debug   | R/W    | 0     |
| 0          | DecEn        | Enable the decoding of next state<br>0: disable<br>1: enable<br>This bit will be automatically cleared by the decoder. Writing “0” takes no effect.  | R/W    | 0     |

### 8.2.7.7 AuCodecDecStateH

**AuCodecDecStateH (Audio Codec Decoding State Register High byte, SFR Address 0x97, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | StateH       | The bit 15 to 8 of Audio Codec Decoding State Register | R      | 00    |

### 8.2.7.8 AuCodecDecStateL

**AuCodecDecStateL (Audio Codec Decoding State Register Low byte, SFR Address 0x98, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | StateL       | The bit 7 to 0 of Audio Codec Decoding State Register | R      | 00    |

## 8.2.8 Mp3 Decoder Register Description

### 8.2.8.1 MP3IE

**MP3IE (MP3 Decoder Interrupt Enable Register, SFR Address 0xa9, SFR bank 04)**

| Bit Number | Bit Mnemonic  | Function  | Access | Reset |
|------------|---------------|---|--------|-------|
| 7          | LackPcmEn     | Lack PCM interrupt enable<br>0: Disable<br>1: Enable  | R/W    | 0     |
| 6          | EnergyEn      | 0: Disable new energy ready interrupt;<br>1: Enable new energy ready interrupt.<br>When both EnergyEn and EnergyIP are set, Post processor will send interrupt signal.                            | R/W    | 0     |
| 5          | WordIntEn     | 0: Disable sentence detection for interrupt control;<br>1: Enable sentence detection for interrupt control.<br>When both WordIntEn and WordIP are set, Post processor will send interrupt signal. | R/W    | 0     |
| 4          | InfoChangeEn2 | 0: Disable sampling rate change detection for interrupt control;<br>1: Enable sampling rate change detection for interrupt control.   | R/W    | 0     |
| 3          | InfoChangeEn1 | 0: Disable layer change or channel number change detection for interrupt control;<br>1: Enable layer change or channel number change detection for interrupt control.                             | R/W    | 0     |
| 2          | CRCErrIntEn   | 0: Disable CRC error occur interrupt;   | R/W    | 0     |

|   |              |  |     |   |
|---|--------------|--|-----|---|
|   |              | 1: Enable CRC error occur interrupt.<br>0: Disable information ready interrupt;<br>1: Enable information ready interrupt.  |     |   |
| 1 | InfoRdyIntEn | 0: Disable new PCM ready interrupt;<br>1: Enable new PCM ready interrupt.<br><br>When both PCMRdyIntEn and PCMRdyIP are set, MP3 decoder will send interrupt signal. | R/W | 0 |
| 0 | PCMRdyIntEn  | 0: Disable new PCM ready interrupt;<br>1: Enable new PCM ready interrupt.<br><br>When both PCMRdyIntEn and PCMRdyIP are set, MP3 decoder will send interrupt signal. | R/W | 0 |

### 8.2.8.2 MP3IP

**MP3IP (MP3 Decoder Interrupt Pending Register, SFR Address 0xaaa, SFR bank 04)**

| Bit Number | Bit Mnemonic  | Function   | Access | Reset |
|------------|---------------|--|--------|-------|
| 7          | LackIP        | Lacking PCM interrupt pending<br>0: Having enough PCM to play;<br>1: Having not enough PCM to play.<br><br>When decoding is too slow or file finished, LackIP will be set. Write 1 will clear this bit.                            | R/W    | 0     |
| 6          | EnergyIP      | New energy ready interrupt pending<br>0: No New energy ready for accessing;<br>1: New energy ready for accessing.<br><br>EnergyIP will be set for every 2048 samples per channel has been decoded.<br>Write 1 will clear this bit. | R/W    | 0     |
| 5          | WordIP        | Interrupt pending for detecting a sentence.<br><br>If the energy less than EnergyIn for SilenceT times, WordIP will be set. Write 1 will clear this bit.   | R/W    | 0     |
| 4          | InfoChangeIP2 | Interrupt pending for sampling rate change.<br>0: Sampling rate has not been changed;<br>1: Sampling rate has been changed;  | R/W    | 0     |
| 3          | InfoChangeIP1 | Interrupt pending for layer change or channel number change.<br>0: Layer or channel number has not been changed;<br>1: Layer or channel number has been changed;   | R/W    | 0     |
| 2          | CRCerrIP      | CRC error interrupt pending<br>0: No CRC error occurred;   | R/W    | 0     |

|   |           |   |     |   |
|---|-----------|---|-----|---|
|   |           | 1: CRC error occurred.<br>Write 1 will clear this bit.  |     |   |
| 1 | InfoRdyIP | Information ready interrupt pending<br>0: No new information is ready;<br>1: New information is ready;<br>InfoRdyIP asserts once every nowtime updated or bitrate updated. Write 1 will clear this bit. | R/W | 0 |
| 0 | PCMRdyIP  | PCM ready interrupt pending<br>0: PCM is ready for DAC;<br>1: New PCM is ready for DAC.<br>Write 1 will clear this bit.   | R/W | 0 |

### 8.2.8.3 MP3Ctl

**MP3Ctl (MP3 Decoder Control Register, SFR Address 0xab, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | NewFile      | Write 1 this bit to initial for a new file decoder, and it will be cleared by hardware itself.  | R/W    | 0     |
| 6          | En           | MP3 Decoder Enable(play/pause)*<br>0: Pause MP3 Decoder<br>1: Enable MP3 Decoder  | R/W    | 0     |
| 5          | FFEn         | Fast Forward Enable<br>0: Normal play or Reverse;<br>1: Fast Forward;   | R/W    | 0     |
| 4          | REVEN        | Reversed Enable<br>0: Normal play or FF;<br>1: Reverse(Fast Backward);  | R/W    | 0     |
| 3          | CRCSkipEn    | CRC Skip Enable<br>0: Disable processing of the frame which contains CRC error, frame is skipped;<br>1: Enable processing of the frame which contains CRC error, frame is played whatever containing CRC error. | R/W    | 0     |
| 2          | Reserved     | Be read as "0"  | -      | -     |
| 1          | BKPlayEn     | Breakpoint play Enable<br>0: Disable Breakpoint play<br>1: Enable Breakpoint play   | R/W    | 0     |
| 0          | Reserved     | Be read as "0"  | -      | -     |

\* To start a new file decoding the “NewFile” bit should be set before “En” bit set. Set the “En” bit can trigger

fade in if “FadeIn” bit is “1”. Clear the “En” bit can trigger fade out if “FadeOut” bit is “0”. The decoding is stop until fade out finished.

### 8.2.8.4 MP3HeaderInfo

**MP3HeaderInfo (Header information register, SFR Address 0xac, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | SampFrq      | Sample frequency (kHz)<br>0000: 44.1;<br>0001: 48;<br>0010: 32;<br>0100: 22.05;<br>0101: 24;<br>0110: 16;<br>1000: 11.025;<br>1001: 12;<br>1010: 8;<br>Others: reserved;<br>If bit 7 has been set, the file version is MPEG2.5; if bit 6 has been set, the file version is MPEG2.0; if both bit 7 and bit 6 have been cleared, the file version is MPEG1.0. | R      | 1000  |
| 3:2        | Layer        | 00: reserved;<br>01: Layer III;<br>10: Layer II;<br>11: Layer I.  | R      | 00    |
| 1          | Errprotect   | CRC protection indicator<br>0: Have no CRC protection in the file;<br>1: Have CRC protection in the file.   | R      | 0     |
| 0          | AudioMode    | Audio mode<br>0: dual channel;<br>1: single channel   | R      | 0     |

### 8.2.8.5 MP3BitRateH

**MP3BitRateH (High byte of bit rate index register, SFR Address 0xad, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                     | Access | Reset |
|------------|--------------|------------------------------|--------|-------|
| 7:4        | BitRateIdx3  | Bit [15:12] of bitrate index | R      | xxxx  |
| 3:0        | BitRateIdx2  | Bit [11:8] of bitrate index  | R      | xxxx  |

### 8.2.8.6 MP3BitRateL

**MP3BitRateL (Low byte of bit rate index register, SFR Address 0xae, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                   | Access | Reset |
|------------|--------------|----------------------------|--------|-------|
| 7:4        | BitRateIdx1  | Bit [7:4] bitrate index    | R      | xxxx  |
| 3:0        | BitRateIdx0  | Bit [3:0] of bitrate index | R      | xxxx  |

Note: BitRate = BitRateIdx[15:12]\*100 + BitRateIdx[11:8]\*10 + BitRateIdx[7:4] + BitRateIdx[3:0]\*0.1 (kbps)

### 8.2.8.7 MP3TtimeH

**MP3TtimeH (Total time hours register, SFR Address 0xaf, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset  |
|------------|--------------|--|--------|--------|
| 7          | XingTag      | Hit Xing Tag<br>0: Has not hit Xing Tag;<br>1: Has hit Xing Tag in current file. | R      | 0      |
| 6          | Reserved     | Be read as “0”   | -      | -      |
| 5:0        | TtimeH       | Total time hours [0-59]  | R      | xxxxxx |

### 8.2.8.8 MP3TtimeM

**MP3TtimeM (Total time minutes register, SFR Address 0xb0, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                  | Access | Reset  |
|------------|--------------|---------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”           | --     | --     |
| 5:0        | TtimeM       | Total time minutes [0-59] | R      | xxxxxx |

### 8.2.8.9 MP3TtimeS

**MP3TtimeS (Total time seconds register, SFR Address 0xb1, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                 | Access | Reset  |
|------------|--------------|--------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”          | --     | --     |
| 5:0        | TtimeS       | Total time second [0-59] | R      | xxxxxx |

### 8.2.8.10 MP3CtimeH

**MP3CtimeH (Current time hours register, SFR Address 0xb2, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                  | Access | Reset  |
|------------|--------------|---------------------------|--------|--------|
| 31:6       | Reserved     | Be read as 26-zeros       | -      | -      |
| 5:0        | CtimeH       | Current time hours [0-59] | R      | xxxxxx |

### 8.2.8.11 MP3CtimeM

MP3CtimeM (Current time minutes register, SFR Address 0xb3, SFR bank 04)

| Bit Number | Bit Mnemonic | Function                    | Access | Reset  |
|------------|--------------|-----------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”             | --     | --     |
| 5:0        | CtimeM       | Current time minutes [0-59] | R      | xxxxxx |

### 8.2.8.12 MP3CtimeS

MP3CtimeS (Current time seconds register, SFR Address 0xb4, SFR bank 04)

| Bit Number | Bit Mnemonic | Function                   | Access | Reset  |
|------------|--------------|----------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”            | --     | --     |
| 5:0        | CtimeS       | Current time second [0-59] | R      | xxxxxx |

### 8.2.8.13 MP3FrameNumH

MP3FrameNumH (High byte of Frame Number register, SFR Address 0xb5, SFR bank 04)

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | FrameNumH    | High byte of Frame Number register<br>BlockNum [23:16] | R/W    | x     |

### 8.2.8.14 MP3FrameNumM

MP3FrameNumM (Middle byte of Frame Number register, SFR Address 0xb6, SFR bank 04)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | FrameNumM    | Middle byte of Frame Number register<br>BlockNum [15:8] | R/W    | x     |

### 8.2.8.15 MP3FrameNumL

**MP3FrameNumL (Low byte of Frame Number register, SFR Address 0xb7, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | FrameNumL    | Low byte of Frame Number register<br>BlockNum [7:0] | R/W    | x     |

Note: normal play and write MP3FrameNum will refresh the current time register.

### 8.2.8.16 MP3FileLen3

**MP3FileLen3 (Byte 3 of File Length register, SFR Address 0xb8, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function           | Access | Reset |
|------------|--------------|--------------------|--------|-------|
| 7:0        | FileLength3  | FileLength [31:24] | R/W    | -     |

### 8.2.8.17 MP3FileLen2

**MP3FileLen2 (Byte 2 of File Length register, SFR Address 0xb9, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function           | Access | Reset |
|------------|--------------|--------------------|--------|-------|
| 7:0        | FileLength2  | FileLength [23:16] | R/W    | -     |

### 8.2.8.18 MP3FileLen1

**MP3FileLen1 (Byte 1 of File Length register, SFR Address 0xba, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:0        | FileLength1  | FileLength [15:8] | R/W    | -     |

Note: The current time can be set by writing most 3 high bytes of file length to the register of MP3FileLen3, MP3FileLen2 and MP3FileLen1.

### 8.2.8.19 MP3HeaderSynCtl

**MP3HeaderSynCtl (MP3 header synchronization control register, SFR Address 0xbb, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | IDEn         | 0:ID bit for header synchronization<br>1:disable | R/W    | 1     |

|   |                 |   |     |   |
|---|-----------------|---|-----|---|
|   |                 | 1:ID bit header synchronization enable  |     |   |
| 6 | LayerEn         | 0: Layer information bits for header synchronization disable<br>1: Layer information bits for header synchronization enable   | R/W | 1 |
| 5 | ProtectionBitEn | 0: Protection bit for header synchronization disable<br>1: Protection bit header synchronization enable                       | R/W | 0 |
| 4 | SampFreqEn      | 0: Sampling frequency bits for header synchronization disable<br>1: Sampling frequency bits for header synchronization enable | R/W | 1 |
| 3 | ModeEn          | 0:Mode bits for header synchronization disable<br>1:Mode bits for header synchronization enable                               | R/W | 1 |
| 2 | CopyrightEn     | 0: Copyright bit for header synchronization disable<br>1: Copyright bit for header synchronization enable                     | R/W | 1 |
| 1 | OriginalEn      | 0: Original/copy bit for header synchronization disable<br>1: Original/copy bit for header synchronization enable             | R/W | 0 |
| 0 | EmphasisEn      | 0: Emphasis bit for header synchronization disable<br>1: Emphasis bit for header synchronization enable                       | R/W | 1 |

### 8.2.8.20 SynHeaderData1

**SynHeaderData1 (MP3 header synchronization Data register 1, SFR Address 0xbc, SFR bank 04)**

| Bit Number | Bit Mnemonic  | Function  | Access | Reset |
|------------|---------------|---|--------|-------|
| 7          | copyright     | copyright bit for header synchronization<br>0: no copyright<br>1: copyright protected | R/W    | 0     |
| 6          | original_copy | Copy or original information bit for header synchronization<br>0: copy<br>1: original | R/W    | 0     |
| 5:4        | emphasis      | Emphasis bits for header synchronization  | R/W    | 00    |

|     |                 |  |   |    |
|-----|-----------------|--|---|----|
|     |                 | 00: none<br>01: 50/15 microseconds<br>10: reserved<br>11: CCITT J.17   |   |    |
| 3   | Copyright_r     | copyright bit for header synchronization<br>(read from stream)<br>0: no copyright<br>1: copyright protected                            | R | 0  |
| 2   | original_copy_r | Copy or original information bit for header synchronization (read from stream)<br>0: copy<br>1: original                               | R | 0  |
| 1:0 | Emphasis_r      | Emphasis bits for header synchronization<br>(read from stream)<br>00: none<br>01: 50/15 microseconds<br>10: reserved<br>11: CCITT J.17 | R | 00 |

### 8.2.8.21 SynHeaderData2

**SynHeaderData2 (MP3 header synchronization Data register 2, SFR Address 0xbd, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | SampFrq      | Sample frequency (kHz)<br>0000: 44.1;<br>0001: 48;<br>0010: 32;<br>0100: 22.05;<br>0101: 24;<br>0110: 16;<br>1000: 11.025;<br>1001: 12;<br>1010: 8;<br>Others: reserved;<br>If bit 7 has been set, the file version is MPEG2.5; if bit 6 has been set, the file version is MPEG2.0; if both bit 7 and bit 6 have been cleared, the file version is MPEG1.0. | R/W    | 0000  |
| 3:2        | Layer        | 00: reserved;<br>01: Layer III;   | R/W    | 00    |

|   |            |   |     |   |
|---|------------|---|-----|---|
|   |            | 10: Layer II;<br>11: Layer I.   |     |   |
| 1 | Errprotect | CRC protection indicator<br>0: Have no CRC protection in the file;<br>1: Have CRC protection in the file. | R/W | 0 |
| 0 | AudioMode  | Audio mode<br>0: dual channel;<br>1: single channel   | R/W | 0 |

## 8.2.9 WMA Decoder Register Description

### 8.2.9.1 WMAIE

**WMAIE (WMA Decoder Interrupt Enable Register, SFR Address 0xa9, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | LackIntEn    | Lacking PCM interrupt enable<br>0: Disable;<br>1: Enable.<br><br>When both LackIntEn and LackIP are set,<br>Post processor will send interrupt signal.  | R/W    | 0     |
| 6          | EnergyIntEn  | 0: Disable new energy ready interrupt;<br>1: Enable new energy ready interrupt.<br><br>When both EnergyIntEn and EnergyIP<br>are set, Post processor will send interrupt<br>signal.   | R/W    | 0     |
| 5          | WordIntEn    | 0: Disable sentence detected interrupt;<br>1: Enable sentence detected interrupt.<br><br>When both WordIntEn and WordIP are<br>set, Post processor will send interrupt<br>signal.   | R/W    | 0     |
| 4          | Reserved     | Be read as "0"  | --     | --    |
| 3          | DRMInfoIntEn | 0: Disable DRM information checking<br>interrupt<br>1: Enable DRM information checking<br>interrupt   | R/W    | 0     |
| 2          | ErrIntEn     | 0: Disable error occur interrupt;<br>1: Enable error occur interrupt.<br><br>These errors include legal wma file or<br>opaque data present. WMA decoder will<br>send interrupt signal while both ErrIntEn<br>and ErrIP are set. | R/W    | 0     |

|   |              |  |     |   |
|---|--------------|--|-----|---|
| 1 | InfoRdyIntEn | 0: Disable information ready interrupt;<br>1: Enable information ready interrupt.<br><br>When both InfoIntEn and InfoIP are set, WMA decoder will send interrupt signal. | R/W | 0 |
| 0 | PCMRdyIntEn  | 0: Disable new PCM ready interrupt;<br>1: Enable new PCM ready interrupt.<br><br>When both PCMRdyIntEn and PCMRdyIP are set, WMA decoder will send interrupt signal.     | R/W | 0 |

### 8.2.9.2 WMAIP

**WMAIP (WMA Decoder Interrupt Pending Register, SFR Address 0xaa, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | LackIP       | Lacking PCM interrupt pending<br>0: Having enough PCM to play;<br>1: Having not enough PCM to play.<br><br>When decoding is slow or file finished, LackIP will be set. Write 1 will clear this bit.  | R/W    | 0     |
| 6          | EnergyIP     | New energy ready interrupt pending<br>0: No New energy ready for accessing;<br>1: New energy ready for accessing.<br><br>EnergyIP will be set for every 2048 samples per channel has been decoded.<br>Write 1 will clear this bit.                   | R/W    | 0     |
| 5          | WordIP       | Interrupt pending for detecting a sentence.<br><br>If the energy less than EnergyIn for SilenceT times, WordIP will be set. Write 1 will clear this bit.   | R/W    | 0     |
| 4          | Reserved     | Be read as “0”   | -      | -     |
| 3          | DRMInfoIP    | 0: DRM information is not detected .<br>1: DRM information is detected .   | R/W    | 0     |
| 2          | ErrIP        | Error interrupt pending<br>0: No error occurred;<br>1: Error occurred.<br><br>These errors include legal wma file or opaque data present. WMA decoder will send interrupt signal while both ErrIntEn and ErrIP are set. Write 1 will clear this bit. | R/W    | 0     |

|   |           |   |     |   |
|---|-----------|---|-----|---|
| 1 | InfoRdyIP | Information ready interrupt pending<br>0: No new information is ready;<br>1: New information is ready;<br>InfoRdyIP asserts once every current time updated or bitrate updated Write 1 will clear this bit. | R/W | 0 |
| 0 | PCMRdyIP  | PCM ready interrupt pending<br>0: New PCM is ready for DAC;<br>1: New PCM is ready for DAC.<br>Write 1 will clear this bit.   | R/W | 0 |

### 8.2.9.3 WMACtl

**WMACtl (WMA Decoder Control Register, SFR Address 0xab, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | NewFile      | Write 1 this bit to initial for a new file decoder, and it will be cleared by hardware itself. | R/W    | 0     |
| 6          | En           | WMA Decoder Enable(play/pause)*<br>0: Pause WMA Decoder<br>1: Enable WMA Decoder               | R/W    | 0     |
| 5          | FF_REV_En    | Fast Forward or Reverse Enable<br>0: Normal play<br>1: Fast Forward or Reverse                 | R/W    | 0     |
| 4          | REVEN        | Reverse Enable<br>0: Normal play or FF;<br>1: Reverse(Fast Backward);                          | R/W    | 0     |
| 3:2        | Reserved     | Be read as “00”  | --     | --    |
| 1          | REV_SEL      | Reverse mode selection bit<br>0:Reverse mode is not selected<br>1:Reverse mode selected        | R/W    | 0     |
| 0          | Reserved     | Be read as “0”   | -      | -     |

\* To start a new file decoding the “NewFile” bit should be set before “En” bit set. Set the “En” bit can trigger fade in if “FadeIn” bit is “1”. Clear the “En” bit can trigger fade out if “FadeOut” bit is “0”. The decoding is stop until fade out finished.

### 8.2.9.4 WMAMediaInfo

**WMAMediaInfo (Header information register, SFR Address 0xac, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function | Access | Reset |
|------------|--------------|----------|--------|-------|
|            |              |          |        |       |

|     |            |  |   |      |
|-----|------------|--|---|------|
| 7:4 | SampFrq    | Sample frequency (kHz)<br>0000: 44.1;<br>0001: 48;<br>0010: 32;<br>0100: 22.05;<br>0110: 16;<br>1000: 11.025;<br>1010: 8;<br>Others: reserved; | R | 0000 |
| 3   | AllowSuper | 0: Do not allow super frame;<br>1: Allow super frame.  | R | 0    |
| 2   | WeightMode | Weighting Mode<br>0 : LPC mode ;<br>1 : Bark mode ;  | R | 0    |
| 1   | NoiseSub   | Noise Substitution<br>0 : Have no noise substitution ;<br>1: Have noise substitution.  | R | 1    |
| 0   | AudioMode  | Audio mode<br>0: dual channel;<br>1: single channel.   | R | 1    |

### 8.2.9.5 WMABitRateH

**WMABitRateH (High byte of bit rate index ffuse , SFR Address 0xad, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                     | Access | Reset |
|------------|--------------|------------------------------|--------|-------|
| 7:4        | BitRateIdx3  | Bit [15:12] of bitrate index | R      | xxxx  |
| 3:0        | BitRateIdx2  | Bit [11:8] of bitrate index  | R      | xxxx  |

### 8.2.9.6 WMABitRateL

**WMABitRateL (Low byte of bit rate index ffuse , SFR Address 0xae, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                   | Access | Reset |
|------------|--------------|----------------------------|--------|-------|
| 7:4        | BitRateIdx1  | Bit [7:4] bitrate index    | R      | xxxx  |
| 3:0        | BitRateIdx0  | Bit [3:0] of bitrate index | R      | xxxx  |

Note: BitRate = BitRateIdx[15:12]\*100 + BitRateIdx[11:8]\*10 + BitRateIdx[7:4] + BitRateIdx[3:0]\*0.1 (kbps)

### 8.2.9.7 WMATTimeH

**WMATTimeH (Total time hours register, SFR Address 0xaf, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                | Access | Reset  |
|------------|--------------|-------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”         | --     | --     |
| 5:0        | TtimeH       | Total time hours [0-59] | R      | xxxxxx |

### 8.2.9.8 WMATTimeM

WMATTimeM (Total time minutes register, SFR Address 0xb0, SFR bank 04)

| Bit Number | Bit Mnemonic | Function                  | Access | Reset  |
|------------|--------------|---------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”           | --     | --     |
| 5:0        | TtimeM       | Total time minutes [0-59] | R      | xxxxxx |

### 8.2.9.9 WMATTimeS

WMATTimeS (Total time seconds register, SFR Address 0xb1, SFR bank 04)

| Bit Number | Bit Mnemonic | Function                 | Access | Reset  |
|------------|--------------|--------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”          | --     | --     |
| 5:0        | TtimeS       | Total time second [0-59] | R      | xxxxxx |

### 8.2.9.10 WMACTimeH

WMACTimeH (Current time hours register, SFR Address 0xb2, SFR bank 04)

| Bit Number | Bit Mnemonic | Function                  | Access | Reset  |
|------------|--------------|---------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”           | --     | --     |
| 5:0        | CtimeH       | Current time hours [0-59] | R      | xxxxxx |

### 8.2.9.11 WMACTimeM

WMACTimeM (Current time minutes register, SFR Address 0xb3, SFR bank 04)

| Bit Number | Bit Mnemonic | Function                    | Access | Reset  |
|------------|--------------|-----------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”             | --     | --     |
| 5:0        | CtimeM       | Current time minutes [0-59] | R      | xxxxxx |

### 8.2.9.12 WMACTimeS

**WMACTimeS (Current time seconds register, SFR Address 0xb4, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                   | Access | Reset  |
|------------|--------------|----------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”            | --     | --     |
| 5:0        | CtimeS       | Current time second [0-59] | R      | xxxxxx |

### 8.2.9.13 WMAPackNumH

**WMAPackNumH (High byte of Packet Number register, SFR Address 0xb5, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function             | Access | Reset |
|------------|--------------|----------------------|--------|-------|
| 7:0        | PacketNumH   | PacketNumber [23:16] | R/W    | x     |

### 8.2.9.14 WMAPackNumM

**WMAPackNumM (Middle byte of Packet Number register, SFR Address 0xb6, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function            | Access | Reset |
|------------|--------------|---------------------|--------|-------|
| 7:0        | PacketNumM   | PacketNumber [15:8] | R/W    | x     |

### 8.2.9.15 WMAPackNumL

**WMAPackNumL (Low byte of Packet Number register, SFR Address 0xb7, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function           | Access | Reset |
|------------|--------------|--------------------|--------|-------|
| 7:0        | PacketNumL   | PacketNumber [7:0] | R/W    | x     |

### 8.2.9.16 WMAPackSizeH

**WMAPackSizeH (High byte of Packet Size register, SFR Address 0xb8, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | PacketSizeH  | High byte of PacketSize register<br>PacketSizeH[23:16] | R      | 0     |

### 8.2.9.17 WMAPackSizeM

WMAPackSizeM (Middle byte of Packet Size register, SFR Address 0xb9, SFR bank 04)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | PacketSizeM  | Middle byte of PacketSize register<br>PacketSizeM[15:8] | R      | 0     |

### 8.2.9.18 WMAPackSizeL

WMAPackSizeL (Low byte of Packet Size register, SFR Address 0xba, SFR bank 04)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | PacketSizeL  | Low byte of PacketSize register<br>PacketSizeL[7:0] | R      | 0     |

### 8.2.9.19 VirtualAddrH

VirtualAddrH (High byte of virtual address of current frame, SFR Address 0xbb, SFR bank 04)

| Bit Number | Bit Mnemonic            | Function   | Access | Reset |
|------------|-------------------------|--|--------|-------|
| 7:4        | Reserved                | Be read as 4 zeros.                                  | -      | -     |
| 3:0        | VirtualAddr_bit_8_to_11 | The bit 8 to 11 of virtual address of current frame. | R/W    | 0     |

### 8.2.9.20 VirtualAddrL

VirtualAddrL (Low byte of virtual address of current frame, SFR Address 0xbc, SFR bank 04)

| Bit Number | Bit Mnemonic           | Function  | Access | Reset |
|------------|------------------------|---|--------|-------|
| 7:0        | VirtualAddr_bit_0_to_7 | The bit 0 to 7 of virtual address of current frame. | R/W    | 0     |

### 8.2.9.21 PhysicalAddrH

PhysicalAddrH (High byte of physical address of current frame, SFR Address 0xbd, SFR bank 04)

| Bit Number | Bit Mnemonic | Function            | Access | Reset |
|------------|--------------|---------------------|--------|-------|
| 7:4        | Reserved     | Be read as 4 zeros. | -      | -     |

|     |                              |   |   |   |
|-----|------------------------------|---|---|---|
| 3:0 | PhysicalAddr_bit<br>_8_to_11 | The bit 8 to 11 of physical address of current frame.<br><br>The physical address is valid only | R | - |
|-----|------------------------------|---|---|---|

### 8.2.9.22 PhysicalAddrL

**PhysicalAddrL (Low byte of physical address of current frame, SFR Address 0xbf, SFR bank 04)**

| Bit Number | Bit Mnemonic                | Function   | Access | Reset |
|------------|-----------------------------|--|--------|-------|
| 7:0        | PhysicalAddr_bit<br>_0_to_7 | The bit 0 to 7 of physical address of current frame. | R      | -     |

### 8.2.9.23 DiscardLen

**DiscardLen (Discard length of current subframe, SFR Address 0xc1, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                                | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | Reserved     | Be read as one zero.                    | -      | -     |
| 6:0        | DiscardLen   | The discard length of current subframe. | R      | -     |

## 8.2.10 WAV Decoder Register Description

### 8.2.10.1 WAVIE

**WAVIE (WAV Decoder Interrupt Enable Register, SFR Address 0xa9, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | LackIntEn    | Lacking PCM interrupt enable<br>0: Disable;<br>1: Enable.<br><br>When both LackIntEn and LackIP are set,<br>Post processor will send interrupt signal.                              | R/W    | 0     |
| 6          | EnergyIntEn  | 0: Disable new energy ready interrupt;<br>1: Enable new energy ready interrupt.<br><br>When both EnergyIntEn and EnergyIP<br>are set, post processor will send interrupt<br>signal. | R/W    | 0     |
| 5          | WordIntEn    | 0: Disable sentence detected interrupt;<br>1: Enable sentence detected interrupt.<br><br>When both WordIntEn and WordIP are   | R/W    | 0     |

|     |              |  |     |    |
|-----|--------------|--|-----|----|
|     |              | set, post processor will send interrupt signal.  |     |    |
| 4:3 | Reserved     | Be read as “00”  | --  | -- |
| 2   | ErrIntEn     | 0: Disable error occur interrupt;<br>1: Enable error occur interrupt.<br><br>When both ErrIntEn and ErrIP are set, WAV decoder will send interrupt signal..              | R/W | 0  |
| 1   | InfoRdyIntEn | 0: Disable information ready interrupt;<br>1: Enable information ready interrupt.<br><br>When both InfoIntEn and InfoIP are set, WAV decoder will send interrupt signal. | R/W | 0  |
| 0   | PCMRdyIntEn  | 0: Disable new PCM ready interrupt;<br>1: Enable new PCM ready interrupt.<br><br>When both PCMRdyIntEn and PCMRdyIP are set, WAV decoder will send interrupt signal.     | R/W | 0  |

### 8.2.10.2 WAVIP

**WAVIP (WAV Decoder Interrupt Pending Register, SFR Address 0xaaa, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | LackIP       | Lacking PCM interrupt pending<br>0: Having enough PCM to play;<br>1: Having not enough PCM to play.<br><br>When decoding is slow or file finished, LackIP will be set. Write 1 will clear this bit.                                | R/W    | 0     |
| 6          | EnergyIP     | New energy ready interrupt pending<br>0: No New energy ready for accessing;<br>1: New energy ready for accessing.<br><br>EnergyIP will be set for every 2048 samples per channel has been decoded.<br>Write 1 will clear this bit. | R/W    | 0     |
| 5          | WordIP       | Interrupt pending for detecting a sentence.<br><br>If the energy less than EnergyIn for SilenceT times, WordIP will be set. Write 1 will clear this bit.   | R/W    | 0     |
| 4:3        | Reserved     | Be read as “00”  | --     | --    |
| 2          | ErrIP        | Error interrupt pending <sup>(1)</sup><br>0: No error occurred;<br>1: Error occurred.  | R/W    | 0     |

|   |           |  |     |   |
|---|-----------|--|-----|---|
|   |           | The error conditions include being not legal WAV/pcm file, having DRM, or opaque data present. Write 1 will clear this bit.  |     |   |
| 1 | InfoRdyIP | Information ready interrupt pending <sup>(2)</sup><br>0: No new information is ready;<br>1: New information is ready;<br>InfoRdyIP asserts once every current time updated or bitrate updated Write 1 will clear this bit. | R/W | 0 |
| 0 | PCMRdyIP  | PCM ready interrupt pending<br>0: New PCM is ready for DAC;<br>1: New PCM is ready for DAC.<br>Write 1 will clear this bit.  | R/W | 0 |

(1) The ErrIP will be set in the following cases:

- ① unsupported format
- ② the decoder not in decoding state while **InputSelect == 1'b1**

(2) The InfoRdyIP will be set in the following cases:

- ① state transition from STATE\_LOAD\_HEADER to STATE\_DECODE
- ② Increment or decrement of time is more than 1 second.

### 8.2.10.3 WAVCtl

**WAVCtl (WAV Decoder Control Register, SFR Address 0xab, SFR bank 04)**

| Bit Number | Bit Mnemonic       | Function  | Access | Reset |
|------------|--------------------|---|--------|-------|
| 7          | NewFile            | Write 1 this bit to initial for a new file decoder, and it will be cleared by hardware itself. *** ***  | R/W    | 0     |
| 6          | En                 | WAV Decoder Enable(play/pause)*<br>0: Pause WAV Decoder<br>1: Enable WAV Decoder  | R/W    | 0     |
| 5:4        | mode               | The mode of wave decoder:<br>00: Normal play<br>01: Fast Reverse<br>10: Fast Forward<br>11: Normal play<br><b>The AuCodecFIFO must be in empty state while mode transition.</b> | R/W    | 00    |
| 3:2        | <b>InputSelect</b> | PCM input select.<br><b>00: from AUIP FIFO 0;</b><br><b>01: from ADC</b><br><b>10: from IIS RX</b>  | R/W    | 0     |

|   |            | 11: SPDIF RX  |     |   |
|---|------------|---|-----|---|
| 1 | ChannelSel | <p>If the encoder is in mono ffuse mode:</p> <p>0: select left channel as input source.</p> <p>1: select right channel as input source</p> <p>If the encoder is in stero ffuse mode:</p> <p>0: select left channel as first input source.</p> <p>1: select right channel as first input source</p> <p>The left/right channel indicator is necessary for both ADC/IIS FIFO and SPDIF FIFO.</p> <p>The ChannelSel bit is valid only in direct input mode(InputSelect != 2'b00).</p> | R/W | 0 |
| 0 | Reserved   | Be read as "0"  | -   | - |

\* To start a new file decoding the “NewFile” bit should be set before “En” bit set. Set the “En” bit can trigger fade in if “FadeIn” bit is “1”. Clear the “En” bit can trigger fade out if “FadeOut” bit is “0”. The decoding is stop until fade out finished.

### 8.2.10.4 WAVHeaderInfo1

**WAVHeaderInfo1 (Header information register 1, SFR Address 0xac, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | SampFrq      | <p>Sample frequency (kHz)</p> <p>0000: 44.1;</p> <p>0001: 48;</p> <p>0010: 32;</p> <p>0011: 192</p> <p>0100: 22.05;</p> <p>0101: 24;</p> <p>0110: 16;</p> <p>0111: 176.4</p> <p>1000: 11.025;</p> <p>1001: 12;</p> <p>1010: 8;</p> <p>1011: 96</p> <p>1100: 88.2</p> <p>1101: reserved</p> <p>1110: reserved</p> | R      | xxxx  |

|     |                    |  |    |      |
|-----|--------------------|--|----|------|
|     |                    | 1111: reserved   |    |      |
| 3   | Reserved           | Be read as "0"   | -- | --   |
| 2:0 | Channel_assignment | Channel assignment: * * * * *<br>0000: 1 channel: mono<br>0001: 2 channels: left, right<br>0010: 3 channels: left, right, center<br>0011: 4 channels: left, right, back left, back right<br>0100: 5 channels: left, right, center, back/surround left, back/surround right<br>0101: 6 channels: left, right, center, LFE, back/surround left, back/surround right<br>0110: 7 channels: not defined<br>0111: 8 channels: not defined<br><br><b>Only first two channels of the stream is decoded and send to PCM FIFO.</b> | R  | 3'd7 |

### 8.2.10.5 WAVHeaderInfo2

**WAVHeaderInfo2 (Header information register 2, SFR Address 0xad, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:5        | Format       | Type of wave format:<br>000: Unknown Wave Format<br>001: LPCM (little endian)<br>010: MS-ADPCM<br>011: A-LAW<br>100: U-LAW<br>101: IMA-ADPCM<br>110: BPCM (big endian)<br>111: EXTENSIBLE | R      | -     |
| 4:0        | SampleSize   | Sample size in bits (valid bit per sample):<br>00000 : 32 bits per sample<br>00001 : 1 bits per sample<br>00001 : 2 bits per sample<br>...<br>11111 : 31 bits per sample                  | R      | -     |

### 8.2.10.6 WAVTTimeH

**WAVTTimeH (Total time hours register, SFR Address 0xaf, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                | Access | Reset  |
|------------|--------------|-------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”         | --     | --     |
| 5:0        | TtimeH       | Total time hours [0-59] | R      | xxxxxx |

### 8.2.10.7 WAVTTimeM

**WAVTTimeM (Total time minutes register, SFR Address 0xb0, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                  | Access | Reset  |
|------------|--------------|---------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”           | --     | --     |
| 5:0        | TtimeM       | Total time minutes [0-59] | R      | xxxxxx |

### 8.2.10.8 WAVTTimeS

**WAVTTimeS (Total time seconds register, SFR Address 0xb1, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                 | Access | Reset  |
|------------|--------------|--------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”          | --     | --     |
| 5:0        | TtimeS       | Total time second [0-59] | R      | xxxxxx |

Note: total time in seconds = DataChunkSize/nAvgBytesPerSec/nSamplesPerSec

### 8.2.10.9 WAVCTimeH

**WAVCTimeH (Current time hours register, SFR Address 0xb2, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                  | Access | Reset  |
|------------|--------------|---------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”           | --     | --     |
| 5:0        | CtimeH       | Current time hours [0-59] | R      | xxxxxx |

### 8.2.10.10 WAVCTimeM

**WAVCTimeM (Current time minutes register, SFR Address 0xb3, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                    | Access | Reset  |
|------------|--------------|-----------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”             | --     | --     |
| 5:0        | CtimeM       | Current time minutes [0-59] | R      | xxxxxx |

## 8.2.10.11 WAVCTimeS

**WAVCTimeS (Current time seconds register, SFR Address 0xb4, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function                   | Access | Reset  |
|------------|--------------|----------------------------|--------|--------|
| 7:6        | Reserved     | Be read as “00”            | --     | --     |
| 5:0        | CtimeS       | Current time second [0-59] | R      | xxxxxx |

## 8.2.10.12 WAVBlockNumH

**WAVBlockNumH (High byte of Block Number register, SFR Address 0xb5, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | BlockNumH    | High byte of BlockNumber register<br>BlockNumber [23:16] | R/W    | x     |

## 8.2.10.13 WAVBlockNumM

**WAVBlockNumM (Middle byte of Block Number register, SFR Address 0xb6, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | BlockNumM    | Middle byte of BlockNumber register<br>BlockNumber [15:8] | R/W    | x     |

## 8.2.10.14 WAVBlockNumL

**WAVBlockNumL (Low byte of Block Number register, SFR Address 0xb7, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | BlockNumL    | Low byte of BlockNumber register<br>BlockNumber [7:0] | R/W    | x     |

## 8.2.11 WAV Encoder Register Description

### 8.2.11.1 WAVEncCtl

**WAVEncCtl (WAV Encoder Control Register 1, SFR Address 0xab, SFR bank 04)**

| Bit | Bit | Function | Access | Reset |
|-----|-----|----------|--------|-------|
|     |     |          |        |       |

| Number | Mnemonic           |   |     |   |
|--------|--------------------|---|-----|---|
| 7      | NewFile            | Write 1 this bit to initial for a new file decoder, and it will be cleared by hardware itself. <sup>(1)</sup>   | R/W | 0 |
| 6      | En                 | WAV Encoder Enable(play/pause)<br>0: Pause WAV Decoder<br>1: Enable WAV Decoder   | R/W | 0 |
| 5      | ChannelMode        | 0: stero; <sup>(2)</sup><br>1: mono.  | R/W | 0 |
| 4      | EncodeMode         | 0 : ADPCM encoder mode<br>1 : PCM encoder mode  | R/W | 0 |
| 3:2    | InputSelect        | PCM input select.<br>00: from AUPIP FIFO 0;<br>01: from ADC<br>10: from IIS RX<br>11: SPDIF RX  | R/W | 0 |
| 1:0    | PunctuateDetection | The encoder will stop encoding once below the <a href="#">energy threshold</a> for a specified <a href="#">time threshold</a><br>00: disable auto-stop encode and mute output<br>01: enable auto-stop encode and disable mute output<br>10: disable auto-stop encode and enable mute output<br>11: disable auto-stop encode and mute output | R/W | 0 |

Note:

- (1) To start a new file decoding the “NewFile” bit should be set before “En” bit set. The encoding is pause immediately if this bit is cleared.
- (2) The block size is 0x400 bytes for stero encoding and 0x200 for mono encoding. The sample size is 16 bit for LPCM encoding, and 4 bit for IMA-ADPCM encoding. The number of valid sample pairs is 0x100 per block for PCM encoding and 0x3f9 per block for IMA-ADPCM encoding.

### 8.2.11.2 WAVEncInfo

**WAVEncInfo (WAV Encoder I Register, SFR Address 0xac, SFR bank 04)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7          | BlockFIFOFullIP | 0: block buffer full had never appeared<br>1: block buffer full had appeared<br>Writing 1 will clear this bit.                 | R/W    | 0     |
| 6          | ChannelSel      | If the encoder is in mono encoding mode:<br>0: select left channel as input source.<br>1: select right channel as input source | R/W    | 0     |

|     |               |  |   |   |
|-----|---------------|--|---|---|
|     |               | If the encoder is in stereo encoding mode:<br>0: select left channel as first input source.<br>1: select right channel as first input source<br><br><b>The left/right channel indicator is necessary for both ADC/IIS FIFO and SPDIF FIFO.</b>   |   |   |
| 5   | Energy_stat e | 0: above energy threshold or Auto_stop_enc is disable<br>1: below energy threshold and Auto_stop_enc is enable   | R | 0 |
| 4:0 | FIFOInfo      | The number of encoded blocks in the encode output FIFO. The encoding will stop encoding automatically if the fifo is full.<br><br>(1) The maximum number of blocks can be stored in the encode output FIFO is $(0x1400*3-0x3f9*2*2)/0x400=7$ (about 148ms @48ksps) if the encoding mode is IMA-ADPCM stereo.<br><br>(2) The maximum number of blocks can be stored in the encode output FIFO is $(0x1400*3-0x3f9*2*2)/0x200=22$ (about 466ms @48ksps) if the encoding mode is IMA-ADPCM mono.<br><br>(3) The maximum number of blocks can be stored in the encode output FIFO is $(0x1400*3-0x400*2)/0x400=13$ (about 69ms @48ksps) if the encoding mode is LPCM stereo.<br><br>(4) The maximum number of blocks can be stored in the encode output FIFO is $(0x1400*3-0x200*2)/0x200=28$ (about 149ms @48ksps) if the encoding mode is LPCM mono. | R | 0 |

## 8.2.12 MP3 Encoder Register Description

### 8.2.12.1 MP3EncCtl

**MP3EncCtl (MP3 Encoder Control Register 1, SFR Address 0xab, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | NewFile      | Write 1 this bit to initial for a new file decoder, and it will be cleared by hardware itself. <sup>(1)</sup> | R/W    | 0     |

|     |                    |   |     |   |
|-----|--------------------|---|-----|---|
| 6   | En                 | WAV Encoder Enable(play/pause)<br>0: Pause WAV Decoder<br>1: Enable WAV Decoder   | R/W | 0 |
| 5   | ChannelMode        | 0: stero; <sup>(2)</sup><br>1: mono.  | R/W | 0 |
| 4   | bitrate            | 0: 64k bits/s if mono, 128kbits/s if stero<br>1: 128k bits/s if mono, 256kbits/s if stero   | R/W | 0 |
| 3:2 | InputSelect        | PCM input select.<br>00: from AUIP FIFO 0;<br>01: from ADC<br>10: from IIS RX<br>11: SPDIF RX   | R/W | 0 |
| 1:0 | PunctuateDetection | The encoder will stop encoding once below the <a href="#">energy threshold</a> for a specified <a href="#">time threshold</a><br>00: disable auto-stop encode and mute output<br>01: enable auto-stop encode and disable mute output<br>10: disable auto-stop encode and enable mute output<br>11: disable auto-stop encode and mute output | R/W | 0 |

Note:

- (1) To start a new file decoding the “NewFile” bit should be set before “En” bit set. The encoding is pause immediately if this bit is cleared.
- (2) The block size is 192 bytes if bitrate==64kbits/s and 384 bytes if bitrate==128kbits/s. The number of valid sample pairs is 0x480 per frame for mp3 encoding.

### 8.2.12.2 MP3EncInfo

**MP3EncInfo (MP3 Encoder I Register, SFR Address 0xac, SFR bank 04)**

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7          | BlockFIFOFullIP | 0: block buffer full had never appeared<br>1: block buffer full had appeared<br>Writing 1 will clear this bit.   | R/W    | 0     |
| 6          | ChannelSel      | If the encoder is in mono encoding mode:<br>0: select left channel as input source.<br>1: select right channel as input source<br>If the encoder is in stero encoding mode:<br>0: select left channel as first input source.<br>1: select right channel as first input source<br><a href="#">The left/right channel indicator is necessary for both ADC/IIS FIFO and SPDIF FIFO.</a> | R/W    | 0     |

|     |               |   |   |   |
|-----|---------------|---|---|---|
| 5   | Energy_stat_e | 0: above energy threshold or Auto_stop_enc is disable<br>1: below energy threshold and Auto_stop_enc is enable  | R | 0 |
| 4:0 | FIFOInfo      | The number of encoded blocks in the encode output FIFO. The encoding will stop encoding automatically if the FIFO is full.<br><br>The size of block FIFO is 7296 bytes in mono encode mode. The size of block FIFO is 3264 bytes in stero encode mode<br><br>(1) The maximum number of blocks can be stored in the encode output FIFO is floor(3264/384)=8 in stero encode mode @128kbps.<br>(2) The maximum number of blocks can be stored in the encode output FIFO is floor(3264/768)=4 in stero encode mode @256kbps..<br>(3) The maximum number of blocks can be stored in the encode output FIFO is floor(7296/192)=38 in mono encode mode @64kbps.<br>(4) The maximum number of blocks can be stored in the encode output FIFO is floor(7296/384)=19 in mono encode mode @128kbps. | R | 0 |

## 8.2.13 Software Decoder Register Description

### 8.2.13.1 SoftIE

**SoftIE (Software Decoding Enable Register, SFR Address 0xa9, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 31:8       | Reserved     | Be read as 24 zeros.  | -      | -     |
| 7          | LackIntEn    | Lacking PCM interrupt enable<br>0: Disable;<br>1: Enable.<br><br>When both LackIntEn and LackIP are set, post processor will send interrupt signal. | R/W    | 0     |
| 6          | EnergyIntEn  | 0: Disable new energy ready interrupt;<br>1: Enable new energy ready interrupt.<br><br>When both EnergyIntEn and EnergyIP                           | R/W    | 0     |

|     |               |  |     |    |
|-----|---------------|--|-----|----|
|     |               | are set, post processor will send interrupt signal.  |     |    |
| 5   | WordIntEn     | 0: Disable sentence detected interrupt;<br>1: Enable sentence detected interrupt.<br><br>When both WordIntEn and WordIP are set, post processor will send interrupt signal.  | R/W | 0  |
| 4   | Reserved      | Be read as 1 zero.   | --  | -- |
| 3   | FrameRdyIntEn | The pending bit to show whether the post processing of one frame's audio stream is complete and has been loaded to the PCM buffer.<br><br>0: The decoding of one frame's audio stream is not complete.<br>1: The decoding of one frame's audio stream is complete.<br><br>Write 1 will clear this bit. | R/W | 0  |
| 2:0 | Reserved      | Be read as 3 zeros.  | -   | -  |

### 8.2.13.2 SoftIP

**SoftIP (Software Decoding Pending Register, SFR Address 0xaa, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 31:8       | Reserved     | Be read as 24 zeros.   | -      | -     |
| 7          | LackIP       | Lacking PCM interrupt pending<br>0: The PCM buffer is not empty;<br>1: The PCM buffer is empty.<br><br>When decoding is slow or file finished, LackIP will be set. Write 1 will clear this bit.                                    | R/W    | 0     |
| 6          | EnergyIP     | New energy ready interrupt pending<br>0: No New energy ready for accessing;<br>1: New energy ready for accessing.<br><br>EnergyIP will be set for every 2048 samples per channel has been decoded.<br>Write 1 will clear this bit. | R/W    | 0     |
| 5          | WordIP       | Interrupt pending for detecting a sentence.<br><br>If the energy less than EnergyIn for SilenceT times, WordIP will be set. Write 1 will clear this bit.   | R/W    | 0     |
| 4          | Reserved     | Be read as 1 zero.   | -      | -     |

|     |            |  |     |   |
|-----|------------|--|-----|---|
| 3   | FrameRdyIP | The pending bit to show whether the post processing of one frame's audio stream is complete and has been loaded to the PCM buffer.<br>0: The decoding of one frame's audio stream is not complete.<br>1: The decoding of one frame's audio stream is complete.<br>Write 1 will clear this bit. | R/W | 0 |
| 2:0 | Reserved   | Be read as 3 zeros.  | -   | - |

### 8.2.13.3 SoftCtl

**SoftCtl (Software Decoder Control Register, SFR Address 0xab, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 31:8       | Reserved     | Be read as 24 zeros.   | -      | -     |
| 7          | NewFile      | Write 1 this bit to initial for a new file decoder, and it will be cleared by hardware itself.   | R/W    | 0     |
| 6          | En           | Soft Decoder Enable(play/pause)*<br>0: Pause Soft Decoder<br>1: Enable Soft Decoder  | R/W    | 0     |
| 5:2        | SampFrq      | Sample frequency (kHz)<br>0000: 44.1;<br>0001: 48;<br>0010: 32;<br>0100: 22.05;<br>0101: 24;<br>0110: 16;<br>1000: 11.025;<br>1001: 12;<br>1010: 8;<br>Others: reserved; | R/W    | 0000  |
| 1          | ChannelNum   | The number of channels of current frame.<br>0: mono<br>1: stereo   | R/W    | 0     |
| 0          | NewFrame     | Write 1 to this bit to start a new frame decoder, and it will be cleared by hardware itself.   | R/W    | 0     |

\* To start a new file decoding the "NewFile" bit should be set before "En" bit set. Set the "En" bit can trigger fade in if "FadeIn" bit is "1". Clear the "En" bit can trigger fade out if "FadeOut" bit is "0". The decoding is

stop until fade out finished.

\*\* There is no frame debug for soft decoding mode. Both the value of AuDebugLength and AuCodecDebug registers are invalid in this mode.

### 8.2.13.4 FrameLen1

**FrameLen1 (Frame length register 1, SFR Address 0xb8, SFR bank 04)**

| Bit Number | Bit Mnemonic  | Function  | Access | Reset |
|------------|---------------|---|--------|-------|
| 7:4        | Reserved      | Be read as 4 zeros.                               | -      | -     |
| 3:0        | FrameLen_11_8 | The bit 11:8 of the sample pairs of current frame | R/W    | -     |

### 8.2.13.5 FrameLen0

**FrameLen0 (Frame length register 0, SFR Address 0xb9, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | FrameLen_7_0 | The bit 7:0 of the sample pairs of current frame | R/W    | -     |

The audio data should put in the muram1 like this if sample size is 24 bit:

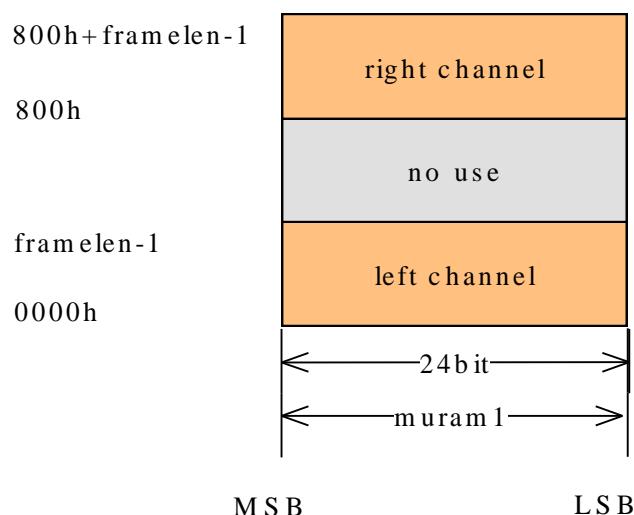


Figure 8-17 24bit sample size audio data in muram1

The audio data should put in the muram1 like this if sample size is 16 bit:

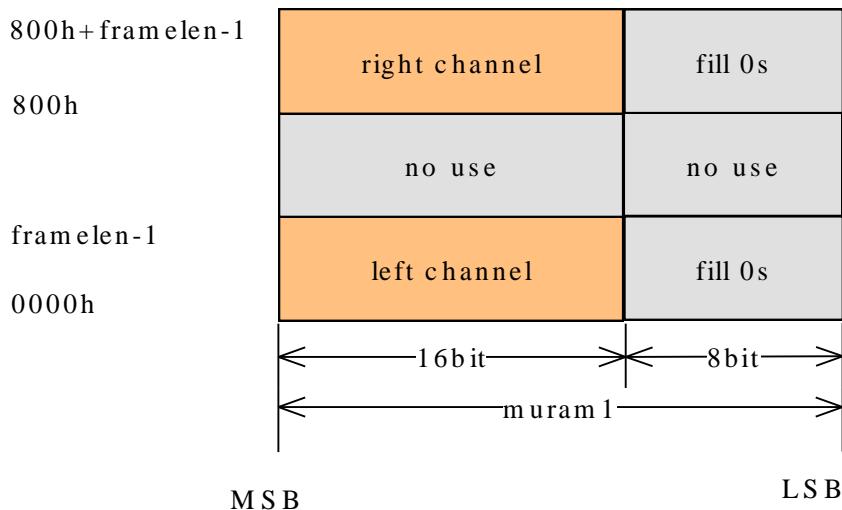


Figure 8-18 16bit sample size audio data in muram1

The audio data should put in the muram1 like this if sample size is 8 bit:

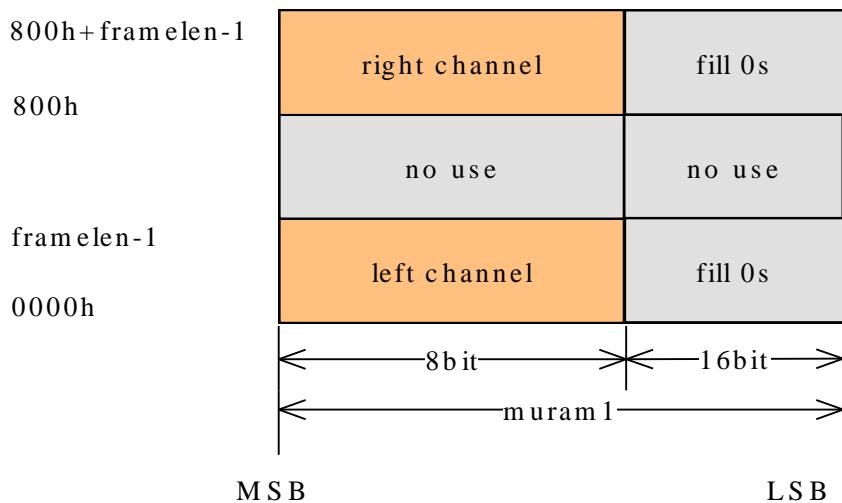


Figure 8-19 8bit sample size audio data in muram1

All audio data should be aligned to the MSB of MURAM1.

### **8.2.14 Post Processor Register Description**

## 8.2.14.1 BEPCtl

**BEPCT1 (Post Processor Control Register 1, SFR Address 0xc2, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | EffectEn     | SRS or Karaoke sound effect enable bit<br>0: Disable;<br>1: Enable. | R/W    | 0     |
| 6          | EnergyEn     | Enable estimating PCM energy for both encode and decode mode;       | R/W    | 0     |

|   |               |  |     |   |
|---|---------------|--|-----|---|
|   |               | 0: Disable;<br>1: Enable.  |     |   |
| 5 | WhishtEn      | whisht enable<br>0: Normal play;<br>1: Mute 256~2048 PCM samples, its high level width must longer than one clock.<br>During fadein/fadeout, Whisht must not be enable, otherwise fadein/fadeout will be restarted again while Whisht is finished. | R/W | 0 |
| 4 | FadeInEn      | 0: Normal play, no fadein is in effect;<br>1: Fadein enable, it must keep active to make the effect valid for play-pause-play switch.  | R/W | 0 |
| 3 | FadeOutEn     | 0: Normal play, no fadeout is in effect;<br>1: Fadein enable, it must keep active to make the effect valid for play-pause-play switch.   | R/W | 0 |
| 2 | AutoFadeOutEn | Auto fadeout enable<br>0: auto fadeout at the last second<br>1: disable auto fadeout   | R/W | 0 |
| 1 | FadeOutP      | FadeOutPending<br>0: FadeOut is not finished<br>1: FadeOut is finished<br>If write 1 to this bit, it will be cleared.  | R/W | 0 |
| 0 | Halt2_en      | 0: The audio codec will continue decoding after stream decoding state is complete.<br>1: The audio codec will enter “HALT2” state stream decoding state is complete.   | R/W | 0 |

### 8.2.14.2 BEPCtl2

**BEPCtl2 (Post Processor Control Register 2, SFR Address 0xc3, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:5        | EqBandSel    | Number of bands that equalizer selected<br>000: 0 band EQ<br>001: 1 band EQ<br>010: 2 band EQ<br>011: 3 band EQ<br>100: 4 band EQ<br>101: 5 band EQ<br>110: 6 band EQ<br>111: 7 band EQ | R/W    | 111   |

|     |              |   |     |     |
|-----|--------------|---|-----|-----|
| 4:2 | FadeDuration | The duration of fade in or fade out is set by FadeDuration[2:0] register:<br>Duration of fade in or fade out<br>000: 0 second<br>001: 0.1 second<br>010: 0.2 second<br>011: 0.4 second<br><b>100: 0.025 second</b><br><b>101: 0.05 second</b> | R/W | 011 |
| 1   | PCMFull      | PCM Buffer Full:<br>0: Not Full<br>1: Full  | R   | 0   |
| 0   | PCMEmpty     | PCM Buffer Empty<br>0: Not Empty<br>1: Empty  | R   | 1   |

### 8.2.14.3 BEPCtl3

**BEPCtl3 (Post Processor Control Register 3, SFR Address 0xc4, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | MuteEn       | Enable Mute  | R/W    | 0     |
| 6          | MuteL        | Mute left channel  | R/W    | x     |
| 5          | MuteR        | Mute right channel   | R/W    | x     |
| 4:0        | SilenceT     | The threshold of silence time to for sentence punctuation detection.<br>It is an unsigned Q2.3 value which means “00001” denotes 0.125 seconds and “11111” denotes 3.875 seconds.<br><b>In encoding mode the time threshold is:</b><br><b>0: 32*16 blocks below energy threshold</b><br><b>1: 1*16* blocks below energy threshold</b><br><b>2: 2*16* blocks below energy threshold</b><br><b>...</b><br><b>2: 31*16* blocks below energy threshold</b><br><b>The block size is 512 sample pairs.</b> | R/W    | xxxxx |

### 8.2.14.4 BEPCtl4

**BEPCtl4 (Post Processor Control Register 4, SFR Address 0xc5, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function | Access | Reset |
|------------|--------------|----------|--------|-------|
|            |              |          |        |       |

|     |                  |   |     |    |
|-----|------------------|---|-----|----|
| 7:6 | WhishtSample     | WhishtSamples [2 :0]<br>00 : 2048 samples<br>01 : 1024 samples<br>10 : 512 samples<br>11 : 256 samples  | R/W | 00 |
| 5   | PCM_almost_full  | PCM FIFO almost full bit is set if the PCM FIFO can receive less than 0x100 sample pairs from the decoder:<br>0: PCM FIFO is not almost full<br>1: PCM FIFO is almost full                | R   | 0  |
| 4   | PCM_almost_empty | PCM FIFO almost empty information bit is set if the PCM FIFO can receive more than 0x300 sample pairs from the decoder:<br>0: PCM FIFO is not almost empty<br>1: PCM FIFO is almost empty | R   | 1  |
| 3   | EffectMode       | Sound effect Mode<br>0: SRS effect selecte;<br>1: karaoke effect selected;  | R/W | 0  |
| 2   | EQEn             | EQ Enable bit:<br>0: disable EQ<br>1: enable EQ   | R/W | 0  |
| 1   | Halt_en          | 0: the audio codec will continue decode after the state “BLOCK_END” or the state “HALT”.<br>1: the audio codec will hold the state “HALT” after the state “BLOCK_END”.                    | R/W | 0  |
| 0   | FadeInActive     | 0: The fade in process is complete.<br>1: The fade in process is not complete.  | R   | 0  |

### 8.2.14.5 BEPCtl5

**BEPCtl5 (Post Processor Control Register 5, SFR Address 0xc6, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset       |
|------------|--------------|---|--------|-------------|
| 7          | EffectInBits | The input data resolution for audio effect<br>0: 16bits<br>1: 24bits  | R/W    | 0           |
| 6:0        | EnergyIn     | The threshold of the energy of one frame’s (512) PCM sample pairs for sentence punctuation detection. The sum add 2 times if the audio is mono. | R/W    | xxxxxx<br>x |

### 8.2.14.6 GlobalGainH

**GlobalGainH (Global Gain High Byte Register, SFR Address 0xc7, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | GlobalGainH  | The bits 23:16 of Global Gain register.<br><small>(1)(2)</small> | R/W    | 10h   |

### 8.2.14.7 GlobalGainM

**GlobalGainM (Global Gain Middle Byte Register, SFR Address 0xc8, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | GlobalGainM  | The bits 15:8 of Global Gain register.<br><small>(1)(2) (3)</small> | R/W    | 00h   |

### 8.2.14.8 GlobalGainL

**GlobalGainL (Global Gain Low Byte Register, SFR Address 0xc9, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | GlobalGainL  | The bits 7:0 of Global Gain register.<br><small>(1)(2) (3)</small> | R/W    | 00h   |

Note:

- (2) GlobalGain is unsigned Q4.20 format value.
- (3) The volume of the audio effect output is adjusted by GlobalGain in decode mode
- (4) The volume of input PCM is adjusted by GlobalGain in encode mode

### 8.2.14.9 CurrentEnergy

**CurrentEnergy (Current Energy Register , SFR Address 0xca, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset         |
|------------|--------------|--|--------|---------------|
| 7:0        | Energy       | The energy of current frame's PCM sample pairs.<br>The most 7 significant bits of the sum of the absolute value of one frame's PCM sample pairs. (512-sample pairs per one frame for both decode and encode mode). | R      | 000000<br>00b |

## 8.2.1 Karaoke Register Description

### 8.2.1.1 KaraokeCtl

**KaraokeCtl (Karaoke Control Register, SFR Address 0xcb, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | Reserved     | Be read as 1 zero.   | -      | -     |
| 6          | MonoEn       | Single channel output enable bit:<br>0: disable<br>1: enable   | R/W    | 0     |
| 5          | Lrselect     | Left or right channel select:<br>0: left channel selected<br>2: right channel selected   | R/W    | 0     |
| 4:3        | LPF          | 3Db cut-off frequency of low pass filter:<br>00: 200Hz<br>01: 250Hz<br>10: 300Hz<br>11: 350Hz  | R/W    | 00    |
| 2:0        | HPF          | 3Db cut-off frequency of high pass filter:<br>000: 4000Hz<br>001: 5000Hz<br>010: 6000Hz<br>011: 7000Hz<br>100: 8000Hz<br>101: 9000Hz<br>110: 10000Hz<br>111: 11000Hz | R/W    | 000   |

### 8.2.1.2 LCHGain

**LCHGain (Left channel gain, SFR Address 0xcc, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | LCHgain      | Unsigned Q2.6.<br>This gain coefficient is active only when KaraokeEn bit is set. | R/W    | 71h   |

### 8.2.1.3 RCHGain

**RCHGain (Right channel gain, SFR Address 0xcd, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | RCHgain      | Unsigned Q2.6.<br>This gain coefficient is active only when KaraokeEn bit is set. | R/W    | 0f8h  |

### 8.2.1.4 LPFGain

LPFGain (Low pass filter gain, SFR Address 0xce, SFR bank 04)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | LPFgain      | Unsigned Q2.6.<br>This gain coefficient is active only when KaraokeEn bit is set. | R/W    | 19h   |

### 8.2.1.5 HPFGain

HPFGain (High pass filter gain, SFR Address 0xcf, SFR bank 04)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | HPFgain      | Unsigned Q2.6.<br>This gain coefficient is active only when KaraokeEn bit is set. | R/W    | 00h   |

## 8.2.2 EQ Coefficients Description

Both left and right channel have separate EQ band gain coefficients. These coefficients are stored in the FIR\_MPX\_RAM at the start address of (1900h-1400h). The figure below shows the location of these coefficients:

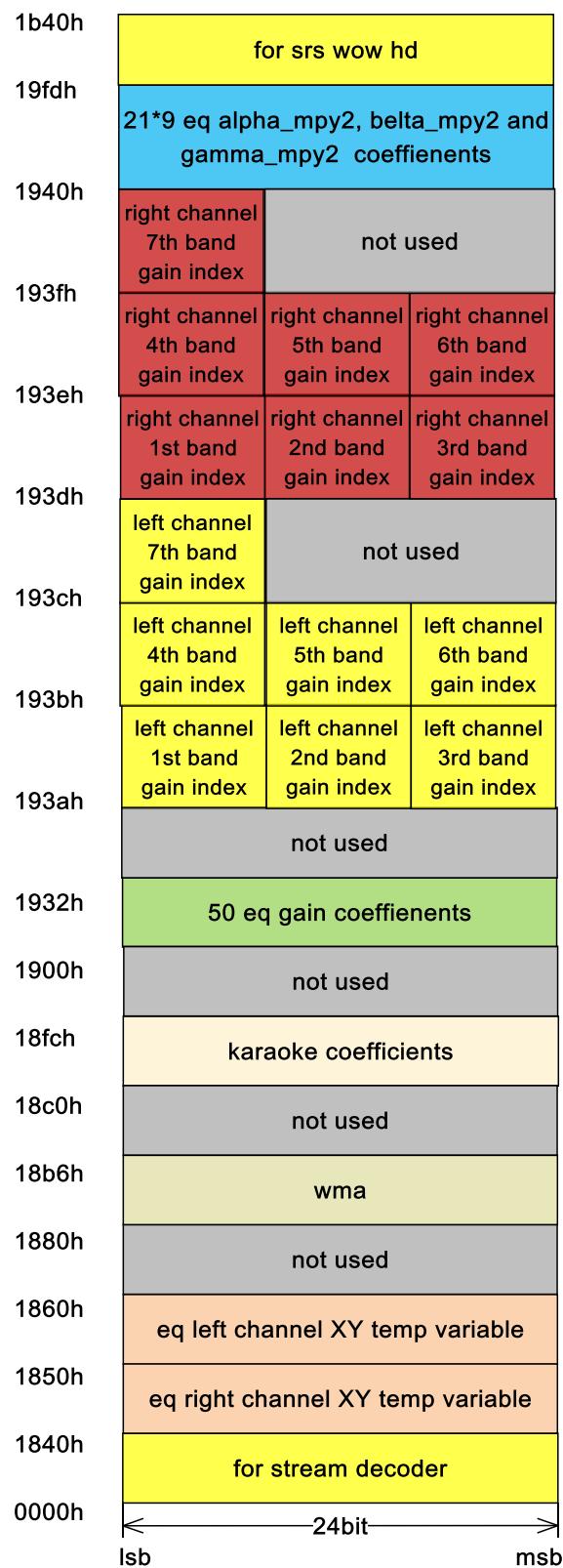


Figure 8-20 memory map of EQ coefficients

The contents of EQ gain coefficients is show below:

#### EQCoe(0~6) (NO.0~6 Equalizer Coefficients for both left and right channel)

| Bit Number | Bit Mnemonic | Function        | Access | Reset |
|------------|--------------|-----------------|--------|-------|
| 7          | Reserved     | Be read as "0". | -      | -     |

| 6      | SignSel         | 0: positive gain:<br>Gain coefficient should be Multiplied by 1<br>1: Negative gain:<br>Gain coefficient should be Multiplied by -1  | R/W    | 0       |      |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
|--------|-----------------|--|--------|---------|------|-------|---------|------|--------|------|---|--------|------|---|--------|----------|---|--------|--------|---|--------|----------|---|--------|--------|---|--------|----------|---|--------|--------|---|--------|----------|---|--------|--------|---|--------|----------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|--------|---|--------|---------|---|--------|---------|---|--------|---------|---|--------|---------|---|--------|---------|---|--------|---------|---|--------|---------|---|--------|---------|---|--------|---------|---|--------|---------|---|-----|--------|
| 5:0    | GainCoefficient | For each band the 6 bits coefficient represents -12 to 12 Db gain, EQ gain table (signed Q0.23)<br><br><table> <thead> <tr> <th>index</th> <th>EQ gain</th> <th>sign</th> <th>index</th> <th>EQ gain</th> <th>sign</th> </tr> </thead> <tbody> <tr><td>6'd00:</td><td>0Db;</td><td>0</td><td>6'd25:</td><td>0Db;</td><td>0</td></tr> <tr><td>6'd01:</td><td>-12.0Db;</td><td>1</td><td>6'd26:</td><td>0.5Db;</td><td>0</td></tr> <tr><td>6'd02:</td><td>-11.5Db;</td><td>1</td><td>6'd27:</td><td>1.0Db;</td><td>0</td></tr> <tr><td>6'd03:</td><td>-11.0Db;</td><td>1</td><td>6'd28:</td><td>1.5Db;</td><td>0</td></tr> <tr><td>6'd04:</td><td>-10.5Db;</td><td>1</td><td>6'd29:</td><td>2.0Db;</td><td>0</td></tr> <tr><td>6'd05:</td><td>-10.0Db;</td><td>1</td><td>6'd30:</td><td>2.5Db;</td><td>0</td></tr> <tr><td>6'd06:</td><td>-9.5Db;</td><td>1</td><td>6'd31:</td><td>3.0Db;</td><td>0</td></tr> <tr><td>6'd07:</td><td>-9.0Db;</td><td>1</td><td>6'd32:</td><td>3.5Db;</td><td>0</td></tr> <tr><td>6'd08:</td><td>-8.5Db;</td><td>1</td><td>6'd33:</td><td>4.0Db;</td><td>0</td></tr> <tr><td>6'd09:</td><td>-8.0Db;</td><td>1</td><td>6'd34:</td><td>4.5Db;</td><td>0</td></tr> <tr><td>6'd10:</td><td>-7.5Db;</td><td>1</td><td>6'd35:</td><td>5.0Db;</td><td>0</td></tr> <tr><td>6'd11:</td><td>-7.0Db;</td><td>1</td><td>6'd36:</td><td>5.5Db;</td><td>0</td></tr> <tr><td>6'd12:</td><td>-6.5Db;</td><td>1</td><td>6'd37:</td><td>6.0Db;</td><td>0</td></tr> <tr><td>6'd13:</td><td>-6.0Db;</td><td>1</td><td>6'd38:</td><td>6.5Db;</td><td>0</td></tr> <tr><td>6'd14:</td><td>-5.5Db;</td><td>1</td><td>6'd39:</td><td>7.0Db;</td><td>0</td></tr> <tr><td>6'd15:</td><td>-5.0Db;</td><td>1</td><td>6'd40:</td><td>7.5Db;</td><td>0</td></tr> <tr><td>6'd16:</td><td>-4.5Db;</td><td>1</td><td>6'd41:</td><td>8.0Db;</td><td>0</td></tr> <tr><td>6'd17:</td><td>-4.0Db;</td><td>1</td><td>6'd43:</td><td>8.5Db;</td><td>0</td></tr> <tr><td>6'd18:</td><td>-3.5Db;</td><td>1</td><td>6'd43:</td><td>9.0Db;</td><td>0</td></tr> <tr><td>6'd19:</td><td>-3.0Db;</td><td>1</td><td>6'd44:</td><td>9.5Db;</td><td>0</td></tr> <tr><td>6'd20:</td><td>-2.5Db;</td><td>1</td><td>6'd45:</td><td>10.0Db;</td><td>0</td></tr> <tr><td>6'd21:</td><td>-2.0Db;</td><td>1</td><td>6'd46:</td><td>10.5Db;</td><td>0</td></tr> <tr><td>6'd22:</td><td>-1.5Db;</td><td>1</td><td>6'd47:</td><td>11.0Db;</td><td>0</td></tr> <tr><td>6'd23:</td><td>-1.0Db;</td><td>1</td><td>6'd48:</td><td>11.5Db;</td><td>0</td></tr> <tr><td>6'd24:</td><td>-0.5Db;</td><td>1</td><td>6'd49:</td><td>12.0Db;</td><td>0</td></tr> </tbody> </table> | index  | EQ gain | sign | index | EQ gain | sign | 6'd00: | 0Db; | 0 | 6'd25: | 0Db; | 0 | 6'd01: | -12.0Db; | 1 | 6'd26: | 0.5Db; | 0 | 6'd02: | -11.5Db; | 1 | 6'd27: | 1.0Db; | 0 | 6'd03: | -11.0Db; | 1 | 6'd28: | 1.5Db; | 0 | 6'd04: | -10.5Db; | 1 | 6'd29: | 2.0Db; | 0 | 6'd05: | -10.0Db; | 1 | 6'd30: | 2.5Db; | 0 | 6'd06: | -9.5Db; | 1 | 6'd31: | 3.0Db; | 0 | 6'd07: | -9.0Db; | 1 | 6'd32: | 3.5Db; | 0 | 6'd08: | -8.5Db; | 1 | 6'd33: | 4.0Db; | 0 | 6'd09: | -8.0Db; | 1 | 6'd34: | 4.5Db; | 0 | 6'd10: | -7.5Db; | 1 | 6'd35: | 5.0Db; | 0 | 6'd11: | -7.0Db; | 1 | 6'd36: | 5.5Db; | 0 | 6'd12: | -6.5Db; | 1 | 6'd37: | 6.0Db; | 0 | 6'd13: | -6.0Db; | 1 | 6'd38: | 6.5Db; | 0 | 6'd14: | -5.5Db; | 1 | 6'd39: | 7.0Db; | 0 | 6'd15: | -5.0Db; | 1 | 6'd40: | 7.5Db; | 0 | 6'd16: | -4.5Db; | 1 | 6'd41: | 8.0Db; | 0 | 6'd17: | -4.0Db; | 1 | 6'd43: | 8.5Db; | 0 | 6'd18: | -3.5Db; | 1 | 6'd43: | 9.0Db; | 0 | 6'd19: | -3.0Db; | 1 | 6'd44: | 9.5Db; | 0 | 6'd20: | -2.5Db; | 1 | 6'd45: | 10.0Db; | 0 | 6'd21: | -2.0Db; | 1 | 6'd46: | 10.5Db; | 0 | 6'd22: | -1.5Db; | 1 | 6'd47: | 11.0Db; | 0 | 6'd23: | -1.0Db; | 1 | 6'd48: | 11.5Db; | 0 | 6'd24: | -0.5Db; | 1 | 6'd49: | 12.0Db; | 0 | R/W | 000000 |
| index  | EQ gain         | sign   | index  | EQ gain | sign |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd00: | 0Db;            | 0  | 6'd25: | 0Db;    | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd01: | -12.0Db;        | 1  | 6'd26: | 0.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd02: | -11.5Db;        | 1  | 6'd27: | 1.0Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd03: | -11.0Db;        | 1  | 6'd28: | 1.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd04: | -10.5Db;        | 1  | 6'd29: | 2.0Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd05: | -10.0Db;        | 1  | 6'd30: | 2.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd06: | -9.5Db;         | 1  | 6'd31: | 3.0Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd07: | -9.0Db;         | 1  | 6'd32: | 3.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd08: | -8.5Db;         | 1  | 6'd33: | 4.0Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd09: | -8.0Db;         | 1  | 6'd34: | 4.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd10: | -7.5Db;         | 1  | 6'd35: | 5.0Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd11: | -7.0Db;         | 1  | 6'd36: | 5.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd12: | -6.5Db;         | 1  | 6'd37: | 6.0Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd13: | -6.0Db;         | 1  | 6'd38: | 6.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd14: | -5.5Db;         | 1  | 6'd39: | 7.0Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd15: | -5.0Db;         | 1  | 6'd40: | 7.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd16: | -4.5Db;         | 1  | 6'd41: | 8.0Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd17: | -4.0Db;         | 1  | 6'd43: | 8.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd18: | -3.5Db;         | 1  | 6'd43: | 9.0Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd19: | -3.0Db;         | 1  | 6'd44: | 9.5Db;  | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd20: | -2.5Db;         | 1  | 6'd45: | 10.0Db; | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd21: | -2.0Db;         | 1  | 6'd46: | 10.5Db; | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd22: | -1.5Db;         | 1  | 6'd47: | 11.0Db; | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd23: | -1.0Db;         | 1  | 6'd48: | 11.5Db; | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |
| 6'd24: | -0.5Db;         | 1  | 6'd49: | 12.0Db; | 0    |       |         |      |        |      |   |        |      |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |          |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |        |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |        |         |   |     |        |

### 8.2.3 SRS Processor Register Description

#### 8.2.3.1 SRSCtl0

**SRSCtl0 (SRS Control Register 0, SFR Address 0xcc, SFR bank 04)**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | Reserved     | Be read as one zero.                               | -      | -     |
| 6          | SRSTrBass    | Determines which of two operating modes TruBass HD | R/W    | 1     |

|   |                 |   |     |   |
|---|-----------------|---|-----|---|
|   | sProcessMode    | operates in, Mono or Stereo.<br>0: Mono<br>1: Stereo  |     |   |
| 5 | DialogClarityEn | Dialog Clarity Enable:<br>0: disable<br>1: enable   | R/W | 1 |
| 4 | LimiterEn       | limiter enable:<br>0: disable<br>1: enable  | R/W | 1 |
| 3 | OutputMode      | Only 2 output mode is supported:<br>0: kSrsTruSurroundHDOutput2_0_0<br>1: kSrsTruSurroundHDOutputHeadphone  | R/W | 0 |
| 2 | InputMode       | Only 2 inputMode of audio stream is supported:<br>0: kSrsTruSurroundHDInput2_0_1 // L/R<br>1: kSrsTruSurroundHDInputLtRt  | R/W | 0 |
| 1 | TSHDEn          | TruSurroundHD channel enable bit:<br>0: SRSTruSurroundHD bypass<br>1: SRSTruSurroundHD process  | R/W | 0 |
| 0 | SurroundEn      | Surround Effects Enable.<br><br>Determines if surround effects are enabled or disabled.<br>This control is only functional with SRS_IO_LtRt input<br>and SRS_IO_2_0_0 output mode.<br><br>If the Surround Effects are disabled, TruBass HD<br>processing is applied to the signal.<br><br>If enabled, general TruSurround HD processing is applied. | R/W | 1 |

### 8.2.3.2 SRSCtl1

**SRSCtl1 (SRS control register 1, SFR Address 0xcd, SFR bank 04)**

| Bit Number | Bit Mnemonic  | Function  | Access | Reset |
|------------|---------------|---|--------|-------|
| 7          | FocusREn      | Focus processing right channel enable bit<br>0: disable<br>1: enable      | R/W    | 1     |
| 6          | FocusLEn      | Focus processing left channel enable bit<br>0: disable<br>1: enable       | R/W    | 1     |
| 5          | DefinitionREn | Definition processing right channel enable bit<br>0: disable<br>1: enable | R/W    | 1     |
| 4          | DefinitionLEn | Definition processing left channel enable bit<br>0: disable<br>1: enable  | R/W    | 1     |

|     |             |   |     |   |
|-----|-------------|---|-----|---|
| 3   | TruBassEn   | TruBass enable:<br>0: disable<br>1: enable  | R/W | 1 |
| 2:0 | SpeakerSize | TruBass Speaker Low Pass Filter Response:<br>000: 40 Hz<br>001: 60 Hz<br>010: 100 Hz<br>011: 150 Hz<br>100: 200 Hz<br>101: 250 Hz<br>110: 300 Hz<br>111: 400 Hz | R/W | 0 |

### 8.2.3.3 SRSCtl2

**SRSCtl2 (SRS control register 2, SFR Address 0xce, SFR bank 04)**

| Bit Number | Bit Mnemonic   | Function  | Access | Reset |
|------------|----------------|---|--------|-------|
| 7:5        | Reserved       | Be read as “000”  | -      | -     |
| 4          | SRS3Den        | SRS3D enable:<br>0: disable<br>1: enable  | R/W    | 1     |
| 3:2        | SRS3Dmod       | SRS3D mode selection :<br>2'd0 : kSrsSRS3Dmono,<br>2'd1: kSrsSRS3DsinglSpeaker,<br>2'd2: kSrsSRS3Dstereo,<br>2'd3: kSrsSRS3Dextreme | R/W    | 2     |
| 1          | SRS3Dheadphone | 3D Headphone is selected or not selected:<br>0: false<br>1: true  | R/W    | 0     |
| 0          | HighBitRate    | HighBitRate is selected or not selected:<br>0: false<br>1: true   | R/W    | 1     |

### 8.2.3.4 SRS control variable

These variables are initialized in MURAM1 before SRS sound effect is enable:

| module                    | Mnemonic                             | Default value   | Description        |
|---------------------------|--------------------------------------|-----------------|--------------------|
| global<br>gain<br>control | nTruSurroundHDChannel.mInputGain     | 0x400000        | @Base+0            |
|                           | nTruSurroundHDChannel.mOutputGain    | 0x400000        | @Base+1            |
|                           | nTruSurroundHDChannel.mBypassGain    | 0x5ae148(0.71)  | @Base+2            |
|                           | nTruSurroundHDChannel.mSurroundLevel | 0x 4ccccd (0.6) | @Base+3 (TSHD use) |

|            |   |                |          |
|------------|---|----------------|----------|
|            |   |                | (only)   |
| limiter    | nTruSurroundHDChannel.mWowHDChannel.mLimiterChannel.mLimiterControl   | 0x600000(0.75) | @Base+4  |
| SRS3D      | nTruSurroundHDChannel.mWowHDChannel.SRS3Dchannel.mSpaceControl  | 0x666666(0.8)  | @Base+5  |
|            | nTruSurroundHDChannel.mWowHDChannel.SRS3Dchannel.mCenterControl   | 0x400000(0.5)  | @Base+6  |
| TruBass    | nTruSurroundHDChannel.mWowHDChannel.mTruBassChannel.mTruBassControl   | 0x266666(0.3)  | @Base+7  |
| Definition | nTruSurroundHDChannel.mWowHDChannel.mDefinitionChannelLeft.mDefinitionControl<br>mDefinitionChannelRight.mDefinitionControl | 0x266666(0.3)  | @Base+8  |
| Focus      | nTruSurroundHDChannel.mWowHDChannel.mFOCUSChannelLeft.mFOCUSControl<br>mFOCUSChannelRight.mFOCUSControl                     | 0x333333(0.4)  | @Base+9  |
|            | nTruSurroundHDChannel.MDialogClarityChannel.mFOCUSControl   | 0x 400000(0.5) | @Base+10 |

Note:

- (1) All these variables are signed Q0.23.
- (2) The total number of control variables is 11.

### 8.2.3.5 SRS filter temporary variables

These variables is in register groups during the WOWHD or Trusurround process. These variable should be popped into register group before the process of one frame and pushed into memory once the process of one frame is complete.

| Module     | Mnemonic  | Description     |
|------------|---|-----------------|
| TruBass    | mWowHDState.mTruBassState.mTruBassLowPassFilter1        | Number : only 1 |
|            | mWowHDState.mTruBassState.mTruBassLowPassFilter2        | Number : only 1 |
|            | mWowHDState.mTruBassState.mTruBassLowFilter             | Number : 4      |
|            | mWowHDState.mTruBassState.mTruBassMidFilter             | Number : 4      |
|            | mWowHDState.mTruBassState mFasdLevel                    | Number : 1      |
|            | mWowHDState.mTruBassState mIntegrator                   | Number : 1      |
|            | mWowHDState.mTruBassState mIntegrationLimit             | Number : 1      |
|            | mWowHDState.mTruBassState mAttack                       | Number : 1      |
|            | mWowHDState.mTruBassState mDecay                        | Number : 1      |
| SRS3D      | mWowHDState.Msrs3Dstate.mInitialPerspectiveFilter       | Number : 2      |
|            | mWowHDState.Msrs3Dstate.mFinalPerspectiveFilter         | Number : 1      |
|            | mWowHDState.Msrs3Dstate.mSinglePerspectiveAllPassFilter | Number : 1      |
|            | mWowHDState.Msrs3Dstate.mSinglePerspectiveFirstFilter   | Number : 2      |
|            | mWowHDState.Msrs3Dstate.mSinglePerspectiveSecondFilter  | Number : 2      |
|            | mWowHDState.Msrs3Dstate.mSinglePerspectiveThirdFilter   | Number : 2      |
| definition | mWowHDState mDefinitionStateLeft.mDefinitionFilter      | Number : 2      |

|                       |  |                     |
|-----------------------|--|---------------------|
|                       | mWowHDState.mDefinitionStateRight.mDefinitionFilter  | Number : 2          |
| focus                 | mWowHDState.mFOCUSStateLeft.mFOCUSFilter             | Number : 1          |
|                       | mWowHDState.mFOCUSStateLeft.mFOCUSFilter             | Number : 1          |
|                       | nTruSurroundHDState.mDialogClarityState.mFOCUSFilter | Number : 1          |
| limiter               | mWowHDState.MLimiterState.mLimiterLevel              | Number : 1*2(64bit) |
|                       | mWowHDState.MLimiterState.mDecay                     | Number : 1          |
| PerspectiveFront      | mPerspectiveFrontState.mHighPass48                   | Number : 1          |
|                       | mPerspectiveFrontState.mLowPass200                   | Number : 1          |
|                       | mPerspectiveFrontState.mHighPass7k                   | Number : 1          |
| PerspectiveRearCenter | mPerspectiveRearCenterState.mHighPass13k             | Number : 1          |
|                       | mPerspectiveRearCenterState.mLowPass8k               | Number : 1          |
|                       | mPerspectiveRearCenterState.mLowPass950              | Number : 1          |
| PerspectiveRearSpace  | mPerspectiveRearSpaceState.mHighPass13k              | Number : 1          |
|                       | mPerspectiveRearSpaceState.mLowPass8k                | Number : 1          |
|                       | mPerspectiveRearSpaceState.mLowPass950               | Number : 1          |

Note:

- (1) All these variables are signed Q23.0.
- (2) There is no need to store mWowHDState.Msrs3Dstate.mNormalizeGain, as this variable is refresh each one frame process.
- (3) There is no need to store mWowHDState.mTruBassState.mReferenceLevel as it is 0x0 if mLimiterChannel.mEnable == true or 0x400000(signed Q3.20) if mLimiterChannel.mEnable == false.
- (4) The total number of temporary variables should be stored in.muram1 is 44.

### 8.2.3.6 SRS constant table

These constants is store in murom1.

| Module  | Mnemonic                      | Qvalue       | Num | Description                                    |
|---------|-------------------------------|--------------|-----|--|
| TruBass | kTruBass40HzLowPassFilter_ph  | Signed Q0.23 | 2*8 | Low Pass Filter, -3Db at 40Hz                  |
|         | kTruBass40HzLowFilter_ph      | Signed Q0.23 | 3*8 | 40Hz Band Pass Filter, Gain = -2.5Db, Q = 2.1  |
|         | kTruBass40HzMidFilter_ph      | Signed Q0.23 | 3*8 | 70Hz Band Pass Filter, Gain = 6.1Db, Q = 1.34  |
|         | kTruBass60HzLowPassFilter_ph  | Signed Q0.23 | 2*8 | Low Pass Filter, -3Db at 60Hz                  |
|         | kTruBass60HzLowFilter_ph      | Signed Q0.23 | 3*8 | 61Hz Band Pass Filter, Gain = -2.5Db, Q = 2.1  |
|         | kTruBass60HzMidFilter_ph      | Signed Q0.23 | 3*8 | 105Hz Band Pass Filter, Gain = 6.1Db, Q = 1.34 |
|         | kTruBass100HzLowPassFilter_ph | Signed Q0.23 | 2*8 | Low Pass Filter, -3Db at 100Hz                 |
|         | kTruBass100HzLowFilter_ph     | Signed Q0.23 | 3*8 | 101Hz Band Pass Filter, Gain = -2.5Db, Q = 2.1 |

|         |                                |              |     |  |
|---------|--------------------------------|--------------|-----|--|
|         | kTruBass100HzMidFilter_ph      | Signed Q0.23 | 3*8 | 175Hz Band Pass Filter, Gain = 6.1Db, Q = 1.34                                       |
|         | kTruBass150HzLowPassFilter_ph  | Signed Q0.23 | 2*8 | Low Pass Filter, -3Db at 150Hz   |
|         | kTruBass150HzLowFilter_ph      | Signed Q0.23 | 3*8 | 151Hz Band Pass Filter, Gain = -2.5Db, Q = 2.1                                       |
|         | kTruBass150HzMidFilter_ph      | Signed Q0.23 | 3*8 | 263Hz Band Pass Filter, Gain = 6.1Db, Q = 1.34                                       |
|         | kTruBass200HzLowPassFilter_ph  | Signed Q0.23 | 2*8 | Low Pass Filter, -3Db at 200Hz   |
|         | kTruBass200HzLowFilter_ph      | Signed Q0.23 | 3*8 | 202Hz Band Pass Filter, Gain = -2.5Db, Q = 2.1                                       |
|         | kTruBass200HzMidFilter_ph      | Signed Q0.23 | 3*8 | 351Hz Band Pass Filter, Gain = 6.1Db, Q = 1.34                                       |
|         | kTruBass250HzLowPassFilter_ph  | Signed Q0.23 | 2*8 | Low Pass Filter, -3Db at 250Hz   |
|         | kTruBass250HzLowFilter_ph      | Signed Q0.23 | 3*8 | 252Hz Band Pass Filter, Gain = -2.5Db, Q = 2.1                                       |
|         | kTruBass250HzMidFilter_ph      | Signed Q0.23 | 3*8 | 439Hz Band Pass Filter, Gain = 6.1Db, Q = 1.34                                       |
|         | kTruBass300HzLowPassFilter_ph  | Signed Q0.23 | 2*8 | Low Pass Filter, -3Db at 300Hz   |
|         | kTruBass300HzLowFilter_ph      | Signed Q0.23 | 3*8 | 315Hz Band Pass Filter, Gain = 0.1Db, Q = 1.75                                       |
|         | kTruBass300HzMidFilter_ph      | Signed Q0.23 | 3*8 | 462Hz Band Pass Filter, Gain = 5.6Db, Q = 1.66                                       |
|         | kTruBass400HzLowPassFilter_ph  | Signed Q0.23 | 2*8 | Low Pass Filter, -3Db at 400Hz   |
|         | kTruBass400HzLowFilter_ph      | Signed Q0.23 | 3*8 | 420Hz Band Pass Filter, Gain = 1.5Db, Q = 1.77                                       |
|         | kTruBass400HzMidFilter_ph      | Signed Q0.23 | 3*8 | 568Hz Band Pass Filter, Gain = 5.4Db, Q = 1.82                                       |
|         | kAttack_ph                     | Signed Q0.23 | 8   | Fast Attack Coefficients   |
|         | kDecay_ph                      | Signed Q0.23 | 8   | Slow Decay Coefficients  |
| limiter | kLimiterDecay_ph               | Signed Q0.23 | 8   |  |
| SRS3D   | kInitialPerspectiveHHFilter_ph | -            | 5*8 | Initial 2 <sup>nd</sup> -order perspective curve filter, Headphone, High Bit Rate    |
|         | kInitialPerspectiveHHScale_ph  | -            | 8   |  |
|         | kInitialPerspectiveHLFilter_ph | -            | 5*8 | Initial 2n-order perspective curve filter, Headphone, Low Bit Rate                   |
|         | kInitialPerspectiveHLScale_ph  | -            | 8   |  |
|         | kInitialPerspectiveNHFILTER_ph | -            | 5*8 | Initial 2 <sup>nd</sup> -order perspective curve filter, No-Headphone, High Bit Rate |
|         | kInitialPerspectiveNHScale_ph  | -            | 8   |  |

|            |                                    |              |     |  |
|------------|------------------------------------|--------------|-----|--|
|            | kInitialPerspectiveNLFilter_ph     | -            | 5*8 | Initial 2 <sup>nd</sup> -order perspective curve filter, |
|            | kInitialPerspectiveNLScale_ph      | -            | 8   | No Headphone, Low Bit Rate                               |
|            | kFinalPerspectiveFilter_ph         | -            | 2*8 | Final first-order perspective curve filter               |
|            | kSinglePerspectiveAllPassFilter_ph | -            | 2*8 | First-order all pass perspective curve filter            |
|            | kSinglePerspectiveFirstFilter_ph   | -            | 5*8 | First second-order perspective curve filter              |
|            | kSinglePerspectiveSecondFilter_ph  | -            | 5*8 | Second second-order perspective curve filter             |
|            | kSinglePerspectiveThirdFilter_ph   | -            | 5*8 | Third second-order perspective curve filter              |
| definition | kDefinitionFilter_ph               | Signed Q1.22 | 5*8 | Definition Filter  |
| focus      | kFOCUSFilter_ph                    | Signed Q1.22 | 5*8 | FOCUS Filter   |
| TSHD       | kHighPass48Filter_ph               | -            | 3*3 | 48Hz HighPass for tshd                                   |
|            | kLowPass200Filter_ph               | -            | 3*3 | 200Hz LowPass  |
|            | kHighPass7kFilter_ph               |              | 3*3 | 7kHz HighPass  |
|            | kHighPass13kFilter_ph              |              | 3*3 | 13kHz HighPass   |
|            | kLowPass8kFilter_ph                |              | 3*3 | 8kHz LowPass   |
|            | kLowPass950Filter_ph               |              | 3*3 | 950 Hz LowPass   |
|            | kConstPhase200HzFilter             |              | 2*3 |  |
|            | kConstPhase1675HzFilter            |              | 2*3 |  |
|            | kConstPhase18kHzFilter             |              | 2*3 |  |
|            | kConstPhase50HzFilter              |              | 2*3 |  |
|            | kConstPhase600HzFilter             |              | 2*3 |  |
|            | kConstPhase4850HzFilter            |              | 2*3 |  |

Note:

- (1) The total number of constant coefficients is 1050.

### 8.2.3.7 SRS ram allocation

The Ram allocation for TrusurroundHD and WOWHD are listed below

| function      | Mnemonic | allocation         | Description  |
|---------------|----------|--------------------|--|
| TruSurroundHD | Iobuffer | sTempBuffer[4][32] | Input and Output buffer                            |
|               | tshdTemp | tshdTemp[5][32]    | temporary buffers required for internal processing |
| WowHD         | Iobuffer | sTempBuffer[2][32] | Input and Output buffer                            |
|               | tshdTemp | tshdTemp[5][32]    | temporary buffers required for internal processing |

## 8.3 文件系统加速器模块（黄少彬、蔡瑞仁）

| 日期         | 版本     | 描述      | 修订人 |
|------------|--------|---------|-----|
| 2012-07-24 | V1. 00 | initial | 黄少彬 |

### 8.3.1 描述

文件系统加速器的目的是处理文件系统中一些单一而重复的运算。

加速单元功能有两个：

1、对 RAM 中的一段数据，按照一定的数据单元（2byte 或者 4byte）进行无符号数累加（最高位无进位）。累加和放在寄存器中。此功能可应用于固件校验中。

2、对 RAM 中的一段数据(长度为 block\_length 的倍数)，在其中搜索一个匹配数据(2byte 或者 4byte，内容可配置)，并输出：

(1) 以数据单元长度 block\_length 为单位，在一个单元长度中如果搜索到匹配数据，则向寄存器输出 1 个 bit。

(2) 在 length 长度中，有多少个匹配数据。

此功能可应用于磁盘空间簇的收集中。

#### 8.3.1.1 文件系统加速器框图

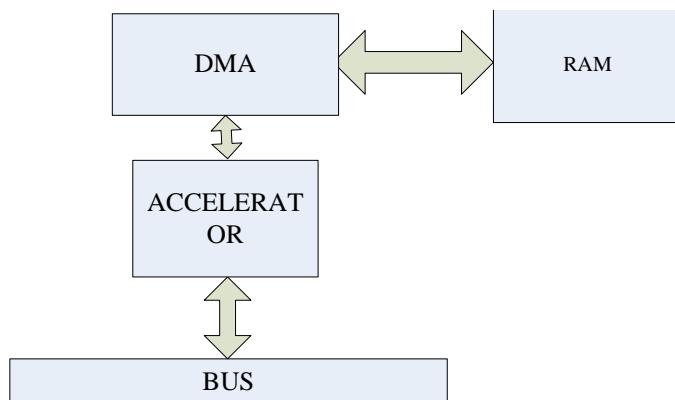


Figure 8-21 加速器共用 Flash Controller 的数据通道 DMA

#### 8.3.1.2 加速器时钟

加速器的时钟由 CMU 提供。

### 8.3.2 ACC Register List

| Index | Mnemonic                    | Description  | BANK |
|-------|-----------------------------|--|------|
| 0x89  | <b>ACC_CTRL</b>             | Control Register   | Bank |
| 0x8a  | <b>ACC_BLKLEN_LENHI</b>     | Search block length or the low byte of data length of unsigned accumulation unit number  | 0x08 |
| 0x8b  | <b>ACC_BLKNUM_LENLO</b>     | Search block number or the high byte of data length of unsigned accumulation unit number | 0x08 |
| 0xf1  | <b>ACC_CMPDATA0_SU_MHIE</b> | The first byte of data searched or the higher byte of unsigned accelerator's result      | 0x08 |
| 0xf2  | <b>ACC_CMPDATA1_SU_MHI</b>  | The second byte of data searched or the high byte of unsigned accelerator's result       | 0x08 |
| 0xf3  | <b>ACC_CMPDATA2_SU_MLO</b>  | The third byte of data searched or the low byte of unsigned accelerator's result         | 0x08 |
| 0xf4  | <b>ACC_CMPDATA3_SU_MLOE</b> | The fourth byte of data searched or the lower byte of unsigned accelerator's result      | 0x08 |
| 0xf5  | <b>ACC_MATCNTHI</b>         | The high byte of bitmap  | 0x08 |
| 0xf6  | <b>ACC_MATCNTLO</b>         | The low byte of match number   | 0x08 |
| 0xf7  | <b>ACC_BITMAPPHI</b>        | The high byte of bitmap  | 0x08 |
| 0xf8  | <b>ACC_BITMAPLO</b>         | The low byte of bitmap   | 0x08 |
| 0x8c  | <b>ACC_FIFO</b>             | ACC FIFO address   | 0x08 |

### 8.3.3 Registers Description

#### 8.3.3.1 ACC\_CTRL

Compute Accelerator: **Control Register**

**offset= 0x89**

| Bits | Name        | Description  | Access | Reset |
|------|-------------|--|--------|-------|
| 7    | CLKEN       | 0: clock disable<br>1: clock enable                          | RW     | 0     |
| 6:4  | /           | Reserved   | R      | 0     |
| 3    | ENDIAN_MODE | 0: little<br>1: big (only sensitive to unsigned accumulator) | RW     | 0     |
| 2    | UNIT        | Unit of accumulation or searching<br>0: 2 byte<br>1: 4 byte  | RW     | 0     |
| 1    | MODE        | 0: unsigned accumulator<br>1: data search                    | RW     | 0     |

|   |       |   |    |   |
|---|-------|---|----|---|
| 0 | START | 1: enable compute accelerator<br>When setting this bit high start dma5 and the compute accelerator start work, when computation finished, the bit auto clear. | RW | 0 |
|---|-------|---|----|---|

### 8.3.3.2 ACC\_BLKLEN\_LENHI

Compute Accelerator: **Search block length or the high byte of data length of unsigned accumulation unit number.**

Offset= 0x8a

| Bits | Name              | Description   | Access | Reset |
|------|-------------------|---|--------|-------|
| 7:0  | BLK_LEN/<br>LENHI | When used for data search, means block length:<br>0: Block length = 512 bytes;<br>1: Block length = 1024 bytes;<br>2: Block length = 2048 bytes;<br>3: Block length = 4096 bytes;<br>Others: reserved.<br>When used for unsigned data accelerator, means the high byte of accumulation byte number. | RW     | 0     |

### 8.3.3.3 ACC\_BLKNUM\_LENLO

Compute Accelerator: **Search block number or the low byte of data length of unsigned accumulation unit number**

offset= 0x8b

| Bits | Name              | Description   | Access | Reset |
|------|-------------------|---|--------|-------|
| 7:0  | BLK_NUM/<br>LENLO | When used for data search, means block numbers(the max block num is 16);<br>When used for unsigned accelerator, means the low byte of accumulation byte number. | RW     | 1     |

### 8.3.3.4 ACC\_CMPDATA0\_SUMHIE

Compute Accelerator: **The first byte of data searched or the lower byte of unsigned accelerator's result**

offset= 0xf1

| Bits | Name                | Description   | Access  | Reset |
|------|---------------------|---|---|-------|
| 7:0  | CMPDATA0/<br>SUMHIE | When used for non-signal adder, the register mapping as PIC 10-13<br>When used for data searching the register mapping as PIC 10-14 | ACC_CTRL.MODE=0 :R ;<br>ACC_CTRL.MODE=1 :RW ; | 0     |

ATTENTION:

When ACC\_CTRL.acc\_mode = 0, the SUMDATA configure as:

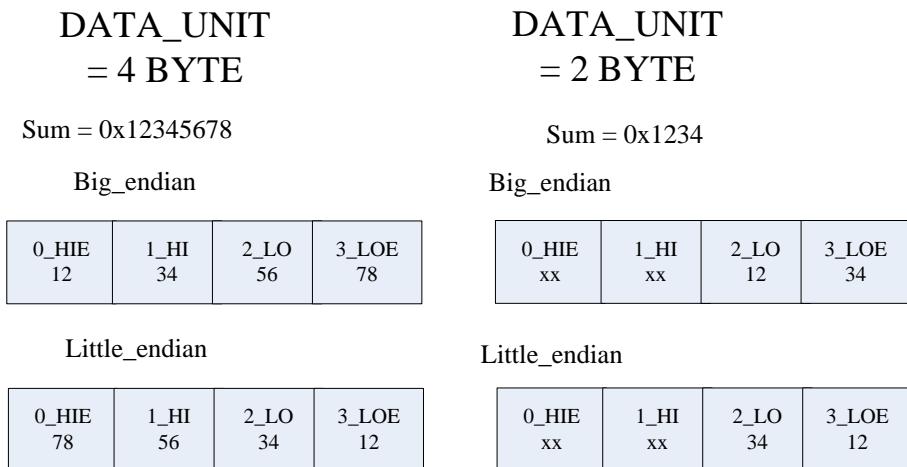


Figure 8-22 The register and sum result mapping picture

When ACC\_CTRL.acc\_mode = 1, the CMPDATA configure as:

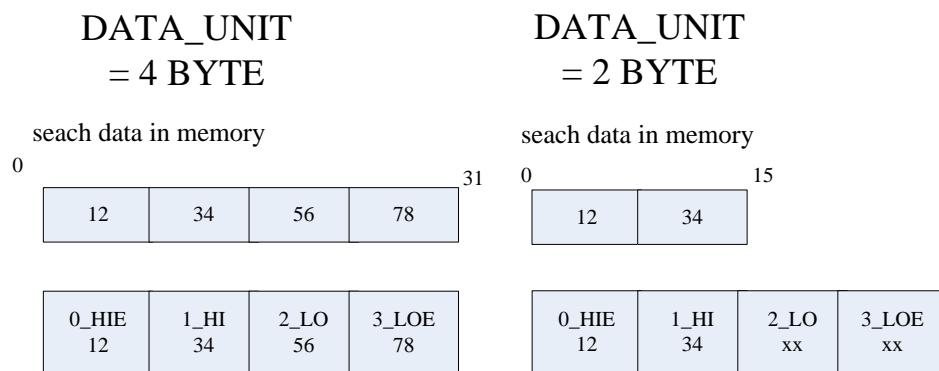


Figure 8-23 The register and comparing data mapping picture

### 8.3.3.5 ACC\_CMPDATA1\_SUMHI

Compute Accelerator: **The second byte of data searched or the high byte of unsigned accelerator's result offset= 0xf2**

| Bits | Name               | Description   | Access  | Reset |
|------|--------------------|---|---|-------|
| 7:0  | CMPDATA1/<br>SUMHI | When used for non-signal adder, the register mapping as PIC 10-13<br><br>When used for data searching the register mapping as PIC 10-14 | ACC_CTRL.MODE=0 :R ;<br>ACC_CTRL.MODE=1 :RW ; | 0     |

### 8.3.3.6 ACC\_CMPDATA2\_SUMLO

Compute Accelerator: **The third byte of data searched or the low byte of unsigned accelerator's result offset= 0xf3**

| Bits | Name               | Description   | Access  | Reset |
|------|--------------------|---|---|-------|
| 7:0  | CMPDATA2/<br>SUMLO | When used for non-signal adder, the register mapping as PIC 10-13 | ACC_CTRL.MODE=0 :R ;<br>ACC_CTRL.MODE=1 :RW ; | 0     |

|  |   |  |  |
|--|---|--|--|
|  | When used for data searching the register mapping as PIC<br>10-14 |  |  |
|--|---|--|--|

### 8.3.3.7 ACC\_CMPDATA3\_SUMLOE

Compute Accelerator: **The fourth byte of data searched or the lower byte of unsigned accelerator's result**  
**offset= 0xf4**

| Bits | Name                | Description   | Access  | Reset |
|------|---------------------|---|---|-------|
| 7:0  | CMPDATA3/<br>SUMLOE | When used for non-signal adder, the register mapping as<br>PIC 10-13<br><br>When used for data searching the register mapping as PIC<br>10-14 | ACC_CTRL.MODE=0 :R ;<br>ACC_CTRL.MODE=1 :RW ; | 0     |

### 8.3.3.8 ACC\_MATCNTHI

Compute Accelerator: **The high byte of match number**  
**offset= 0xf5**

| Bits | Name     | Description                          | Access | Reset |
|------|----------|--------------------------------------|--------|-------|
| 7:0  | MATCNTHI | The <b>high</b> byte of match number | R      | 0     |

### 8.3.3.9 ACC\_MATCNTLO

Compute Accelerator: **The low byte of match number**  
**offset= 0xf6**

| Bits | Name     | Description                  | Access | Reset |
|------|----------|------------------------------|--------|-------|
| 7:0  | MATCNTLO | The low byte of match number | R      | 0     |

### 8.3.3.10 ACC\_BITMAPPHI

Compute Accelerator: **The high byte of bitmap**  
**offset= 0xf7**

| Bits | Name     | Description                             | Access | Reset |
|------|----------|---|--------|-------|
| 7:0  | BITMAPHI | The high byte of empty cluster's bitmap | R      | 0     |

### 8.3.3.11 ACC\_BITMAPPLO

Compute Accelerator: **The low byte of bitmap**  
**offset= 0xf8**

| Bits | Name     | Description                            | Access | Reset |
|------|----------|--|--------|-------|
| 7:0  | BITMAPLO | The low byte of empty cluster's bitmap | R      | 0     |

### 8.3.3.12 ACC\_FIFO

Compute Accelerator: **The low byte of bitmap**

**offset= 0x8c**

| Bits | Name     | Description           | Access | Reset |
|------|----------|-----------------------|--------|-------|
| 7:0  | ACC_FIFO | ACC data fifo address | R/W    | 0     |

## 8.3.4 加速器软件调用

加速器处理的是 RAM 中的数据，对 RAM 中的数据做无符号加法或者数据搜索，根据不同功能，软件调用如下。

### 8.3.4.1 无符号累加的软件调用

加速器启动的使用需要配置的寄存器：

- 1、ACC\_CTRL, CLKEN 位置 1
  - 2、ACC\_CTRL, ENA 位置 1, MODE 位置 0, 配置 UNIT 位指示当前操作的数据单元，根据不同大  
小端情况配置 ENDIAN\_MODE
  - 3、ACC\_BLKLEN\_LENLO/HI, 累加数据的单元个数长度
- 最后设置 ACC\_CTRL 的 START 位，  
2044、配置并启动 DMA

软件这时候轮询 ACC\_CTRL 的 START 为 0 的话，表示运算结束。

然后软件读 ACC\_CMPDATALO\_SUMLO 等四个寄存器，累加和结果，与在 sram 中的累加和再做  
和运算。

### 8.3.4.2 数据搜索的软件调用

加速器启动的使用需要配置的寄存器：

- 1、ACC\_CTRL, CLKEN 位置 1
  - 2、ACC\_CTRL, ENA 位置 1, MODE 位置 1, 配置 UNIT 位指示当前操作的数据单元
  - 3、ACC\_BLKLEN\_LENLO, BLOCK 的长度
  - 4、ACC\_BLKNUM\_LENHI, BLOCK NUMBER
  - 5、ACC\_CMPDATALO\_SUMLO 等四个寄存器, 搜索目的数据
- 最后设置 ACC\_CTRL 的 START 位，  
2045、配置并启动 DMA

软件这时候轮询 ACC\_CTRL 的 START 为 0 的话，表示运算结束。

然后软件读 ACC\_MATCNTHI/ACC\_MATCNTLO, 表示空簇的数目, 与在 sram 中的累加和再做和运算。

然后软件读 ACC\_BITMAPLO/ACC\_BITMAPHI, 表示空簇的位图, 加在 sram 中的位图表中。

## 9 Touch Key Controller (刘惠民、陈国安、黄俏)

| 日期         | 版本     | 描述   | 修订人 |
|------------|--------|--|-----|
| 2012-10-20 | V2. 00 | 新增 touch-key 模块  | 刘惠民 |
| 2012-11-07 | V2. 01 | <ol style="list-style-type: none"> <li>因为 PWM 的 N 可调, 因此在 TK 电流自动调节模块中需要增加 TK1 和 TK2 可调, 在 spec 当中加入了 4 个寄存器来设置 TK1 和 TK2:<br/> <u>TK_AUTO_LOWTH_H</u><br/> <u>TK_AUTO_LOWTH_L</u><br/> <u>TK_AUTO_HIGHTH_H</u><br/> <u>TK_AUTO_HIGHTH_L</u></li> <li>根据第 1 点, 调整了寄存器的地址, 更新了 register list;</li> <li>调整了 TK_PRESS_TH 各个档位的阈值, 并将默认值修改为 01101;<br/>         将 TK_NOISE_TH 的默认值修改为 1000;</li> <li>调整了 debug 信号的排列, 这样 coding 时好例化</li> </ol>  | 刘惠民 |
| 2012-12-12 | V2. 02 | <ol style="list-style-type: none"> <li>增加 <u>TK_RAW_PD</u> 和以下 24 个寄存器, 不然 FPGA 和 IC 无法验证 baseline 的自动跟踪功能:<br/> <u>TK_KEY0_RAWH</u><br/> <u>TK_KEY0_RAWL</u><br/> <u>TK_KEY1_RAWH</u><br/> <u>TK_KEY1_RAWL</u><br/> <u>TK_KEY2_RAWH</u><br/> <u>TK_KEY2_RAWL</u><br/> <u>TK_KEY3_RAWH</u><br/> <u>TK_KEY3_RAWL</u><br/> <u>TK_KEY4_RAWH</u><br/> <u>TK_KEY4_RAWL</u><br/> <u>TK_KEY5_RAWH</u><br/> <u>TK_KEY5_RAWL</u><br/> <u>TK_KEY0_BL_H</u><br/> <u>TK_KEY0_BL_L</u><br/> <u>TK_KEY1_BL_H</u><br/> <u>TK_KEY1_BL_L</u><br/> <u>TK_KEY2_BL_H</u><br/> <u>TK_KEY2_BL_L</u></li> </ol> | 刘惠民 |

|            |       |  |     |
|------------|-------|--|-----|
|            |       | <a href="#">TK_KEY3_BL_H</a><br><a href="#">TK_KEY3_BL_L</a><br><a href="#">TK_KEY4_BL_H</a><br><a href="#">TK_KEY4_BL_L</a><br><a href="#">TK_KEY5_BL_H</a><br><a href="#">TK_KEY5_BL_L</a> |     |
| 2013-07-05 | V2.04 | 2. 一共 25 个寄存器，但是只有一个寄存器是可读写的，其他的 24 个寄存器是只读的，对于寄存器资源消耗微弱；<br>3. 增加软件 operation manual 说明部分<br><br>在 operation manual 部分，增加按键使能控制的说明，这是由于实际设计中按键组合逻辑存在问题导致的；                                  | 刘惠民 |

## 9.1 Features

- ◆ Support cap sense key ,standard value from 5Pf to 50 Pf
- ◆ Support six independent capacitive touch keys
- ◆ LP (low power operation) mode and normal mode
- ◆ Auto-calibration
- ◆ Debounce prevention
- ◆ Baseline Correction

## 9.2 Function Description

This TKC detects the TK base on the change of parasitic capacity after touching, it can support cap value on PCB from 5Pf to 50 Pf. When the cap was touched, the cap variance will be 0.5Pf~5Pf, the least variance ratio is 1/100. So, the most important thing is to find a method to detect the variance of cap but not how much. For this TKC, we use relaxation oscillation method.

This TKC support five independent Cap Sense keys with Auto-calibration, debounce prevention and Baseline Correction function, and It has two operation mod including LP mode and normal mode.

### 9.2.1 Auto-calibration

TKC can calibrate Cap Sense parameter of every key automatically to compensate PCB difference and environment changes, and it can judge the calibrated value is effective or not ( $|\Delta n|$  sum bigger than BUCKET\_TH effective, otherwise not effective). If effective, update the Cap Sense parameter by adjusting Baseline through adding or subtracting one step value determined by BL\_STEP, otherwise maintain the parameter. Auto-calibration is done during initiation of the system and for a constant time that can be set by CAL\_CYC in TK\_PRM, and it can be enabled by setting AUTO\_CAL\_EN of register TK\_CTL.

## 9.2.2 Baseline Correction

When user touches TK for a mistake during initiation of the system, the Baseline of the TK detected by TKC will be incorrect. Baseline Correction function can detect the TK for several cycles in order to avoid this problem. It is enabled by setting BL\_CAL\_EN.

$$\Delta n = n_{\text{later}} - n_{\text{first}}$$

$\Delta n$  is the value of latter Baseline( $n_{\text{later}}$ ) subtract the Baseline( $n_{\text{first}}$ ) got for the first time. If  $\Delta n$  is bigger than BL\_CAL\_TH. The Baseline will be updated by the latter Baseline.

## 9.2.3 Debounce prevention

This TKC has Debounce prevention function. It confirms key touching only when a key touching was detected for DB\_NUM scan cycles.

## 9.2.4 Operation mode

This TKC can enter LP mode when not key was detected for several Normal detect cycles, and then use LP\_CYC as detecting clock during LP mode. The LP\_CYC Duty cycle is very low, so the power consumption can be very low. Once any key was detected during LP mode, TKC can return to normal mode immediately. LP mode can be enabled by setting LPEN.

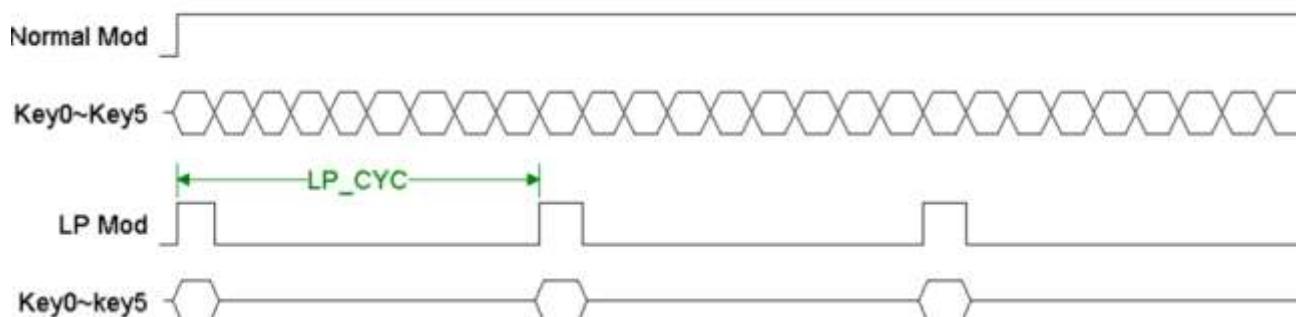


Figure 9-1 Operation Timing

Note: LP\_CYC is the LP mode detecting cycle, It can be set by LP\_CYC in register TK\_PRM

## 9.3 Module Description

### 9.3.1 Block Diagram

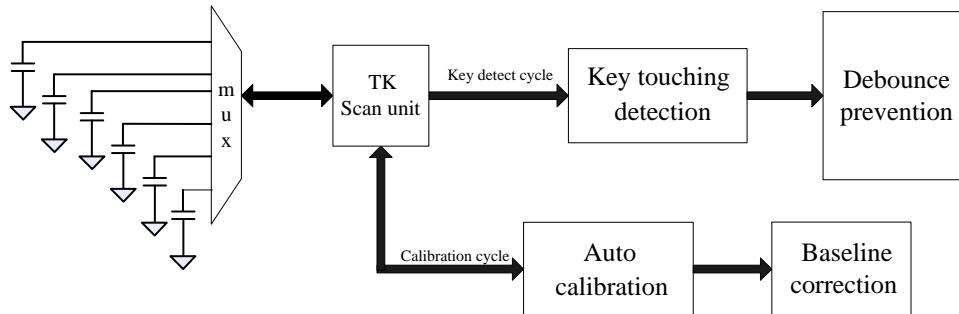


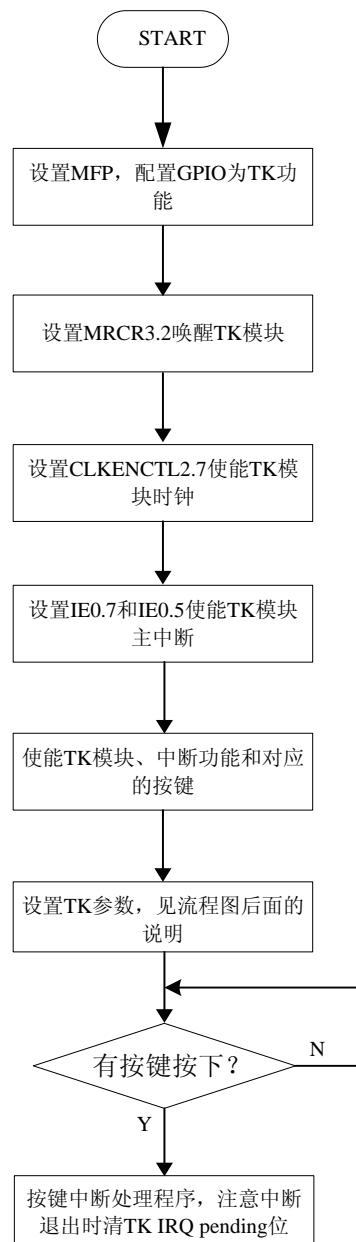
Figure 9-2 Touch Key Controller Block Diagram

### 9.3.2 Signal list

| Signal    | Input/Output | Description              |
|-----------|--------------|--------------------------|
| Tkey0     | I/O          | Touch Key 0 input signal |
| Tkey1     | I/O          | Touch Key 1 input signal |
| Tkey2     | I/O          | Touch Key 2 input signal |
| Tkey3     | I/O          | Touch Key 3 input signal |
| Tkey4     | I/O          | Touch Key 4 input signal |
| Tkey5     | I/O          | Touch Key 5 input signal |
| TK_SHIELD | I/O          | Touch Key Shield signal  |

## 9.4 Operation Manual

### 9.4.1 软件控制流程



1. TK模块一共有6个触摸按键，软件可以根据需要确定使用个数；6个触摸按键的中断入口都是一样的，都是中断向量表入口地址偏置0x2b；在中断处理程序中，可以通过TK\_IRQ\_PD来判断是哪个KEY发生了触摸事件，注意退出中断处理程序时，需要清TK\_IRQ\_PD标志位，否则会不停进入TK中断处理程序；
2. 由于实际设计的缘故，TK按键的组合逻辑存在问题；例如，希望使用TK0/1/2/3/5，那么对TK\_EN\_CTL写入0x5f，但是实际上TK3不可用；所以使用按键时，建议将全部触摸按键都使能，通过GPIO的MFP切

换来决定使用哪个按键，这样就能够避免这个问题；

3. TK模块提供了触摸按键LOCK功能，如果使能LOCK功能，那么任何触碰都将无法触发按键中断响应；只有软件取消LOCK功能才能够让触控按键恢复响应；
4. 软件可以在使用之前，配置触摸按键的控制参数；在应用中软件必须配置的参数有：

TK\_AUTO\_LOWTH\_H/L和TK\_AUTO\_HIGHTH\_H/L；

- (1) 如果应用采用TK充电电流手动配置模式，那么需要将TK\_AUTO\_LOWTH\_H/L配置为零，这是因为状态机中TK初始化时会将第一次充电时候得到的baseline值和LOWTH、HIGHTH进行比较：如果baseline < LOWTH或者baseline > HIGHTH，那么就会不断执行将新的raw值赋给baseline；如果此时因为充电电流的设置，使得硬件内部计算得到的raw值不满于大于LOWTH或者小于HIGHTH，那么这个时候TK的baseline就会不断地跟随raw值更新，识别不到手指的触控；TK充电电流手动配置时，应该将TK\_AUTO\_LOWTH\_H/L配置为0，TK\_AUTO\_HIGHTH\_H/L可以不变，按照默认值配置；
- (2) 如果应用采用TK充电电流自动配置模式，那么也可以将TK\_AUTO\_LOWTH\_H/L设置为零；当然根据实测效果，将LOWTH和HIGHTH设置成合适的数值；

可能需要调整的参数有：

- (1) 如果需要调整按键的灵敏度，可以设置TK\_DB\_CTL寄存器中的DB\_NUM，有四档参数可以配置，设置值越大，灵敏度越低，越不容易误触发；
- (2) 如果需要调整按键的触摸阈值，可以设置TK\_PRESS\_TH，有32档参数可以配置，设置值越大，同样灵敏度越低，越不容易误触发；

其余的可配置参数是为了后续TK优化而设置的，软件不用关心；

## 9.5 Register List

**Table TK Controller Registers**

| Index | Mnemonic     | Description                            | BANK |
|-------|--------------|--|------|
| 0x90  | TK_EN_CTL    | Touch Key Control Register.            | 0x14 |
| 0x91  | TK_FUN_CTL   | Touch Key Function Control Register.   | 0x14 |
| 0x92  | TK_IRQ_PD    | Touch Key IRQ Pending Status Register. | 0x14 |
| 0x93  | TK_PRESS_STA | Touch Key Press Status Register.       | 0x14 |

|      |                  |   |      |
|------|------------------|---|------|
| 0x94 | TK_DB_CTL        | Touch Key Debounce Control Register.                          | 0x14 |
| 0x95 | TK_PWM_CNTH      | Touch Key PWM Counter High Register.                          | 0x14 |
| 0x97 | TK_PWM_CNTL      | Touch Key PWM Counter Low Register.                           | 0x14 |
| 0x99 | TK_PRESS_TH      | Touch Key Press Threshold Register.                           | 0x14 |
| 0x9A | TK_BL_CAL_TH     | Touch Key Baseline Calibration Threshold Register.            | 0x14 |
| 0x9B | TK_NOISE_TH      | Touch Key Noise Threshold Register.                           | 0x14 |
| 0x9C | TK_BUCKET_TH     | Touch Key Bucket Threshold And Baseline Step Adjust Register. | 0x14 |
| 0x9D | TK_AUTO_LOWTH_H  | Touch Key idac auto adjust low threshold high bits Register.  |      |
| 0x9E | TK_AUTO_LOWTH_L  | Touch Key idac auto adjust low threshold low bits Register.   |      |
| 0xa2 | TK_AUTO_HIGHTH_H | Touch Key idac auto adjust high threshold high bits Register. |      |
| 0xa3 | TK_AUTO_HIGHTH_L | Touch Key idac auto adjust high threshold low bits Register.  |      |
| 0xa4 | TK_KEY0_IDA_CTL  | Touch Key0 charging current Control Register.                 | 0x14 |
| 0xa5 | TK_KEY1_IDA_CTL  | Touch Key1 charging current Control Register.                 | 0x14 |
| 0xa6 | TK_KEY2_IDA_CTL  | Touch Key2 charging current Control Register.                 | 0x14 |
| 0xa9 | TK_KEY3_IDA_CTL  | Touch Key3 charging current Control Register.                 | 0x14 |
| 0xaa | TK_KEY4_IDA_CTL  | Touch Key4 charging current Control Register.                 | 0x14 |
| 0xab | TK_KEY5_IDA_CTL  | Touch Key5 charging current Control Register.                 | 0x14 |
| 0xad | TK_UPDAT_PD      | Touch Key New Data Update Pending Register.                   | 0x14 |
| 0xae | TK_KEY0_DATH     | Touch Key0 current data high bits.                            | 0x14 |
| 0xaf | TK_KEY0_DATL     | Touch Key0 current data low bits.                             | 0x14 |
| 0xb0 | TK_KEY1_DATH     | Touch Key1 current data high bits.                            | 0x14 |
| 0xb1 | TK_KEY1_DATL     | Touch Key1 current data low bits.                             | 0x14 |
| 0xb2 | TK_KEY2_DATH     | Touch Key2 current data high bits.                            | 0x14 |
| 0xb3 | TK_KEY2_DATL     | Touch Key2 current data low bits.                             | 0x14 |
| 0xb4 | TK_KEY3_DATH     | Touch Key3 current data high bits.                            | 0x14 |
| 0xb5 | TK_KEY3_DATL     | Touch Key3 current data low bits.                             | 0x14 |

|      |              |   |      |
|------|--------------|---|------|
| 0Xb6 | TK_KEY4_DATH | Touch Key4 current data high bits.              | 0x14 |
| 0Xb7 | TK_KEY4_DATL | Touch Key40 current data low bits.              | 0x14 |
| 0Xb8 | TK_KEY5_DATH | Touch Key5 current data high bits.              | 0x14 |
| 0Xb9 | TK_KEY5_DATL | Touch Key5 current data low bits.               | 0x14 |
| 0Xba | TK_RAW_PD    | Touch Key New RAW Data Update Pending Register. | 0x14 |
| 0Xbc | TK_KEY0_RAWH | Touch Key0 current RAW high bits.               | 0x14 |
| 0Xbd | TK_KEY0_RAWL | Touch Key0 current RAW low bits.                | 0x14 |
| 0Xbf | TK_KEY1_RAWH | Touch Key1 current RAW high bits.               | 0x14 |
| 0Xc1 | TK_KEY1_RAWL | Touch Key1 current RAW low bits.                | 0x14 |
| 0Xc2 | TK_KEY2_RAWH | Touch Key2 current RAW high bits.               | 0x14 |
| 0Xc3 | TK_KEY2_RAWL | Touch Key2 current RAW low bits.                | 0x14 |
| 0Xc4 | TK_KEY3_RAWH | Touch Key3 current RAW high bits.               | 0x14 |
| 0Xc5 | TK_KEY3_RAWL | Touch Key3 current RAW low bits.                | 0x14 |
| 0Xc6 | TK_KEY4_RAWH | Touch Key4 current RAW high bits.               | 0x14 |
| 0Xc7 | TK_KEY4_RAWL | Touch Key4 current RAW low bits.                | 0x14 |
| 0Xc8 | TK_KEY5_RAWH | Touch Key5 current RAW high bits.               | 0x14 |
| 0Xc9 | TK_KEY5_RAWL | Touch Key5 current RAW low bits.                | 0x14 |
| 0Xca | TK_KEY0_BL_H | Touch Key0 current baseline high bits.          | 0x14 |
| 0Xcc | TK_KEY0_BL_L | Touch Key0 current baseline low bits.           | 0x14 |
| 0Xcd | TK_KEY1_BL_H | Touch Key1 current baseline high bits.          | 0x14 |
| 0Xce | TK_KEY1_BL_L | Touch Key1 current baseline low bits.           | 0x14 |
| 0Xd2 | TK_KEY2_BL_H | Touch Key2 current baseline high bits.          | 0x14 |
| 0Xd3 | TK_KEY2_BL_L | Touch Key2 current baseline low bits.           | 0x14 |
| 0Xd4 | TK_KEY3_BL_H | Touch Key3 current baseline high bits.          | 0x14 |
| 0Xd5 | TK_KEY3_BL_L | Touch Key3 current baseline low bits.           | 0x14 |
| 0Xd6 | TK_KEY4_BL_H | Touch Key4 current baseline high bits.          | 0x14 |
| 0Xd7 | TK_KEY4_BL_L | Touch Key4 current baseline low bits.           | 0x14 |
| 0Xd8 | TK_KEY5_BL_H | Touch Key5 current baseline high bits.          | 0x14 |

|      |              |                                       |      |
|------|--------------|---------------------------------------|------|
| 0Xd9 | TK_KEY5_BL_L | Touch Key5 current baseline low bits. | 0x14 |
| 0Xda | TK_DEBUG     | Touch Key Debug Register.             | 0x14 |

## 9.6 Register Description

### 9.6.1 TK\_EN\_CTL

Touch Key Control Register.

Offset = 0x90

| Bits | Name   | Description   | R/W | Reset |
|------|--------|---|-----|-------|
| 7    | -      | Reserved  | R   | 0     |
| 6    | TK5_EN | KEY5 sense enabled<br>0: Disable<br>1: Enable           | R/W | 0     |
| 5    | TK4_EN | KEY4 sense enabled<br>0: Disable<br>1: Enable           | R/W | 0     |
| 4    | TK3_EN | KEY3 sense enabled<br>0: Disable<br>1: Enable           | R/W | 0     |
| 3    | TK2_EN | KEY2 sense enabled<br>0: Disable<br>1: Enable           | R/W | 0     |
| 2    | TK1_EN | KEY1 sense enabled<br>0: Disable<br>1: Enable           | R/W | 0     |
| 1    | TK0_EN | KEY0 sense enabled<br>0: Disable<br>1: Enable           | R/W | 0     |
| 0    | TK_EN  | TK controller Enable enabled<br>0: Disable<br>1: Enable | R/W | 0     |

### 9.6.2 TK\_FUN\_CTL

Touch Key Function Control Register.

Offset = 0x91

| Bits | Name | Description | R/W | Reset |
|------|------|-------------|-----|-------|
|      |      |             |     |       |

|     |               |   |     |   |
|-----|---------------|---|-----|---|
| 7:3 | -             | Reserved  | R   | 0 |
| 2   | IDA_ADJUST_EN | IDAC adjust (Baseline range Correction) enabled<br>0: Disable<br>1: Enable<br><b>Note:</b> It is used to adjust the charging current of internal current source.  | R/W | 0 |
| 1   | TK_INT_EN     | TK Interrupt enabled<br>0: Disable<br>1: Enable   | R/W | 0 |
| 0   | TK_LOCK       | TKC Lock Bit<br>0: Normal<br>When system quit Lock state, this bit should be cleared by writing 0;<br>1: Lock TKC<br>When system enter Lock state, this bit should be set.<br>When Lock TKC, TKC do not respond any TK but Auto-calibration is working. | R/W | 0 |

### 9.6.3 TK\_IRQ\_PD

Touch Key IRQ Pending Status Register.

Offset = 0x92

| Bits | Name       | Description   | R/W | Reset |
|------|------------|---|-----|-------|
| 7:6  | -          | Reserved  | R   | 0     |
| 5    | TK5_IRQ_PD | TKEY5 IRQ Pending Bit, Write 1 to clear this bit;<br>0: Not IRQ Pending<br>1: IRQ Pending | R/W | 0     |
| 4    | TK4_IRQ_PD | TKEY4 IRQ Pending Bit, Write 1 to clear this bit;<br>0: Not IRQ Pending<br>1: IRQ Pending | R/W | 0     |
| 3    | TK3_IRQ_PD | TKEY3 IRQ Pending Bit, Write 1 to clear this bit;<br>0: Not IRQ Pending<br>1: IRQ Pending | R/W | 0     |
| 2    | TK2_IRQ_PD | TKEY2 IRQ Pending Bit, Write 1 to clear this bit;<br>0: Not IRQ Pending<br>1: IRQ Pending | R/W | 0     |
| 1    | TK1_IRQ_PD | TKEY1 IRQ Pending Bit, Write 1 to clear this bit;<br>0: Not IRQ Pending<br>1: IRQ Pending | R/W | 0     |

|   |            |   |     |   |
|---|------------|---|-----|---|
| 0 | TK0_IRQ_PD | TKEY0 IRQ Pending Bit, Write 1 to clear this bit;<br>0: Not IRQ Pending<br>1: IRQ Pending | R/W | 0 |
|---|------------|---|-----|---|

## 9.6.4 TK\_PRESS\_STA

Touch Key Press Status Register.

Offset = 0x93

| Bits | Name      | Description   | R/W | Reset |
|------|-----------|---|-----|-------|
| 7:6  | -         | Reserved  | R   | 0     |
| 5    | TK5_PRESS | TKEY5 Status in Normal Mode<br>0: key not pressed<br>1: key pressed | R   | 0     |
| 4    | TK4_PRESS | TKEY4 Status in Normal Mode<br>0: key not pressed<br>1: key pressed | R   | 0     |
| 3    | TK3_PRESS | TKEY3 Status in Normal Mode<br>0: key not pressed<br>1: key pressed | R   | 0     |
| 2    | TK2_PRESS | TKEY2 Status in Normal Mode<br>0: key not pressed<br>1: key pressed | R   | 0     |
| 1    | TK1_PRESS | TKEY1 Status in Normal Mode<br>0: key not pressed<br>1: key pressed | R   | 0     |
| 0    | TK0_PRESS | TKEY0 Status in Normal Mode<br>0: key not pressed<br>1: key pressed | R   | 0     |

## 9.6.5 TK\_DB\_CTL

Touch Key Debounce Control Register.

Offset = 0x94

| Bits | Name  | Description                             | R/W | Reset |
|------|-------|---|-----|-------|
| 7    | -     | Reserved                                | R   | 0     |
| 6:2  | PWM_N | PWM timer N select, timer N = PWM_N + 1 | R/W | 0x10  |

|     |        |                     |  |     |    |
|-----|--------|---------------------|--|-----|----|
| 1:0 | DB_NUM | Key Debounce Number |  | R/W | 00 |
|     |        | 00 1                |  |     |    |
|     |        | 01 2                |  |     |    |
|     |        | 2046、 3             |  |     |    |
|     |        | 11 4                |  |     |    |

## 9.6.6 TK\_PWM\_CNTH

Touch Key PWM Counter High Register.

Offset = 0x95

| Bits | Name        | Description                      | R/W | Reset |
|------|-------------|----------------------------------|-----|-------|
| 7:0  | TK_PWM_CNTH | Touch Key PWM Counter High bits. | R/W | 0x1f  |

## 9.6.7 TK\_PWM\_CNTL

Touch Key PWM Counter Low Register.

Offset = 0x97

| Bits | Name        | Description                     | R/W | Reset |
|------|-------------|---------------------------------|-----|-------|
| 7:0  | TK_PWM_CNTL | Touch Key PWM Counter Low bits. | R/W | 0xff  |

## 9.6.8 TK\_PRESS\_TH

Touch Key Press Threshold Register.

Offset = 0x99

| Bits | Name | Description | R/W | Reset |
|------|------|-------------|-----|-------|
| 7:5  |      |             |     |       |

|     |                 |   |     |       |
|-----|-----------------|---|-----|-------|
|     |                 |   |     |       |
| 4:0 | TK_PRESS_T<br>H | Touch key sensitivity sel(It is the threshold used to judge whether a key has been touched. )<br>00000 $\Delta n = 50$<br>00001 $\Delta n = 60$<br>00010 $\Delta n = 70$<br>00011 $\Delta n = 80$<br>00100 $\Delta n = 90$<br>00101 $\Delta n = 100$<br>00110 $\Delta n = 110$<br>00111 $\Delta n = 120$<br>01000 $\Delta n = 130$<br>01001 $\Delta n = 140$<br>01010 $\Delta n = 155$<br>01011 $\Delta n = 170$<br>01100 $\Delta n = 185$<br>01101 $\Delta n = 200$<br>01110 $\Delta n = 215$<br>01111 $\Delta n = 230$<br>10000 $\Delta n = 245$<br>10001 $\Delta n = 260$<br>10010 $\Delta n = 275$<br>10011 $\Delta n = 290$<br>10100 $\Delta n = 305$<br>10101 $\Delta n = 320$<br>10110 $\Delta n = 340$<br>10111 $\Delta n = 360$<br>11000 $\Delta n = 380$<br>11001 $\Delta n = 400$<br>11010 $\Delta n = 420$<br>11011 $\Delta n = 440$<br>11100 $\Delta n = 460$<br>11101 $\Delta n = 480$<br>11110 $\Delta n = 500$<br>11111 $\Delta n = 520$<br><b>Note:</b> Its value is determined by the capacitance variance of the key that has been touched. It should be adjusted for the best effect according to different project.<br>The smaller this value, the easier to detect key touching, but the more sensitive to noise. | R/W | 01101 |

## 9.6.9 TK\_BL\_CAL\_TH

Touch Key Baseline Calibration Threshold Register.

Offset = 0x9a

| Bits   | Name      | Description  | R/W    | Reset |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
|--------|-----------|--|--------|-------|--------|----|--------|----|--------|----|--------|----|--------|----|--------|----|--------|----|--------|-----|--------|-----|--------|-----|--------|-----|--------|-----|--------|-----|--------|-----|--------|-----|-----|------|
| 7:4    | -         | Reserved   | R      | 0     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 3:0    | BL_CAL_TH | <p>Baseline Correction <math>\Delta n</math> value. It is used to correct the Baseline.</p> <table> <tr><td>0000 :</td><td>20</td></tr> <tr><td>0001 :</td><td>30</td></tr> <tr><td>0010 :</td><td>40</td></tr> <tr><td>0011 :</td><td>50</td></tr> <tr><td>0100 :</td><td>60</td></tr> <tr><td>0101 :</td><td>70</td></tr> <tr><td>0110 :</td><td>80</td></tr> <tr><td>0111 :</td><td>90</td></tr> <tr><td>1000 :</td><td>100</td></tr> <tr><td>1001 :</td><td>110</td></tr> <tr><td>1010 :</td><td>120</td></tr> <tr><td>1011 :</td><td>130</td></tr> <tr><td>1100 :</td><td>140</td></tr> <tr><td>1101 :</td><td>150</td></tr> <tr><td>1110 :</td><td>160</td></tr> <tr><td>1111 :</td><td>170</td></tr> </table> <p><b>Note:</b> for details, please refer to Baseline Correction of function Description.</p> | 0000 : | 20    | 0001 : | 30 | 0010 : | 40 | 0011 : | 50 | 0100 : | 60 | 0101 : | 70 | 0110 : | 80 | 0111 : | 90 | 1000 : | 100 | 1001 : | 110 | 1010 : | 120 | 1011 : | 130 | 1100 : | 140 | 1101 : | 150 | 1110 : | 160 | 1111 : | 170 | R/W | 0110 |
| 0000 : | 20        |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 0001 : | 30        |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 0010 : | 40        |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 0011 : | 50        |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 0100 : | 60        |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 0101 : | 70        |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 0110 : | 80        |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 0111 : | 90        |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 1000 : | 100       |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 1001 : | 110       |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 1010 : | 120       |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 1011 : | 130       |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 1100 : | 140       |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 1101 : | 150       |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 1110 : | 160       |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |
| 1111 : | 170       |  |        |       |        |    |        |    |        |    |        |    |        |    |        |    |        |    |        |     |        |     |        |     |        |     |        |     |        |     |        |     |        |     |     |      |

### 9.6.10 TK\_NOISE\_TH

Touch Key Noise Threshold Register.

Offset = 0x9b

| Bits | Name | Description | R/W | Reset |
|------|------|-------------|-----|-------|
| 7:3  | -    | Reserved    | R   | 0     |

|     |          |  |     |      |
|-----|----------|--|-----|------|
|     |          | Effective Calibration delta range(  $\Delta n$   bigger than this range will be discarded by TKC, which will not be added to BUCKET_TH)  |     |      |
|     |          | 2047、 10   |     |      |
|     |          | 2048、 15   |     |      |
|     |          | 0010 20  |     |      |
|     |          | 0011 25  |     |      |
|     |          | 0100 30  |     |      |
|     |          | 0101 35  |     |      |
|     |          | 0110 40  |     |      |
|     |          | 0111 45  |     |      |
| 3:0 | NOISE_TH | 1000 50  | R/W | 1000 |
|     |          | 1001 55  |     |      |
|     |          | 1010 60  |     |      |
|     |          | 1011 65  |     |      |
|     |          | 1100 70  |     |      |
|     |          | 1101 75  |     |      |
|     |          | 1110 80  |     |      |
|     |          | 1111 85  |     |      |
|     |          | <b>Note:</b> This value is used during Auto-calibration. During Auto-calibration, when the detecting Baseline variance absolute value is bigger than it, the detecting Baseline will be judged as noise, and be discarded. |     |      |

### 9.6.11 TK\_BUCKET\_TH

Touch Key Bucket Threshold And Baseline Step Adjust Register.

Offset = 0x9c

| Bits | Name      | Description   | R/W | Reset |
|------|-----------|---|-----|-------|
| 7    | -         | Reserved  | R   | 0     |
|      |           | Effective Calibration delta absolute value sum range                                |     |       |
|      |           | 000 30 delta = 0.1961   |     |       |
|      |           | 001 40 delta = 0.2614   |     |       |
|      |           | 010 50 delta = 0.3268   |     |       |
|      |           | 011 60 delta = 0.3922   |     |       |
| 6:4  | BUCKET_TH | 100 70 delta = 0.4575   | R/W | 010   |
|      |           | 101 80 delta = 0.5229   |     |       |
|      |           | 110 100 delta = 0.6536  |     |       |
|      |           | 111 110 delta = 0.7190  |     |       |
|      |           | <b>Note:</b> for details, please refer to Auto-calibration of function Description. |     |       |
| 3    | -         | Reserved  | R   | 0     |

|   |         |   |  |     |     |
|---|---------|---|--|-----|-----|
| 2:0   | BL_STEP | Baseline adjust step. When $ \Delta n $ sum bigger than |  | R/W | 010 |
|   |         | BUCKET_TH, adjust Baseline by one step.                 |  |     |     |
|   |         | 2049、 1   |  |     |     |
|   |         | 2050、 2   |  |     |     |
|   |         | 010 3   |  |     |     |
|   |         | 011 4   |  |     |     |
|   |         | 100 5   |  |     |     |
|   |         | 101 6   |  |     |     |
|   |         | 110 7   |  |     |     |
|   |         | 111 8   |  |     |     |
| <b>Note:</b> for details, please refer to Auto-calibration of function Description. |         |   |  |     |     |

### 9.6.12 TK\_AUTO\_LOWTH\_H

Touch Key idac auto adjust low threshold high bits Register.

Offset=0x9d

| Bits | Name           | Description                                    | R/W | Reset |
|------|----------------|--|-----|-------|
| 7:0  | TK_ADJUST_TH1H | TKEY idac auto adjust low threshold bits[15:8] | R/W | 0x02  |

### 9.6.13 TK\_AUTO\_LOWTH\_L

Touch Key idac auto adjust low threshold low bits Register.

Offset=0x9e

| Bits | Name           | Description                                   | R/W | Reset |
|------|----------------|---|-----|-------|
| 7:0  | TK_ADJUST_TH1L | TKEY idac auto adjust low threshold bits[7:0] | R/W | 0xbc  |

### 9.6.14 TK\_AUTO\_HIGHTH\_H

Touch Key idac auto adjust high threshold high bits Register.

Offset=0xa2

| Bits | Name           | Description                                     | R/W | Reset |
|------|----------------|---|-----|-------|
| 7:0  | TK_ADJUST_TH2H | TKEY idac auto adjust high threshold bits[15:8] | R/W | 0x1e  |

### 9.6.15 TK\_AUTO\_HIGHTH\_L

Touch Key idac auto adjust high threshold low bits Register.

Offset=0xa3

| Bits | Name           | Description                                    | R/W | Reset |
|------|----------------|--|-----|-------|
| 7:0  | TK_ADJUST_TH2L | TKEY idac auto adjust high threshold bits[7:0] | R/W | 0x78  |

### 9.6.16 TK\_KEY0\_IDA\_CTL

Touch Key0 charging current Control Register.

Offset = 0xa4

| Bits | Name         | Description   | R/W | Reset |
|------|--------------|---|-----|-------|
| 7:4  | -            | Reserved  | R   | 0     |
| 3:0  | TK0_IDA[3:0] | TKEY0 changing current control:<br>2051、 0Ua<br>2052、 0.5Ua<br>0010 1Ua<br>0011 1.5Ua<br>0100 2Ua<br>0101 2.5Ua<br>0110 3Ua<br>0111 3.5Ua<br>1000 4Ua<br>1001 4.5Ua<br>1010 5Ua<br>1011 5.5Ua<br>1100 6Ua<br>1101 6.5Ua<br>1110 7Ua<br>1111 7.5Ua<br><b>Note:</b> This value is used to set the initial charging current to external cap key. | R/W | 1000  |

### 9.6.17 TK\_KEY1\_IDA\_CTL

Touch Key1 charging current Control Register.

Offset = 0xa5

| Bits | Name | Description | R/W | Reset |
|------|------|-------------|-----|-------|
|      |      |             |     |       |

|      |              |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
|------|--------------|--|------|-----|------|-------|------|-----|------|-------|------|-----|------|-------|------|-----|------|-------|------|-----|------|-------|------|-----|------|-------|------|-----|------|-------|------|-----|------|-------|-----|------|
| 7:4  | -            | Reserved   | R    | 0   |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 3:0  | TK1_IDA[3:0] | <p>KEY1 changing current control:</p> <table> <tr><td>2053</td><td>0Ua</td></tr> <tr><td>2054</td><td>0.5Ua</td></tr> <tr><td>0010</td><td>1Ua</td></tr> <tr><td>0011</td><td>1.5Ua</td></tr> <tr><td>0100</td><td>2Ua</td></tr> <tr><td>0101</td><td>2.5Ua</td></tr> <tr><td>0110</td><td>3Ua</td></tr> <tr><td>0111</td><td>3.5Ua</td></tr> <tr><td>1000</td><td>4Ua</td></tr> <tr><td>1001</td><td>4.5Ua</td></tr> <tr><td>1010</td><td>5Ua</td></tr> <tr><td>1011</td><td>5.5Ua</td></tr> <tr><td>1100</td><td>6Ua</td></tr> <tr><td>1101</td><td>6.5Ua</td></tr> <tr><td>1110</td><td>7Ua</td></tr> <tr><td>1111</td><td>7.5Ua</td></tr> </table> <p><b>Note:</b> This value is used to set the initial charging current to external cap key.</p> | 2053 | 0Ua | 2054 | 0.5Ua | 0010 | 1Ua | 0011 | 1.5Ua | 0100 | 2Ua | 0101 | 2.5Ua | 0110 | 3Ua | 0111 | 3.5Ua | 1000 | 4Ua | 1001 | 4.5Ua | 1010 | 5Ua | 1011 | 5.5Ua | 1100 | 6Ua | 1101 | 6.5Ua | 1110 | 7Ua | 1111 | 7.5Ua | R/W | 1000 |
| 2053 | 0Ua          |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 2054 | 0.5Ua        |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 0010 | 1Ua          |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 0011 | 1.5Ua        |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 0100 | 2Ua          |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 0101 | 2.5Ua        |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 0110 | 3Ua          |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 0111 | 3.5Ua        |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 1000 | 4Ua          |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 1001 | 4.5Ua        |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 1010 | 5Ua          |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 1011 | 5.5Ua        |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 1100 | 6Ua          |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 1101 | 6.5Ua        |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 1110 | 7Ua          |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |
| 1111 | 7.5Ua        |  |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |      |     |      |       |     |      |

### 9.6.18 TK\_KEY2\_IDA\_CTL

Touch Key2 charging current Control Register.

Offset = 0xa6

| Bits | Name | Description | R/W | Reset |
|------|------|-------------|-----|-------|
| 7:4  | -    | Reserved    | R   | 0     |

|     |              |   |     |      |
|-----|--------------|---|-----|------|
|     |              | TKEY2 changing current control:<br>2055、 0Ua<br>2056、 0.5Ua<br>0010 1Ua<br>0011 1.5Ua<br>0100 2Ua<br>0101 2.5Ua<br>0110 3Ua<br>0111 3.5Ua<br>1000 4Ua<br>1001 4.5Ua<br>1010 5Ua<br>1011 5.5Ua<br>1100 6Ua<br>1101 6.5Ua<br>1110 7Ua<br>1111 7.5Ua<br><b>Note:</b> This value is used to set the initial charging current to external cap key. |     |      |
| 3:0 | TK2_IDA[3:0] | 1000  | R/W | 1000 |

### 9.6.19 TK\_KEY3\_IDA\_CTL

Touch Key3 charging current Control Register.

Offset = 0xa9

| Bits | Name | Description | R/W | Reset |
|------|------|-------------|-----|-------|
| 7:4  | -    | Reserved    | R   | 0     |

|     |              |   |     |      |
|-----|--------------|---|-----|------|
|     |              | TKEY3 changing current control:<br>2057、 0Ua<br>2058、 0.5Ua<br>0010 1Ua<br>0011 1.5Ua<br>0100 2Ua<br>0101 2.5Ua<br>0110 3Ua<br>0111 3.5Ua<br>1000 4Ua<br>1001 4.5Ua<br>1010 5Ua<br>1011 5.5Ua<br>1100 6Ua<br>1101 6.5Ua<br>1110 7Ua<br>1111 7.5Ua<br><b>Note:</b> This value is used to set the initial charging current to external cap key. |     |      |
| 3:0 | TK3_IDA[3:0] | 1000  | R/W | 1000 |

### 9.6.20 TK\_KEY4\_IDA\_CTL

Touch Key4 charging current Control Register.

Offset = 0xaa

| Bits | Name | Description | R/W | Reset |
|------|------|-------------|-----|-------|
| 7:4  | -    | Reserved    | R   | 0     |

|     |              |   |     |      |
|-----|--------------|---|-----|------|
|     |              | TKEY4 changing current control:<br>2059、 0Ua<br>2060、 0.5Ua<br>0010 1Ua<br>0011 1.5Ua<br>0100 2Ua<br>0101 2.5Ua<br>0110 3Ua<br>0111 3.5Ua<br>1000 4Ua<br>1001 4.5Ua<br>1010 5Ua<br>1011 5.5Ua<br>1100 6Ua<br>1101 6.5Ua<br>1110 7Ua<br>1111 7.5Ua<br><b>Note:</b> This value is used to set the initial charging current to external cap key. |     |      |
| 3:0 | TK4_IDA[3:0] | 1000  | R/W | 1000 |

### 9.6.21 TK\_KEY5\_IDA\_CTL

Touch Key5 charging current Control Register.

Offset = 0xab

| Bits | Name | Description | R/W | Reset |
|------|------|-------------|-----|-------|
| 7:4  | -    | Reserved    | R   | 0     |

|     |              |   |     |      |
|-----|--------------|---|-----|------|
|     |              | TKEY5 changing current control:<br>2061、 0Ua<br>2062、 0.5Ua<br>0010 1Ua<br>0011 1.5Ua<br>0100 2Ua<br>0101 2.5Ua<br>0110 3Ua<br>0111 3.5Ua<br>1000 4Ua<br>1001 4.5Ua<br>1010 5Ua<br>1011 5.5Ua<br>1100 6Ua<br>1101 6.5Ua<br>1110 7Ua<br>1111 7.5Ua<br><b>Note:</b> This value is used to set the initial charging current to external cap key. |     |      |
| 3:0 | TK5_IDA[3:0] | 1000<br>1001<br>1010<br>1011<br>1100<br>1101<br>1110<br>1111  | R/W | 1000 |

### 9.6.22 TK\_UPDAT\_PD

Touch Key New Data Update Pending Register.

Offset = 0xad

| Bits | Name         | Description  | R/W | Reset |
|------|--------------|--|-----|-------|
| 7:6  | -            | Reserved   | R   | 0     |
| 5    | TK5_UPDAT_PD | TKEY5 new data update pending bit, Write 1 to clear this bit.<br>0: no new data update<br>1: new data update | R/W | 0     |
| 4    | TK4_UPDAT_PD | TKEY4 new data update pending bit, Write 1 to clear this bit.<br>0: no new data update<br>1: new data update | R/W | 0     |
| 3    | TK3_UPDAT_PD | TKEY3 new data update pending bit, Write 1 to clear this bit.<br>0: no new data update<br>1: new data update | R/W | 0     |
| 2    | TK2_UPDAT_PD | TKEY2 new data update pending bit, Write 1 to clear this bit.<br>0: no new data update<br>1: new data update | R/W | 0     |

|   |              |  |     |   |
|---|--------------|--|-----|---|
| 1 | TK1_UPDAT_PD | TKEY1 new data update pending bit, Write 1 to clear this bit.<br>0: no new data update<br>1: new data update | R/W | 0 |
| 0 | TK0_UPDAT_PD | TKEY0 new data update pending bit, Write 1 to clear this bit.<br>0: no new data update<br>1: new data update | R/W | 0 |

### 9.6.23 TK\_KEY0\_DATH

Touch Key0 current data high bits.

Offset=0xae

| Bits | Name     | Description                   | R/W | Reset |
|------|----------|-------------------------------|-----|-------|
| 7:0  | TK0_DATH | TKEY0 current data bits[15:8] | R   | 0     |

### 9.6.24 TK\_KEY0\_DATL

Touch Key0 current data low bits.

Offset=0xaf

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK0_DATL | TKEY0 current data bits[7:0] | R   | 0     |

### 9.6.25 TK\_KEY1\_DATH

Touch Key1 current data high bits.

Offset=0xb0

| Bits | Name     | Description                   | R/W | Reset |
|------|----------|-------------------------------|-----|-------|
| 7:0  | TK1_DATH | TKEY1 current data bits[15:8] | R   | 0     |

### 9.6.26 TK\_KEY1\_DATL

Touch Key1 current data low bits.

Offset=0xb1

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK1_DATL | TKEY1 current data bits[7:0] | R   | 0     |

### 9.6.27 TK\_KEY2\_DATH

Touch Key2 current data high bits.

Offset=0xb2

| Bits | Name     | Description                   | R/W | Reset |
|------|----------|-------------------------------|-----|-------|
| 7:0  | TK2_DATH | TKEY2 current data bits[15:8] | R   | 0     |

### 9.6.28 TK\_KEY2\_DATL

Touch Key2 current data low bits.

Offset=0xb3

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK2_DATL | TKEY2 current data bits[7:0] | R   | 0     |

### 9.6.29 TK\_KEY3\_DATH

Touch Key3 current data high bits.

Offset=0xb4

| Bits | Name     | Description                   | R/W | Reset |
|------|----------|-------------------------------|-----|-------|
| 7:0  | TK3_DATH | TKEY3 current data bits[15:8] | R   | 0     |

### 9.6.30 TK\_KEY3\_DATL

Touch Key3 current data low bits.

Offset=0xb5

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK3_DATL | TKEY3 current data bits[7:0] | R   | 0     |

### 9.6.31 TK\_KEY4\_DATH

Touch Key4 current data high bits.

Offset=0xb6

| Bits | Name     | Description                   | R/W | Reset |
|------|----------|-------------------------------|-----|-------|
| 7:0  | TK4_DATH | TKEY4 current data bits[15:8] | R   | 0     |

### 9.6.32 TK\_KEY4\_DATL

Touch Key4 current data low bits.

Offset=0xb7

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK4_DATL | TKEY4 current data bits[7:0] | R   | 0     |

### 9.6.33 TK\_KEY5\_DATH

Touch Key5 current data high bits.

Offset=0xb8

| Bits | Name     | Description                   | R/W | Reset |
|------|----------|-------------------------------|-----|-------|
| 7:0  | TK5_DATH | TKEY5 current data bits[15:8] | R   | 0     |

### 9.6.34 TK\_KEY5\_DATL

Touch Key5 current data low bits.

Offset=0xb9

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK5_DATL | TKEY5 current data bits[7:0] | R   | 0     |

### 9.6.35 TK\_RAW\_PD

Touch Key New RAW Data Update Pending Register.

Offset = 0Xba

| Bits | Name       | Description   | R/W | Reset |
|------|------------|---|-----|-------|
| 7:6  | -          | Reserved  | R   | 0     |
| 5    | TK5_RAW_PD | TKEY5 new RAW update pending bit, Write 1 to clear this bit.<br>0: no new baseline update<br>1: new baseline update | R/W | 0     |
| 4    | TK4_RAW_PD | TKEY4 new RAW update pending bit, Write 1 to clear this bit.<br>0: no new baseline update<br>1: new baseline update | R/W | 0     |
| 3    | TK3_RAW_PD | TKEY3 new RAW update pending bit, Write 1 to clear this bit.<br>0: no new baseline update<br>1: new baseline update | R/W | 0     |

|   |            |   |     |   |
|---|------------|---|-----|---|
| 2 | TK2_RAW_PD | TKEY2 new RAW update pending bit, Write 1 to clear this bit.<br>0: no new baseline update<br>1: new baseline update | R/W | 0 |
| 1 | TK1_RAW_PD | TKEY1 new RAW update pending bit, Write 1 to clear this bit.<br>0: no new baseline update<br>1: new baseline update | R/W | 0 |
| 0 | TK0_RAW_PD | TKEY0 new RAW update pending bit, Write 1 to clear this bit.<br>0: no new baseline update<br>1: new baseline update | R/W | 0 |

### 9.6.36 TK\_KEY0\_RAWH

Touch Key0 current RAW high bits.

Offset=0Xbc

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK0_RAWH | TKEY0 current RAW bits[15:8] | R   | 0     |

### 9.6.37 TK\_KEY0\_RAWL

Touch Key0 current RAW low bits.

Offset=0Xbd

| Bits | Name     | Description                 | R/W | Reset |
|------|----------|-----------------------------|-----|-------|
| 7:0  | TK0_RAWL | TKEY0 current RAW bits[7:0] | R   | 0     |

### 9.6.38 TK\_KEY1\_RAWH

Touch Key1 current RAW high bits.

Offset=0Xbf

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK1_RAWH | TKEY1 current RAW bits[15:8] | R   | 0     |

### 9.6.39 TK\_KEY1\_RAWL

Touch Key1 current RAW low bits.

Offset=0Xc1

| Bits | Name     | Description                 | R/W | Reset |
|------|----------|-----------------------------|-----|-------|
| 7:0  | TK1_RAWL | TKEY1 current RAW bits[7:0] | R   | 0     |

### 9.6.40 TK\_KEY2\_RAWH

Touch Key2 current RAW high bits.

Offset=0Xc2

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK2_RAWH | TKEY2 current RAW bits[15:8] | R   | 0     |

### 9.6.41 TK\_KEY2\_RAWL

Touch Key2 current RAW low bits.

Offset=0Xc3

| Bits | Name     | Description                 | R/W | Reset |
|------|----------|-----------------------------|-----|-------|
| 7:0  | TK2_RAWL | TKEY2 current RAW bits[7:0] | R   | 0     |

### 9.6.42 TK\_KEY3\_RAWH

Touch Key3 current RAW high bits.

Offset=0Xc4

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK3_RAWH | TKEY3 current RAW bits[15:8] | R   | 0     |

### 9.6.43 TK\_KEY3\_RAWL

Touch Key3 current RAW low bits.

Offset=0Xc5

| Bits | Name     | Description                 | R/W | Reset |
|------|----------|-----------------------------|-----|-------|
| 7:0  | TK3_RAWL | TKEY3 current RAW bits[7:0] | R   | 0     |

### 9.6.44 TK\_KEY4\_RAWH

Touch Key4 current RAW high bits.

Offset=0Xc6

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK4_RAWH | TKEY4 current RAW bits[15:8] | R   | 0     |

### 9.6.45 TK\_KEY4\_RAWL

Touch Key4 current RAW low bits.

Offset=0Xc7

| Bits | Name     | Description                 | R/W | Reset |
|------|----------|-----------------------------|-----|-------|
| 7:0  | TK4_RAWL | TKEY4 current RAW bits[7:0] | R   | 0     |

### 9.6.46 TK\_KEY5\_RAWH

Touch Key5 current RAW high bits.

Offset=0Xc8

| Bits | Name     | Description                  | R/W | Reset |
|------|----------|------------------------------|-----|-------|
| 7:0  | TK5_RAWH | TKEY5 current RAW bits[15:8] | R   | 0     |

### 9.6.47 TK\_KEY5\_RAWL

Touch Key5 current RAW low bits.

Offset=0Xc9

| Bits | Name     | Description                 | R/W | Reset |
|------|----------|-----------------------------|-----|-------|
| 7:0  | TK5_RAWL | TKEY5 current RAW bits[7:0] | R   | 0     |

### 9.6.48 TK\_KEY0\_BL\_H

Touch Key0 current baseline high bits.

Offset=0Xca

| Bits | Name     | Description                       | R/W | Reset |
|------|----------|-----------------------------------|-----|-------|
| 7:0  | TK0_BL_H | TKEY0 current baseline bits[15:8] | R   | 0     |

### 9.6.49 TK\_KEY0\_BL\_L

Touch Key0 current baseline low bits.

Offset=0Xcc

| Bits | Name     | Description                      | R/W | Reset |
|------|----------|----------------------------------|-----|-------|
| 7:0  | TK0_BL_L | TKEY0 current baseline bits[7:0] | R   | 0     |

### 9.6.50 TK\_KEY1\_BL\_H

Touch Key1 current baseline high bits.

Offset=0Xcd

| Bits | Name     | Description                       | R/W | Reset |
|------|----------|-----------------------------------|-----|-------|
| 7:0  | TK1_BL_H | TKEY1 current baseline bits[15:8] | R   | 0     |

### 9.6.51 TK\_KEY1\_BL\_L

Touch Key1 current baseline low bits.

Offset=0Xce

| Bits | Name     | Description                      | R/W | Reset |
|------|----------|----------------------------------|-----|-------|
| 7:0  | TK1_BL_L | TKEY1 current baseline bits[7:0] | R   | 0     |

### 9.6.52 TK\_KEY2\_BL\_H

Touch Key2 current baseline high bits.

Offset=0Xd2

| Bits | Name     | Description                       | R/W | Reset |
|------|----------|-----------------------------------|-----|-------|
| 7:0  | TK2_BL_H | TKEY2 current baseline bits[15:8] | R   | 0     |

### 9.6.53 TK\_KEY2\_BL\_L

Touch Key2 current baseline low bits.

Offset=0Xd3

| Bits | Name     | Description                      | R/W | Reset |
|------|----------|----------------------------------|-----|-------|
| 7:0  | TK2_BL_L | TKEY2 current baseline bits[7:0] | R   | 0     |

### 9.6.54 TK\_KEY3\_BL\_H

Touch Key3 current baseline high bits.

Offset=0Xd4

| Bits | Name     | Description                       | R/W | Reset |
|------|----------|-----------------------------------|-----|-------|
| 7:0  | TK3_BL_H | TKEY3 current baseline bits[15:8] | R   | 0     |

### 9.6.55 TK\_KEY3\_BL\_L

Touch Key3 current baseline low bits.

Offset=0Xd5

| Bits | Name     | Description                      | R/W | Reset |
|------|----------|----------------------------------|-----|-------|
| 7:0  | TK3_BL_L | TKEY3 current baseline bits[7:0] | R   | 0     |

### 9.6.56 TK\_KEY4\_BL\_H

Touch Key4 current baseline high bits.

Offset=0Xd6

| Bits | Name     | Description                       | R/W | Reset |
|------|----------|-----------------------------------|-----|-------|
| 7:0  | TK4_BL_H | TKEY4 current baseline bits[15:8] | R   | 0     |

### 9.6.57 TK\_KEY4\_BL\_L

Touch Key4 current baseline low bits.

Offset=0Xd7

| Bits | Name     | Description                      | R/W | Reset |
|------|----------|----------------------------------|-----|-------|
| 7:0  | TK4_BL_L | TKEY4 current baseline bits[7:0] | R   | 0     |

### 9.6.58 TK\_KEY5\_BL\_H

Touch Key5 current baseline high bits.

Offset=0Xd8

| Bits | Name     | Description                       | R/W | Reset |
|------|----------|-----------------------------------|-----|-------|
| 7:0  | TK5_BL_H | TKEY5 current baseline bits[15:8] | R   | 0     |

### 9.6.59 TK\_KEY5\_BL\_L

Touch Key5 current baseline low bits.

Offset=0Xd9

| Bits | Name     | Description                      | R/W | Reset |
|------|----------|----------------------------------|-----|-------|
| 7:0  | TK5_BL_L | TKEY5 current baseline bits[7:0] | R   | 0     |

## 9.6.60 TK\_DEBUG

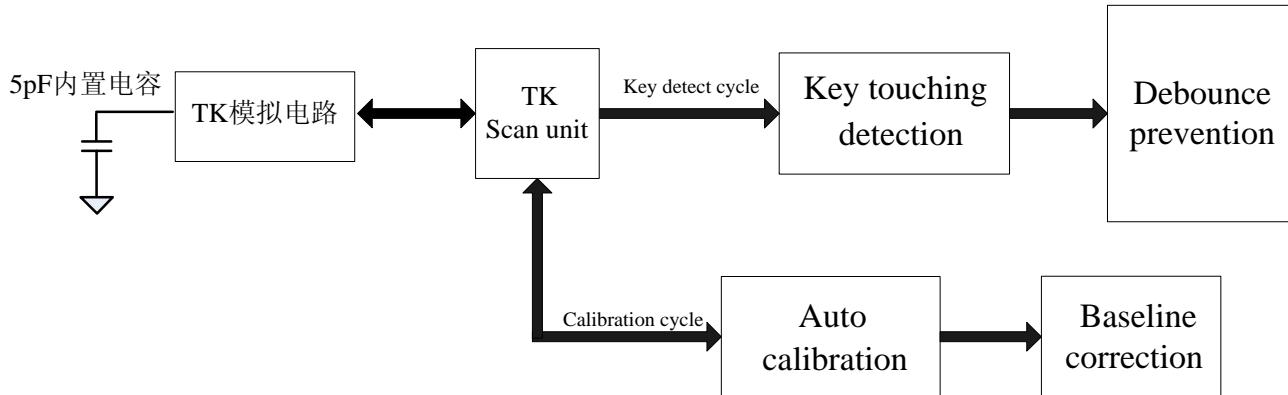
Touch Key Debug Register.

Offset=0Xda

| Bits | Name        | Description  | R/W | Reset |
|------|-------------|--|-----|-------|
| 7:4  | -           | Reserved   | R   | 0     |
| 3    | TK_TCAP_EN  | TK test CAP enable bit:<br>0: disable.<br>1: enable.   | R/W | 0     |
| 2    | TK_SP_REF   | TK Shield Protect Reference current select bit.  | R/W | 0     |
| 1    | TK_SP_EN    | TK Shield Protect enable bit.<br>0: disable.<br>1: enable.   | R/W | 0     |
| 0    | TK_DEBUG_EN | TK Debug mode enable bit.<br>0: TK normal mode.<br>1: TK debug mode, bypass analog phy, data input from debug ram, test digital state machine. | R/W | 0     |

## 9.7 TESTMODE

### 9.7.1 BLOCK DIAGRAM



## 5. Touch Key Controller test mode block Diagram

## 9.7.2 SIGNAL DESCRIPTION

### 9.7.3 TEST METHOD

在 CP/FT 测试时，使能 TK 的内置电容（约 5Pf），此时 6 个 KEY 跟外部模拟 PAD 的通路开关断开，内置电流源只是对内置电容充放电，MCU 从任意一个 KEY 对应的 DATA 寄存器读取充放电原始数据，就可以判断 TK 的 analog 电路和一部分 digital 电路是否正常了；其余的 digital 电路是用于按键判断的，如果需要，可以在 FT 阶段加测；

CP 测试时，假设从 HOSCO 灌入的时钟为 4M，TK 充电电流为 4Ua，N 为 17，Vth 为 1.5V，内置电容为 5Pf，那么 DATA 保存的数值为 127.5；

FT 测试时，HOSC 为 24M，TK 充电电流为 4Ua，N 为 17，Vth 为 1.5V，内置电容为 5Pf，那么 DATA 保存的数值为 765；

### 9.7.4 DEBUG METHOD

TK 有两个 debug 模式：

1. analog debug mode，在此 mode 下输入输出信号定义如下，需要 16 个 GPIO；此模式，是用于单独测试 analog 电路的，也可以将 analog 电路作为 phy 板；

| GPIO   | Signal         | Type |
|--------|----------------|------|
| GPIOA0 | KEY_EN[0]      | I    |
| GPIOA1 | KEY_EN[1]      | I    |
| GPIOA2 | KEY_EN[2]      | I    |
| GPIOA3 | KEY_EN[3]      | I    |
| GPIOA4 | KEY_EN[4]      | I    |
| GPIOA7 | KEY_EN[5]      | I    |
| GPIOB0 | TK_IDA_CTL0[0] | I    |
| GPIOB1 | TK_IDA_CTL0[1] | I    |
| GPIOB2 | TK_IDA_CTL0[2] | I    |
| GPIOB3 | TK_IDA_CTL0[3] | I    |
| GPIOB4 | TKEN/TK_ANA_EN | I    |
| GPIOB5 | TK_OSC_RESET   | I    |
| GPIOB6 | TK_SP_EN       | I    |
| GPIOB7 | TK_SP_REF      | I    |
| GPIOD0 | TK_OSC         | O    |

2. digital debug mode，在此 mode 下 digital bypass 掉 analog 电路，用于测试 digital 电路；此模式只是用于 pattern 仿真，在 IC 端无需定义 GPIO 作为输入和输出；

# 10 Storage

## 10.1 SD/MMC 模块（赵天亮、郑思）

| 日期         | 版本     | 描述                                | 修订人 |
|------------|--------|-----------------------------------|-----|
| 2012-07-24 | V1. 00 | initial                           | 赵天亮 |
| 2012-11-07 | V2. 00 | 在超时计数表格中添加了一个时间单位：(S)             | 赵天亮 |
| 2012-12-12 | V2. 02 | 完善 opertaion manual 章节            | 赵天亮 |
| 2013-07-05 | V2. 04 | Operation manual 添加了新的内容:添加了说明项 8 | 赵天亮 |

### 10.1.1 Features

- ◆ Fully compliant with MMC Specification 4.3
- ◆ Fully compliant with SDIO card Specification 2.0
- ◆ Integrated Clock Delay Chain Technique to Regulate Card Interface Timing: Latching Delay Chain for input Signal, Output Delay Chain for output signal.。
- ◆ Integrated Watchdog Counter to report Exception happening.
- ◆ Integrated Pull up resistance (value 51Komh) for Data and CMD Line.
- ◆ Integrated CRC calculate and check circuit.
- ◆ Send continuous clock to support SDIO card.
- ◆ Support 3.1V CLK PAD voltage.
- ◆ Support 3.1V CMD PAD voltage.
- ◆ Support 3.1V DAT PAD voltage.
- ◆ Band Width: 50Mbyte/S
- ◆ Maximal Clock: 50MHz

### 10.1.2 Function Description

This Specification describes the SDMMC card Host Controller and How data is transferred to SDMMC CARD device and discusses how to configure and program the SDMMC CARD Host Controller (SDC) module.

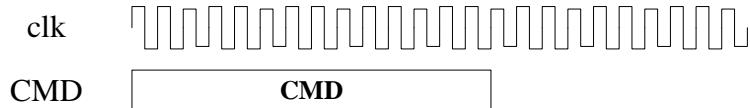
#### 10.1.2.1 Transfer Mode Description

All SD card operation is ranged into 10 transfer modes, on which the design of SD controller is based. Use

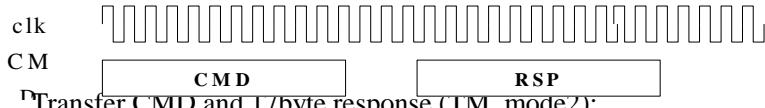
the propriety mode , can manage every SD bus operation.

The transfer modes are described as following:

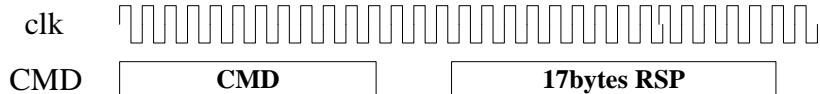
Transfer only CMD(TM\_mode0):



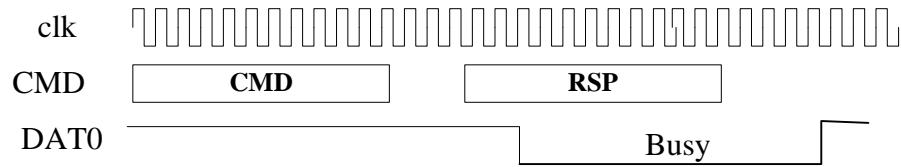
Transfer CMD and 6 byte response (TM\_mode1):



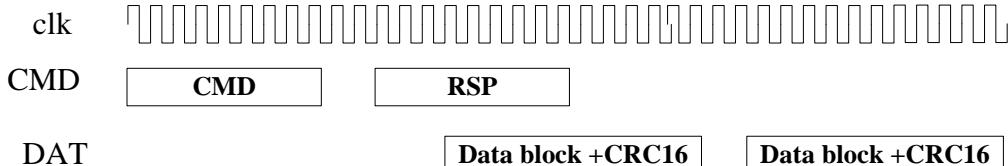
Transfer CMD and 17 byte response (TM\_mode2):



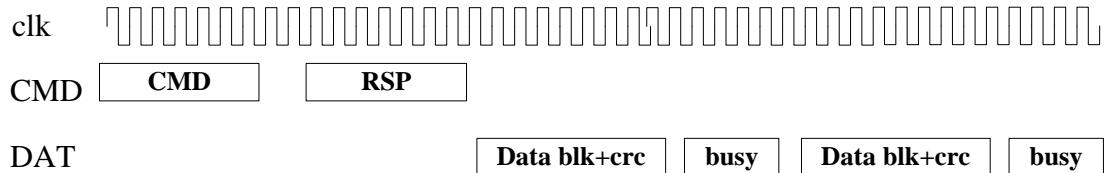
Transfer command and 6byte response with busy signal (TM\_Mode3):



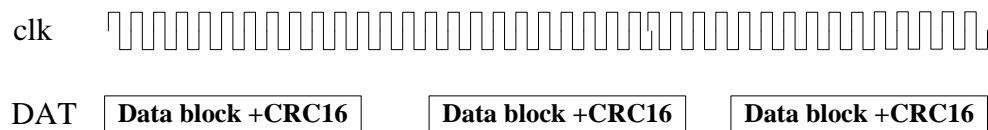
Transfer command and 6byte response and data in mode.(TM\_mode4):



Transfer command and 6byte response and data out mode.(TM\_mode5):



Transfer data in without command and response(TM\_mode6):



Transfer data out without command and response (TM\_mode7):

The timing diagram shows the `clk` signal starting at logic 0. It remains at 0 for one clock cycle, then alternates between 0 and 1 for the subsequent 29 cycles.

DAT Data blk+crc busy Data blk+crc busy Data blk+crc busy

Transfer only CLK.(TM\_mode8):

clk |

**Note, If register Transfer clock number is 00B, sdc controller send continuous clocks to card; if software clear the transfer start bit, sdc controller stops sending clocks.**

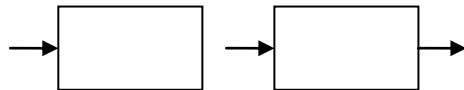
BOOT mode Transfer data in (TM\_mode9 ).(hardware check data crc16 ,do not check boot mode acknowledge.):

The diagram illustrates the timing sequence for booting an SD card. The CLK signal is a continuous square wave. The CMD signal starts high, then goes low for the boot sequence. The DAT[0] signal shows three data frames being sent. Each frame consists of a start bit (S), 10 bytes of data (010), and an end bit (E). The first two frames are labeled "512bytes + CRC". After the second frame, the CMD signal goes high again. A vertical arrow points to this transition with the label "Boot terminated". Below the DAT[0] signal, a horizontal double-headed arrow spans the time between the end of the second frame and the start of the next command. A callout box indicates "Min 8 clocks + 48 clocks = 56 clocks required from CMD signal high to next MMC command." A vertical double-headed arrow on the left side of the first frame is labeled "50ms max". A horizontal double-headed arrow at the bottom of the first frame is labeled "1 sec. max".

BOOT mode Transfer data in mode.(TM\_mode10)(hardware check data crc16 , check boot modeacknowledge).TM\_mode10.the same as mode9 except the controller need to check boot acknowledge.

### 10.1.2.2 Hardware WatchDog

Hardware timeout was only available or used in mode3(busy timeout) ,mode4/6/9/10(data not coming timeout), mode5/7(busy timeout).



Before card Busy state end ok or data start bit coming, INT Timeout Counter decease 1 for every Timeout Clock Period.

If Hardwate Timeout Counter equal 1, the timeout time is 1\*(Timeout Clock Period), Min Timeout time.  
If Hardwate Timeout Counter equal 63, the timeout time is 63\*(Timeout Clock Period), Max Timeout time.

If Hardwate Timeout Counter equal 0, the timeout time is 0\*(Timeout Clock Period), reserved Timeout time.

| Card Clock | Min Timeout time (s) | Max Timeout time(s) |
|------------|----------------------|---------------------|
| 10MHZ      | 0.2097152/1          | 13.42177/1          |
| 20MHZ      | 0.2097152/2          | 13.42177/2          |
| 30MHZ      | 0.2097152/3          | 13.42177/3          |
| 40MHZ      | 0.2097152/4          | 13.42177/4          |
| 50MHZ      | 0.2097152/5          | 13.42177/5          |

Table timeout time maximum and minimum value

### 10. 1. 2. 3 Delay Chain Regulated Unit

| SDC        | output      | Latch        | f <sub>max</sub> |
|------------|-------------|--------------|------------------|
| Controller | Rising edge | Rising edge  | 100M             |
| device     | DS          | Falling edge | rising           |
|            | HS          | rising edge  | Rising edge      |
|            | SDR         | Rising edge  | 100M             |

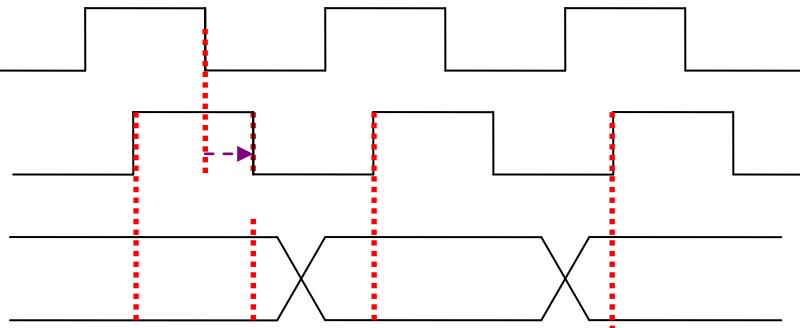


Figure 10-1 Lathing Delay Regulation

In order to delay the inside latching data clock to compensate signal transmission delay, The SDC use the Latch input DATA/CMD delay Time select to regulate the CD, when host controller latching input signal.

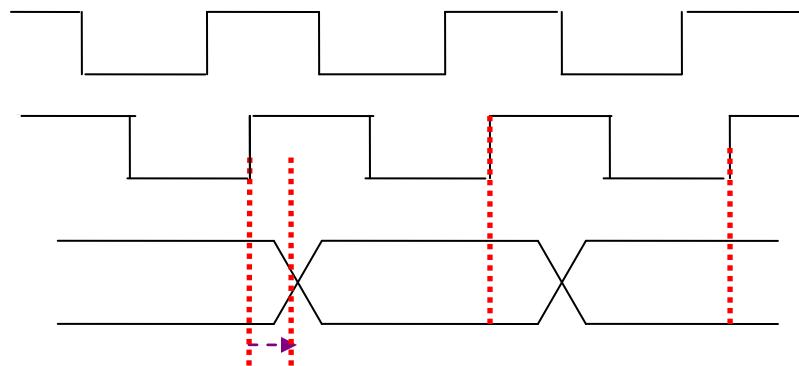


Figure 10-2 Output Delay Regulation

The SDC use OUTPUT DATA/CMD delay Time select to regulate the OD when output signal.

The MSHC use OUTPUT DATA/BS delay Time select to regulate the OD when output signal.

| Delay Selection | Delay Time | Delay Selection | Delay Time |
|-----------------|------------|-----------------|------------|
| 0x0             | 0ns        | 0x8             | 3.2ns      |
| 0x1             | 0.4ns      | 0x9             | 3.6ns      |
| 0x2             | 0.8ns      | 0xa             | 4.6ns      |
| 0x3             | 1.2.ns     | 0xb             | 5.6ns      |
| 0x4             | 1.6ns      | 0xc             | 6.6ns      |
| 0x5             | 2.0ns      | 0xd             | 7.6ns      |
| 0x6             | 2.4ns      | 0xe             | 8.6ns      |
| 0x7             | 2.8ns      | 0xf             | 12.6ns     |

### 10.1.3 Module Description

#### 10.1.3.1 Block Diagram

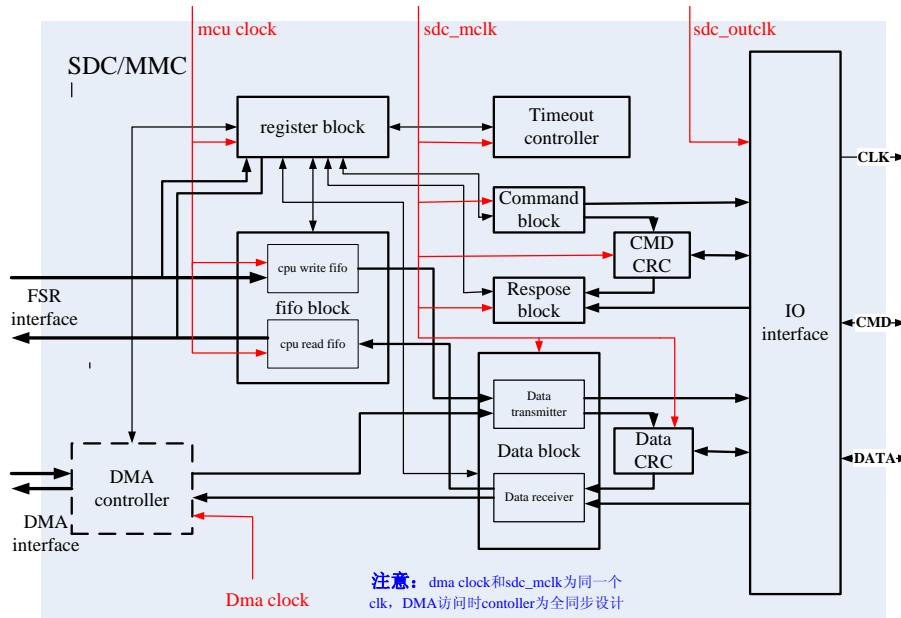


Figure 10-3 模块 SDC controller 接口信号图

SD 卡控制分为小模块的功能如下: register block 负责寄存器读写及与 CPU/FSR 等模块交互; timeout controller 负责超时计算与报错及复位模块的功能; cpu read/write fifo 负责 CPU 访问控制器时候的数据交互缓存; command block response block 负责命令发送、响应接收及报错; data block 负责数据传输控制; I/O interface 负责与卡 PAD 相关的接口时序交互操作。

#### 10.1.3.2 Signal List

表 模块 SDC controller 接口信号

| 信号名         | 位宽 | 方向     | 时钟域      | 描述                           |
|-------------|----|--------|----------|------------------------------|
| sd_clk0     | 1  | output | sdc_mclk | SD card operate clock output |
| sd_clk1     | 1  | output | sdc_mclk | SD card operate clock output |
| sdc_clk0_oe | 1  | output | sdc_mclk | SDC clk0 output enable       |
| sdc_clk1_oe | 1  | output | sdc_mclk | SDC clk1 output enable       |
| cmdi        | 1  | input  | sdc_mclk | SD card response input       |
| cmdo        | 1  | output | sdc_mclk | SD card cmd output           |
| cmdoe       | 1  | output | sdc_mclk | SD card cmd output enable    |
| dati        | 8  | input  | sdc_mclk | SD card data input           |
| dato        | 8  | output | sdc_mclk | Host data output             |
| datoe       | 8  | output | sdc_mclk | Host data output enable      |

### 10.1.3.3 Clock Description

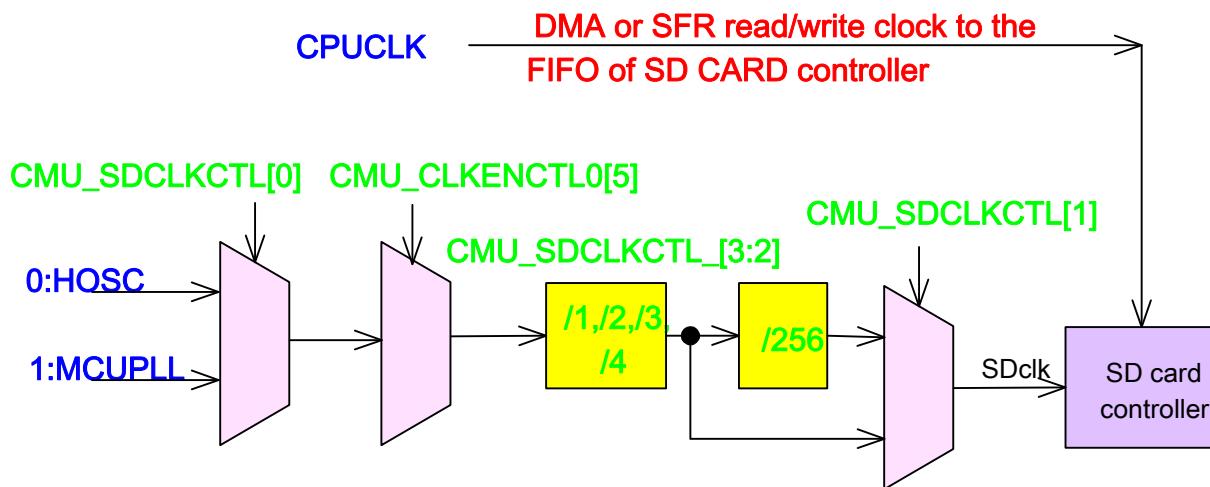


Figure 10-4 SDC clock Description

#### 10.1.4 AC Parameter

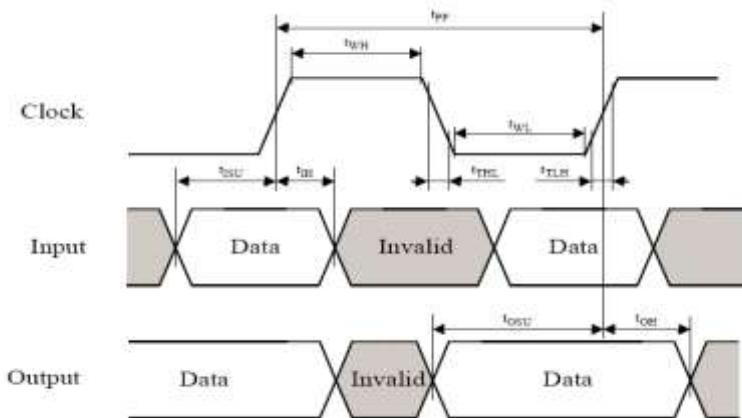


Figure 10-5 3.3V signaling default speed timing diagram

Table Threshold level for 3.3V voltage range

| Parameter           | Symbol | Min       | Max     | Unit | Remark |
|---------------------|--------|-----------|---------|------|--------|
| Supply voltage      | VDD    | 2.7       | 3.6     | V    |        |
| Output high voltage | VOH    | 0.625*VDD | VDD+0.3 | V    |        |
| Output low voltage  | VOL    | VSS-0.3   | 0.25VDD | V    |        |

|                     |     |         |          |    |                   |
|---------------------|-----|---------|----------|----|-------------------|
| In put high voltage | VIH | 0.75VDD |          | V  | IOH=-2Ma          |
| Input low voltage   | VIL |         | 0.125VDD | V  | IOL=2Ma           |
| Power up time       |     |         | 250      | ms | From 0 to VDD min |

CL=Chost+Cbus+Ccard

**Table 3.3V signals timing Default speed mode**

| Parameter   | symbol | Min | Max   | unit | Remark                      |
|---|--------|-----|-------|------|-----------------------------|
| Clock CLK   |        |     |       |      |                             |
| Clock frequency data Transfer Mode<br>(Push Pull) | fpp    | 0   | 25/26 | MHz  | CL<=30Pf(tolerance +100KHz) |
| Clock frequency identification Mode(Open Drain)   | Fod    | 0   | 400   | KHz  | Tolerance:+20KHz            |
| Clock low time                                    | Twl    | 10  |       | ns   | Chost+Cbus<=30pf            |
| Clock low time                                    | Twh    | 10  |       | ns   | Chost+Cbus<=30pf            |
| Clock rise time                                   | Ttlh   |     | 10    | ns   | Chost+Cbus<=30pf            |
| Clock fall time                                   | Tthl   |     | 10    | ns   | Chost+Cbus<=30pf            |
| Inputs CMD DAT(reference to CLK)                  |        |     |       |      |                             |
| Input setup time                                  | Tisu   | 3   |       | ns   | CL<=30Pf                    |
| Input hold time                                   | Tih    | 3   |       | ns   | CL<=30Pf                    |
| Output CMD DAT(reference to CLK)                  |        |     |       |      |                             |
| Output setup time                                 | Tosu   | 5   |       | ns   | CL<=30Pf                    |
| Output hold time                                  | Toh    | 5   |       | ns   | CL<=30Pf                    |

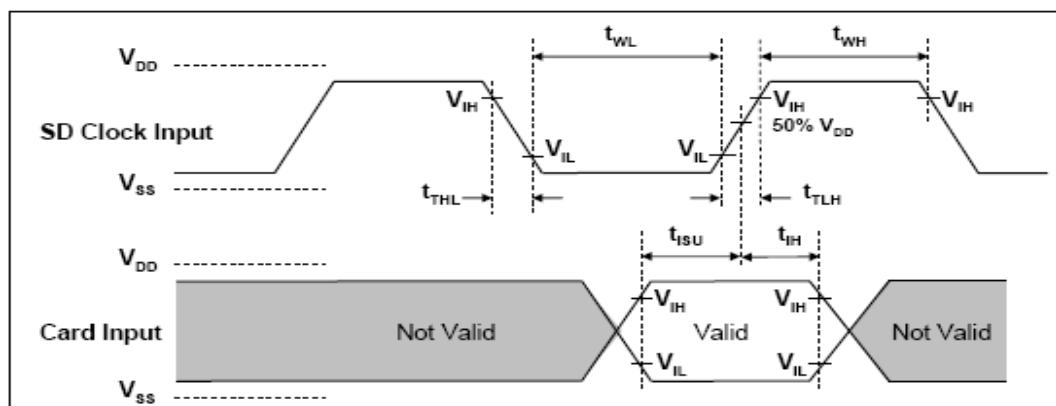


Figure 10-6 3.3V signaling high speed card input timing.(SDC output)

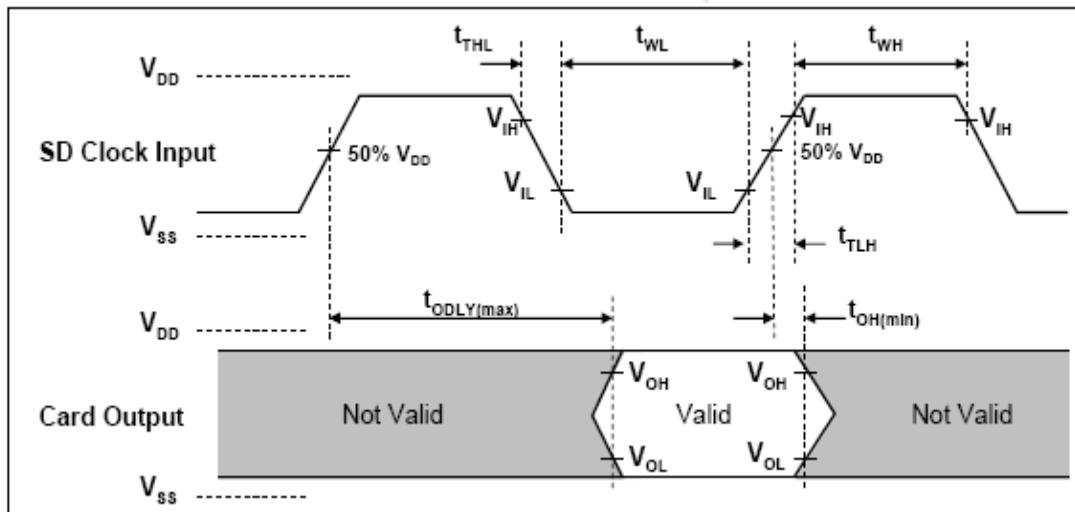


Figure 10-7 3.3V signaling high speed card output timing.(SDC input)

**Table 3.3V signals timing high speed mode**

| Parameter  | symbol | Min     | Max   | unit | Remark                         |
|--|--------|---------|-------|------|--------------------------------|
| Clock CLK  |        |         |       |      |                                |
| Clock frequency data Transfer Mode<br>(Push Pull)  | fpp    | 0       | 50/52 | MHz  | CL<=30Pf(tolerance<br>+100KHz) |
| Clock frequency identification Mode(Open<br>Drain) | Fod    | 0       | 400   | KHz  | Tolerance:+20KHz               |
| Clock low time                                     | Twl    | 7       |       | ns   | Chost+Cbus<=30pf               |
| Clock low time                                     | Twh    | 7       |       | ns   | Chost+Cbus<=30pf               |
| Clock rise time                                    | Ttlh   |         | 3     | ns   | Chost+Cbus<=30pf               |
| Clock fall time                                    | Tthl   |         | 3     | ns   | Chost+Cbus<=30pf               |
| Peak voltage on all lines                          |        | -0.3    | 0.3   | V    |                                |
| Inputs CMD DAT(reference to CLK)                   |        |         |       |      |                                |
| Input setup time                                   | Tisu   | 6-delay |       | ns   | CL<=30Pf                       |
| Input hold time                                    | Tih    | 2.5     |       | ns   | CL<=30Pf                       |
| Output CMD DAT(reference to CLK)                   |        |         |       |      |                                |
| Output setup time                                  | Tosu   | 6       |       | ns   | CL<=30Pf                       |
| Output hold time                                   | Toh    | 2       |       | ns   | CL<=30Pf                       |

## Bus signal Line Load

| Parameter                      | Symbol | Min | Normal | Max | Unit | Remark                  |
|--------------------------------|--------|-----|--------|-----|------|-------------------------|
| Pull up resistance for CMD     | Rcmd   | 4.7 | 50     | 100 | Kohm | To prevent bus floating |
| Pull up resistance for dat0-7  | Rdat   | 50  | 50     | 100 | Kohm | To prevent bus floating |
| Bus signal line capacitance    | CL     |     |        | 30  | Pf   | Single card             |
| Signal card capacitance        | Ccard  |     |        | 7   | Pf   |                         |
| Maximum signal line inductance |        |     |        | 16  | Nh   |                         |

**CL=Chost+Cbus+Ccard**    **Chost+Cbus<=30pf**

## 10.1.5 Operation Manual

SD 卡控制全部采用基于模式的传输；不同模式对应不同的时序，请参考 Transfer Mode Description 的说明；每次传输都通过使能位来开始，结束标志就是该使能位变为 0。

### 10.1.5.1 SDC 寄存器使用的一些注意事项

#### 1, SD\_TF\_CTL

这个寄存器是最重要的一个 SDC 状态机控制寄存器。其 bit7 即是状态机启动的控制位；也是状态机是否运行的状态位。还能通过往这个 bit 写“0”，终止 SDC 的状态机的当前操作，并将状态机复位。（此复位不复位一些配置寄存器，比如 CMD, BLKSIZE 等寄存器，而全局 reset(MRCR 里面的 SDC reset) 则会把 SDC 模块的全部寄存器都复位。软件可以根据实际灵活使用）。在传输过程中，如果 SDC 发现有任何错误，比如 CRC7, CMD\_NO\_RSP, CRC16 等错误，控制会立刻停止，并清除此位；此时，如果 DMA 在等待 SDC 传输数据，则 DMA 一定会一致等待下去，所以推荐软件先检查 SDC 是否结束，在检查 DMA 是否结束。或者说先给等 SDC 结束一个较长时间的超时，再给 DMA 设置一个较短时间的超时。

建议软件在使用中，不要关闭 CRC7 和 CRC16 检查机制。比如在 CMD1 和 ACMD41 的时候，软件知道这个命令会报 CRC7 错误即可。

- 2, SD\_PAD\_CTL 有一点修改，原先的设计是 CMD\_LOW\_enable 只能在 Boot 模式 (TM\_mode9/10) 的时候配置，现在修改为不管在何时，只要是使能，则 CMD 线被 SDC 拉到低电平。减少硬件的强制逻辑，这一位可以在调试的时候提供一定帮助，可以通过使能此位，再检查 SD\_STATE 的 CMD\_STATE，或者万用表测量 CMD 的电平，以验证模块设置是否 OK。
- 3, SD\_DATA\_FIFO 这个寄存器在使用的时候需要注意：如果是 CPU 模式操作，不要在调试工具的 memory 窗口上显示这个寄存器。一些调试工具或实时刷新，就会去读取这个寄存器，则这样会导致 SDC 认为 FIFO 被访问过，指针移位，后续的我们的 IC 真实的动作就被被这个调试工具的动作所干扰，导致数据出错。解决的办法是 SDC 在 CPU 模式下传输数据不开 memroy 观察窗口，或者开了，也不把这个寄存器的位置显示出来。
- 4, SDC 模块超时计数是一个很大的单位，在 10MHz 的 SD\_CLK 的时候，其写 1 就是 200+Ms（频率越高，则时间越短，成简单的反比关系，请根据实际 SD\_CLK 的值换算。）。故软件需要仔细分析，等待的超时需求是多大，以免设置不当。超时也会导致 SD\_TF\_CTL 的 bit7 结束，故超时错误和 SD\_STATE 中的错误位的关系是等价的。（和 CRC7\_ERR, CMD\_NO\_RSP, CRC16\_ERR 是并列的，只要有一个错误，SDC 就停止状态机）。
- 5, SD 卡的时钟源选择和分频比选择寄存器移动到 CMU DIG 模块。原因是：哪个模块做的事情，就将 REG 分配到那个模块，以免 SDC 模块的 REG 控制 CMU 模块的工作状态。
- 6, 发送的 CLK 的数目，被认为是模块自己的动作，所保留在本模块寄存器上。请注意寄存器的不同。
- 7, SD\_CLK\_CTL 的 LBE 建议在一般的情况下不要使能。其设计的初衷是：在我们的需求中，可能会采用 TM\_mode4+6+6+6----的这种方式读取数据，如果在 TM\_mode4 或者 6 的任何一个中多发送额外的 8 个 clk 则可能会导致卡开始放下一个 block 的数据，而在这些 8 个额外的时钟器件，SDC 是不收数据的，这样就会导致 SDC 收到的数据和 SD 卡送的数据不对其，导致最终传输错误。所以我们控制器在 TM\_mode4, 6 结束的时候，是默认不发送额外的 8 个时钟的。但是为了和 SPEC 兼容，我们还是保留了 SDC 在完成最后一个 Block 的读取动作的时候，多发送 8 个时钟的功能，但是按照目前来看，

还未发现不发送有问题的情形，故可以不发送。

- 8, 在 GL5115 的 SDC 中存在一个 bug，在如 7 中的情形下，如果多发送 8 个时钟，则会导致控制器状态异常，下一笔的写动作无法正常发出。此时的解决办法有 2 种，1，在任何读取动作的时候，都不发送多余的 8 个 clk; 2，在任何读写动作之前，先 reset 一下 SDC 的 fifo。推荐采取第一个解决办法。

### 10.1.5.2 SDC 常见应用情景软件控制流程

#### 1、SDC 命令传输

SDCMD 配置读写卡的命令 number。

SDARGRSP，配置卡命令的参数。

SDTFCTL，的 BIT0~3 配置命令模式 (0,, 1, 2, 3)

SDTFCTL 的使能位 (BIT7)，启动 SDC；如果该 BIT 变都为 0，表示传输结束。

最后软件读 SDSTAT/SD\_TIMEOUT\_CNT 的状态寄存器，检查传输是否正确。

#### 2、SDC 数据传输

SDCMD 配置读写卡的命令 number;

SDARGRSP，配置读写卡内部数据的地址；

SDBLKSZH/SDBLKSZL，配置传输 BLOCK 的大小；

SDBLKNUM 配置传输 BLOCK 的数目。

SDTFCTL，的 BIT0~3 配置传输模式 (4, 5, 6, 7);

配置 DMAAnSADDR0, DMAAnDADDR0, RAM 中的数据起首地址。

配置 DMAAnFrameLen，数据传输的 Byte number

配置 DMAAnCTL1，DMA 的设备选择 SD 卡设备

SDTFCTL 的使能位 (BIT7), DMAAnCTL0 的使能位 (BIT0), 启动 SDC, 启动 DMA, 开始传输；

如果该两 BIT 变都为 0，表示传输结束。

最后软件读 SDSTAT/SD\_TIMEOUT\_CNT 的状态寄存器，检查传输是否正确。

#### 3、SDC 的 BOOT 模式

SDBLKSZH/SDBLKSZL，配置传输卡传输 BLOCK 的大小；

SDTFCTL，的 BIT0~3 配置传输模式 (9, 10);

配置 DMAAnSADDR0, DMAAnDADDR0, RAM 中的数据起首地址。

配置 DMAAnFrameLen，数据传输的 Byte number

配置 DMAAnCTL1，DMA 的设备选择 SD 卡设备

然后配置 SDTFCTL 的使能位 (BIT7), 设置 DMAAnCTL0 的使能位 (BIT0), 启动 SDC, 启动 DMA, 开始传输；如果该两 BIT 变都为 0，表示传输结束。

最后软件读 SDSTAT/SD\_TIMEOUT\_CNT 的状态寄存器，检查传输是否正确。

### 10.1.5.3 MFP 的注意事项

- 1, GL5115 中的 mfp 和相关的模块的优先级别如下：analog>EJTAG>DBG>GPIO>mfp. 所以要保证 mfp 将相关 pad 分配给 SDC 模块，则必须先保证在 mfp 前面的各个模块 SDC 相关的功能都未开启。否则即使 mfp 设置正确，SDC 也无法输出。

- 2, GPIO\_C3 既能分配为 SDC\_D1 也能分配为 SD\_CLKB 二者不可兼顾，故在需要 SD\_CLKB 输出的时候，只能配置卡为 1 线。
- 1, SD 模块的 CMD, DAT 信号线都需要上拉，IC 上请选择内部上拉。请在 GPIO 模块的 PUPD 寄存器上选择。
- 2, SD 模块的驱动能力，CM,CLK 能单独设置，数据线只能一起设置。故 SD\_CLKB (SD\_D0 共 pad) 只能和 D0, D2, D3 一起设置，这一点需要注意了。
- 3, 别的模块，如果要用 SD 卡模块的 GPIOC0~C5，首先，尽量不要用 CLK，其次不要用 CMD，再次不用 D3。如果开启了 SDIO 中断，D1 也不要用。

#### 10.1.5.4 卡初始化注意事项

- 1, 初始化的时钟配置： SD 和 MMC 卡在初始化的时候需要配置卡的时钟为低频。需求为 400K 以下，其和 CMD 线上的上拉电阻的阻值密切相关。根据经验，我们的项目是一般设置为 93K 左右。(24MHz/256)
- 2, 初始化发送 CMD0 之前，需要注意将 D3 上拉。否则卡可能会导致卡进入 SPI 模式，一旦进入 SPI 模式则对任何 SD 模式的命令都不能正常响应。
- 3, 在 SD 卡的 ACMD41, MMC 卡的 CMD1 检查卡上电是否 ready 的命令的时候，经验显示，一些现在的大容量的卡，需要的 ready 的时间为接近 900Ms, 这一点在一些新的卡上表现明显，BROM 驱动和 SDK 的等卡 ready 的时间需要相应的调整，以免出现未等到卡 ready 就超时的问题。

#### 10.1.5.5 卡数据传输注意事项

- 1, 一般情况下，SDC 模块和 DMA 传输，需要等待 2 个都结束。
- 2, 一般情况下，SDC 的 FIFO 位宽为 8 位，和 USB 直传的时候(通过 DMA 搬数据)，需要设置 SD\_FIFO 位宽为 32 位，以提高效率。
- 3, 如果传输中有超时机制，请先等 SDC 结束，原因就寄存说明 1.
- 4, 写数据的 CRC7/W\_CRC16 错误的时候，调节 R\_delay 是有用的。原因如下：  
如果用 TM\_mode5 发送数据，则 SDC 的完整流程如下： send\_CMD(W\_CLK), reseve\_RSP(R\_CLK), send\_data(W\_CLK), reseve\_CRC\_status(R\_CLK), wait\_BUSY(W\_CLK). 可见，R\_CLK 的阶段是存在于写动作中的，故调节 R\_delay 对写数据的 W\_CRC16\_err 是有调节作用的。

#### 10.1.6 SDC Register List

**Table SD CARD Controller Registers**

| Index | Mnemonic      | Description                     | BANK |
|-------|---------------|---------------------------------|------|
| 0x90  | SD_CMD        | SD/MMC CMD Register             | 0x09 |
| 0x91  | SD_CMD_ARGRSP | SD/MMC Argument or RSP Register | 0x09 |
| 0x92  | SD_RSP_POIN   | SD/MMC RSP Point Register       | 0x09 |

|      |                |                                   |      |
|------|----------------|-----------------------------------|------|
| 0x93 | SD_TF_CTL      | SD/MMC control register           | 0x09 |
| 0x94 | SD_STATE       | MMC/SD status Register            | 0x09 |
| 0x97 | SD_BLK_SIZE_H  | SD/MMC High Block size Register   | 0x09 |
| 0x98 | SD_BLK_SIZE_L  | SD/MMC Low Block size Register    | 0x09 |
| 0x99 | SD_BLK_NUM     | SD/MMC BLOCK number Register      | 0x09 |
| 0x9a | SD_CLK_CTL     | SD/MMC Clock Control Register     | 0x09 |
| 0x9b | SD_PAD_CTL     | SD/MMC PAD CONTROL Register       | 0x09 |
| 0x9c | SD_INT_CTL     | SD/MMC INTERRUPT Control Register | 0x09 |
| 0x9d | SD_DATA_FIFO   | SD/MMC Data FIFO Register         | 0x09 |
| 0x9e | SD_TIMEOUT_CTL | Data Timeout Counter Register     | 0x09 |
| 0xad | SD_TIMING_CTL  | CARD TIMING CONTROL               | 0x09 |
| 0xae | SD_DBG_CTL     | CARD Debug Signal Output Register | 0x09 |

### 10.1.7 Register Description

#### 10.1.7.1 SD\_CMD

**SD/MMC CMD Register, SFR BANK 0x9, SFR Address 0x90**

| Bit Number | Bit Mnemonic | Function         | Access | Reset |
|------------|--------------|------------------|--------|-------|
| 7:6        | Reserved     | Fixed value: 01  | R      | 01    |
| 5:0        | CMD_INDEX    | Command register | R/W    | 0     |

#### 10.1.7.2 SD\_CMD\_ARGRSP

**SD/MMC Argument or RSP Register, SFR BANK 0x9, SFR Address 0x91**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | CMDRSP       | Argument or RSP value. When write, the register value is Argument, from MSB byte to LSB byte (serial write 4 times); When read, the register Value is that SD_RSP_P0IN value refers to. | R/W    | 0     |

#### 10.1.7.3 SD\_RSP\_P0IN

**SD/MMC RSP Point Register, SFR BANK 0x9, SFR Address 0x92**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | Reserved     | Reserved  | /      | /     |
| 4:0        | RSP_POINT    | Write Argument or Read RSP point (byte)<br>00000b RSP byte0<br>00001b RSP byte1<br>...<br>01111b RSP byte15<br>10000b RSP byte16<br>others reserved | R/W    | 0     |

#### 10.1.7.4 SD\_TF\_CTL

**SD/MMC transfer control register, SFR BANK 0x9, SFR Address 0x93**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | TF_START     | Transfer start.<br>When set, SD/MMC transfer starts according to the transfer mode, access mode, and other control field. It will automatically clear after transfer complete or Error occurred. When clear, transfer stop immediately. | R/W    | 0     |
| 6          | CRC7_DIS     | CRC7DIS<br>command response CRC7 Check disable: When set this bit, indicates don't check CRC7   | R/W    | 0     |
| 5          | CRC16_DIS    | CRC16DIS<br>Write and read Data CRC16 Check disable. When set this bit indicates don't check data CRC16   | R/W    | 0     |
| 4          | CPU_DMA_SEL  | DATA TRANSFER CHANNEL:<br>1: CPU CHANNEL mode<br>0: DMA CHANNEL mode.   | R/W    | 0     |

|     |         |  |     |      |
|-----|---------|--|-----|------|
| 3:0 | TM_MODE | Transfer Mode[3:0] Specifies the transfer mode when transfer start bit is set<br>0000b: Transfer command without response<br>0001b: Transfer command with 6 bytes response(not including Data transfer)<br>0010b: Transfer command with 17 bytes response(not including Data transfer)<br>0011b: Transfer command with 6 bytes response, and with busy(not including Data transfer)<br>0100b: Transfer data in mode with command(include crc16 checked)<br>0101b: Transfer data out mode with command (include CRC and busy checked)<br>0110b: Transfer data in mode without command<br>0111b: Transfer data out mode without command (include CRC and busy checked)<br>1000b: Transfer only clock(without any command, response, and data)<br>1001b: BOOT mode ,Transfer data in mode with Command (hardware check data crc16 , do not check boot mode acknowledge).<br>1010b: BOOT mode ,Transfer data in mode with Command (hardware check data crc16 , also check boot mode acknowledge).<br>1011b~1111b: reserved | R/W | 1111 |
|     |         |  |     |      |
|     |         |  |     |      |
|     |         |  |     |      |
|     |         |  |     |      |
|     |         |  |     |      |
|     |         |  |     |      |
|     |         |  |     |      |
|     |         |  |     |      |
|     |         |  |     |      |

### 10.1.7.5 SD\_STATE

**MMC/SD status Register, SFR BANK 0x9, SFR Address 0x94**

| Bit Number | Bit Mnemonic     | Function  | Access | Reset |
|------------|------------------|---|--------|-------|
| 7          | D0_STA           | DAT0 Status:<br>This bit reflects the level of the DAT0 Signal of SD/MMC card   | R      | X     |
| 6          | D1_STA           | DAT1 Status:<br>This bit reflects the level of the DAT1 Signal of SD/MMC card   | R      | X     |
| 5          | CMD_STA          | CMD Status:<br>This bit reflects the level of the COMMAND Signal of SD/MMC card   | R      | X     |
| 4          | NO_RSP           | Command Line No response<br>(only for command with response)<br>. Write 1 clear. This bit is also cleared when transfer start is set.                     | R/W    | 0     |
| 3          | CMD_COMP<br>LETE | Command Line transfer Complete:<br>Write 1 clear. This bit is auto cleared when Transfer Start is set, and is set when command line transfer is complete. | R/W    | 0     |

|   |           |  |     |   |
|---|-----------|--|-----|---|
| 2 | W_CRC16_E | Write CRC16 Error:<br>When set, this indicated a CRC write error detected over the data line 0. Write 1 clear. This bit is auto cleared when Transfer Start is set.  | R/W | 0 |
| 1 | R_CRC16_E | Read CRC16 Error:<br>When set, this indicated a CRC16 error detected over the received data. It is also set when receiving the wrong boot mode transfer gement pattern. Write 1 clear.<br>This bit is auto cleared when Transfer Start is set. | R/W | 0 |
| 0 | CRC7_E    | CRC7 Error:<br>When Set, this indicated CRC7 error detected over the response. Write 1 clear. This bit is auto cleared when Transfer Start is set.   | R/W | 0 |

### 10.1.7.6 SD\_BLK\_SIZE\_H

**SD/MMC High Block size Register, SFR BANK 0x9, SFR Address 0x97**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | Reserved     | reserved  | x      | x     |
| 3:0        | BLK_SIZ_H    | Block Count[11:8] this field determines A block size, that is how many bytes counter in a block | R/W    | 02h   |

### 10.1.7.7 SD\_BLK\_SIZE\_L

**SD/MMC Low Block size Register, SFR BANK 0x9, SFR Address 0x98**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | BLK_SIZ_L    | Block Count[7:0] this field determines A block size, that is how many bytes counter in a block | R/W    | 00h   |

Note: 0 meaning zero bytes to transfer.

### 10.1.7.8 SD\_BLK\_NUM

**SD/MMC BLOCK number Register, SFR BANK 0x9, SFR Address 0x99**

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:0        | BLK_SIZ_NUM  | This field determines block number in once read or write operations. Default value is 1, The Max is 256(when this reg is set as "0",the block number is the max 256). | R/W    | 1     |

### 10. 1. 7. 9 SD\_CLK\_CTL

**SD/MMC Clock Control Register, SFR BANK 0x9, SFR Address 0x9a**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:6        | Reserved     | reserved   | -      | -     |
| 5          | LBE          | Sending 8clock ENABLE:<br>0 : Sending 8clock more at the end of data or command end.<br>1 : Stop immediately at the end of data or command end.<br>EX: this bit is only used in data transfer mode 4 and mode 6 when reading data.   | R/W    | 0     |
| 4:3        | Reserved     | reserved   | -      | -     |
| 3:2        | CLK_NUM_SEL  | Transfer clock number. Used in transfer mode 8.<br>00b: clock number is infinite.<br>01b: clock number is 64 clocks<br>10b: clock number is 128 clocks<br>11b: clock number is 256 clocks<br><br>Note:<br>1,frist :set SD_TF_CTL bit3:0 as "1000b",select SDC "send only CLK"<br>2, second: set here, select send CLK mode.<br>3, third: set SD_TF_CTL bit7 start SDC. | R/W    | 0     |
| 1          | Reserved     | reserved   | -      | -     |
| 0          | Reserved     | reserved   | -      | -     |

### 10. 1. 7. 10 SD\_PAD\_CTL

**SD/MMC PAD control Register, SFR BANK 0x9, SFR Address 0x9b**

| Bit Number | Bit Mnemonic | Function | Access | Reset |
|------------|--------------|----------|--------|-------|
| 7          | Reserved     | reserved | R/W    | 0     |

|     |            |   |     |   |
|-----|------------|---|-----|---|
| 6   | CMD_LOW_EN | CMD LOW Enable<br>Enable CMD line drive low level by Software<br>1: Drive CMD line to low level<br>0: not drives CMD line to low.<br><br>NOTE:<br>The software can set this bit to 1 to drive CMD line to low level.<br>After power up, the card can be maintained in boot mode if the<br>CMD line is always low level. So you can read boot data using<br>mode 9 or mode 10. | R/W | 0 |
| 5   | Reserved   | reserved  | -   | - |
| 4   | Reserved   | reserved  | -   | - |
| 3   | SDIO_EN    | SDIO mode enable , 1:enable sdio mode ; 0:disable sdio mode .   | R/W | 0 |
| 2:1 | BUS_WIDTH  | Data Bus Width.<br>00: SD/MMC 1-bit data mode<br>01: SD/MMC 4-bit data mode<br>10: MMC 8-bit data mode<br>11: Reserved  | R/W | 0 |
| 0   | CLK_SEL    | SDMMC CLK0 PIN enable control.<br>0: SDMMC clock0 select and output.<br>1: SDMMC clock1 select and output.  | R/W | 0 |

### 10.1.7.11 SD\_INT\_CTL

**SD/MMC INTERRUPT Control Register, SFR BANK 0x9, SFR Address 0x9c**

| Bit Number | Bit Mnemonic   | Function  | Access | Reset |
|------------|----------------|---|--------|-------|
| 7          | SDIO_INT_EN    | SDIO mode IRQ enable: 1: enable IRQ; 0: disable IRQ   | R/W    | 0     |
| 6          | SDIO_INT_PD    | SDIO MODE IRQ pending: 1: IRQ ,Write 1 clear  | R/W    | 0     |
| 5          | TF_END_INT_EN  | Transfer end IRQ enable: When set, Enable interrupt Request.  | R/W    | 0     |
| 4          | TF_END_INT_PD  | Transfer end IRQ status. Write 1 clear.   | R/W    | 0     |
| 3          | FIFO_WIDTH_SEL | <b>FIFO DATA TRANSFER WIDTH SELECTION</b><br>1: 32BIT FIFO DATA TRANSFER WIDTH SELECTION for<br>USB transfer. NOTE: output DRQ when in err status in this<br>mode.<br>0: NORMAL 8BIT DATA WIDTH. NOTE: MUST SUPPORT<br>dma0~2 BURST8 TRANSFER MODE. | R/W    | 0     |
| 2          | FIFO_RST       | <b>FIFO Reset.</b> When write '1', reset FIFO, clear status bit   | R/W    | 0     |
| 1          | FIFO_FULL      | FIFO Full Flag 1: Full 0: Not Full  | R      | 0     |
| 0          | FIFO_EMPTY     | FIFO Empty Flag 1: Empty 0: Not Empty   | R      | 1     |

### 10.1.7.12 SD\_DATA\_FIFO

**SD/MMC Data FIFO Register, SFR BANK 0x9, SFR Address 0x9d**

| Bits | Description                    | Access | Reset |
|------|--------------------------------|--------|-------|
| 7:0  | SD/MMC DATA FIFO access window | R/W    | X     |

NOTE: only used under CPU mode

### 10.1.7.13 SD\_TIMEOUT\_CTL

**Data Timeout Counter Register , SFR BANK 0x9, SFR Address 0x9e**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7          | TOUT_EN      | Hardware Timeout enable.<br>0: Disable Hardware Timeout.<br>1: Enable Hardware Timeout.<br><br>The timeout period is configured by <i>Data Timeout Counter</i> | R/W    | 0     |
| 6          | TOU_ERR      | Data Timeout error : When set, this indicated a timeout error has detected in the latest transfer. Write '1' clear. Also clear when next transfer start.       | R/W    | 0     |
| 5:0        | TOUT_CNT     | HARDWARE Timeout Counter : This counter determine the timeout time of SD card data output.   | R/W    | 1     |

### 10.1.7.14 SD\_TIMING\_CTL

**CARD interface control, SFR BANK 0x9, SFR Address 0xAd**

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | R_DELAY      | Latch input DATA delay Time select (when host controller latching data, delay the inside latching data clock to compensate signal transmission delay):<br><br>00h : Don't delay the latch time of input signal.<br>01~09h : Delay the latching clock of input signal, compared to output clock. The Increment Delay Unit is 0.4ns, error 33%.<br>0a~0eh: Delay the latching clock of input signal, compared to output clock. The Increment Delay Unit is 1ns, error 33%.<br>0fh: Delay the latching clock of input signal, compared to output clock. The Increment Delay Unit is 4ns, error 33%. | R/W    | 0     |

|     |         |  |     |   |
|-----|---------|--|-----|---|
| 3:0 | W_DELAY | Output DATA output delay Time select:<br>00h : Don't delay the latch time of input signal.<br>01~09h: Delay the output time of output signal, compared to output clock. The Increment Delay Unit is 0.4ns, error 33%.<br>0a~0eh: Delay the output time of output signal, compared to output clock. The Increment Delay Unit is 1ns, error 33%.<br>0fh: Delay the output time of output signal, compared to output clock. The Increment Delay Unit is 4ns, error 33%. | R/W | 0 |
|-----|---------|--|-----|---|

### 10.1.7.15 SD\_DBG\_CTL

**CARD debug signal control, SFR BANK 0x9, SFR Address 0xAe**

| Bit Number | Bit Mnemonic | Function | Access | Reset |
|------------|--------------|----------|--------|-------|
| 7:4        | Reserved     | reserved | R      | 0     |

|     |  |     |   |
|-----|--|-----|---|
|     |  |     |   |
| 3:0 | <p>DBG_MODE Debug Signal Output selection:</p> <pre> 4'h0: {1'h0, drq, dma_cnt[3:0], clk_dma, dma_wr, dma_do[7:0]}; 4'h1: {1'h0, drq, dma_cnt[3:0], clk_dma, dma_rd, dma_di[7:0]}; 4'h2: {4'b0000, diow_ms, dior_ms, mem_rd, mem_wr, umod_en,         udreq, uack, drq_sdma, blk_end, cnt_end, rd_st[1:0]}; 4'h3: {4'b0000, diow_sdc, dior_sdc, mem_rd, mem_wr,         udreq, uack,         mem_ready, drq_sdma, blk_end, cnt_end, rd_st[1:0]}; 4'h4: {3'b000,         swr_dat_crc_en, swr_dat_crc_cal, sif_handshake_ok,         clk_ms, sdmard, do_ms[7:0]}; 4'h5: {2'b00, srd_dat_crc_en, srd_dat_crc_cal, sif_rd_crc_lat,  sif_handshake_ok, clk_ms, sdmawr, srd_shift[7:0]}; 4'h6: {3'b000,         pwr_dat_crc_en, pwr_dat_crc_cal, pif_handshake_ok,         clk_ms, pdmard, do_ms[7:0]}; 4'h7: {2'b00, prd_dat_crc_en, prd_dat_crc_cal, pif_rd_crc_lat,  pif_handshake_ok, clk_ms, pdmawr, prd_shift[7:0]}; 4'h8: {3'h0, clk_de115, cmd_stu, dmard, wrdat_oen, wr_clk_oen,         dmado[7:0]}; 4'h9: {3'h0, clk_de115, cmd_stu, dmawr, rd_dat_oen, rd_clk_oen,         dmadi[7:0]}; 4'ha: {3'h0, clk_de115, rsp_rec_en, cmd_in2, rsp_st_set, rsp_dat_1d,         rsp_dat[7:0]}; </pre> | R/W | 0 |

## 10.2 External Memory Interface (赵天亮、蔡瑞仁)

| 日期         | 版本     | 描述   | 修订人 |
|------------|--------|--|-----|
| 2012-07-24 | V1. 00 | initial  | 赵天亮 |
| 2012-08-09 | V1. 01 | 为了保证和 CPU 设计一致, 将 EM_DL 寄存器定义地址从 0x06f4 改到 0x06ff。 | 赵天亮 |
| 2012-09-12 | V1. 02 | 1、数据线回复到 16bit16bit<br>2、添加控制信号输出使能位               | 赵天亮 |
| 2012-12-12 | V2. 02 | 完善 operation manual 章节                             | 赵天亮 |

## 10.2.1 Features

- ◆ Support 8bit/16bit 8080i write and read timing
- ◆ Programmable Waite State

## 10.2.2 Function Description

CPU can write or read through EXTMEM\_DH and EXTMEM\_DL to access the extended bus.

When it is set to 8bit interface, cpu writes or reads EXTMEM\_DL, the bus accesses the lower 8bit data bus.

When it is set to 16bit interface, cpu writes the EXTMEM\_DH first, and then EXTMEM\_DL, the bus writes the 16bit data bus.

When it is set to 16bit interface, cpu reads the EXTMEM\_DL first, and then EXTMEM\_DH, the bus reads the 16bit data bus.

Set the wait state to suit the frequency of the external NandFlash or LCM interface.

When wait state is set to more than 0, the busy signal will hold the CPU until a certain time which depends on wait state setting. The CPU wait state will count on the setting of Ext\_Wait (Reg\_Allbank\_0x96)

The total cycles of extended interface write or read must be equal to the cpu write or read cycles. That is,  $\text{EXTMEM\_WT}[7:4]+\text{EXTMEM\_WT}[3:0]+1=\text{Ext\_Wait}[4:0]$ .

## 10.2.3 Module Description

### 10.2.3.1 Block Diagram

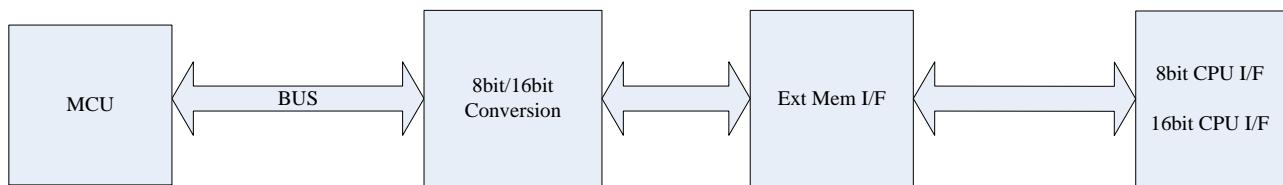


Figure 10-8 Extended Memory IF Block Diagram

### 10.2.3.2 Timing

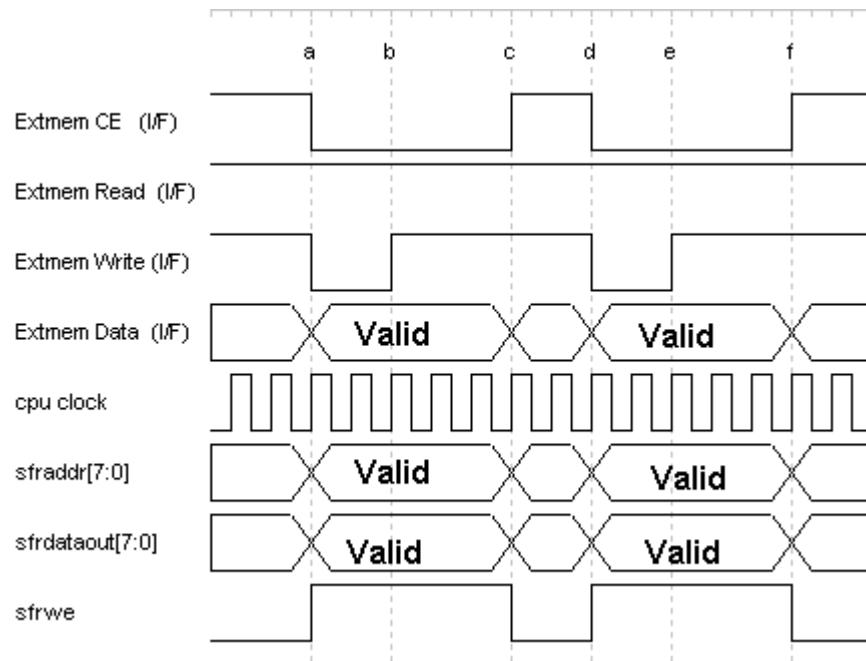


Figure 10-9 8080i Write Timing

Write Timing:

a to b is the low state of writing cycle, the cycles depends on EXTMEM\_WT[3:0]

b to c is the high state of writing cycle. The cycles depends on EXTMEM\_WT[7:4]

a to c is a writing cycle,

When CPU writes the EXTMEM\_DL register, the EXTMEM CEB is driven to low level, the host will drive the EXTMEM Data bus until the EXTMEM Write cycle is over. When the EXTMEM CEB is low level, the device will be chip selected.

The EXTMEM Write signal will be driven to low level until the low state counter is EXTMEM\_WT[3:0], then the write signal will be driven to high level until the high state counter is EXTMEM\_WT[7:4].

The device will latch the data at the rise edge of EXTMEM Write.

If the writing cycles of cpu ends before the extended writing cycle, the extended writing cycle is forced to end.

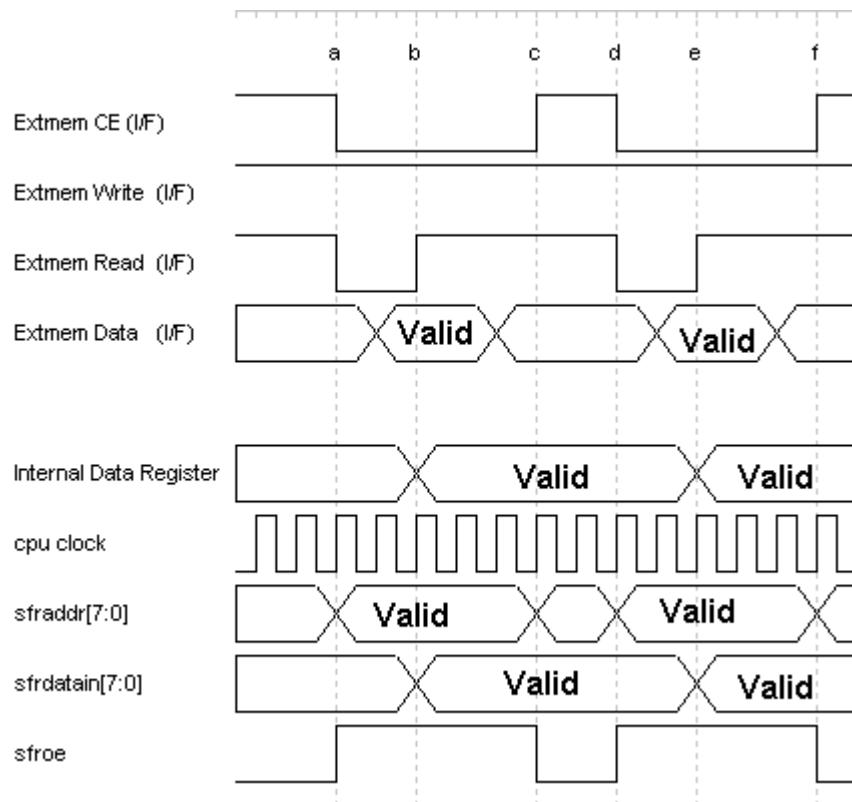


Figure 10-10 8080i Read Timing

#### Read Timing:

a to b is the low state of reading cycle, the cycles depends on EXTMEM\_WT[3:0]

b to c is the high state of reading cycle. The cycles depends on EXTMEM\_WT[7:4]

a to c is a read cycle,

When CPU reads the EXTMEM\_DL register, the EXTMEM CEB is driven to low level until the EXTMEM Read cycle is over. When the EXTMEM CEB is low level, the device will be chip selected.

The EXTMEM Read signal will be driven to low level until the low state counter is EXTMEM\_WT[3:0], then the read signal will be driven to high level until the high state counter is EXTMEM\_WT[7:4].

When EXTMEM Read is low level , the device will drive the EXTMEM Data bus.

The host will sample the EXTMEM Data at the rise edge of EXTMEM Read, the then latch the data in an internal data register and sfrdatain[7:0].

At the fall edge of sfroe, the cpu latch the from sfrdatain[7:0]

If the reading cycles of cpu ends before the extended writing cycle, the extended writing cycle is forced to end.

#### NOTE:

1、 CEB signal should be controller by LCDC ,when writing or reading data, it should be low level, otherwise it should be high level

2063、 Extmem read signal should return to high when Extmem is not used

## 10.2.1 Operation Manual

### 10.2.1.1 Operation Flow for Software

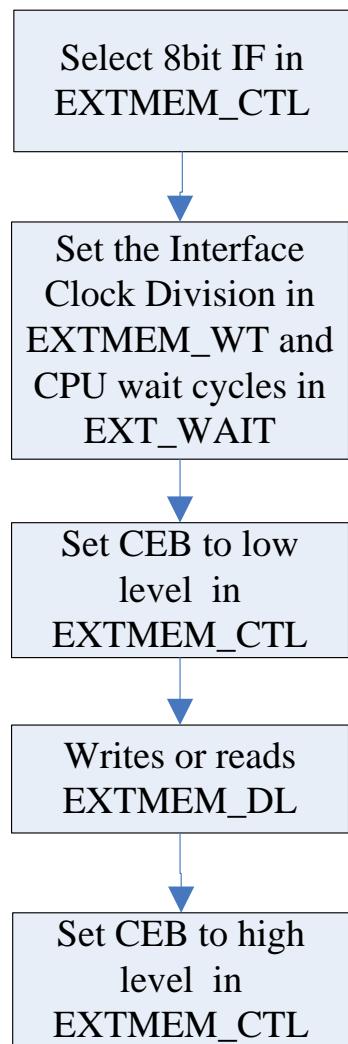


Figure 10-18 Reads or Writes 8bit Interface

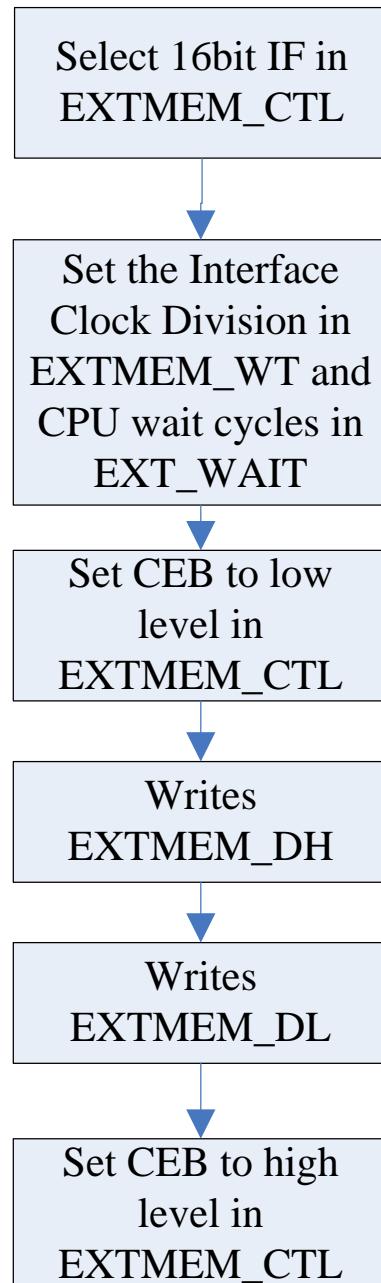


Figure 10-19 Writes 16bit Interface

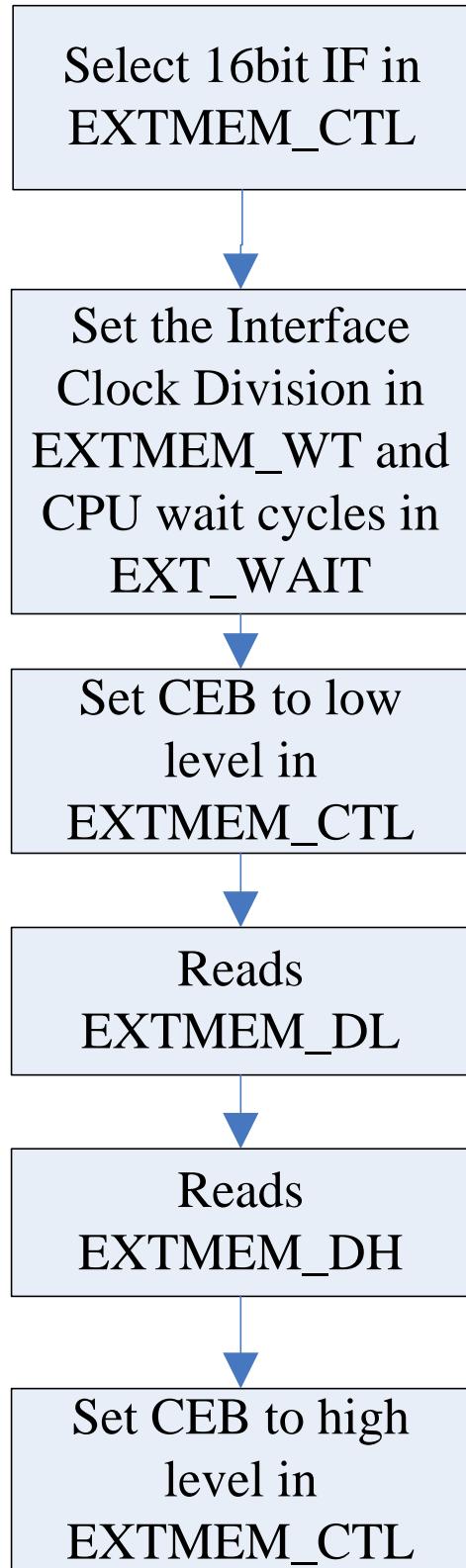


Figure 10-20 Reads 16bit Interface

注意：

在外扩总线的 16bit 位宽的读写过程中，由于 CPU 只对 DL 进行 EX\_wait 等待。所以读取 DL 寄存器动作发生后，只有当 DL 的值是可读的之后，DH 的值才是可读的。故一定要保证先读取 DL 的值，再读取 DH 的值，不能反过来，否则会导致 DH 的值不可靠。同理，写操作的时候，也必须先对 DH 进行写动作，再对 DL 进行写动作，则这样才能将 16bit 的数据正常的发送出去。（可以这么理解，软件只

要对 DL 进行了操作，就相当于启动一次 EM 发送或者接受动作。)

### 10.2.1.2 模块的输出使能信号

注意在 GL5115 的 EM 模块中，添加了 2 个输出使能信号，在 EXTMEM\_CTL 的 bit7 和 bit6 只有当这 2 个位置使能的时候，才能保证模块的信号能送到 mfp。这一点主要是为了防止在 EM 总线不想动作的时候（由于默认 mfp 分配 EM 模块而导致），EM 模块的信号对外有输出，而导致一些比如 led 点亮等非设计所愿的事情发生。请软件人员注意这一点。

### 10.2.2 EM Register List

**Table Extended Memory Interface Registers Address**

| Index | Mnemonic   | Description                                   | BANK |
|-------|------------|---|------|
| 0x9e  | EXTMEM_CTL | Extended Memory Interface Control Register    | 0x06 |
| 0x9f  | EXTMEM_WT  | Extended Memory Interface Wait State Register | 0x06 |
| 0xfe  | EXTMEM_DH  | Extended Memory Interface high Byte Register  | 0x06 |
| 0xff  | EXTMEM_DL  | Extended Memory Interface Low Byte Register   | 0x06 |

### 10.2.3 Register Description

#### 10.2.3.1 EXTMEM\_CTL

Extended Memory Interface Control Register (0x06 0x9e)

| Bit Number | Bit Mnemonic    | Function   | Access | Reset |
|------------|-----------------|--|--------|-------|
| 7          | CE_OUT_EN       | When PAD's mfp select as “EM” function, this bit will controller the output state of the pad:<br>0: EM do not driver the PAD, so the pad should be high Z state<br>1: EM driver the pad according to bit 2:0 of this REG.            | R/W    | 1     |
| 6          | RD_WR_RS_OUT_EN | When PAD's mfp select as “EM” function, this bit will controller the PADS: WR, RD, RS output state<br>0: EM do not driver the PADS, so these pads should be high Z state.<br>1: EM driver these pads according to the timing of EMIF | R/W    | 0     |
| 5          | -               | reserved   | R      | 0     |
| 4          | IFSEL           | 8bit or 16bit bus sel  | R/W    | 0     |

|     |       |  |     |     |
|-----|-------|--|-----|-----|
|     |       | 0: 8bit<br>1: 16bit  |     |     |
| 3   | RS    | RS select<br>0:RS output low voltage level<br>1:RS output high voltage level<br>RS is low or high voltage in the case of writing INDEX/DATA/REG in different LCM (When using NandFlash, this bit should be set to 0 first) | R/W | 0   |
| 2:0 | CESEL | Choose the Chip Select of extended memory interface<br>001 :CE0<br>010 :CE1<br>Others : Reserved   | R/W | 000 |

Note:

- 1, EX\_CE0 will out put “H”, when MFP and this REG is default value.
- 2, EX\_RD,EX\_WR,EX\_RS will out put “high Z” when MFP and this REG is default value.

### 10.2.3.2 EXTMEM\_WT

Extended Memory Interface Wait State Register (0x06: 0x9f)

| Bit Number | Bit Mnemonic | Function                                      | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | WTH          | Write or Read High Voltage Wait State Counter | R/W    | 0000  |
| 3:0        | WTL          | Write or Read Low Voltage Wait State Counter  | R/W    | 0000  |

The total cycles of extended interface write or read must be equal to the cpu write or read cycles. That is, EXTMEM\_WT[7:4]+EXTMEM\_WT[3:0]+1=Ext\_Wait[4:0].

### 10.2.3.3 EXTMEM\_DH

Extended Memory Interface Low Byte Register (0x06: 0xfe)

| Bit Number | Bit Mnemonic | Function                                       | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:0        | EXT_DATH     | The higher 8bit data bus of extended interface | R/W    | 0x0   |

### 10.2.3.4 EXTMEM\_DL

Extended Memory Interface Low Byte Register (0x06: 0xff)

| Bit Number | Bit Mnemonic | Function | Access | Reset |
|------------|--------------|----------|--------|-------|
|            |              |          |        |       |

|     |          |   |     |     |
|-----|----------|---|-----|-----|
| 7:0 | EXT_DATL | The lower 8bit data bus of extended interface | R/W | 0x0 |
|-----|----------|---|-----|-----|

# 11 Transfer and Communication

## 11.1 USB (黄少彬、黄俏)

| 日期         | 版本     | 描    述  | 修订人 |
|------------|--------|---|-----|
| 2012-07-24 | V1. 00 | initial   | 黄少彬 |
| 2012-08-09 | V1. 01 | 1、修改 Endpoint and memory configuration<br>2、将 NTIRQ 寄存器删除，NTIRQ 移到 Usbirq_hcusbirq 寄存器中<br>3、增加 EP1STADDRH/L 和 EP2STADDRH/L 4 个寄存器<br>4、增加 USB 与 DMA 接口 FIFO 的 reset 控制 (0x86_bit[3:2])<br>5、将 EP1、EP2 DMA start 控制位分开分别控制 (0x86_bit[1:0])<br>6、将原先 DMAlenh 寄存器修改为 EP1DMAlenh 寄存器，增加 EP2DMAlenh 寄存器<br>7、删除 SoftVBUS 控制位，VBUS 一直有效。<br>8、删除 OTGIRQ 和 OTGIEN bit3 和 bit1，删除 Otgctrl bit[6:1]<br>9、删除 Otgstatus bit0<br>10、删除 OTG 时间相关寄存器 (0xdf,e1,e2,e3,f4)<br>11、删除 BKDOOR bit3 | 黄少彬 |
| 2012-11-07 | V2. 01 | 1、<br>2、Usbirq_hcusbirq 的 bit6 默认值改为 1  | 黄少彬 |
| 2012-12-12 | V2. 02 | 1、完善 operation manual 章节  | 黄少彬 |
| 2013-01-17 | V2. 03 | 1、增加 PHY 描述<br>2、增加 APHY 寄存器描述<br>3、增加 DPHY 寄存器描述<br>4、增加 PHY BUDEG 寄存器描述   | 黄少彬 |

### 11.1.1 Features

- Complies with On-The-Go Supplement to the USB2.0 Specification Revision 1.0a.

- UTMI+ level2 Transceiver Macrocell Interface.
- Supports point-to-point communication with full-speed/high-speed device in Host mode(no HUB support).
- Supports full-speed or high-speed in peripheral mode.
- Supports 2 IN endpoint and 1 OUT endpoint except endpoint0.
- Supports bulk Isochronous and Interrupt transfer.
- Partially configurable endpoint endpoint type and single, double triple or quad buffering.
- Integrated synchronous RAM as endpoint FIFOs.
- Supports suspend, resume and power managements function.
- Support remote wakeup.
- Support Udisk mode high speed DMA panel

## 11.1.2 Function Description

The AOTG can be used as a dual-role device and can act as a USB host or a USB peripheral device. The soft id controls the default role. If the soft id=1, it means that the mini-B plug was connected and the AOTG becomes a B-device. When the soft id=0 it means that a mini-A plug was connected and the AOTG becomes an A-device. Figure 2-1 shows the A-device FSM and Figure 2-4 shows the B-device FSM. For details, see On-The-Go Supplement to the USB2.0 Specification Rev. 1.0a.

### 11.1.2.1 OTG state machine

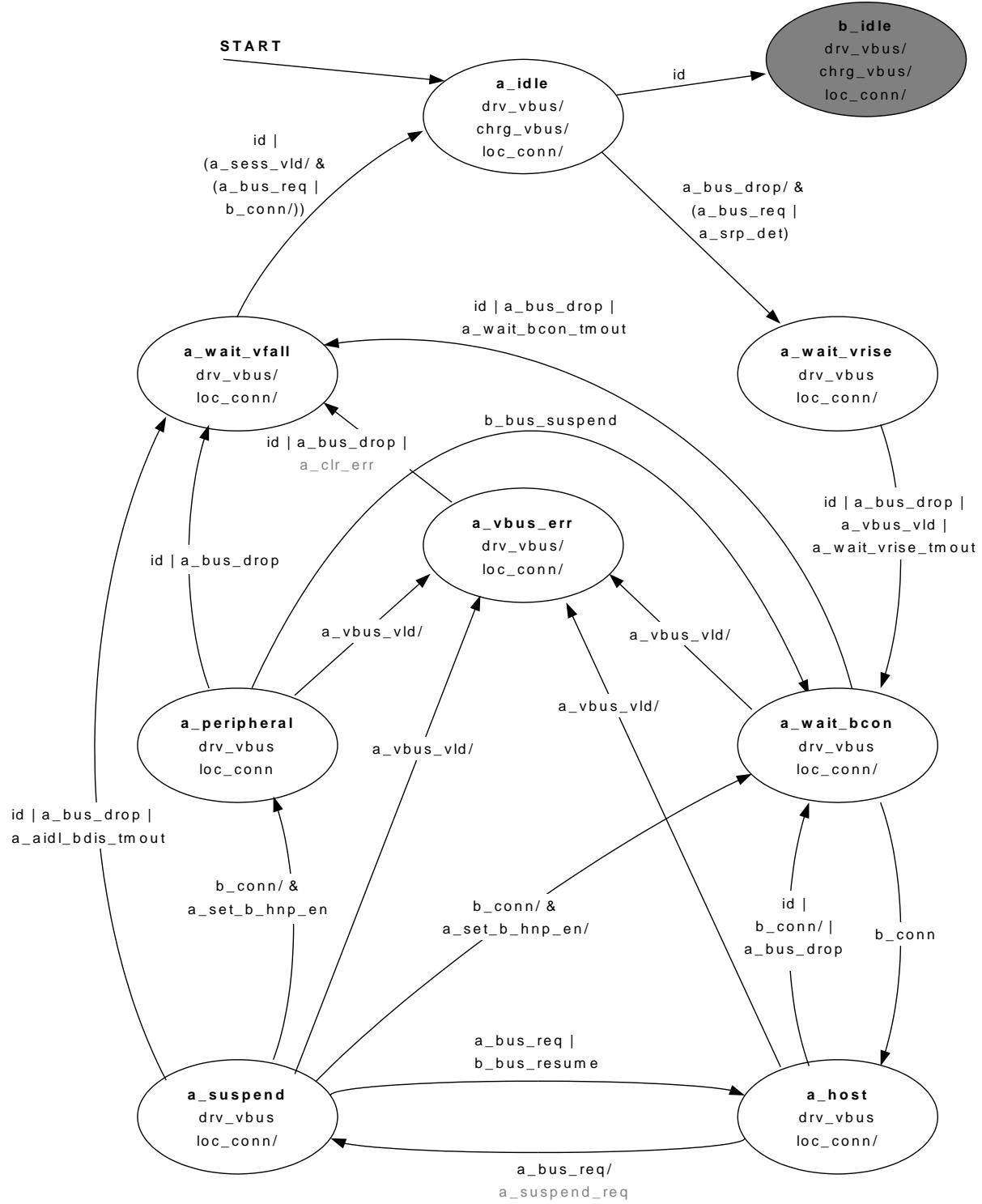


Figure 11-1 Dual Role A-Device FSM

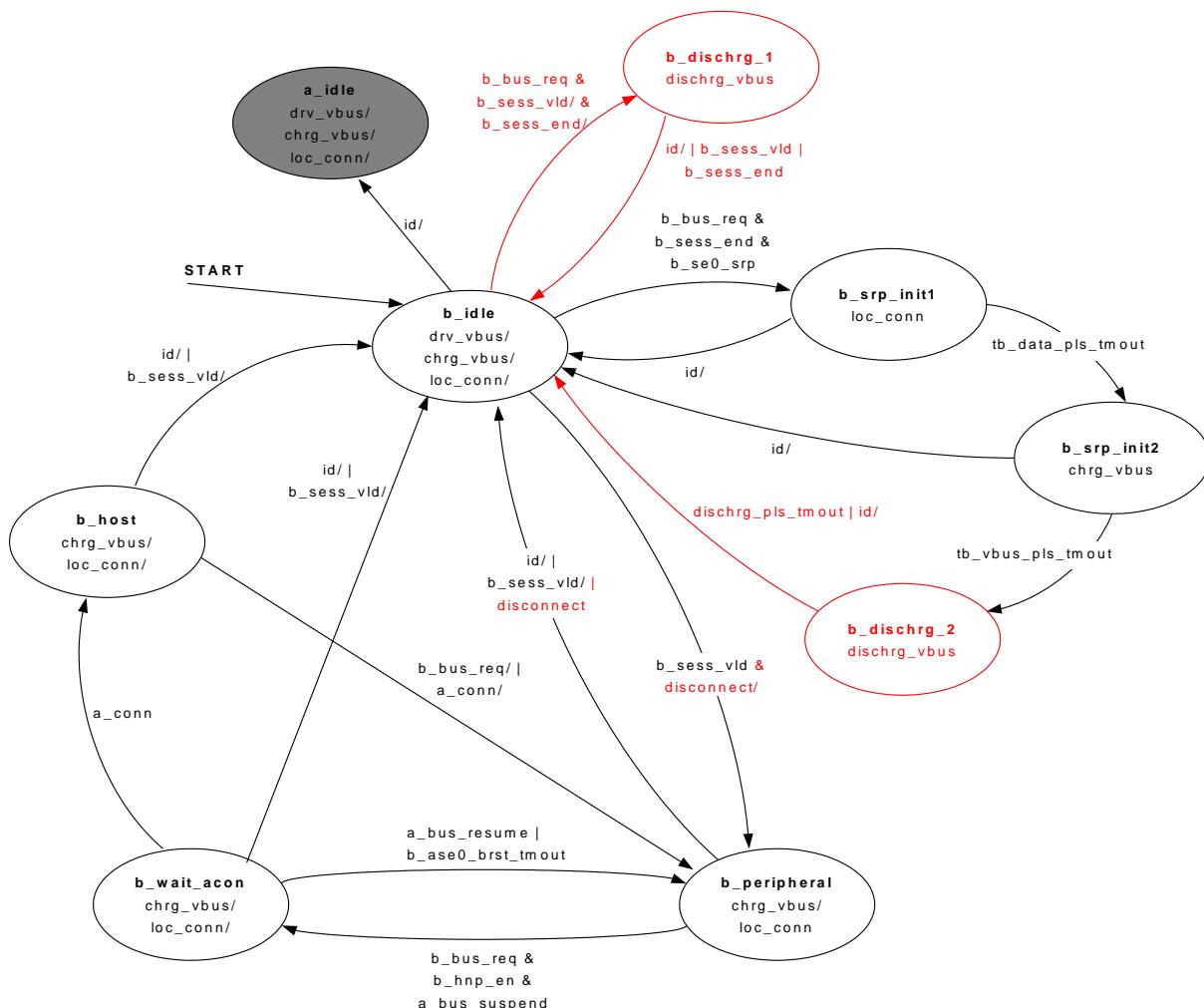


Figure 11-2 Dual Role B-Device FSM

Using a set of the Special Function Registers, microprocessor can control and monitor the A-device and the B-device FSMs.

There are few differences between AOTG FSM and FSM from the OTG Supplement. Marked in gray, are not implemented because they concern software behavior and are not necessary in the hardware. State and transitions, marked in red, are additional to the OTG Supplement.

Table 2-1 shows the SFRs that are provided for the microprocessor to control the OTG FSMs.

| SFR Name  | Bit Name   | OTG FSM Function  |
|-----------|------------|---|
| otgctrl.0 | busreq     | A_bus_req (for A-device FSM) or<br>b_bus_req (for B-device FSM) |
| otgctrl.1 | abusdrop   | A_bus_drop (for A-device FSM)                                   |
| otgctrl.2 | asetbhnpen | A_set_b_hnp_en (for A-device FSM)                               |
| otgctrl.3 | bhnpen     | B_hnp_en (for B-device FSM)                                     |
| usbcs.6   | discon     | Disconnect (for B-device FSM)                                   |

Table 2-1 OTG FSM Control Signals

OTG specification defines several timer. Table 2-2 shows AOTG FSM Programmable timer, other timer in the OTG FSM are fixed and they can not be programmed by the microprocessor.

| SFR Name     | OTG FSM Function  |
|--------------|-------------------|
| taaidlbdis   | a_aidl_bdis_tmout |
| tawaitbcon   | a_wait_bcon_tmout |
| tbvbuspls    | b_vbus_pls_tmout  |
| tbvbusdispls | dischrg_pls_tmout |

Table 2-2 OTG FSM Programmable Timing Constants

### 11.1.2.2 Endpoint and memory configuration

There are 5 endpoint totally. Endpoint0 is fixed to 64\*2 bytes, and it's mainly to implement control transfer. Endpoint1/2 are used for USB bulk, interrupt or isochronous transfer, endpoint 3 is used for bulk and interrupt only.

There are three ram as USB buffer for all endpoints. First one is URAM, which is 512 bytes. The second one is FIR\_RDS\_RAM, which is 192\*2 bytes. The third one is PCMRAM, which is a 4KB RAM. Endpoint1/2 can be configured as single, double, triple or quad buffered. And also can be configured to URAM or FIR\_RDS\_RAM or PCMRAM. Data transmission during multiple buffering is the same as for single buffering. The microprocessor loads and arms consecutive sub-buffers of the endpoint x. Advantage of the multiple buffering is that data can be loaded/unload to the one sub-buffer while the controller sends/receive from another sub-buffer. Multiple buffering allows for smoother data transmission.

Endpoint 0(in/out) and endpoint 3 have a fixed start address and buffer size in URAM0. EP0IN buffer (0x8d80~0x8dbf), EP0OUT buffer (0x8dc0~0x8dff), EP3IN buffer (0x8e00~0x8e3f).

When the AOTG is in peripheral mode for mass storage or in host mode for data transmission to SDC, the endpoint 1/2 FIFO can be configured to URAM or FIR\_RDS\_RAM or PCMRAM. You should set the EP1ADDR and the EP2ADDR to configure the start address of the EP1/2, such as Figure 2-3

When the AOTG is in peripheral mode for audio, the endpoint 1/2 FIFO can only be configured to URAM or FIR\_RDS\_RAM and use 320+192\*2 bytes memory. You should set the EP1ADDR and the EP2ADDR to configure the start address of the EP1/2, such as Figure 2-4

When the AOTG is in host mode for data transmission for audio, the endpoint 1/2 FIFO can only be configured to URAM and use the 320 byte memory. You should set the EP1ADDR and the EP2ADDR to configure the start address of the EP1/2, such as Figure 2-5

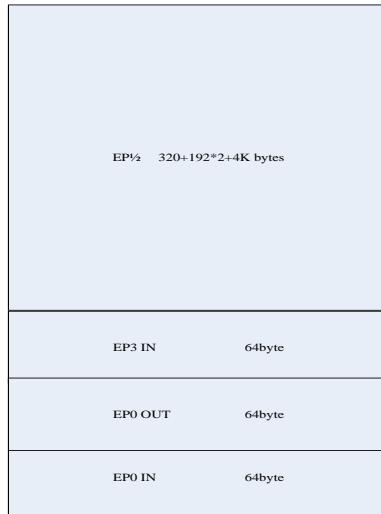


Figure 11-3 Example of endpoint buffer configuration in peripheral mode for mass storage

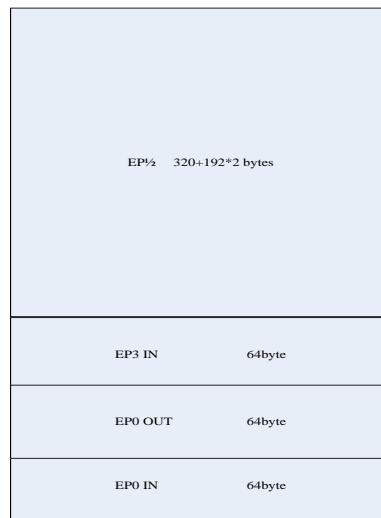


Figure 11-4 Example of endpoint buffer configuration in peripheral mode for audio

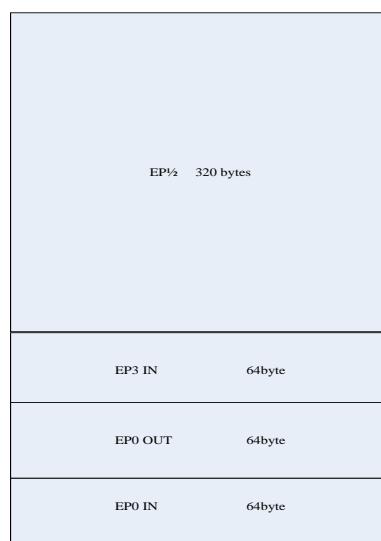


Figure 11-5 Example of endpoint buffer configuration in host mode for audio

Busy (sub-busy) = '1' – buffer (sub-buffer) means the bit is busy and the microprocessor can not access (write or read) this buffer.

Busy (sub-busy) = '0' – buffer (sub-buffer) means the bit can be accessed by the microprocessor.

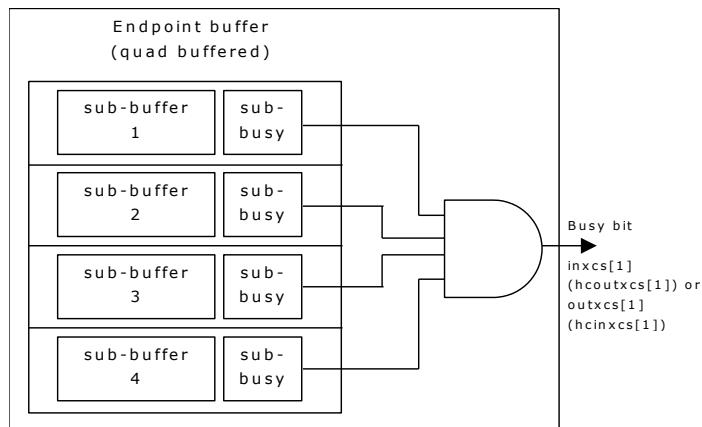


Figure 11-6 Multiple Buffered Endpoint (IN or OUT)

For OUT(HCIN) endpoint, busy=1 indicates that the buffer is empty (all sub-buffers are empty). The microprocessor should wait until the data are received by the OUT (HC IN) endpoint. The busy=1 to busy=0 transaction generates the outxirq (hcinxirq) interrupt request. The busy=0 indicates that the OUT endpoint buffer is not empty (one or more sub-buffers are full). After reading data from the OUT (HC IN) buffer (sub-buffer), if autoout bit in outxcs(hcinxcs) register is not set, the microprocessor should reload the outxcs (hcinxcs) register with any value to arm (set sub-busy bit) the sub-buffer for the next transaction. When the microprocessor arms all sub-buffers, the sets the busy bit to 1.

For IN(HCOUT) endpoints, the busy=1 indicates that the buffer is full (all sub-buffers are full). The microprocessor should wait until data from the IN (HC OUT) endpoint are sent. The busy=1 to busy=0 transaction generates the inxirq (hcouthxirq) interrupt request. The busy=0 indicates that the IN endpoint buffer is not full (one or more sub-buffers are empty). After writing data to the IN (HC OUT) buffer (sub-buffer), if autoin bit in inxcs(hcouthxcs) register is not set, the microprocessor should reload the inxcs (hcouthxcs) register with any value to arm (set sub-busy bit) the ff-buffer for next transaction. When the microprocessor arms all sub-buffers, the sets the busy bit to 1.

The npak bits (see inxcs (hcouthxcs), outxcs (hcinxcs) registers description) provide information about the endpoint buffer state. The meaning of the npak bits is different for the IN (HC OUT) and OUT (HC IN) endpoints.

For OUT endpoints, NPAK indicates the number of packets received and ready to be read by the microprocessor. See table 2-3.

| Buffering                                     | Sub-Busy Bits | Number of Filled Sub-Buffers | outxcs[1] (hcinxcs [1]) | Outxcs[3..2] (hcinxcs [3..2])<br>npak1,npak0 |
|---|---------------|------------------------------|-------------------------|--|
| 2064、single,<br>2-double<br>3-triple, 4- quad |               |                              | busy bit                |  |

|    |   |   |                                 |         |
|----|---|---|---------------------------------|---------|
| 00 | 1 | 0 | All sub-busy bits are set       | 1,2,3,4 |
| 00 | 0 | 1 | One sub-busy bit is not set     | 1,2,3,4 |
| 01 | 0 | 2 | Two sub-busy bits are not set   | 2,3,4   |
| 10 | 0 | 3 | Three sub-busy bits are not set | 3,4     |
| 11 | 0 | 4 | Four sub-busy bits are not set  | 4       |

Table 2-3 npak Bits of OUT (HC IN) Endpoint

For IN (HC OUT) Endpoints, NPAK indicates the number of IN (HC OUT) sub-buffers that can be loaded by the microprocessor and armed for the USB transfer. See table 2-4

| inxcs[3..2]<br>(hcoutxcs[3..2])<br>npak1,npak0 | inxcs[1]<br>(hcoutxcs[1])<br>busy bit | Number of<br>empty<br>sub-buffers | Sub-busy bits                   | Buffering<br>2065、single,<br>2-double<br>3-triple, 4- quad |
|--|---------------------------------------|-----------------------------------|---------------------------------|--|
| 00   | 1                                     | 0                                 | All sub-busy bits are set       | 1,2,3,4  |
| 00   | 0                                     | 1                                 | One sub-busy bit is not set     | 1,2,3,4  |
| 01   | 0                                     | 2                                 | Two sub-busy bits are not set   | 2,3,4  |
| 10   | 0                                     | 3                                 | Three sub-busy bits are not set | 3,4  |
| 11   | 0                                     | 4                                 | Four sub-busy bits are not set  | 4  |

Table 2-4 npak bits of IN (HC OUT) Endpoint

### 11.1.2.3 Device Controller

#### 11.1.2.3.1 Control endpoint0

Each USB device is allocated by endpoint numbers. Endpoint 0 (EP0) is reserved for control transfers. The host uses EP0 for reading device descriptors. The host configures the device according to the information obtained by sending commands via EP0.

The microprocessor uses endpoint 0 to support all standard and class-specific USB requests. The SET\_ADDRESS and SET\_FEATURE-Test Mode requests are the only USB requests that are supported by the endpoint 0 hardware and the microprocessor can ignore those requests.

To operate control endpoint0, relative SFR registers are list in table 2-5. For detailed information, please see SFR section.

| SFR Name      | Bit Name     | Bit Description                    |
|---------------|--------------|------------------------------------|
| usbien(0)     | sudavie      | Setup data valid interrupt enable  |
| usbien(2)     | sutokie      | Setup token interrupt enable       |
| usbirq(0)     | sudavir      | Setup data valid interrupt request |
| usbirq(2)     | sutokir      | Setup token interrupt request      |
| Setupdat      | setupdat7-0  | Setup data                         |
| EP03TOKIRQ(0) | Ep0intokirq  | IN 0 endpoint interrupt request    |
| EP03TOKIRQ(1) | Ep0outtokirq | OUT 0 endpoint interrupt request   |
| EP03TOKIEN(0) | Ep0intokien  | IN 0 endpoint interrupt enable     |
| EP03TOKIEN(1) | Ep0outtokien | OUT 0 endpoint interrupt enable    |
| ep0cs(0)      | ep0stall     | Endpoint 0 STALL bit               |
| ep0cs(1)      | hsnak        | Handshake NAK                      |
| ep0cs(2)      | inbsy        | IN 0 buffer busy flag              |
| ep0cs(3)      | outbsy       | OUT 0 buffer busy flag             |
| in0bc         | Register     | IN 0 byte counter                  |
| out0bc        | Register     | OUT 0 byte counter                 |

Table 2-5 Endpoint 0 Special Functions Registers

A control transfer consists of two or three stages: Setup stage, Data stage (optional), and Status stage.

During control write transfer, Host will issue an SETUP packet to initiate this transfer, After receiving the SETUP token the AOTG hardware sets the hsnak and sutokir bits. If an 8-byte data packet is received correctly, the AOTG sets the sudavir bit. Setting the sutokir and (or) sudavir bits generates the appropriate interrupts request.

The data stage consists of one or more OUT bulk-like transactions. The microprocessor should reload the out0bc register with any value (setting the busy bit) to arm the OUT 0 endpoint for the first OUT transaction during the data stage. The AOTG generates an OUT 0 interrupt request by setting the out0ir bit after each correct OUT transaction during the data stage.

The out0bc register contains the number of data bytes received in the last OUT transaction. The microprocessor should service the interrupt request and then prepare the endpoint for the

next transaction by reloading the out0bc register with any value (setting the busy bit).

The status stage of a control transfer is the last operation in the sequence. The microprocessor should clear the hsnak bit (by writing a 1 to it) to instruct the AOTG controller to ACK the status stage. The AOTG sends STALL handshake when both hsnak and stall bits are set.

Control Read Transfer is similar to Control Write Transfer. The difference is in the data stage. During the data stage of control read transfers, the AOTG generates an IN 0 interrupt request by setting the in0ir bit. This is done after each acknowledge received from the host. The microprocessor should load new data into the IN 0 buffer and then reload the in0bc register with a valid number of loaded data. Reloading the in0bc register sets the busy bit and arms the endpoint for the next IN transaction. The status stage of a control transfer is the last operation in the sequence. The microprocessor should clear the hsnak bit (by writing a 1 to it) to instruct the AOTG to ACK the status stage. The AOTG sends STALL handshake when both hsnak and stall bits are set.

Some control transfers do not have a data stage. In this case, the status stage consists of the IN data packet. The microprocessor should clear the hsnak bit (by writing a 1 to it) to instruct the AOTG to ACK the status stage.

### 11.1.2.3.2 Non-Control endpoints

Each USB transaction is formed as a token packet, optional data packet and optional handshake packet. Data transfer consists of two or three phases: token packet, data packet and optional handshake packet. Only control, bulk and interrupt transfers have their own handshake phase. Isochronous (ISO) transfers do not contain a handshake phase. Data is transferred during the data packet phase. Four PID types are available for this: DATA0, DATA1, DATA2 (ISO only) and MDATA (ISO only). Table 2-6 and table 2-7 is the IN endpoints and OUT endpoints SFR.

| SFR Name       | Bit Name    | Bit Description                       |
|----------------|-------------|---------------------------------------|
| Inxcs          | register    | IN x endpoint control/status register |
| Inxcon         | register    | IN x endpoint configuration register  |
| inxbch, inxbcl | register    | IN x endpoint byte counter            |
| EP03TOKIEN(3)  | ep1intokien | IN 1 endpoint interrupt enable bit    |
| EP03TOKIEN(6)  | ep3intokien | IN 3 endpoint interrupt enable bit    |
| EP03TOKIRQ(3)  | ep1intokirq | IN 1 endpoint interrupt request bit   |
| EP03TOKIRQ(6)  | ep3intokirq | IN 3 endpoint interrupt request bit   |
| FIFO1DAT       | Register    | Ep1 data port                         |
| FIFO3DAT       | Register    | EP3 data port                         |

Table 2-6 IN Endpoints Special Function Registers

| SFR Name      | Bit Name      | Bit Description                        |
|---------------|---------------|--|
| Out2cs        | register      | OUT 2 endpoint control/status register |
| EP03TOKIEN(4) | ep2outtokien  | endpoint 2 out interrupt enable bit    |
| EP03TOKIEN(5) | ep2pingtokien | endpoint 2 ping interrupt enable bit   |
| EP03TOKIRQ(4) | ep2outtokirq  | endpoint 2 out interrupt request bit   |
| EP03TOKIRQ(5) | ep2pingtokirq | endpoint 2 ping interrupt request bit  |
| FIFO2DAT      | Register      | Ep2 data port                          |

Table 2-7 OUT Endpoints Special Function Registers

Every endpoint should be initialized before USB transfer.

To enable the IN (HC OUT) and OUT (HCIN) endpoints for normal operation, the microprocessor should set the appropriate configuration bits in the inxcon (hcouthcon) and outxcon (hcinxcon) Special Function Registers.

The microprocessor can individually enable/disable each endpoint by writes the inxcon[7]-“val” or outxcon[7]-“val” (hcinxcon[7]) bits.

Each endpoint can be programmed as BULK, INTERRUPT or ISO type. To select appropriate endpoint type microprocessor writes inxcon[3..2]-“type1,type0” (hcouthcon[3..2]) and outxcon[3..2]-“type1,type0” (hcinxcon[3..2]) bits.

The buffering can be programmed for each endpoint. Each endpoint can be programmed as single, double, triple or quad buffered. To select appropriate buffering the microprocessor writes inxcon[1..0] – “buf1,buf0” (hcouthcon[1..0]) and outxcon[1..0] – “buf1,buf0” (hcinxcon[1..0]) bits. The In endpoint 3 is configurable as bulk and interrupt mode only, and no support multi-buffering

After the microprocessor initializes the endpoint, it should reset the endpoint by following those steps:

Select the endpoint, by writing the value 000xxxee to the endprst register.

The value -01xxxee has the following format:

ee is the selected endpoint number, a binary value from 00 to 11

Reset the endpoint by writing the value, 011xxxee to the endprst register

Note: Setting ee=0000 selects all the endpoints from 01 to 11.

During bulk or interrupt data transfer, for IN endpoint, when Host send an IN token, and if the busy bit (inxcs[1]) is set, the AOTG will respond by returning a data packet. If the host receives a valid data packet, it responds with an ACK handshake. After having received a valid ACK handshake from the host, the AOTG sets the inxirq bit and clears the busy bit. Setting the inxirq bit generates an interrupt request for IN x endpoint (x = appropriate number of endpoint). The microprocessor services the interrupt request and load new data into the fifoxdat buffer – the endpoint logic keeps track of number of loaded bytes – and then the microprocessor reloads the inxcs register with any value to set the busy bit. IN x endpoint is armed for the next transfer when

---

the busy bit is set. When the busy bit is not set, the AOTG returns NAK handshake for each IN token from the host. When the stall bit is set, the AOTG returns STALL handshake.

During bulk or interrupt data transfer, for OUT endpoint, When the host wants to transmit bulk data, it issues an OUT Token packet followed by a data packet. When the AOTG receives an error free OUT data packets and the busy (outxcs[1]) bit is set, it will return an ACK handshake to the host and set the outxirq bit. Setting the outxirq bit generates an interrupt request for the OUT x endpoint (x = appropriate number of endpoint). The microprocessor should service the OUT x interrupt request. The received data packet is available in the fifoxdat buffer. After servicing an interrupt request, the microprocessor should reload the outxcs register with any value to set the busy bit. The OUT x endpoint is armed for the next OUT transfer when the busy bit is set. When the AOTG receives an error free OUT data packets but the busy bit is not set, it will return a NAK handshake to the host. When the AOTG receives an error free OUT and data packets and the stall bit (outxcon[6]) is set it will return a STALL handshake to the host. If any transmission error occurs during an OUT token or data phase, the AOTG will not return a handshake.

Note that in HS mode, additional PING and NYET protocol is supported.

During ISO transfer, the ISO IN endpoint does not support the data toggle synchronization and handshake phase. For ISO IN transfer, the microprocessor should load new data into the ISO IN endpoint buffer(s) every start-of-frame interrupt request. The ISO IN endpoint is accessed via the fifoxdata register. The AOTG keeps track of the number of bytes that the microprocessor writes into the fifoxdat register. After loading the IN endpoint, the microprocessor should reload the inxcs register with any value to arm the endpoint for the next transaction. When the host wants to receive ISO data, it issues an IN token for the specific endpoint. If the specific endpoint selected by the host IN endpoint buffer contains data, the AOTG will respond by returning a data packet. When the IN buffer is empty (not armed), the AOTG will send a 0-length data packet.

During ISO OUT transfer, With every start-of-frame interrupt request, the microprocessor reads data that was sent by the host in the previous frame. The outxbch and the outxbcl registers contain the number of received bytes. The data are accessible via the fifoxdata register. After reading the fifoxdat buffer, the microprocessor should reload the outxcs register with any value to arm endpoint for the next transfer.

High-Speed mode allows ISO packets up to 1024 bytes long. Full-Speed mode allows ISO packets up to 1023 bytes long. The microprocessor should keep track of the mode and should not send packets that are too long.

High-Speed ISO IN data uses DATA0, DATA1, DATA2 and MDATA PIDs.

In Full-Speed mode only one ISO IN packet can be transferred per endpoint, per frame. In High-Speed mode, up to three ISO IN packets can be transferred per endpoint, per microframe. Microprocessor writes inxcon[5..4] – “isod1,isod0” bits to select the number of ISO IN packets per microframe.

When the host does not read the programmed number of ISO IN packets or when the firmware does not arm the appropriate number of ISO IN packets during the single microframe, then error is reported by the isoerr bit (inxcs[0]).

### 11.1.2.3.3 Data Packet Synchronization

Data packet synchronization for bulk and interrupt endpoints is achieved by using the data sequence toggle bits and the DATA0/DATA1 PIDs. The microprocessor automatically toggles DATA0/DATA1 PIDs every bulk or interrupt transfer. The microprocessor can directly clear data toggle bits using the endprst register. The microprocessor should clear the toggle bits when the host issues Clear\_Feature, Set\_Interface or selects alternate settings.

To clear a toggle bit, the microprocessor should execute the following sequence:

Select the endpoint, by writing the value 000xxxee to the endprst register.

The value -01xxxee has the following format:

ee is the selected endpoint number, a binary value from 01 to 11

Clear the toggle bit by writing the value, 001xxxee to the endprst register

### 11.1.2.4 Host Controller

The Host Controller (HC) is a part of the AOTG when it works as a USB Host. The main tasks of HC are:

Generation of suspend/resume signaling and USB reset

Generation of SOF tokens

USB data transactions

### 11.1.2.4.1 Frame Generator and Transaction Scheduler

When the HC is in USB Operational state, it sends SOF tokens. SOF tokens are generated every 1 ms in the Full-Speed or every 1/8 ms in the High-Speed. In the Low-Speed mode Keep-Alive is sent every 1 ms. Current frame number is stored in the hcfrmnrh and hcfrmnr1 register. Additionally, the microprocessor can monitor the time remaining to the end of the current frame using the hcfrmremainh&hcfrmremainl register. The hcfrmremainh&hcfrmremainl register contains information about how many bytes can be transmitted or received before the next SOF token.

The Host Transaction Scheduler (HTS) is a part of the AOTG Host Controller. It analyzes how many endpoints wait for service and decides which endpoints will be serviced in the current (m)frame and which ones will be serviced in the next (m)frame. The HC initializes a new USB transaction if it can be completed before start of the next (m)frame. To determine if a transaction can be completed before the end of the current frame, the HTS compares the size of the data packet and the time remaining to the end of frame.

### 11.1.2.4.2 Control Transfer in the Host Mode

The endpoint 0 in the Host mode supports only CONTROL transfers. To initialize endpoint 0 the microprocessor should write the hcin0maxpck register. Endpoint 0 is always single buffered. When the AOTG enters the Host mode, endpoint 0 HC IN and 0 HC OUT are automatically unarmed by the HC hardware (busy bits are set to 0).

The microprocessor must control all phases of the CONTROL transfer. Below is a detailed description of the CONTROL transfer using endpoint 0.

Control transfer starts when the USB Host sends a SETUP token. To initialize the SETUP stage of the CONTROL transfer, the microprocessor should write a '1' to the hcset bit (ep0cs[4]). Writing a '1' to the hcset bit causes endpoint 0 to be initialized to start the CONTROL transfer:

Toggle bit of the endpoint 0 HC IN is set

Toggle bit of the endpoint 0 HC OUT is cleared

Busy bits of the endpoint 0 HC IN and 0 HC OUT are cleared

HC will send SETUP token instead OUT token during nearest transaction of endpoint 0

After writing the hcset bit the microprocessor should write 8 bytes of setup data to the hcep0outdat buffer. To arm the endpoint 0 HC OUT buffer and send the SETUP token, the microprocessor should write a 08H value to the hcout0bc register.

The next stage of the control transfer is the DATA stage. The microprocessor can send or receive data packets during the DATA stage using endpoint 0 HC IN (hcep0indat data buffer) or endpoint 0 HC OUT (hcep0outdat data buffer). The hcset bit does not need to be cleared before the DATA stage but during the DATA stage the microprocessor should not set this bit.

The last stage of the control transfer is the STATUS stage. The STATUS stage starts with DATA1 packet identifier. Therefore the microprocessor should set the toggle bit for the endpoint 0. To set the toggle bit, the microprocessor should write a '1' to the hcsettoggle bit (ep0cs[6]). Next, the microprocessor should do one of the following:

Write any value to the hcin0bc register to arm endpoint 0 HC IN (for control write transfers and no-data control transfers) or

Write a 00H value to the hcout0bc register to arm endpoint 0 HC OUT (for control read transfers).

The STATUS stage is successfully completed when the microprocessor receives hcin0irq (for control write transfers and no-data control transfers) or hcout0irq (for control read transfers).

### 11.1.2.4.3 Data Transfer

In Host mode, endpoints initialize and data transfer is the similar to peripheral mode, but should note that error deal with:

If transaction completed without errors, after the USB transaction is successfully completed,

the HC generates the hcouthirq/hcinxirq interrupt request and the microprocessor can unload/load next data to the fifoxdat buffer for the next USB transaction.

If transactions completed with errors, the HC automatically repeats this transaction. When three consecutive transactions for the endpoint x are completed with USB errors, the HC disables the endpoint x and generates a usbintreq interrupt request (hcin(out)xerrirq) for the microprocessor. For more details about USB error types see the description of the hcoutherr and hcinxerr registers. The microprocessor should read the hcoutherr register and determine the type of the USB error. The microprocessor can halt the endpoint (endpoint halted) or resume transmission (write hcoutherr[5] – resend bit).

### 11.1.2.5 Interrupt

The AOTG provides two interrupt request signals for microprocessor: [ewakeupirq](#) and [usbintirq](#).

#### 11.1.2.5.1 Wakeup Interrupt

Only the DP change event will make AOTG generate wake up signal to microprocessor. Then ewakeupirq interrupt will happen , if this interrupt is enabled.

#### 11.1.2.5.2 USB Interrupt

The USB interrupt request is provided via the usbintreq signal and includes:

- 5 endpoint interrupts (ep03irq),
- 5 transmission error interrupts( hcoutherr.hcinxerr),
- 6 endpoint token interrupts (ep03tokirq)
- start of frame interrupt (sofir),
- suspend interrupt (suspir),
- USB reset interrupt (uresir),
- high-speed interrupt (hspeedir),
- setup token interrupt (sutokir) and
- setup data valid interrupt (sudavir).

The AOTG will prioritize the USB interrupts if two or more occurs simultaneously. The vector of the active interrupt is available in the ivec register. Table 3-8 shows all the interrupt sources and their natural priority.

| Source of Interrupt | SFR Bit   | ivec Register |
|---------------------|-----------|---------------|
| sudav               | usbirq(0) | 00H           |

| Source of Interrupt | SFR Bit                       | ivec Register |
|---------------------|-------------------------------|---------------|
| sof                 | usbirq(1)                     | 04H           |
| sutok               | usbirq(2)                     | 08H           |
| suspend             | usbirq(3)                     | 0CH           |
| usbreset            | usbirq(4)                     | 10H           |
| hspeed              | usbirq(5)                     | 14H           |
| hcout0err           | hcep03errirq(1)               | 16H           |
| ep0in<br>(hcep0out) | ep03irq(0)<br>(hcep03irq(0))  | 18H           |
| hcin0err            | hcep03errirq(0)               | 1AH           |
| ep0out<br>(hcep0in) | ep03irq(1)<br>(hcep03irq(1))  | 1CH           |
| ep0ping             | ep03tokirq(2)                 | 20H           |
| hcout1err           | hcep03errirq(2)               | 22H           |
| ep1in<br>(hcep1out) | ep03irq(2)<br>(hcep03irq(2))  | 24H           |
| hcin1err            | hcin04errirq(1)               | 26H           |
| ep1out<br>(hcep1in) | out04irq(1)<br>(hcin04irq(1)) | 28H           |
| ep1ping             | out04pngirq(1)                | 2CH           |
| hcout2err           | hcout04errirq(2)              | 2EH           |
| ep2in<br>(hcep2out) | in04irq(2)<br>(hcou04irq(2))  | 30H           |
| hcin2err            | hcep03errirq(3)               | 32H           |
| ep2out<br>(hcep2in) | ep03irq(3)<br>(hcep03irq(3))  | 34H           |
| ep2ping             | ep03tokirq(6)                 | 38H           |
| hcout3err           | hcep03errirq(4)               | 3AH           |
| ep3in<br>(hcep3out) | ep03irq(4)<br>(hcep03irq(4))  | 3CH           |
| hcin3err            | hcin04errirq(3)               | 3EH           |
| ep3out<br>(hcep3in) | out04irq(3)<br>(hcin04irq(3)) | 40H           |
| ep3ping             | out04pngirq(4)                | 44H           |
| hcout4err           | hcout04errirq(4)              | 46H           |

| Source of Interrupt   | SFR Bit                       | ivec Register |
|-----------------------|-------------------------------|---------------|
| ep4in<br>(hccep43out) | in04irq(4)<br>(hcou04irq(4))  | 48H           |
| hcin4err              | hcin04errirq(4)               | 4AH           |
| ep4out<br>(hccep4in)  | out04irq(4)<br>(hcin04irq(4)) | 4CH           |
| ep4ping               | out04pngirq(4)                | 50H           |
| otgirq                | otgirq register               | D8H           |

Note: When the otgirq (vector = D8H) is requested, the microprocessor should read the otgirq register to determine what is the source of interrupt.

Table 2-8 USB interrupt vectors

The ext.USBirq bit must be cleared first and then the int.USBirq bit, when clearing the USB irq.

### 11.1.2.6 Speed Detection

In Host mode, The AOTG recognizes the speed of the connected peripheral device. Speed detection is performed during the AOTG sends the USB reset. The AOTG can operate with low-speed (1.5Mb/s), full-speed (12Mb/s) and high-speed (480Mb/s) peripheral devices. If HS chirp handshake is successful, a hspeedir interrupt will generate (if this interrupt is enabled), else, Host can read lsmode to know whether current peripheral is LS or HS/FS mode.

In peripheral mode, the AOTG can response chirp handshake during USB bus reset, and automatically switches to corresponding speed mode. But note AOTG can also be HS disabled by write '1' to hsdisable bit (register VDCTRL[7]).

### 11.1.2.7 USB RESET

In Host mode, the HC automatically generates the USB reset when the OTG FSM enters the a\_host or b\_host states. Additionally the microprocessor may generate USB reset by I a '1' to the portrst bit (hcportctrl[5]). When the HC sends USB reset, the USB reset interrupt request (usbirq[4]) is generated.

In peripheral mode, a host signals the USB reset by driving the D+ and D- lines low for a minimum of 10ms. After reset is removed, the AOTG will be in the default state. When the AOTG detects the USB reset, it will generate an USB reset interrupt request (usbirq[4]) .

### 11.1.2.8 USB SUSPEND

In Host mode, when the OTG FSM is in the a\_host state the microprocessor can suspend the USB bus. To force a suspend state, the microprocessor clears the busreq (otgctrl[0]) bit. After the busreq bit is cleared the OTG FSM enters the a\_suspend state and the HC stops any bus activity

(data transfers and SOF generation). When the OTG FSM is in the a\_suspend state less than 1 ms and the microprocessor sets the busreq bit, then the OTG FSM immediately returns to the a\_host state. When the OTG FSM is in the a\_suspend state for more than 1 ms and the microprocessor writes the busreq bit then the AOTG initializes resume ffuse n. When the OTG FSM enters the a\_suspend state the microprocessor can switch off the clkusb clock by I the clkgate register. After entering a\_suspend the microprocessor should wait for at least 1 ms and then it can write the clkgate register and switch off the clkusb clock.

In peripheral mode, the AOTG generates the suspend interrupt request (usbirq[3] – suspir) bit after it detects a constant Idle state on their upstream facing bus lines for 3.0 ms. When the processor detects the suspend interrupt request it writes any value to the clkgate register to switch off the clkusb clock. Note if want AOTG can wakeup microprocessor, software should set wkupdpen, wkupiden,wkupvbusen (register USBEIRQ) and/or wkupbvlden(register USBEKUP2), then set wuiden,wudpen,wuvbusen (clkgate register). If wake up event(s) happened, clkgate register will be cleared by hardware.

After the processor writes the *clkgate* register, the *utmisuspendm* signal goes low and the external UTMI+ component switches to the suspend state – at this moment the *clkusb* is switched off. The processor should switch off all power-consuming components and then switch off the *clkusb* clock to enter the power-down state.

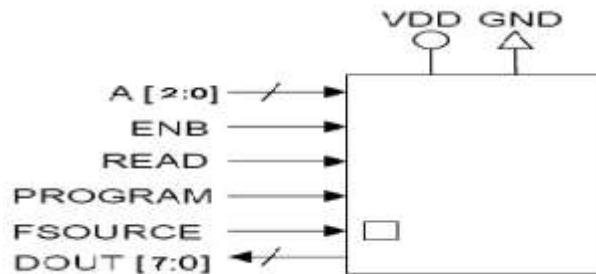
### 11.1.2.9 USB RESUME

In Host mode, if busreq (otgctrl[0]) is cleared, Host will stops all bus activity and peripheral will enter suspend. If Host software set busreq (otgctrl[0]), a resume signal will generate on USB bus to wakeup peripheral.

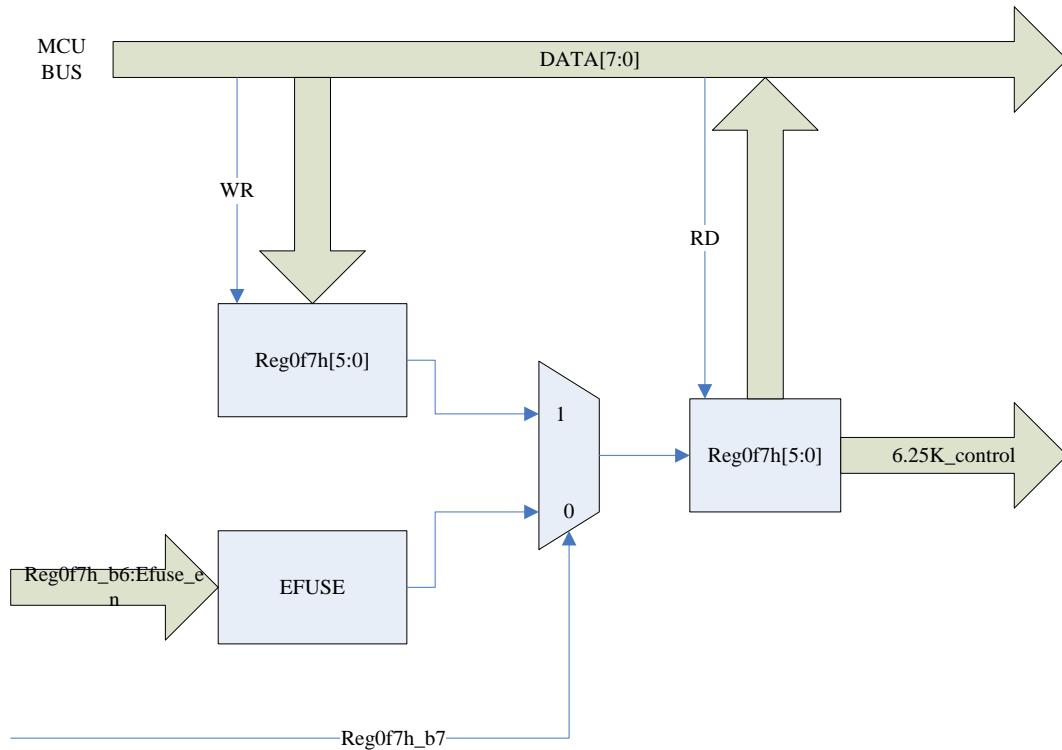
In peripheral mode, if USB bus is on suspend state, peripheral can generate remote wake up signal by set sigrsum (UsbCTRL\_STUS [5]) bit, and Host will response with resume signal.

### 11.1.2.10 Efuse

There is a build-in 8bits electrical poly-fuse macro organized in 1-8 liner array, whose standby current is 5Ua, operating current is 4Ma. Its block diagram as follow:



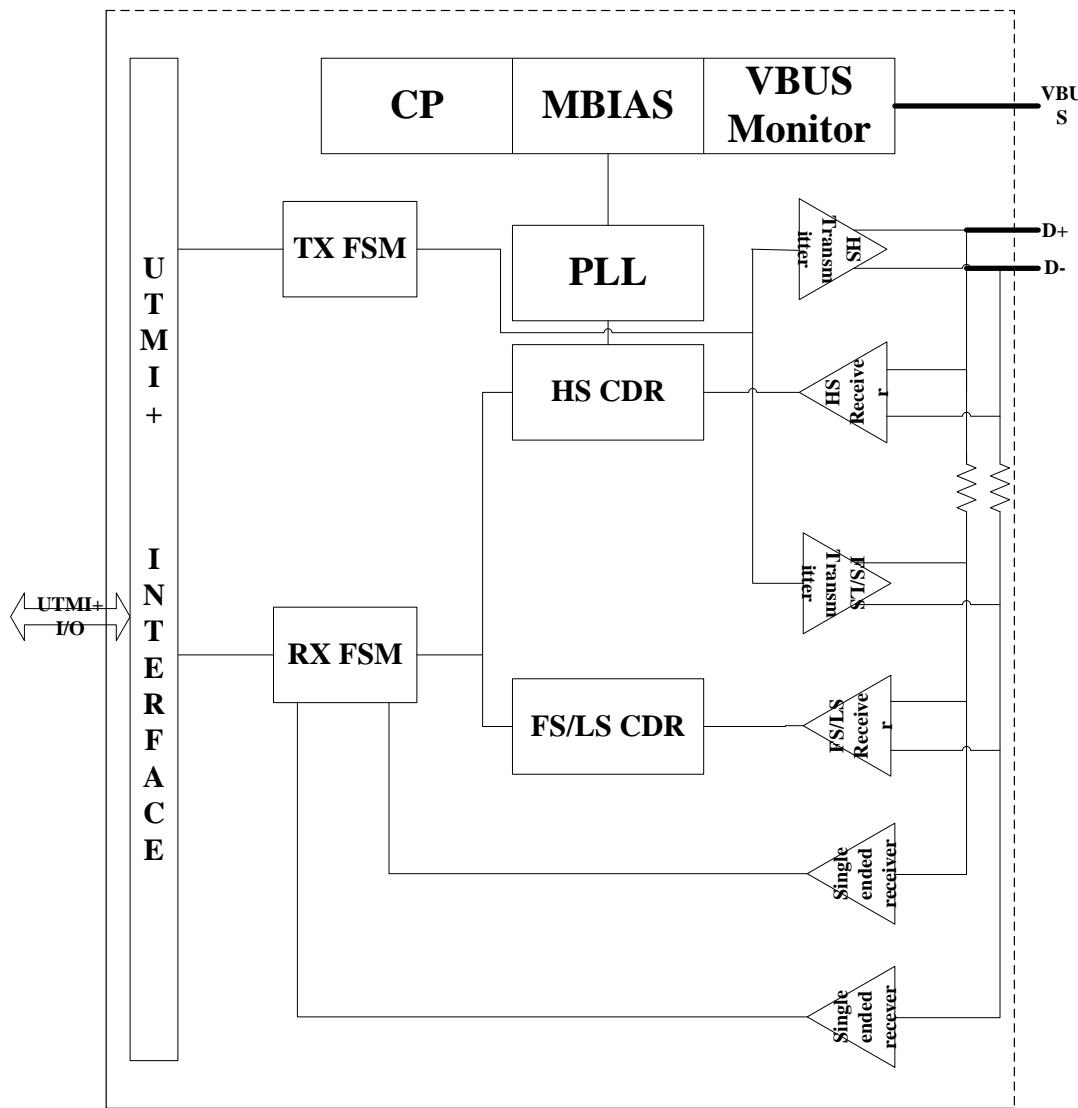
### 11.1.2.11 Efuse Control



### 11.1.2.12 PHY

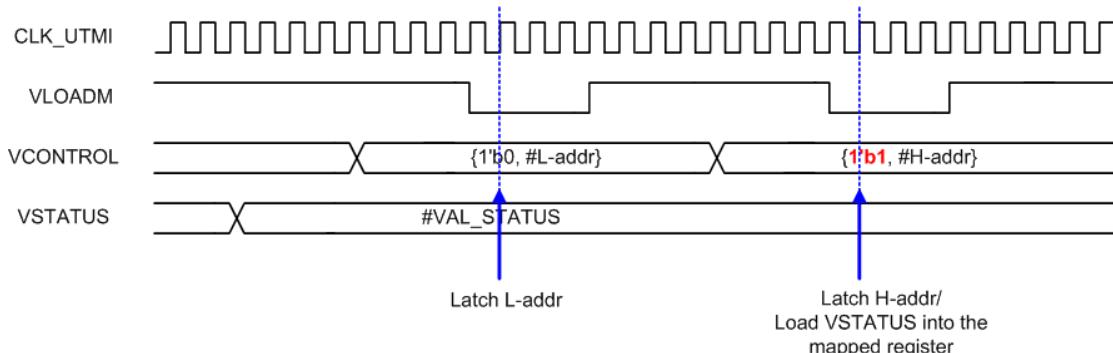
USB2.0 OTG PHY uses UTMI+ level2 Transceiver Macrocell Interface. It has several module, such as:

- HS/FS transceiver
- 24MHz input 480MHz output PLL
- HS clock data recovery circuit
- MBIAS circuit, impendence control circuit
- VBUS monitor
- 10Ma output Charge pump
- High speed digital control logic (NRZI, transmitter FSM, receiver FSM)



We use the vendor interface to configure the registers of the PHY. Also we output the signal of PHY to PAD for debugging.

We use VSTATUS[7:0] as data bus. It put data into the PHY register when VLOADM is low. And when VLOADM is high, it uses as debug interface. The address information is transmitted by VCONTROL[3:0]. Because the address of the register of PHY is 6 bits and the VCONTROL[3:0] has only 4 lines, the address of register is divided to tow part. When VLOADM is low and VCONTROL[3] is 0, VCONTROL[2:0] are the low address. When VLOADM is low and VCONTROL[3] is 1, VCONTROL[2:0] are the high address.



## 11.1.3 Module Description

### 11.1.3.1 Block Diagram

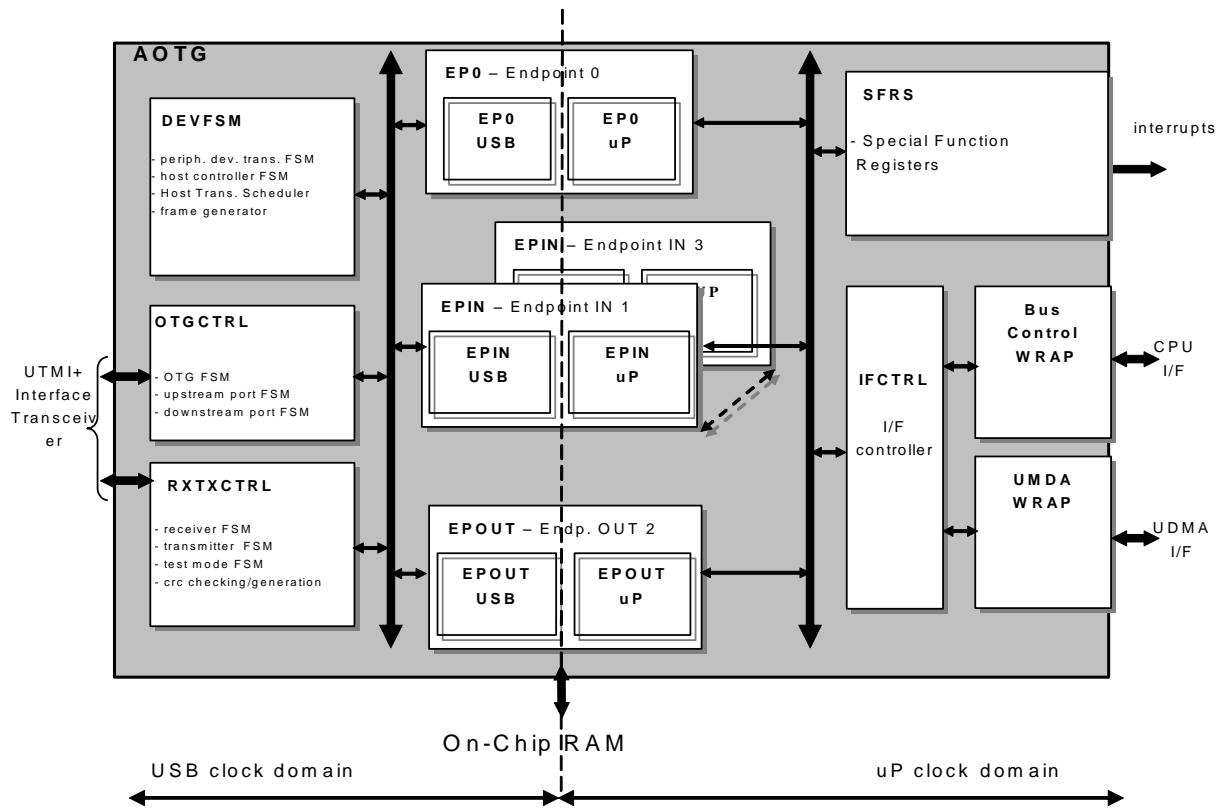


Figure 11-6 AOTG Controller block diagram

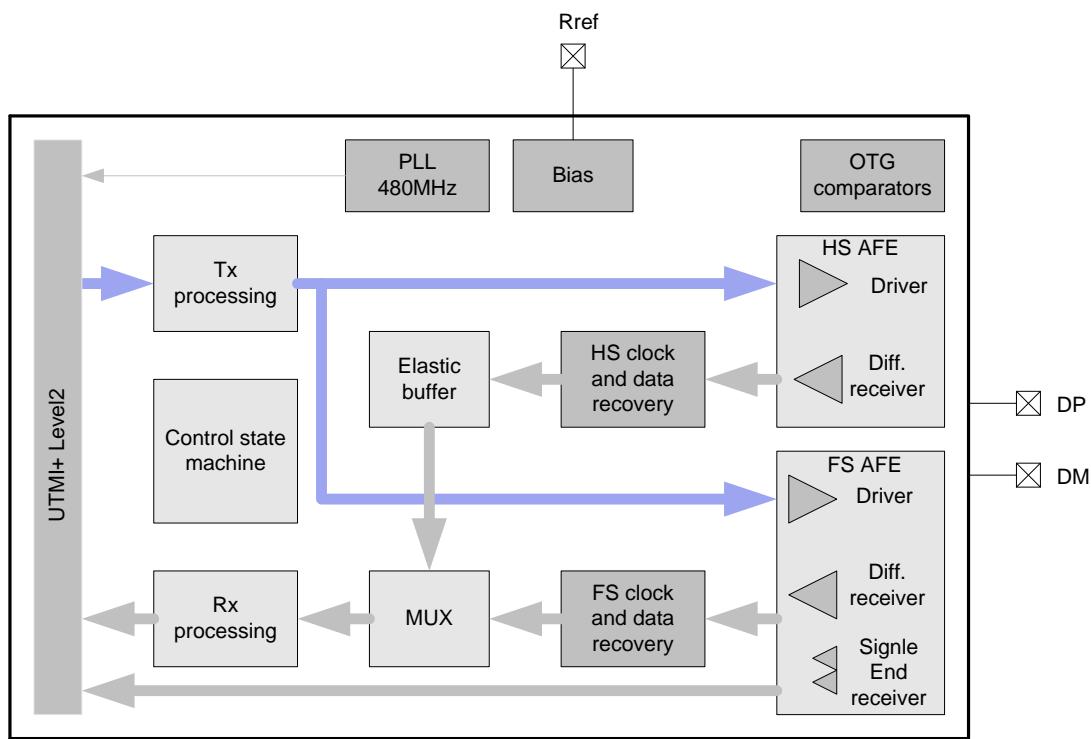


Figure 11-7 UTMI+ transceiver block diagram

### 11.1.3.2 Signal List

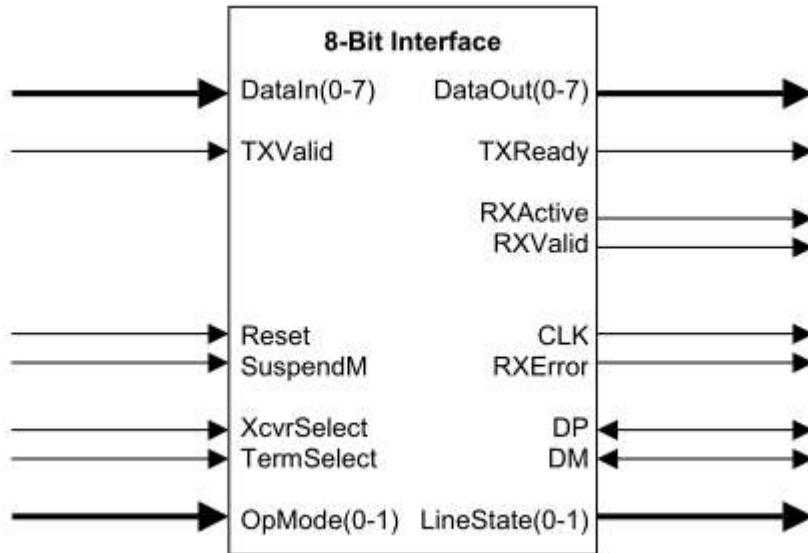


Figure 11-8 UTMI+ 8bit interface

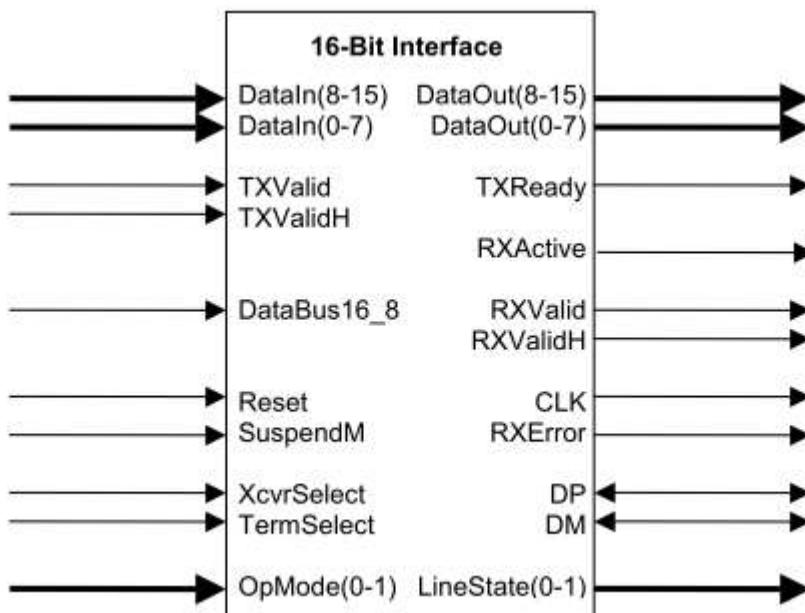


Figure 11-9 UTMI+ 16bit interface

| SIGNAL Name | I/O Type | Short Description  |
|-------------|----------|--|
| CLK         | Output   | This output is used for clocking receive and transmit parallel data.   |
| Reset       | Input    | Reset all state machines in the UTM.   |
| Xcvr Select | Input    | This signal selects between the FS and HS transceivers:<br>0: HS transceiver enabled<br>1: FS transceiver enabled  |
| Term Select | Input    | This signal selects between the FS and HS terminations:<br>0: HS termination enabled<br>1: FS termination enabled  |
| SuspendM    | Input    | Places the Macrocell in a mode that draws minimal power from supplies.<br>Shuts down all blocks not necessary for Suspend/Resume operation.<br>While suspended, TermSelect must always be in FS mode to ensure that the 1.5K pull-up on DP remains powered.<br>0: Macrocell circuitry drawing suspend current<br>1: Macrocell circuitry drawing normal current |

| LineState(0-1) | Output | These signals reflect the current state of the single ended receivers. They are combinatorial until a “usable” CLK is available then they are synchronized to CLK. They directly reflect the current state of the DP (LineState[0]) and DM (LineState[1]) signals:<br><br><table border="0"> <thead> <tr> <th>DM</th><th>DP</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0: SE0</td></tr> <tr> <td>0</td><td>1</td><td>1: ‘J’ State</td></tr> <tr> <td>1</td><td>0</td><td>2: ‘K’ State</td></tr> <tr> <td>1</td><td>1</td><td>3: SE1</td></tr> </tbody> </table> | DM  | DP  | Description | 0 | 0 | 0: SE0              | 0 | 1 | 1: ‘J’ State   | 1 | 0 | 2: ‘K’ State                              | 1 | 1 | 3: SE1      |
|----------------|--------|--|-----|-----|-------------|---|---|---------------------|---|---|----------------|---|---|---|---|---|-------------|
| DM             | DP     | Description  |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |
| 0              | 0      | 0: SE0   |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |
| 0              | 1      | 1: ‘J’ State   |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |
| 1              | 0      | 2: ‘K’ State   |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |
| 1              | 1      | 3: SE1   |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |
| OpMode(0-1)    | Input  | These signals select between various operational modes:<br><br><table border="0"> <thead> <tr> <th>[1]</th><th>[0]</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0: Normal Operation</td></tr> <tr> <td>0</td><td>1</td><td>1: Non-Driving</td></tr> <tr> <td>1</td><td>0</td><td>2: Disable Bit Stuffing and NRZI encoding</td></tr> <tr> <td>1</td><td>1</td><td>3: Reserved</td></tr> </tbody> </table>   | [1] | [0] | Description | 0 | 0 | 0: Normal Operation | 0 | 1 | 1: Non-Driving | 1 | 0 | 2: Disable Bit Stuffing and NRZI encoding | 1 | 1 | 3: Reserved |
| [1]            | [0]    | Description  |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |
| 0              | 0      | 0: Normal Operation  |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |
| 0              | 1      | 1: Non-Driving   |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |
| 1              | 0      | 2: Disable Bit Stuffing and NRZI encoding  |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |
| 1              | 1      | 3: Reserved  |     |     |             |   |   |                     |   |   |                |   |   |   |   |   |             |

Table 2-9 System Interface Signals

| SIGNAL Name | I/O Type         | Short Description  |
|-------------|------------------|--------------------|
| DP          | Input/<br>Output | USB data pin Data+ |
| DM          | Input/<br>Output | USB data pin Data- |

Table 2-10 USB Interface Signals

| SIGNAL Name   | I/O Type | Short Description  |
|---------------|----------|--|
| VcontrolLoadM | Input    | Assertion of this signal loads the Vendor Control register:<br>0: Load Vendor Control Register<br>1: NOP             |
| Vcontrol0-3   | Input    | Vendor defined 4-bit parallel input bus. Note these pins are optional, a macrocell may define partial sets as well.  |
| Vstatus0-7    | Output   | Vendor defined 8-bit parallel output bus. Note these pins are optional, a macrocell may define partial sets as well. |

Table 2-11 Vendor Control Signals

| SIGNAL Name | I/O Type | Short Description   |
|-------------|----------|---|
| DataIn0-7   | Input    | 8-bit parallel USB data input bus. When DataBus16_8 = 1 this bus transfers the low byte of 16-bit transmit data. When DataBus16_8 = 0 all transmit data is transferred over this bus. |

|            |        |   |
|------------|--------|---|
| DataIn8-15 | Input  | An 8-bit parallel USB data input bus that transfers the high byte of 16-bit transmit data. These signals are only valid when DataBus16_8 = 1.   |
| TXValid    | Input  | Indicates that the DataIn bus is valid. The assertion of Transmit Valid initiates SYNC on the USB. The negation of Transmit Valid initiates EOP on the USB. In HS (XcvrSelect = 0) mode, the SYNC pattern must be asserted on the USB between 8 and 16 bit times after the assertion of TXValid is detected by the Transmit State Machine. See section 6.4 for more information. In FS (XcvrSelect = 1), FS Only, or LS Only modes, the SYNC pattern must be asserted on the USB no less than 1 CLK and no more than 5 10 CLKs3 after the assertion of TXValid is detected by the Transmit State Machine. |
| TXValidH   | Input  | When DataBus16_8 = 1, this signal indicates that the DataIn(8-15) bus contains valid transmit data. This signal is ignored when DataBus16_8 = 0. This signal is not provided in 8-Bit transceiver implementations.  |
| TXReady    | Output | If TXValid is asserted, the SIE must always have data available for clocking in to the TX Holding Register on the rising edge of CLK. If TXValid is TRUE and TXReady is asserted at the rising edge of CLK, the UTM will load the data on the DataIn bus into the TX Holding Register on the next rising edge of CLK, at that time, SIE should immediately present the data for next transfer on the DataIn bus. If TXValid is asserted and TXReady is negated, the SIE must hold the previously asserted data on the DataIn bus. From the time TXValid is negated, TXReady is a don't care for the SIE.  |

Table 2-12 Data Interface Signals (Transmit)

| SIGNAL Name | I/O Type | Short Description   |
|-------------|----------|---|
| DataIn0-7   | Output   | 8-bit parallel USB data output bus. When DataBus16_8 = 1 this bus transfers the low byte of 16-bit receive data. When DataBus16_8 = 0 all receive data is transferred over this bus.                                    |
| DataIn8-15  | Output   | An 8-bit parallel USB data output bus that transfers the high byte of 16-bit receive data. These signals are only valid when DataBus16_8 = 1.   |
| RXValid     | Output   | Indicates that the DataOut bus has valid data. The Receive Data Holding Register is full and ready to be unloaded. The SIE is expected to latch the DataOut bus on the clock edge.                                      |
| RXValidH    | Output   | When DataBus16_8 = 1 this signals indicates that the DataOut(8-15) bus is presenting valid receive data. This signal is ignored when DataBus16_8 = 0. This signal is not provided in 8-Bit transceiver implementations. |

|          |        |  |
|----------|--------|--|
| RXActive | Output | Indicates that the receive state machine has detected SYNC and is active. RXActive is negated after a Bit Stuff Error or an EOP is detected. See the RX State Machine for more details on the negation conditions for RXActive.<br><br>In HS mode (XcvrSelect = 0), RXActive must be negated no less than 3 and no more than 8 CLKs after an Idle state is detected on the USB. And RXActive must be negated for at least 1 CLK between consecutive received packets.<br><br>In FS (XcvrSelect = 1), FS Only, or LS Only modes, RXActive must be negated no more than 2 CLKs after a FS Idle state4 is detected on the USB. And RXActive must be negated for at least 4 CLKs between consecutive received packets. |
| RXError  | Output | 2066、 Indicates no error.<br>1 Indicates that a receive error has been detected.   |

Table 2-13 Data Interface Signals (Receive)

### 11.1.3.3 Clock Description

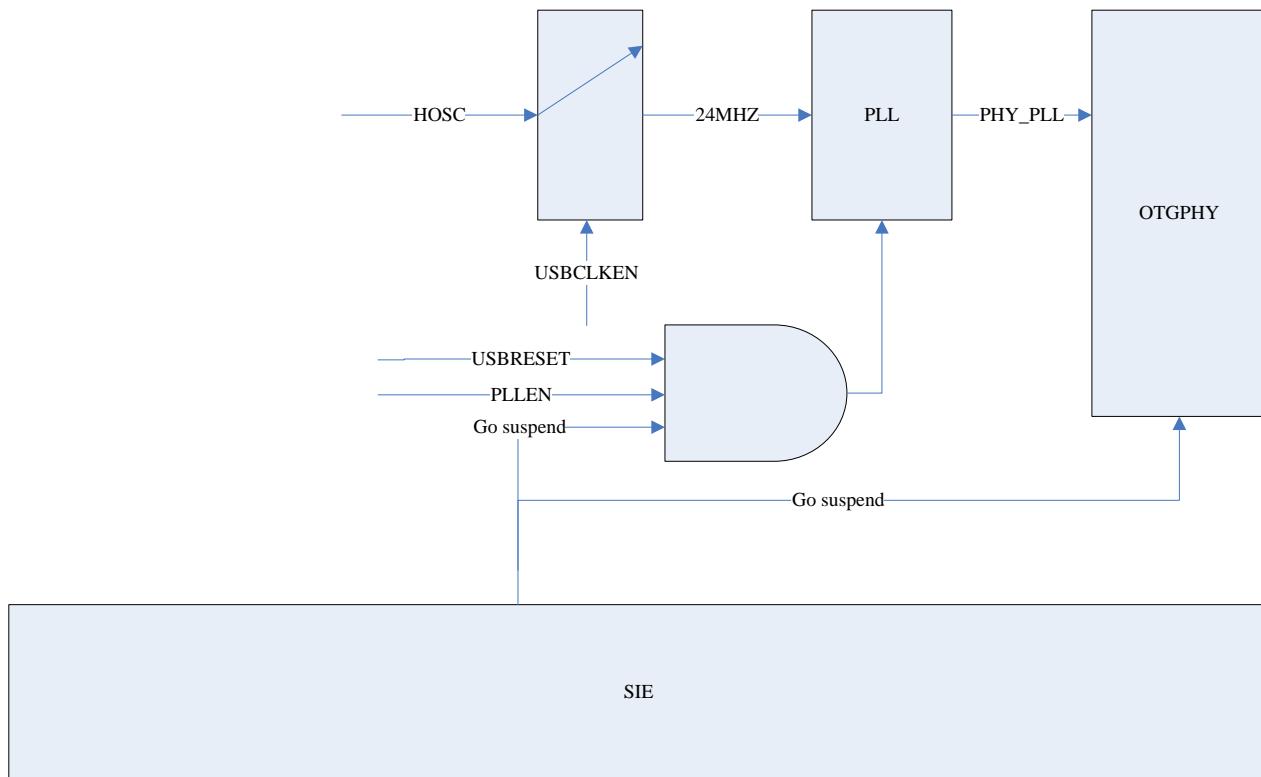


Figure 11-10 USB CLK

## 11.1.4 Operation Manual

### 11.1.4.1 Flow Of Software

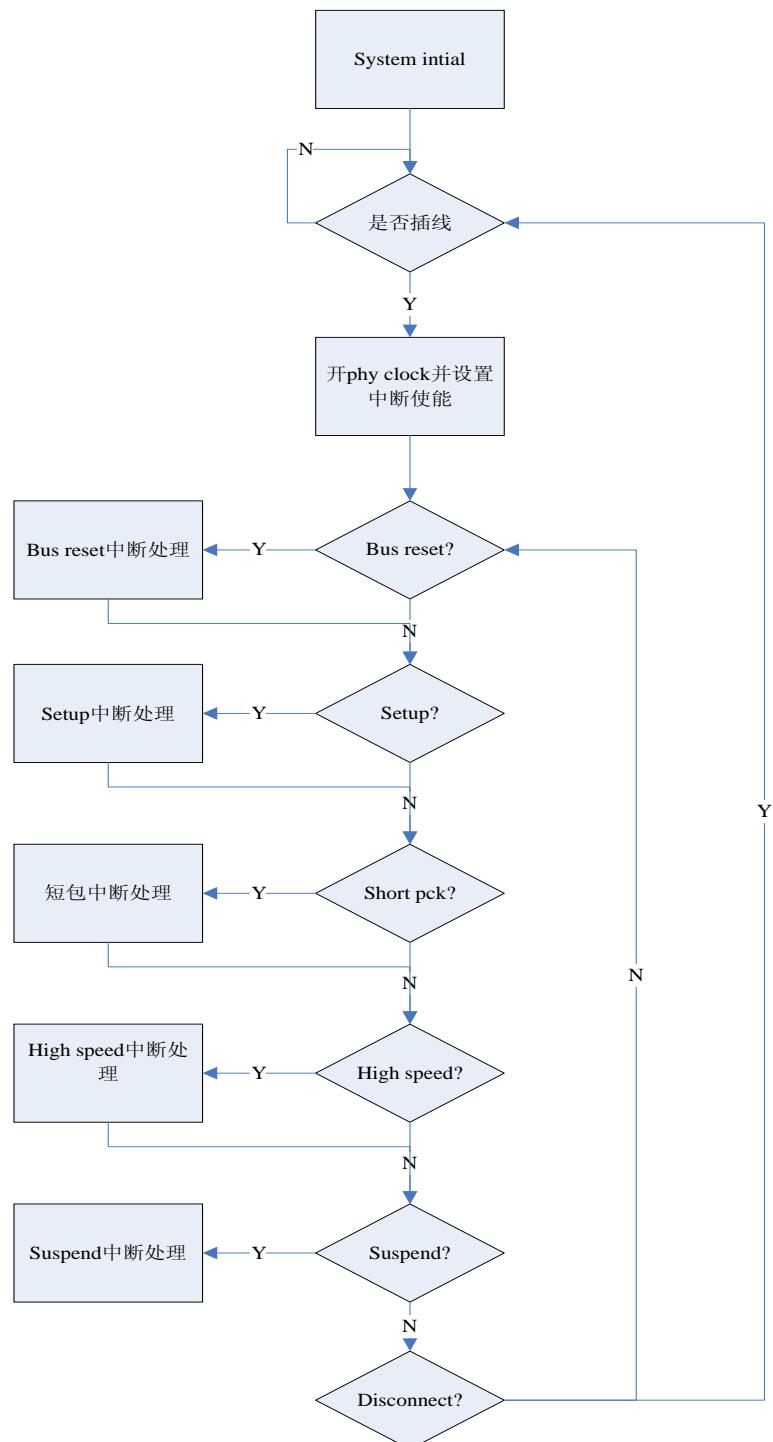


Figure 11-11device only 流程图

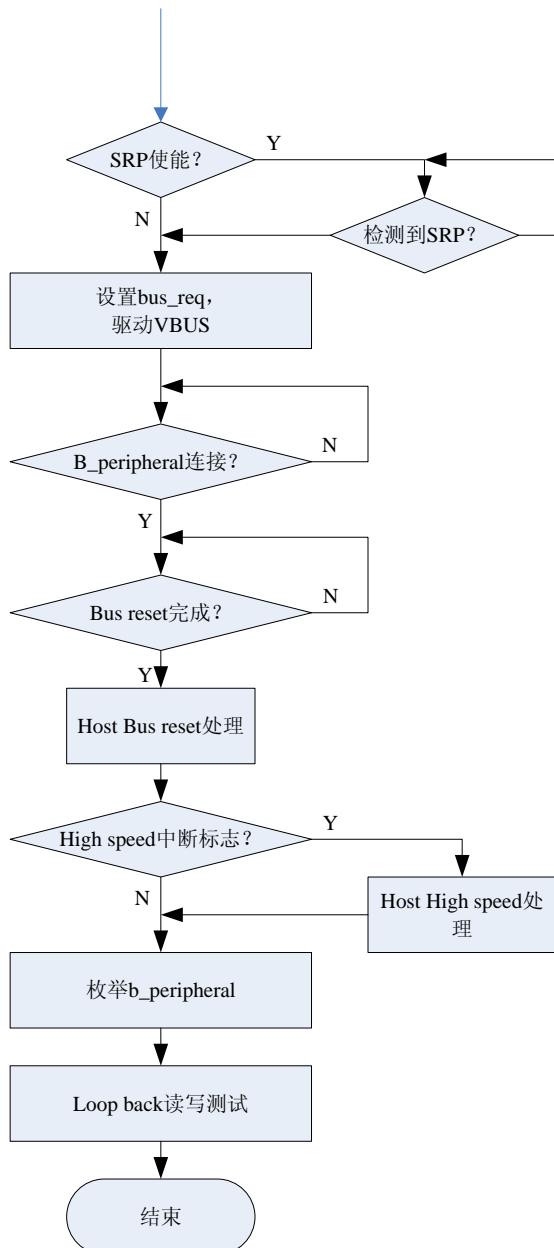


Figure 11-12 Host only 流程图

#### 11.1.4.2 URES AND EFUSE

If the URES has been check, the URES check itself after power on. There is nothing have to do in software.

#### 11.1.4.3 Configuration of EP and Memory

See section Endpoint and memory configuration and make sure that the EP FIFO don't use the same RAM.

#### 11.1.4.4 Interrupt

- The EXT.USBIRQ(USBEIRQ\_bit7) bit must be cleared first and then the INT.USBIRQ bit, when clearing the USB IRQ
- INT.USBIRQ may be endpoint interrupts, transmission error interrupts, endpoint token interrupts, start of frame interrupt, suspend interrupt, USB bus reset interrupt, setup token interrupt, setup data valid interrupt.

#### 11.1.4.5 DMA

- UDMAM register: Start USB DMA transfer or reset USB DMA, for EP1 and EP2.
- EP1DMALEN/EP2DMALEN length register (USB transfer length): DMA transfer length. If it is 0x00, that means there are 64KB data to transferred.
- When USB send out data, the software should check UDMAM register to make sure that the DMA mode transfers is over.

#### 11.1.4.6 Plug in detect

- Device mode:  
In device mode, DPDM 500KR pull up resistance is enabled. And DPDM is high level. When device is plug into host, DPDM is pulled down by host. We can judge whether the device has been connected to the HOST by reading the line status. If enable the line detect, we can use the plugin bit to judge.
- HOST mode:  
In host mode, DPDM 15KR pull down resistance is enabled. And DPDM is low level, when device is connected to host, DPDM of the device should be pulled down and then the device should pull up DP. Now, host can judge whether the device has been connected by reading the line status.

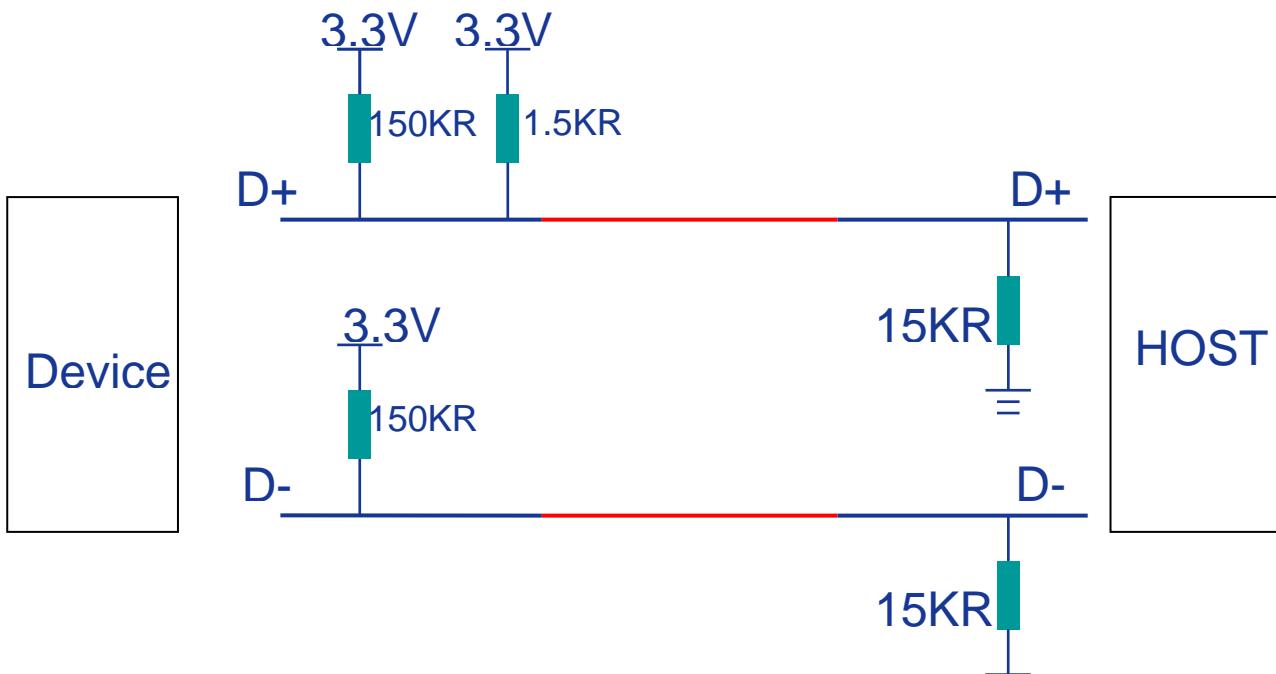


Figure 11-13 外围连接图

#### 11.1.4.7 Initialization

- Make sure that USBCLK is enabled, which is in CMU module.
- Make sure that the USB module is set to normal, which is in RMU module.
- USB reset must set reset2 to normal first and then reset1.
- In device mode, the IDVBUSCTRL register is set to 0x0c and the DPDMCTRL register is set to 0x1f.
- In HOST mode, the IDVBUSCTRL register is set to 0x04 and the DPDMCTRL register is set to 0x10.
- After detect plug in(DEVICE or HOST), enable the PHYPPLL and initialize the USB interrupt.

#### 11.1.4.8 AUTO mode

- You can set the USB BULK transfer mode into auto mode by setting the Fifoctrl register.
- In auto mode, the EP FIFO (out) will auto send out data if it is a full packet. If it is a short packet, you must set busy to send out the data. Before moving data to FIFO, you should set the FIFO into auto mode if you want to use the auto mode.
- In auto mode, the EP FIFO (IN) will auto send in token until all data is received. You can use the HCIN2CNT register to configure the number of in token to send. When you read data from the FIFO, you should set busy to make the FIFO pointer point to next buffer if it is a short packet. Long packet is needless.

### 11.1.4.9 Exit

- Save the variable for the system.
- Reset USB RESET1 and then set it to normal
- Close the PHYPLL (USB\_PHYCTRL register)

### 11.1.4.10 Error Detection and Recovery

The hardware show the mistake during the transfers, such as no endpoint handshake, PID error, STALL, data toggle mismatch, CRC error, Data Overrun, Data Under run. All mistake should be deal with.

- no endpoint handshake: the software should try 3 times. Then send bus reset signal and enumerate the device
- PID error: that may be the signal problem. The software should try 3 times to make sure.
- STALL: clear STALL and try again
- data toggle mismatch: resend or try again
- CRC error: resend or try again
- Data Overrun: try again
- Data Under run: try again

## 11.1.5 USB Register List

**Table SFR Registers**

| Index | Mnemonic                       | Description                                 | BANK |
|-------|--------------------------------|---|------|
| 0x86  | <a href="#">UDMAM</a>          | USB DMA MODE                                | 0x07 |
| 0x89  | <a href="#">AUTOTIMER</a>      | Auto in mode in token timer                 | 0x07 |
| 0x8a  | <a href="#">EP1STADDRH</a>     | EP1 FIFO start address high register        | 0x07 |
| 0x8b  | <a href="#">EP1STADDRL</a>     | EP1 FIFO start address low register         | 0x07 |
| 0x8c  | <a href="#">EP1DMALENH</a>     | EP1 DMA transfer length high in normal mode | 0x07 |
| 0x8d  | <a href="#">EP1DMALENL</a>     | EP1 DMA transfer length low in normal mode  | 0x07 |
| 0x90  | <a href="#">OUTPCKCNTH</a>     | Out transaction packet counter high         | 0x07 |
| 0x91  | <a href="#">OUTPCKCNTL</a>     | Out transaction packet counter low          | 0x07 |
| 0x92  | <a href="#">IDVBUSCTRL</a>     | ID&VBus control                             | 0x07 |
| 0x93  | <a href="#">USBSTATUS</a>      | USB status                                  | 0x07 |
| 0x94  | <a href="#">DPDMCTRL</a>       | DP DM control register                      | 0x07 |
| 0x95  | <a href="#">USB_PHYCTRL</a>    | PHY control register                        | 0x07 |
| 0x97  | <a href="#">Out0bc_hcin0bc</a> | Endpoint 0 OUT Byte Count                   | 0x07 |

|      |                           |  |      |
|------|---------------------------|--|------|
| 0x98 | <u>In0bc_hcout0bc</u>     | Endpoint 0 IN Byte Count                   | 0x07 |
| 0x99 | <u>Ep0cs_hcep0cs</u>      | Endpoint 0 Control and Status              | 0x07 |
| 0x9a | <u>In1bch_hcout1bch</u>   | Endpoint 1 IN Byte Count High              | 0x07 |
| 0x9b | <u>In1bcl_hcout1bcl</u>   | Endpoint 1 IN Byte Count Low               | 0x07 |
| 0x9c | <u>In1ctrl_hcout1ctrl</u> | Endpoint 1 IN Control                      | 0x07 |
| 0x9d | <u>In1cs_hcout1cs</u>     | Endpoint 1 IN Control And Status           | 0x07 |
| 0x9e | <u>Out2bch_hcin2bch</u>   | Endpoint 2 OUT Byte Count High             | 0x07 |
| 0x9f | <u>Out2bcl_hcin2bcl</u>   | Endpoint 2 OUT Byte Count Low              | 0x07 |
| 0xa2 | <u>Out2ctrl_hcin2ctrl</u> | Endpoint 2 OUT Control                     | 0x07 |
| 0xa3 | <u>Out2cs_hcin2cs</u>     | Endpoint 2 OUT Control And Status          | 0x07 |
| 0xa4 | <u>In3bc_hcout3bc</u>     | Endpoint 3 IN Byte Count                   | 0x07 |
| 0xa5 | <u>In3ctrl_hcout3ctrl</u> | Endpoint 3 IN Control                      | 0x07 |
| 0xa6 | <u>In3cs_hcout3cs</u>     | Endpoint 3 IN Control And Status           | 0x07 |
| 0xa7 | <u>Fifoldat</u>           | FIFO 1 Data                                | 0x07 |
| 0xa9 | <u>Fifo2dat</u>           | FIFO 2 Data                                | 0x07 |
| 0xaa | <u>Fifo3dat</u>           | FIFO 3 Data                                | 0x07 |
| 0xab | <u>Ep0indata</u>          | EP0 IN DATA                                | 0x07 |
| 0xac | <u>Ep0outdata</u>         | EP0 OUT DATA                               | 0x07 |
| 0xad | <u>Usbirq_hcusbirq</u>    | USB Interrupt                              | 0x07 |
| 0xae | <u>Usbien_hcusbien</u>    | USB interrupt enable                       | 0x07 |
| 0xaf | <u>SHORTPCKIRQ</u>        | Short packets Interrupt request and enable | 0x07 |
| 0xb0 | <u>Hcep0ctrl</u>          | Endpoint 0 Control                         | 0x07 |
| 0xb1 | <u>Hcout0err</u>          | Endpoint 0 HC OUT Error                    | 0x07 |
| 0xb2 | <u>Hcin0err</u>           | Endpoint 0 HC IN Error                     | 0x07 |
| 0xb3 | <u>Hcout1ctrl</u>         | Endpoint 1 HC OUT Control                  | 0x07 |
| 0xb4 | <u>Hcout1err</u>          | Endpoint 1 HC OUT Error                    | 0x07 |
| 0xb5 | <u>Hcin2ctrl</u>          | Endpoint 2 HC IN Control                   | 0x07 |
| 0xb6 | <u>Hcin2err</u>           | Endpoint 2 HC OUT Error                    | 0x07 |
| 0xb7 | <u>EP2STADDRH</u>         | EP2 FIFO start address high register       | 0x07 |
| 0xb8 | <u>EP2STADDRL</u>         | EP2 FIFO start address low register        | 0x07 |

|      |                                     |  |      |
|------|-------------------------------------|--|------|
| 0xb9 | <a href="#"><u>Hcout3ctrl</u></a>   | Endpoint 3 HC OUT Control                      | 0x07 |
| 0xba | <a href="#"><u>Hcout3err</u></a>    | Endpoint 3 HC OUT Error                        | 0x07 |
| 0xbb | <a href="#"><u>Setupdat0</u></a>    | SETUP DATA0                                    | 0x07 |
| 0xbc | <a href="#"><u>Setupdat1</u></a>    | SETUP DATA1                                    | 0x07 |
| 0xbd | <a href="#"><u>Setupdat2</u></a>    | SETUP DATA2                                    | 0x07 |
| 0xbf | <a href="#"><u>Setupdat3</u></a>    | SETUP DATA3                                    | 0x07 |
| 0xc1 | <a href="#"><u>Setupdat4</u></a>    | SETUP DATA4                                    | 0x07 |
| 0xc2 | <a href="#"><u>Setupdat5</u></a>    | SETUP DATA5                                    | 0x07 |
| 0xc3 | <a href="#"><u>Setupdat6</u></a>    | SETUP DATA6                                    | 0x07 |
| 0xc4 | <a href="#"><u>Setupdat7</u></a>    | SETUP DATA7                                    | 0x07 |
| 0xc5 | <a href="#"><u>Ep03irq</u></a>      | Endpoint 0 to 3 Interrupt Request              | 0x07 |
| 0xc6 | <a href="#"><u>Ep03ien</u></a>      | Endpoint 0 to 3 Interrupt Enables              | 0x07 |
| 0xc7 | <a href="#"><u>Ep03tokirq</u></a>   | Endpoint 0 to 3 Token Interrupt Request        | 0x07 |
| 0xc8 | <a href="#"><u>Ep03tokien</u></a>   | Endpoint 0 to 3 Token Interrupt Request Enable | 0x07 |
| 0xc9 | <a href="#"><u>IVECT</u></a>        | Interrupt Vector                               | 0x07 |
| 0xca | <a href="#"><u>EPRST</u></a>        | Endpoint Reset                                 | 0x07 |
| 0xcb | <a href="#"><u>UsbCTRL_STUS</u></a> | USB Control And Status                         | 0x07 |
| 0xcc | <a href="#"><u>FrmCNTH</u></a>      | USB Frame Counter HIGH                         | 0x07 |
| 0xcd | <a href="#"><u>FrmCNTL</u></a>      | USB Frame Counter Low                          | 0x07 |
| 0xce | <a href="#"><u>Fnaddr</u></a>       | Function Address                               | 0x07 |
| 0xcf | <a href="#"><u>Clkgate</u></a>      | Clock Gate                                     | 0x07 |
| 0xd2 | <a href="#"><u>Fifoctrl</u></a>     | FIFO Control                                   | 0x07 |
| 0xd3 | <a href="#"><u>Hcportctrl</u></a>   | HC Port Control                                | 0x07 |
| 0xd4 | <a href="#"><u>Hcfrmnh</u></a>      | HC Frame Number high                           | 0x07 |
| 0xd5 | <a href="#"><u>Hcfrmn1</u></a>      | HC Frame Number low                            | 0x07 |
| 0xd6 | <a href="#"><u>Hcfrmremainh</u></a> | HC Frame Remain high                           | 0x07 |
| 0xd7 | <a href="#"><u>Hcfrmremainl</u></a> | HC Frame Remain Low                            | 0x07 |
| 0xd8 | <a href="#"><u>Hcep03errirq</u></a> | HC 0 to 3 Error Interrupt Request              | 0x07 |
| 0xd9 | <a href="#"><u>Hcep03errien</u></a> | HC 0 to 3 Error Interrupt Enable               | 0x07 |
| 0xda | <a href="#"><u>Otgirq</u></a>       | OTG Interrupt Request                          | 0x07 |
| 0xdb | <a href="#"><u>Otgstate</u></a>     | The OTG FSM State                              | 0x07 |

|      |                               |   |      |
|------|-------------------------------|---|------|
| 0xdc | <a href="#">Otgcctrl</a>      | OTG Control                                 | 0x07 |
| 0xdd | <a href="#">Otgstatus</a>     | OTG Status                                  | 0x07 |
| 0xde | <a href="#">Otgiens</a>       | OTG Interrupt Enable                        | 0x07 |
| 0xe2 | <a href="#">EP2DMALENH</a>    | EP2 DMA transfer length high in normal mode | 0x07 |
| 0xe3 | <a href="#">EP2DMALENL</a>    | EP2 DMA transfer length low in normal mode  | 0x07 |
| 0xe4 | <a href="#">Hcin0maxpck</a>   | HC IN 0 Max Packet Size                     | 0x07 |
| 0xe5 | <a href="#">Hcin2maxpckh</a>  | HC IN 2 max packet high                     | 0x07 |
| 0xe6 | <a href="#">Hcin2maxpckl</a>  | HC IN 2 max packet low                      | 0x07 |
| 0xe7 | <a href="#">Hcout3maxpck</a>  | HC OUT 3 max packet                         | 0x07 |
| 0xe9 | <a href="#">Hcout1maxpckh</a> | HC OUT 1 max packet high                    | 0x07 |
| 0xea | <a href="#">Hcout1maxpckl</a> | HC OUT 1 max packet low                     | 0x07 |
| 0xeb | <a href="#">USBEIRQ</a>       | USB external Interrupt request              | 0x07 |
| 0xec | <a href="#">AUTONAKCTRL</a>   | auto nak control                            | 0x07 |
| 0xed | <a href="#">HCINCTRL</a>      | host in control                             | 0x07 |
| 0xee | <a href="#">DBGMODE</a>       | debug mode                                  | 0x07 |
| 0xef | <a href="#">VDCTRL</a>        | USB PHY vendor control                      | 0x07 |
| 0xf1 | <a href="#">VDSTAT</a>        | USB PHY vendor status                       | 0x07 |
| 0xf3 | <a href="#">BKDOOR</a>        | Test back door                              | 0x07 |
| 0xf5 | <a href="#">OTGTRIEN</a>      | OTR status machine interrupt enable         | 0x07 |
| 0xf6 | <a href="#">OTGTRIRQ</a>      | OTR status machine interrupt request        | 0x07 |
| 0xf7 | <a href="#">USB_Efuse_Ref</a> | Usb Access Efuse_Ref register               | 0x07 |
| 0xfc | <a href="#">FSMPRESTATE</a>   | FSM pre-state register                      | 0x07 |
| 0xfd | <a href="#">HCIN2CNTH</a>     | hcin2 packet counter high                   | 0x07 |
| 0xfe | <a href="#">HCIN2CNTL</a>     | hcin2 packet counter low                    | 0x07 |

## 11.1.6 Register Description

### 11.1.6.1 UDMAM

USB DMA MODE

Offset = 0x86

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|--------|------|-------------|-----|-------|

|     |               |                               |     |     |
|-----|---------------|-------------------------------|-----|-----|
| 7:1 | -             | Reserved                      | R   | 00H |
| 3   | Ep2DMAfiforst | 1: reset<br>Auto clera        | R/w | 0   |
| 2   | Ep1DMAfiforst | 1: reset<br>Auto clera        | R/w | 0   |
| 1   | Ep2dmastart   | 0: cpuread<br>1: ep2dmastart  | R/W | 0   |
| 0   | Ep1dmastart   | 1: Ep1dmastart<br>0: cpuwrite | R/W | 0   |

### 11.1.6.2 AUTOINTIMER

Auto in mode in token timer

Offset = 0x89

| Bit(s) | Name        | Description  | R/W | Reset |
|--------|-------------|--|-----|-------|
| 7:0    | AUTOINTIMER | The time between in token.<br><br>Tauto= AUTOINTIMER*1/30M<br><br>If the time waiting for the condition of in token is longer then Tauto,, send in token when the condition is meet<br><br>If the time waiting for the condition of in token is shorter then Tauto,, send in token after Tauto | R/W | 00H   |

### 11.1.6.3 EP1DMALENL

EP1 DMA transfer length Register low

Offset = 0x8d

| Bit(s) | Name       | Description                                | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | EP1DMALENL | Ep1 DMA transfer length low in normal mode | R/W | 00H   |

Note: when the register NORMALLENL and NORMALLENH are 0x00, that will make DMA transfer 64KB.

### 11.1.6.4 EP1DMALENH

EP1 DMA transfer length Register high

Offset = 0x8c

| Bit(s) | Name       | Description                                 | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | EP1DMALenH | Ep1 DMA transfer length high in normal mode | R/W | 00H   |

Note: when the register NORMALLENL and NORMALENH are 0x00, that will make DMA transfer 64KB.

### 11.1.6.5 EP2DMALENL

EP2 DMA transfer length Register low

Offset = 0xe3

| Bit(s) | Name       | Description                                | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | EP2DMALenL | Ep2 DMA transfer length low in normal mode | R/W | 00H   |

Note: when the register NORMALLENL and NORMALENH are 0x00, that will make DMA transfer 64KB.

### 11.1.6.6 EP2DMALENH

EP2 DMA transfer length Register high

Offset = 0xe2

| Bit(s) | Name       | Description                                 | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | EP2DMALenH | Ep2 DMA transfer length high in normal mode | R/W | 00H   |

Note: when the register NORMALLENL and NORMALENH are 0x00, that will make DMA transfer 64KB.

### 11.1.6.7 OUTPCKCNTL

Out transaction packet counter register low

Offset = 0x91

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | OutPckCntL | Out transaction packet counter low. IF the out packet counter enable, the SIE will send nyet for out token when SIE received the indicated length by this register. | R/W | 00H   |

### 11.1.6.8 OUTPCKCNTH

Out transaction packet counter register high

Offset = 0x90

| Bit(s) | Name         | Description                      | R/W | Reset |
|--------|--------------|----------------------------------|-----|-------|
| 7      | OutPckCnt_EN | 1: enable the out packet counter | R/W | 0B    |

|     |            |  |     |          |
|-----|------------|--|-----|----------|
|     |            | 0:disable the out packet counter   |     |          |
| 6:0 | OutPCKCntH | Out transaction packet counter high IF the out packet counter enable, the sie will send nyet for out token when sie received the indicated length by this register | R/W | 0000000B |

### 11.1.6.9 IDVBUSCTRL

ID&VBUS control register

Offset = 0x92

| Bit(s) | Name      | Description                                    | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:4    | -         | Reserved                                       | R   | 0000B |
| 3      | SoftID    | Software ID,instead of the ID pin              | R/W | 1B    |
| 2      | SoftID_EN | Softid enable bit<br>1: enable      0: disable | R/W | 0B    |
| 1:0    | -         | Reserved                                       | R   |       |

PS: this register use the MCUCLK to accessd

### 11.1.6.10 USBSTATUS

USB status register

Offset = 0x93

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:5    | -        | Reserved  | R   | 000B  |
| 4:3    | USB_ls   | USB linestate[1:0]<br>Linestate0:DP<br>Linestate1:DM                              | R   | -     |
| 2:1    | -        | Reserved  | R   | 00B   |
| 0      | otgreset | USB OTG reset.If AOTG is in reset state, this bit will be set, else it will be 0. | R/W |       |

PS: this register use the MCUCLK to accessd

### 11.1.6.11 DPDMCTRL

DP DM control register

Offset = 0x94

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7      | -         | Reserved  | R   | 0B    |
| 6      | Plugin    | This bit Indicated the usb connection status when Linedeten is enable.<br>1: connect<br>0: disconnect | R   | -     |
| 5      | -         | Reserved  | R   | 0B    |
| 4      | LineDetEN | Line status detect enable<br>1: enable 0: disable   | R/W | 1B    |
| 3      | DMPuEn    | 500Kohm DM pull up resistor enable.<br>1: enable 0: disable   | R/W | 1B    |
| 2      | DPPuEn    | 500Kohm DP pull up resistor enable.<br>1: enable 0: disable   | R/W | 1B    |
| 1      | DMPdDis   | DM pull down disable.<br>1: disable 0: enable   | R/W | 1B    |
| 0      | DPPdDis   | DP pull down disable.<br>1: disable 0: enable   | R/W | 1B    |

PS: this register use the MCUCLK to accessd

### 11.1.6.12 USB\_PHYCTRL

PHY control register

Offset = 0x95

| Bit(s) | Name           | Description          | R/W | Reset |
|--------|----------------|----------------------|-----|-------|
| 7      | Phy_PLLEN      | 1: DISABLE 0: ENABLE | R/W | 0     |
| 6      | Phy_DALLUALLEN | 1: DISABLE 0: ENABLE | R/W | 0     |
| 5:0    | -              | Reserved             | R   | 0     |

PS: this register use the MCUCLK to accessd

### 11.1.6.13 Otgirq

OTG Interrupt Request Register

Offset = 0xda

| Bit(s) | Name      | Description                                       | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:5    | -         | Reserved  | R   | 000B  |
| 4      | periphirq | Interrupt request on a_peripheral or b_peripheral | R/W | 0B    |

|   |           |   |     |      |
|---|-----------|---|-----|------|
|   |           | state entering. If this interrupt is requested, fuse layer should prepare to transmission in the peripheral mode. Microprocessor writes a '1' to clear otgirq bit.                                |     |      |
| 3 | -         | Reserved  | R   | 000B |
| 2 | locsofirq | Interrupt request on a_a_host or b_host state entering  | R/W | 0B   |
| 1 | -         | Reserved  | R   | 000B |
| 0 | idleirq   | Interrupt request on OTG FSM entering a_idle or b_idle state. Microprocessor should read id bit to define which of the states has been reached. Microprocessor writes a '1' to clear idleirq bit. | R/W | 0B   |

### 11.1.6.14 Otgstate

The OTG FSM State Register

Offset = 0xdb

| Bit(s) | Name    | Description  | R/W | Reset |
|--------|---------|--|-----|-------|
| 7:4    | -       | Reserved   | R   | 0000B |
| 3:0    | st3_st0 | 0 – 0000 – A_idle<br>1 – 0001 – A_wait_vrise<br>2 – 0010 – A_wait_bcon<br>3 – 0011 – A_host<br>4 – 0100 – A_suspend<br>5 – 0101 – A_peripheral<br>6 – 0110 – A_vbus_err<br>7 – 0111 – A_wait_vfall<br>8 – 1000 – B_idle<br>9 – 1001 – B_peripheral<br>A – 1010 – B_wait_acon<br>B – 1011 – B_host<br>C – 1100 – B_srp_init1<br>D – 1101 – B_srp_init2<br>E – 1110 – B_dischrg1 | R   | 0000B |

|  |  |                       |  |  |
|--|--|-----------------------|--|--|
|  |  | F – 1111 – B_dischrg2 |  |  |
|--|--|-----------------------|--|--|

### 11.1.6.15 Otgctrl

OTG Control Register

Offset = 0xdc

| Bit(s) | Name       | Description  | R/W | Reset |
|--------|------------|--|-----|-------|
| 7      | forcebconn | Bit used for the core testing purpose. It should be always written with zero. If set, short debounce interval (TA_BCON_SDB) instead of long debounce interval (TA_BCON_LDB) is required to detect B-device connection. | R/W | 0     |
| 6:1    | -          | reserved   | R   | 0     |
| 0      | busreq     | Bit used to start or end the session. Meaning of this bit is, depending on id signal state, the same as a_bus_req or b_bus_req bit from OTG Supplement Specification.  | R/W | 0     |

### 11.1.6.16 Otgstatus

OTG Status Register

Offset = 0xdd

| Bit(s) | Name | Description  | R/W | Reset |
|--------|------|--|-----|-------|
| 7      | -    | Reserved   | R   | 0B    |
| 6      | id   | Type of plug that is inserted to device's receptacle:<br>‘0’ – Mini-A<br>‘1’ – Mini-B  | R   | 0B    |
| 5:2    | -    | Reserved   | R   | 0000B |
| 1      | conn | Device connection to USB line has been detected.<br>Meaning of this bit depends on id bit value:<br>a_conn – for B-device<br>b_conn – for A-device | R   | 0B    |
| 0      | -    | Reserved   | R   | 0B    |

### 11.1.6.17 Otgien

OTG Interrupt Enable Register

Offset = 0xde

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:5    | -         | Reserved   | R   | 000B  |
| 4      | periphien | Periphirq interrupt enable. When periphien bit is '0', the interrupt request bit periphirq is ignored. | R/W | 0B    |
| 3      | -         | Reserved   | R   | 0B    |
| 2      | locsofien | Connirq interrupt enable. When connien bit is '0', the interrupt request bit locsofirq is ignored.     | R/W | 0B    |
| 1      | -         | Reserved   | R   | 0B    |
| 0      | idleien   | Idleirq interrupt enable. When idleien bit is '0', the interrupt request bit idleirq is ignored.       | R/W | 0B    |

### 11.1.6.18 Hcep0ctrl

Endpoint 0 Control Register

Offset = 0xb0

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:4    | -         | Reserved  | R   | 0000B |
| 3:0    | HcEp0Ctrl | <p>Endpoint 0 can be configured for communication with any control endpoint in a peripheral device. To support this feature the hcep0ctrl register was introduced. The hcep0ctrl(3 downto 0) register contains the address of the endpoint in the peripheral device which will communicate with endpoint 0 in HC. For example,</p> <p>when the microprocessor writes 04H to the hcep0ctrl register, the HC endpoint OUT0 (or IN0) will transfer data to the OUT4 (or receive from IN4) endpoint that is located in the peripheral device.</p> | R/W | 0000B |

### 11.1.6.19 Hcout1ctrl

Endpoint 1 HC OUT Control Registers

Offset = 0xb3

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:4    | -         | Reserved   | R   | 0000B |
| 3:0    | HcEp1Ctrl | <p>Each HC endpoint can be configured for communication with any endpoint in a peripheral device. To support this feature, the hcoutxctrl register was introduced. The hcoutxctrl(3 downto 0) register contains the address of the endpoint in the peripheral device which will communicate with endpoint x in HC.</p> <p>For example, when the microprocessor writes 04H to the hcout1ctrl register, the HC endpoint OUT1 will transfer data to the OUT4 endpoint that is located in the peripheral device.</p> | R/W | 0000B |

### 11.1.6.20 Hcin2ctrl

Endpoint 2 HC IN Control Registers

Offset = 0xb5

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:4    | -         | Reserved   | R   | 0000B |
| 3:0    | HcEp2Ctrl | <p>Each HC endpoint can be configured for communication with any endpoint in a peripheral device. To support this feature, the hcoutxctrl register was introduced. The hcoutxctrl(3 downto 0) register contains the address of the endpoint in the peripheral device which will communicate with endpoint x in HC.</p> <p>For example, when the microprocessor writes 04H to the hcout1ctrl register, the HC endpoint OUT1 will transfer data to the OUT4 endpoint that is located in the peripheral device.</p> | R/W | 0000B |

### 11.1.6.21 Hcout3ctrl

Endpoint 3 HC OUT Control Registers

Offset = 0xb9

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |           |  |     |       |
|-----|-----------|--|-----|-------|
| 7:4 | -         | Reserved   | R   | 0000B |
| 3:0 | HcEp3Ctrl | <p>Each HC endpoint can be configured for communication with any endpoint in a peripheral device. To support this feature, the hcoutxctrl register was introduced. The hcoutxctrl(3 downto 0) register contains the address of the endpoint in the peripheral device which will communicate with endpoint x in HC.</p> <p>For example, when the microprocessor writes 04H to the hcout1ctrl register, the HC endpoint OUT1 will transfer data to the OUT4 endpoint that is located in the peripheral device.</p> | R/W | 0000B |

### 11.1.6.22 Hcout0err

Endpoint 0 HC OUT Error Registers

Offset = 0xb1

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7      | -      | Reserved  | R   | 0B    |
| 6      | doping | <p>Do ping transaction. Read/ Write. Writing ‘1’ to this bit causes sending of PING packet instead the OUT packet for HC OUT x endpoint (only for BULK or CONTROL endpoints).</p> <p>0 – Host Controller sends OUT packet<br/>1 – Host Controller sends PING packet</p> <p>This bit is also controlled by Host Controller.</p> <p>Doping bit value is modified according to the table:</p> <p>Current ‘doping’/ Event Host / Event Device / Next ‘doping’</p> <ul style="list-style-type: none"> <li>1 / PING / Nak / 1</li> <li>1 / PING / Ack / 0</li> <li>1 / PING / no response (or response error) / 1</li> <li>1 / PING / Stall / 1</li> <li>0 / OUT / Nak / 1</li> <li>0 / OUT / Nyet / 1</li> </ul> | W   | 0B    |

|     |                |   |   |      |
|-----|----------------|---|---|------|
|     |                | 0 / OUT / Ack / 0<br>0 / OUT / no response (or response error) / 1<br>0 / OUT / Stall / 1   |   |      |
| 5   | Resend         | The resend bit. Write-only. It always returns ‘0’ during reading.<br><br>After three unsuccessful transactions, HC temporarily stops transmission for endpoint x and generates the error interrupt request. Microprocessor decides if transmission from endpoint x will be continued. If not, it disables endpoint x (endpoint halt). When the microprocessor sets the resend bit the endpoint x is resumed and the HC schedules it for next transaction. | W | 0B   |
| 4:2 | Errtype[2..0]  | Transmission error type. Status register – Read only.<br><br>000 – reserved (no error)<br>011 – endpoint sent STALL handshake<br>100 – no endpoint handshake (timeout)<br>101 – PID error (pid check=error or unknown PID – while handshake receiving)  | R | 000B |
| 1:0 | Errcount[1..0] | Transmission error counter. Status register – Read only.<br><br>This counter is incremented on every transmission error (CRC error or no handshake).<br><br>It is cleared on every correct transmission or by microprocessor (writing resend bit).<br><br>In case of STALL error, counter is written by „11” at once and OUT err interrupt is requested.  | R | 00B  |

### 11.1.6.23 Hcout1err

Endpoint 1 HC OUT Error Registers

Offset = 0xb4

| Bit(s) | Name   | Description                                      | R/W | Reset |
|--------|--------|--|-----|-------|
| 7      | -      | Reserved   | R   | 0B    |
| 6      | doping | Do ping transaction. Read/ Write. Writing ‘1’ to | W   | 0B    |

|  |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
|--|---------------|---|---|------|---------------|--|--------------------|--|--------------------|--|--|--|----------------------|--|-------------------|--|--------------------|--|-------------------|--|---|--|---------------------|--|--|--|
|  |               | <p>this bit causes sending of PING packet instead the OUT packet for HC OUT x endpoint (only for BULK or CONTROL endpoints).</p> <p>0 – Host Controller sends OUT packet<br/>1 – Host Controller sends PING packet</p> <p>This bit is also controlled by Host Controller.</p> <p>Doping bit value is modified according to the table:</p> <table border="0"> <tr><td>Current ‘doping’/ Event Host / Event Device /</td><td></td></tr> <tr><td>Next ‘doping’</td><td></td></tr> <tr><td>1 / PING / Nak / 1</td><td></td></tr> <tr><td>1 / PING / Ack / 0</td><td></td></tr> <tr><td>1 / PING / no response (or response error) / 1</td><td></td></tr> <tr><td>1 / PING / Stall / 1</td><td></td></tr> <tr><td>0 / OUT / Nak / 1</td><td></td></tr> <tr><td>0 / OUT / Nyet / 1</td><td></td></tr> <tr><td>0 / OUT / Ack / 0</td><td></td></tr> <tr><td>0 / OUT / no response (or response error) / 1</td><td></td></tr> <tr><td>0 / OUT / Stall / 1</td><td></td></tr> </table> | Current ‘doping’/ Event Host / Event Device / |      | Next ‘doping’ |  | 1 / PING / Nak / 1 |  | 1 / PING / Ack / 0 |  | 1 / PING / no response (or response error) / 1 |  | 1 / PING / Stall / 1 |  | 0 / OUT / Nak / 1 |  | 0 / OUT / Nyet / 1 |  | 0 / OUT / Ack / 0 |  | 0 / OUT / no response (or response error) / 1 |  | 0 / OUT / Stall / 1 |  |  |  |
| Current ‘doping’/ Event Host / Event Device /  |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| Next ‘doping’                                  |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 1 / PING / Nak / 1                             |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 1 / PING / Ack / 0                             |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 1 / PING / no response (or response error) / 1 |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 1 / PING / Stall / 1                           |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 0 / OUT / Nak / 1                              |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 0 / OUT / Nyet / 1                             |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 0 / OUT / Ack / 0                              |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 0 / OUT / no response (or response error) / 1  |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 0 / OUT / Stall / 1                            |               |   |   |      |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 5  | Resend        | <p>The resend bit. Write-only. It always returns ‘0’ during reading.</p> <p>After three unsuccessful transactions, HC temporarily stops transmission for endpoint x and generates the error interrupt request. Microprocessor decides if transmission from endpoint x will be continued. If not, it disables endpoint x (endpoint halt). When the microprocessor sets the resend bit the endpoint x is resumed and the HC schedules it for next transaction.</p>  | W   | 0B   |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |
| 4:2  | Errtype[2..0] | <p>Transmission error type. Status register – Read only.</p> <p>000 – reserved (no error)<br/>011 – endpoint sent STALL handshake<br/>100 – no endpoint handshake (timeout)<br/>101 – PID error (pid check=error or unknown)</p>  | R   | 000B |               |  |                    |  |                    |  |  |  |                      |  |                   |  |                    |  |                   |  |   |  |                     |  |  |  |

|     |                |   |   |     |
|-----|----------------|---|---|-----|
|     |                | PID – while handshake receiving)  |   |     |
| 1:0 | Errcount[1..0] | <p>Transmission error counter. Status register – Read only.</p> <p>This counter is incremented on every transmission error (CRC error or no handshake).</p> <p>It is cleared on every correct transmission or by microprocessor (writing resend bit).</p> <p>In case of STALL error, counter is written by „11” at once and OUT err interrupt is requested.</p> | R | 00B |

### 11.1.6.24 Hcout3err

Endpoint 3 HC OUT Error Registers

Offset = 0xba

| Bit(s)  | Name   | Description  | R/W   | Reset              |                    |  |                      |                   |                    |                   |   |   |    |
|---|--------|--|---|--------------------|--------------------|--|----------------------|-------------------|--------------------|-------------------|---|---|----|
| 7   | -      | Reserved   | R   | 0B                 |                    |  |                      |                   |                    |                   |   |   |    |
| 6   | doping | <p>Do ping transaction. Read/ Write. Writing ‘1’ to this bit causes sending of PING packet instead the OUT packet for HC OUT x endpoint (only for BULK or CONTROL endpoints).</p> <p>0 – Host Controller sends OUT packet<br/>1 – Host Controller sends PING packet</p> <p>This bit is also controlled by Host Controller.</p> <p>Doping bit value is modified according to the table:</p> <table> <tr> <td>Current ‘doping’/ Event Host / Event Device / Next ‘doping’</td> </tr> <tr> <td>1 / PING / Nak / 1</td> </tr> <tr> <td>1 / PING / Ack / 0</td> </tr> <tr> <td>1 / PING / no response (or response error) / 1</td> </tr> <tr> <td>1 / PING / Stall / 1</td> </tr> <tr> <td>0 / OUT / Nak / 1</td> </tr> <tr> <td>0 / OUT / Nyet / 1</td> </tr> <tr> <td>0 / OUT / Ack / 0</td> </tr> <tr> <td>0 / OUT / no response (or response error) / 1</td> </tr> </table> | Current ‘doping’/ Event Host / Event Device / Next ‘doping’ | 1 / PING / Nak / 1 | 1 / PING / Ack / 0 | 1 / PING / no response (or response error) / 1 | 1 / PING / Stall / 1 | 0 / OUT / Nak / 1 | 0 / OUT / Nyet / 1 | 0 / OUT / Ack / 0 | 0 / OUT / no response (or response error) / 1 | W | 0B |
| Current ‘doping’/ Event Host / Event Device / Next ‘doping’ |        |  |   |                    |                    |  |                      |                   |                    |                   |   |   |    |
| 1 / PING / Nak / 1  |        |  |   |                    |                    |  |                      |                   |                    |                   |   |   |    |
| 1 / PING / Ack / 0  |        |  |   |                    |                    |  |                      |                   |                    |                   |   |   |    |
| 1 / PING / no response (or response error) / 1              |        |  |   |                    |                    |  |                      |                   |                    |                   |   |   |    |
| 1 / PING / Stall / 1  |        |  |   |                    |                    |  |                      |                   |                    |                   |   |   |    |
| 0 / OUT / Nak / 1   |        |  |   |                    |                    |  |                      |                   |                    |                   |   |   |    |
| 0 / OUT / Nyet / 1  |        |  |   |                    |                    |  |                      |                   |                    |                   |   |   |    |
| 0 / OUT / Ack / 0   |        |  |   |                    |                    |  |                      |                   |                    |                   |   |   |    |
| 0 / OUT / no response (or response error) / 1               |        |  |   |                    |                    |  |                      |                   |                    |                   |   |   |    |

|     |                |  |   |      |
|-----|----------------|--|---|------|
|     |                | 0 / OUT / Stall / 1  |   |      |
| 5   | Resend         | <p>The resend bit. Write-only. It always returns ‘0’ during reading.</p> <p>After three unsuccessful transactions, HC temporarily stops transmission for endpoint x and generates the error interrupt request. Microprocessor decides if transmission from endpoint x will be continued. If not, it disables endpoint x (endpoint halt). When the microprocessor sets the resend bit the endpoint x is resumed and the HC schedules it for next transaction.</p> | W | 0B   |
| 4:2 | Errtype[2..0]  | <p>Transmission error type. Status register – Read only.</p> <p>000 – reserved (no error)</p> <p>011 – endpoint sent STALL handshake</p> <p>100 – no endpoint handshake (timeout)</p> <p>101 – PID error (pid check=error or unknown PID – while handshake receiving)</p>  | R | 000B |
| 1:0 | Errcount[1..0] | <p>Transmission error counter. Status register – Read only.</p> <p>This counter is incremented on every transmission error (CRC error or no handshake).</p> <p>It is cleared on every correct transmission or by microprocessor (writing resend bit).</p> <p>In case of STALL error, counter is written by „11” at once and OUT err interrupt is requested.</p>  | R | 00B  |

### 11.1.6.25 Hcin0err

Endpoint 0 HC in Error Registers

Offset = 0xb2

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7      | underrien | <p>If Underrien=’1’ packet shorter than maximal (hcinxmaxpck) causes interrupt request (errtype=”111”).</p> <p>If Underrien=’0’ Data Underrun error is not</p> | W   | 0B    |

|     |                |   |   |      |
|-----|----------------|---|---|------|
|     |                | generated.  |   |      |
| 6   | -              | reserved  | R | 0B   |
| 5   | Resend         | <p>Control register – write only.</p> <p>After three unsuccessful transactions, HC temporarily stops transmission for endpoint x and generates the error interrupt request. Microprocessor decides if transmission from endpoint x will be continued. If not, it disables endpoint x (endpoint halt). When the microprocessor sets the resend bit the endpoint x is resumed and the HC schedules it for next transaction.</p> | W | 0B   |
| 4:2 | Errtype[2..0]  | <p>Transmission error type (status register – R).</p> <p>000 – reserved (no error)</p> <p>001 – CRC error</p> <p>010 – data toggle mismatch</p> <p>011 – endpoint sent STALL handshake</p> <p>100 – no endpoint handshake (timeout)</p> <p>101 – PID error (pid check=error or unknown PID)</p> <p>110 – Data Overrun (too long packet – babble)</p> <p>111 – Data Underrun (packet shorter than MaxPacketSize)</p>           | R | 000B |
| 1:0 | Errcount[1..0] | <p>Status register – Read only.</p> <p>Transmission error counter. This counter is incremented on every transmission error (CRC error or no handshake).</p> <p>It is cleared on every correct transmission or by microprocessor (writing resend bit).</p> <p>In case of data underrun, data overrun and STALL error, counter is written by „11” at once and IN err interrupt is requested.</p>                                | R | 00B  |

### 11.1.6.26 Hcin2err

Endpoint 2 HC in Error Registers

Offset = 0xb6

| Bit(s) | Name           | Description  | R/W | Reset |
|--------|----------------|--|-----|-------|
| 7      | underrien      | If Underrien='1' packet shorter than maximal (hcinxmaxpck) causes interrupt request (errtype="111").<br><br>If Underrien='0' Data Underrun error is not generated.   | W   | 0B    |
| 6      | -              | reserved   | R   | 0B    |
| 5      | Resend         | Control register – write only.<br><br>After three unsuccessful transactions, HC temporarily stops transmission for endpoint x and generates the error interrupt request. Microprocessor decides if transmission from endpoint x will be continued. If not, it disables endpoint x (endpoint halt). When the microprocessor sets the resend bit the endpoint x is resumed and the HC schedules it for next transaction. | W   | 0B    |
| 4:2    | Errtype[2..0]  | Transmission error type (status register – R).<br><br>000 – reserved (no error)<br><br>001 – CRC error<br><br>010 – data toggle mismatch<br><br>011 – endpoint sent STALL handshake<br><br>100 – no endpoint handshake (timeout)<br><br>101 – PID error (pid check=error or unknown PID)<br><br>110 – Data Overrun (too long packet – babble)<br><br>111 – Data Underrun (packet shorter than MaxPacketSize)           | R   | 000B  |
| 1:0    | Errcount[1..0] | Status register – Read only.<br><br>Transmission error counter. This counter is incremented on every transmission error (CRC error or no handshake).<br><br>It is cleared on every correct transmission or by microprocessor (writing resend bit).<br><br>In case of data underrun, data overrun and STALL error, counter is written by „11” at once and IN err interrupt is requested.                                | R   | 00B   |

### 11.1.6.27 Hcportctrl

HC Port Control Register

Offset = 0xd3

| Bit(s) | Name          | Description   | R/W | Reset  |
|--------|---------------|---|-----|--------|
| 7:6    | RstLengthCtrl | USB reset length control. Control register – Read/Write.<br><br>00 –signals 10 ms USB reset (testing purpose).<br>01 –signals 55 ms USB reset (default state).<br>10 –signals 1.6 ms USB reset (testing purpose).<br>11 – invalid.  | R/W | 01B    |
| 5      | PortRst       | Port Reset. Control register – Read/Write.<br><br>Microprocessor may set this bit to force reset signalling on the port. Reset signalling is fuse ng ly done as it is written in USB Specification. Microprocessor does not have to clear this bit after end of reset.  | R/W | 0B     |
| 4:0    | Testm[4..0]   | Port Test Control. Control register – Read/Write.<br><br>Microcontroller writes this register to select Test Mode (USB spec 7.1.20 Test Mode Support)<br><br>00000 – test mode disabled<br>----1 – Test_J<br>---10 – Test_K<br>--100 – Test_SE0_NAK<br>-1000 – Test_Packet<br><br>10000 – Test_Force_Enable (SOF packet sending independently whether peripheral is connected to USB po | R/W | 00000B |

### 11.1.6.28 Hcfrmnl

HC Frame Number Register Low

Offset = 0xd5

| Bit(s) | Name    | Description  | R/W | Reset |
|--------|---------|--------------|-----|-------|
| 7:0    | Hcfrmnl | Full speed : | R   | 00H   |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | The lower 8 bits of the frame counter.<br><br>High speed:<br><br>7:3: The lower 5 bits of the frame counter.<br><br>2:0: Microframe counter |  |  |
|--|--|---|--|--|

### 11.1.6.29 Hcfrmnh

HC Frame Number Register high

Offset = 0xd4

| Bit(s) | Name    | Description  | R/W | Reset |
|--------|---------|--|-----|-------|
| 7:0    | Hcfrmnh | The hcfrmnrh register contains the most significant bits of the frame counter. | R   | 00H   |

### 11.1.6.30 Hcfrmremainl

HC Frame remain Register Low

Offset = 0xd7

| Bit(s) | Name    | Description  | R/W | Reset |
|--------|---------|--|-----|-------|
| 7:0    | hcfrmrl | Full speed:<br><br>The hcfrmremainl register contains number of bytes which can be sent in current frame. Initial value of the hcfrmremainl register is 1500 and it is decremented every Full-Speed Byte time. Two low significant bits are always stuck to ‘0’ (hcfrmrm1 … hcfrmrm0 = “00”) (read only)<br><br>High speed:<br><br>The hcfrmremainl register contains number of bytes which can be sent in current microframe. Initial value of the hcfrmremainl register is 7500 and it is decremented every High-Speed Byte time. Seven low significant bits are always stuck to ‘0’ (hcfrmrm6 … hcfrmrm0 = “0000000”) (read only) | R   | 00H   |

### 11.1.6.31 Hcfrmremainh

HC Frame remain Register high

Offset = 0xd6

| Bit(s) | Name     | Description  | R/W | Reset |
|--------|----------|--|-----|-------|
| 7:4    | -        | Reserved   | R   | 0000B |
| 3:0    | hcfrmrmh | The hcfrmremainh register contains the most significant bits of the frame remain register. | R   | 0000B |

### 11.1.6.32 Hcep03errirq

HC 0 to 3 Error Interrupt Request Register

Offset = 0xd8

| Bit(s) | Name        | Description  | R/W | Reset |
|--------|-------------|--|-----|-------|
| 7:0    | HcEpxErrIrq | <p>The HC sets the xerrirq bit to ‘1’ when it receives an HC IN OR OUT data packet and detects transfer error condition. Possible transfer errors: STALL response from endpoint; no response from endpoint; invalid or unknown PID; CRC error; data toggle mismatch; data overrun; data underrun</p> <p>Bit0: HC OUT 0 ERROR IRQ<br/>         Bit1: HC OUT 1 ERROR IRQ<br/>         Bit2: Reserved<br/>         Bit3: HC OUT 3 ERROR IRQ<br/>         Bit4: HC IN 0 ERROR IRQ<br/>         Bit5: Reserved<br/>         Bit6: HC IN 2 ERROR IRQ<br/>         Bit7: Reserved</p> <p>- Write a ‘1’ to this bit to clear the interrupt request</p> | R/W | 00H   |

### 11.1.6.33 Hcep03errien

HC 0 to 3 Error Interrupt Enable Register

Offset = 0xd9

| Bit(s) | Name        | Description   | R/W | Reset |
|--------|-------------|---|-----|-------|
| 7:0    | HcEpxErrIen | <p>HC x error interrupt enable. X – number of endpoint.</p> <p>Bit0: HC OUT 0 ERROR IRQ</p> | R/W | 00H   |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | Bit1: HC OUT 1 ERROR IRQ<br>Bit2: Reserved<br>Bit3: HC OUT 3 ERROR IRQ<br>Bit4: HC IN 0 ERROR IRQ<br>Bit5: Reserved<br>Bit6: HC IN 2 ERROR IRQ<br>Bit7: Reserved<br>- - When inxien bit is '0', the interrupt request bit for the HC x error (inxerrirq) is ignored. |  |  |
|--|--|--|--|--|

### 11.1.6.34 Hcin0maxpck

HC IN 0 Max Packet Size Register

Offset = 0xe4

| Bit(s) | Name      | Description   | R/W | Reset    |
|--------|-----------|---|-----|----------|
| 7      | -         | Reserved  | R   | 0B       |
| 6:0    | HcIn0MaxP | The maximum packet size for the IN 0 endpoint is reported during the USB enumeration procedure. During the USB enumeration the microprocessor should write the correct value to the hcin0maxpck register. The HC uses this value for the USB transaction scheduling. If a peripheral sends a packet larger than the value written into the hcin0maxpck register, the hcxiterrrq interrupt is requested. This value cannot be larger than the implemented size of the endpoint OUT0. | R/W | 0000000B |

### 11.1.6.35 Hcout1maxpckl

HC OUT 1 max packet low Register

Offset = 0xea

| Bit(s) | Name        | Description  | R/W | Reset |
|--------|-------------|--|-----|-------|
| 7:0    | HcOut1MaxPL | The hcxiterrrq stores a low byte of the maximum packet size for the HC out1 endpoint | R/W | 00H   |

### 11.1.6.36 Hcin2maxpckl

HC IN 2 max packet low Register

Offset = 0xe6

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | HcIn2MaxPL | The hcin2maxpckl stores a low byte of the maximum packet size for the HC in2 endpoint | R/W | 00H   |

### 11.1.6.37 Hcout3maxpck

HC OUT 3 max packet Register

Offset = 0xe7

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | HcOut3MaxP | The hcout3maxpckl stores the maximum packet size for the HC out3 endpoint | R/W | 00H   |

### 11.1.6.38 Hcout1maxpckh

HC OUT 1 max packet high register

Offset = 0xe9

| Bit(s) | Name        | Description                                | R/W | Reset  |
|--------|-------------|--|-----|--------|
| 7:3    | -           | Reserved                                   | R   | 00000B |
| 2:0    | HcOut1MaxPH | HC out 1 maximum packet high size register | R/W | 000B   |

### 11.1.6.39 Hcin2maxpckh

HC IN 2 max packet high register

Offset = 0xe5

| Bit(s) | Name       | Description                               | R/W | Reset  |
|--------|------------|---|-----|--------|
| 7:3    | -          | Reserved                                  | R   | 00000B |
| 2:0    | HcIn2MaxPH | HC in 2 maximum packet high size register | R/W | 000B   |

### 11.1.6.40 Out0bc\_hcin0bc

Endpoint 0 OUT Byte Count Register

Offset = 0x97

| Bit(s) | Name     | Description  | R/W | Reset    |
|--------|----------|--|-----|----------|
| 7      | -        | Reserved   | R   | 0B       |
| 6:0    | Ep0Outbc | <p>The out0bc register contains the number of bytes sent during the last OUT transfer from the host to an OUT 0 endpoint. The out0bc is a read-only register, updated by the <b>AOTG</b> hardware.</p> <p>This register is used also in Host Mode. The meaning of the out0bc register in Host Mode is the same as in Peripheral Mode but the name of the register is changed to hcin0bc. The hcin0bc register contains the number of bytes sent during the last IN transfer from the USB peripheral device to an IN 0 endpoint. The hcin0bc is a read-only register, updated by the <b>AOTG</b> hardware</p> | R   | 0000000B |

### 11.1.6.41 In0bc\_hcout0bc

Endpoint 0 in Byte Count Register

Offset = 0x98

| Bit(s) | Name    | Description   | R/W | Reset    |
|--------|---------|---|-----|----------|
| 7      | -       | Reserved  | R   | 0B       |
| 6:0    | Ep0Inbc | <p>After having loaded the IN 0 endpoint buffer, the microprocessor should write the number of loaded bytes to the in0bc register. Writing to the in0bc register arms the IN 0 endpoint (the ep0cs.2 (inbsy) bit is automatically set to 1). When the host sends the IN token for the IN 0 endpoint and the inbsy bit is set, the <b>AOTG</b> will respond with an in0bc size data packet.</p> <p>This register is used also in Host Mode. The meaning of the in0bc register in Host Mode is the same as in Peripheral Mode but name of the register is changed for hcout0bc. After having loaded the OUT 0 (hcep0outdat) endpoint buffer, the microprocessor should write the number of loaded bytes to the hcout0bc register. Writing to the hcout0bc register arms the OUT 0 endpoint (the hcep0cs.2 (hcoutbsy) bit is automatically set to 1). When the hcoutbsy bit is set, the HC logic</p> | R/W | 0000000B |

|  |  |                                      |  |  |
|--|--|--------------------------------------|--|--|
|  |  | will schedule the OUT 0 transaction. |  |  |
|--|--|--------------------------------------|--|--|

### 11.1.6.42 Ep0cs\_hcep0cs

Endpoint 0 Control/Status Register

Offset = 0x99

| Bit(s) | Name           | Description  | R/W | Reset |
|--------|----------------|--|-----|-------|
| 7      | -              | Reserved   | R   | 0B    |
| 6      | hcsettoggle    | Peripheral Mode operation –Not used<br><br>Host Mode operation –<br><br>The hcsettoggle is write-only bit. It always returns ‘0’ during reading. To set toggle bits of the HC IN0 and HC OUT 0 endpoints the microprocessor writes ‘1’ to the hcsettoggle bit.   | R/W | 0B    |
| 5      | hcclr toggle   | Peripheral Mode operation –Not used<br><br>Host Mode operation –<br><br>The hcclr toggle is write-only bit. It always returns ‘0’ during reading. To clear toggle bits of the HC IN0 and HC OUT 0 endpoints the microprocessor writes ‘1’ to the hcclr toggle bit.   | R/W | 0B    |
| 4      | hcset          | Peripheral Mode operation –Not used<br><br>Host Mode operation –<br><br>The hcset is write-only bit. It always returns ‘0’ during reading. To initialize the CONTROL transfer the microprocessor writes ‘1’ to the hcset bit. When the CONTROL transfer is initialized the Host Controller sends SETUP token instead OUT token during nearest OUT transfer from HC OUT0 endpoint.<br><br>When the microprocessor writes ‘1’ to the hcset bit, the endpoint 0 busy flags (hcinbsy and hcoutbsy) and endpoint 0 toggle bits are cleared. For this reason the microprocessor should first write hcset bit and then arm OUT0 endpoint to initialize data transfer. | R/W | 0B    |
| 3      | Outbsy_hcinbsy | Peripheral Mode operation –<br><br>OUT 0 endpoint busy bit.<br><br>If outbsy=’1’ the AOTG hardware takes control of  | R   | 1B    |

|   |                |   |   |    |
|---|----------------|---|---|----|
|   |                | <p>the OUT 0 endpoint buffer. The endpoint 0 buffer is ready to receive data from host. The microprocessor should not access the OUT 0 endpoint buffer when outbsy='1'.</p> <p>If outbsy='0' the microprocessor takes control of the OUT 0 endpoint buffer.</p> <p>Outbsy is a read-only bit that is automatically set when a SETUP token arrives. The outbsy bit is automatically cleared when the AOTG receives an error-free OUT data packet. The microprocessor sets this bit by writing a dummy value to the out0bc register.</p> <p>Host Mode operation –</p> <p>HC IN 0 endpoint busy bit.</p> <p>If hcinbsy='1' the HC takes control of the HC IN 0 (hcep0indat) endpoint buffer and the endpoint 0 buffer is scheduled for transaction. The microprocessor should not access the HC IN 0 (hcep0indat) endpoint buffer when hcinbsy='1'.</p> <p>The hcinbsy is a read-only bit that is cleared by the HC when the <b>AOTG</b> enters the Host Mode or when HC IN 0 data packet is received without errors. The hcinbsy bit is set by the HC when the microprocessor writes a dummy value to the hcin0bc register.</p> |   |    |
| 2 | Inbsy_hcoutbsy | <p>Peripheral Mode operation –</p> <p>IN 0 endpoint busy bit.</p> <p>If inbsy='1' the <b>AOTG</b> hardware takes control of the IN 0 endpoint buffer. The IN 0 buffer is armed to send data to host. The microprocessor should not access the IN 0 endpoint when inbsy='1'.</p> <p>If inbsy='0' the microprocessor takes control of the IN 0 endpoint buffer.</p> <p>Inbsy is a read-only bit that is automatically cleared when a SETUP token arrives. The microprocessor sets this bit by reloading the in0bc register.</p> <p>Host Mode operation –</p> <p>HC OUT 0 endpoint busy bit.</p>   | R | 0B |

|   |       |  |     |    |
|---|-------|--|-----|----|
|   |       | <p>If hcoutbsy='1' the HC hardware takes control of the HC OUT 0 endpoint buffer and the HC OUT 0 buffer (hcep0outdat) is scheduled to send data to peripheral device. The microprocessor should not access the HC OUT 0 endpoint buffer (hcep0outdat) when hcoutbsy='1'.</p> <p>Hcoutbsy is a read-only bit that is automatically cleared by the HC when the <b>AOTG</b> enters the Host Mode or when HC OUT 0 data packet is successfully transmitted to peripheral device. The HC hardware sets this bit when the microprocessor loads the hc0bc.</p> |     |    |
| 1 | hsnak | <p>Peripheral Mode operation –</p> <p>If the hsnak bit is set to '1' (and the stall bit – ep0cs[0] must be set to '0'), the <b>AOTG</b> responds with a NAK handshake for every packet in the status stage.</p> <p>The hsnak bit is automatically set to '1' when a SETUP token arrives.</p> <p>The microprocessor clears the hsnak bit by writing a '1' to it. When the hsnak is cleared (and the stall bit – ep0cs[0] is '0'), the <b>AOTG</b> acknowledges the status stage of the control transfer.</p> <p>Host Mode operation –</p> <p>Not used</p> | R/W | 0B |
| 0 | stall | <p>Peripheral Mode operation –</p> <p>Endpoint 0 stall bit.</p> <p>If the ep0stall bit is set to '1', the <b>AOTG</b> sends a STALL handshake for any IN or OUT token to the endpoint 0 during the data or status stages of control transfer.</p> <p>Ep0stall is automatically cleared when a SETUP token arrives. The microprocessor sets this bit by writing a '1' to it or clears this bit by writing a '0' to it.</p> <p>Host Mode operation –</p> <p>Not used</p>   | R/W | 0B |

### 11.1.6.43 In1bcl\_hcout1bcl

Endpoint 1 IN Byte Count Low Register

Offset = 0x9b

| Bit(s) | Name   | Description                          | R/W | Reset |
|--------|--------|--------------------------------------|-----|-------|
| 7:0    | Ep1bcl | Ep1 endpoint byte count low register | R   | 00H   |

### 11.1.6.44 Out2bcl\_hcin2bcl

Endpoint 2 OUT Byte Count Low Register

Offset = 0x9f

| Bit(s) | Name   | Description                          | R/W | Reset |
|--------|--------|--------------------------------------|-----|-------|
| 7:0    | Ep2bcl | Ep2 endpoint byte count low register | R   | 00H   |

### 11.1.6.45 In3bc\_hcout3bc

Endpoint 3 IN Byte Count Register

Offset = 0xa4

| Bit(s) | Name  | Description                      | R/W | Reset |
|--------|-------|----------------------------------|-----|-------|
| 7:0    | Ep3bc | Ep3 endpoint byte count register | R   | 00H   |

### 11.1.6.46 In1bch\_hcout1bch

Endpoint 1 IN Byte Count HIGH Register

Offset = 0x9a

| Bit(s) | Name   | Description                           | R/W | Reset |
|--------|--------|---------------------------------------|-----|-------|
| 7:4    | -      | Reserved                              | R   | 0000B |
| 3:0    | Ep1bch | Ep1 endpoint byte count high register | R   | 0000B |

### 11.1.6.47 Out2bch\_hcin2bch

Endpoint 2 OUT Byte Count HIGH Register

Offset = 0x9e

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |        |                                       |   |       |
|-----|--------|---------------------------------------|---|-------|
| 7:4 | -      | Reserved                              | R | 0000B |
| 3:0 | Ep2bcH | Ep2 endpoint byte count high register | R | 0000B |

### 11.1.6.48 In1cs\_hcout1cs

Endpoint 1 IN Control And Status Register

Offset = 0x9d

| Bit(s) | Name             | Description   | R/W | Reset |
|--------|------------------|---|-----|-------|
| 7:5    | -                | Reserved  | R   | 000B  |
| 4      | Autoin_hcautoout | Peripheral Mode operation –<br>Auto-IN bit.<br>When autoin='1', the IN buffer is armed automatically when becomes full. When autoin='0', the IN buffer can be armed only by writing in1cs register.<br>Setting or clearing the autoin bit is done using the fifoctrl register.<br>Host Mode operation –<br>Auto-OUT bit.<br>When hcautoout='1', the HC OUT 1 buffer is armed automatically when becomes full. When hcautoout='0', the HC OUT 1 buffer can be armed only by writing hcout1cs register.<br>Setting or clearing the hcautoout bit is done using the fifoctrl register. | R   | 0B    |
| 3:2    | Npak_hcnpak      | Peripheral Mode operation –<br>Npak bits. (Read only)<br>The number of empty sub-buffers of the multiple buffered endpoint.<br>Host Mode operation –<br>Npak bits. (Read only)<br>The number of empty sub-buffers of the multiple buffered HC OUT 1 endpoint.   | R   | 11B   |
| 1      | Busy_hcbusy      | Peripheral Mode operation –<br>IN 1 endpoint busy bit.  | R/W | 0B    |

|   |     |  |   |    |
|---|-----|--|---|----|
|   |     | <p>If busy='1' the AOTG takes control of the IN 1 endpoint buffer.</p> <p>If busy='0' the microprocessor takes control of the IN 1 endpoint buffer.</p> <p>A '1' to '0' transition of the busy bit generates an interrupt request for the IN 1 endpoint. The microprocessor sets the busy bit by reloading the in1cs register with a dummy value.</p> <p>Host Mode operation –</p> <p>HC OUT 1 endpoint busy bit.</p> <p>If hcbusy='1' the OUT 1 endpoint buffer is scheduled to send to peripheral device.</p> <p>If hcbusy='0' the microprocessor takes control of the HC OUT 1 endpoint buffer.</p> <p>Hcbusy is a read-only bit that is automatically cleared by the HC when the AOTG enters the Host Mode or when HC OUT 1 data packet is successfully transmitted to peripheral device. The HC hardware sets this bit when the microprocessor writes the hcout1cs register.</p> <p>A '1' to '0' transition of the hcbusy bit generates an interrupt request for the HC OUT 1 endpoint.</p> |   |    |
| 0 | err | <p>Peripheral Mode operation –</p> <p>Data sequence error for the ISO IN endpoint.<br/>(Read only)</p> <p>The data sequence error occurs when:</p> <ul style="list-style-type: none"> <li>- The AOTG has not received the correct IN token (for ISO 1 endpoint) during last frame (full-speed mode)</li> <li>- The AOTG has not received the expected number of correct IN tokens (for ISO 1 endpoint) during last microframe (high-speed mode)</li> </ul> <p>The err bit is updated by the AOTG hardware once per frame (microframe) when the host sends the SOF packet.</p> <p>Host Mode operation –</p> <p>Err=1 – AOTG couldn't send programmed</p>  | R | 0B |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | number of packets because microprocessor didn't arm proper amount of buffers with data to send. |  |  |
|--|--|---|--|--|

### 11.1.6.49 Out2cs\_hcin2cs

Endpoint 2 OUT Control And Status Register

Offset = 0xa3

| Bit(s) | Name             | Description  | R/W | Reset |
|--------|------------------|--|-----|-------|
| 7:5    | -                | Reserved   | R   | 000B  |
| 4      | Autoin_hcautoout | <p>Peripheral Mode operation –</p> <p>Auto-OUT bit.</p> <p>When autoout='1', the OUT buffer is armed automatically when becomes empty. When the autoout='0', the OUT buffer can be armed only by writing the out2cs register. Setting or clearing this bit is accomplished using the fifoctrl register.</p> <p>Host Mode operation –</p> <p>Auto-IN bit.</p> <p>When hcautoin='1', the HC IN buffer is armed automatically when becomes empty. When the hcautoin='0', the HC IN buffer can be armed only by writing the hcin2cs register. Setting or clearing this bit is accomplished using the fifoctrl register</p> | R   | 0B    |
| 3:2    | Npak[_hcnpak]    | <p>Peripheral Mode operation –</p> <p>Npak bits. (Read only)</p> <p>Number of received data packets that are stored in the OUT 2 buffer memory.</p> <p>Host Mode operation –</p> <p>Npak bits. (Read only)</p> <p>Number of received data packets that are stored in the HC IN 2 buffer memory.</p>  | R   | 00B   |
| 1      | Busy_hcbusy      | <p>Peripheral Mode operation –</p> <p>OUT 2 endpoint busy bit.</p> <p>If busy='1' the AOTG takes control of the OUT 2 endpoint buffer.</p> <p>If busy='0' the microprocessor takes control of the</p>  | R/W | 1B    |

|   |     |   |   |    |
|---|-----|---|---|----|
|   |     | <p>OUT x endpoint buffer.</p> <p>If busy='1', the OUT 2 endpoint is empty and ready to receive the next data packet from the host. When busy='1', the microprocessor should not read the OUT 2 endpoint buffer. A '1' to '0' transition of the busy bit generates an interrupt request for the OUT 2 endpoint. The microprocessor sets the busy bit by reloading the out2cs register with a dummy value</p> <p>Host Mode operation –</p> <p>IN 2 endpoint busy bit.</p> <p>If busy='1' the HC hardware takes control of the HC IN 2 endpoint buffer.</p> <p>If busy='0' the microprocessor takes control of the HC IN 2 endpoint buffer.</p> <p>The hcbusy is a read-only bit that is cleared by the HC when the AOTG enters the Host Mode or when HC IN 2 data packet is received without errors. The hcinbsy bit is set by the HC when the microprocessor writes a dummy value to the hcinxcs register.</p> <p>If busy='1', the HC IN 2 endpoint is empty and scheduled to receive the next data packet from the peripheral device. When busy='1', the microprocessor should not read the HC IN 2 endpoint buffer. A '1' to '0' transition of the busy bit generates an interrupt request for the HC IN 2 endpoint.</p> |   |    |
| 0 | err | <p>Peripheral Mode operation –</p> <p>Data sequence error for ISO endpoints. (Read only)</p> <p>Err='1' – data sequence error or missing data during single frame (microframe)</p> <p>ISO OUT error occurs when:</p> <ul style="list-style-type: none"> <li>- The data packet received during a single frame (microframe) was corrupted</li> <li>- The host sent a data packet but the OUT data buffer was full thus unable to accept the data packet</li> </ul>  | R | 0B |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | <p>The err bit is updated by the AOTG hardware once per frame (microframe).</p> <p>Host Mode operation –</p> <p>Err=1 – number of ISO transactions in the previous microframe was lower than value in hcin2con(isod1, isod0) register (high-speed mode). It means that peripheral sent DATA0 in the first packet and there were more than one programmed packet or microprocessor didn't arm proper number of packets or it was packet error.</p> |  |  |
|--|--|---|--|--|

### 11.1.6.50 In3cs\_hcout3cs

Endpoint 3 IN Control And Status Register

Offset = 0xa6

| Bit(s) | Name             | Description  | R/W | Reset |
|--------|------------------|--|-----|-------|
| 7:5    | -                | Reserved   | R   | 000B  |
| 4      | Autoin_hcautoout | <p>Peripheral Mode operation –</p> <p>Auto-IN bit.</p> <p>When autoin='1', the IN buffer is armed automatically when becomes full. When autoin='0', the IN buffer can be armed only by writing in3cs register.</p> <p>Setting or clearing the autoin bit is done using the fifoctrl register.</p> <p>Host Mode operation –</p> <p>Auto-OUT bit.</p> <p>When hcautoout='1', the HC OUT 3 buffer is armed automatically when becomes full. When hcautoout='0', the HC OUT 3 buffer can be armed only by writing hcout3cs register.</p> <p>Setting or clearing the hcautoout bit is done using the fifoctrl register.</p> | R   | 0B    |
| 3:2    | -                | Reserved   | R   | 00B   |
| 1      | Busy_hcbusy      | <p>Peripheral Mode operation –</p> <p>IN 3 endpoint busy bit.</p> <p>If busy='1' the AOTG takes control of the IN 3</p>  | R/W | 0B    |

|   |   |   |   |    |
|---|---|---|---|----|
|   |   | <p>endpoint buffer.</p> <p>If busy='0' the microprocessor takes control of the IN 3 endpoint buffer.</p> <p>A '1' to '0' transition of the busy bit generates an interrupt request for the IN 3 endpoint. The microprocessor sets the busy bit by reloading the in3cs register with a dummy value.</p> <p>Host Mode operation –</p> <p>HC OUT 3 endpoint busy bit.</p> <p>If hcbusy='1' the OUT 3 endpoint buffer is scheduled to send to peripheral device.</p> <p>If hcbusy='0' the microprocessor takes control of the HC OUT 3 endpoint buffer.</p> <p>Hcbusy is a read-only bit that is automatically cleared by the HC when the aotg enters the Host Mode or when HC OUT 3 data packet is successfully transmitted to peripheral device. The HC hardware sets this bit when the microprocessor writes the hcout3cs register.</p> <p>A '1' to '0' transition of the hcbusy bit generates an interrupt request for the HC OUT 3 endpoint.</p> |   |    |
| 0 | - | Reserved  | R | 0B |

### 11.1.6.51 In1ctrl\_hcout1ctrl

Endpoint 1 IN Control Register

Offset = 0x9c

| Bit(s) | Name | Description  | R/W | Reset |
|--------|------|--|-----|-------|
| 7      | val  | <p>Peripheral Mode operation –</p> <p>IN 1 endpoint valid bit.</p> <p>If val = '0' the IN 1 endpoint is disabled. The IN 1 endpoint is disabled. The AOTG ignores all transfers to the IN 1 endpoint.</p> <p>If val = '1' the IN 1 endpoint is enabled.</p> <p>The microprocessor writes a '1' to set this bit and writes a '0' to clear this bit.</p> | R/W | 0B    |

|     |             |  |     |     |
|-----|-------------|--|-----|-----|
|     |             | Host Mode operation –<br><br>Not used  |     |     |
| 6   | stall       | <p>Peripheral Mode operation –<br/><br/>IN 1 endpoint stall bit.</p> <p>If stall='1', the AOTG returns a STALL handshake for all requests to the IN 1 endpoint. The microprocessor writes a '1' to set this bit and writes a '0' to clear this bit.</p><br><p>Host Mode operation –<br/><br/>Not used</p>  | R/W | 0B  |
| 5:4 | Isod_hcisod | <p>Peripheral Mode operation –<br/><br/>For High-Speed mode and ISO IN endpoints only. These bits determine the number of packets per microframe. The microprocessor can write the in1con[5..4] bits to select the required number of packets.</p> <p>Isod1,isod0="00" – 1 ISO packet per microframe<br/>Isod1,isod0="01" – 2 ISO packets per microframe<br/>Isod1,isod0="10" – 3 ISO packets per microframe</p><br><p>Host Mode operation –<br/><br/>For high bandwidth isochronous transactions only. These bits determine the number of packets per microframe for HC OUT 1 endpoint (see description above). The microprocessor can write the hcout1con[5..4] bits to select the required number of packets. If hcoutsod1, hcoutsod0 &gt; "00" then HC OUT 1 endpoint is scheduled as high bandwidth ISO endpoint.</p> | R/W | 00B |
| 3:2 | Type_Hctype | <p>Peripheral Mode operation –<br/><br/>Every IN 1 endpoint can be configured as isochronous, interrupt or bulk type. The microprocessor can write the in1con[3..2] bits to select the required endpoint type.</p> <p>Type1,type0 = "10" – bulk endpoint<br/>Type1,type0 = "01" – isochronous endpoint</p>   | R/W | 10B |

|     |           |   |     |     |
|-----|-----------|---|-----|-----|
|     |           | Type1,type0 = “11” – interrupt endpoint   |     |     |
| 1:0 | Buf_hdbuf | <p>Peripheral Mode operation –</p> <p>Every IN 1 buffer can be programmed as single, double, triple or quad buffered. The microprocessor can write the in1con[1..0] bits to select the required buffering.</p> <p>Buf1,buf0 = “00” – single buffering</p> <p>Buf1,buf0 = “01” – double buffering</p> <p>Buf1,buf0 = “10” – triple buffering</p> <p>Buf1,buf0 = “11” – quad buffering</p> <p>Host Mode operation –</p> <p>Every HC OUT x buffer can be programmed as single, double, triple or quad buffered. The microprocessor can write the hcout1con[1..0] bits to select the required buffering. See Peripheral Mode Operation above.</p> | R/W | 11B |

### 11.1.6.52 Out2ctrl\_hcin2ctrl

Endpoint 2 OUT Control Register

Offset = 0xa2

| Bit(s) | Name  | Description   | R/W | Reset |
|--------|-------|---|-----|-------|
| 7      | val   | <p>Peripheral Mode operation –</p> <p>OUT 2 endpoint valid bit.</p> <p>If val = ‘0’ the OUT 2 endpoint is disabled. The AOTG ignores all transfers to the OUT 2 endpoint.</p> <p>If val = ‘1’ the OUT 2 endpoint is enabled.</p> <p>The microprocessor writes a ‘1’ to set this bit and writes a ‘0’ to clear this bit.</p> <p>Host Mode operation –</p> <p>HC IN2 endpoint valid bit. See description above.</p> | R/W | 0B    |
| 6      | stall | <p>Peripheral Mode operation –</p> <p>OUT2 endpoint stall bit.</p> <p>If stall=’1’, the AOTG returns a STALL handshake for all requests to the OUT 2 endpoint.</p>  | R/W | 0B    |

|     |             |   |     |     |
|-----|-------------|---|-----|-----|
|     |             | <p>The microprocessor writes a ‘1’ to set this bit and writes a ‘0’ to clear this bit.</p> <p>Host Mode operation –</p> <p>Not used</p>   |     |     |
| 5:4 | Hcinisod    | <p>Peripheral Mode operation –</p> <p>Not used</p> <p>Host Mode operation –</p> <p>For high bandwidth isochronous transactions only. These bits determine the number of packets per microframe for HC IN 2 endpoint.</p> <p>Hcinisod1, hcinisod0=“00” – 1 ISO packet per microframe</p> <p>hcinisod1, hcinisod0=“01” – 2 ISO packets per microframe</p> <p>hcinisod1, hcinisod0=“10” – 3 ISO packets per microframe</p> <p>The microprocessor can write the hcin2con[5..4] bits to select the required number of packets. If hcinisod1, hcinisod0 &gt; “00” then IN 2 endpoint is scheduled as high bandwidth ISO endpoint.</p> | R/W | 00B |
| 3:2 | Type_Hctype | <p>Peripheral Mode operation –</p> <p>Every OUT 2 endpoint can be configured as isochronous, interrupt or bulk type. The microprocessor can write the out2con[3..2] bits to select the required endpoint type.</p> <p>Type1, type0 = “10” – bulk endpoint</p> <p>Type1, type0 = “11” – interrupt endpoint</p> <p>Type1, type0 = “01” – isochronous endpoint</p> <p>Host Mode operation –</p> <p>Every HC IN 2 endpoint can be configured as isochronous, interrupt or bulk type. The microprocessor can write the hcin2con[3..2] bits to select the required endpoint type. (see Peripheral Mode operation above).</p>          | R/W | 10B |
| 1:0 | Buf_hdbuf   | <p>Peripheral Mode operation –</p> <p>Every OUT 2 buffer can be programmed as single, double, triple buffered or quad buffered. The</p>   | R/W | 11B |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | <p>microprocessor can write the out2con(1..0) bits to select the required buffering.</p> <p>Buf1,buf0 = “00” – single buffering</p> <p>Buf1,buf0 = “01” – double buffering</p> <p>Buf1,buf0 = “10” – triple buffering</p> <p>Buf1,buf0 = “11” – quad buffering</p> <p>Host Mode operation –</p> <p>Every HC IN 2 buffer can be programmed as single, double, triple or quad buffered. The microprocessor can write the hcin2con(1..0) bits to select the required buffering. (see Peripheral Mode operation above).</p> |  |  |
|--|--|---|--|--|

### 11.1.6.53 In3ctrl\_hcout3ctrl

Endpoint 3 IN Control Register

Offset = 0xa5

| Bit(s) | Name  | Description   | R/W | Reset |
|--------|-------|---|-----|-------|
| 7      | val   | <p>Peripheral Mode operation –</p> <p>IN 3 endpoint valid bit.</p> <p>If val = ‘0’ the IN 3 endpoint is disabled. The AOTG ignores all transfers to the IN 3 endpoint.</p> <p>If val = ‘1’ the IN 3 endpoint is enabled.</p> <p>The microprocessor writes a ‘1’ to set this bit and writes a ‘0’ to clear this bit.</p> <p>Host Mode operation –</p> <p>Not used.</p> | R/W | 0B    |
| 6      | stall | <p>Peripheral Mode operation –</p> <p>IN 3 endpoint stall bit.</p> <p>If stall=’1’, the AOTG returns a STALL handshake for all requests to the IN 3 endpoint.</p> <p>The microprocessor writes a ‘1’ to set this bit and writes a ‘0’ to clear this bit.</p> <p>Host Mode operation –</p> <p>Not used</p>   | R/W | 0B    |

|     |             |   |     |     |
|-----|-------------|---|-----|-----|
| 5:4 | -           | Reserved  | R   | 00B |
| 3:2 | Type_Hctype | Peripheral Mode operation –<br><br>Every IN 3 endpoint can be configured as interrupt or bulk type. The microprocessor can write the in3con[3..2] bits to select the required endpoint type.<br><br>Type1,type0 = “10” – bulk endpoint<br><br>Type1,type0 = “11” – interrupt endpoint | R/W | 10B |
| 1:0 | -           | Reserved  | R   | 00B |

### 11.1.6.54 Setupdat0

Setup data Register0

Offset = 0xbb

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | Setupdat0 | The setupdat is the 8 bytes SETUP from the latest CONTROL transfer. The microprocessor has read-only access to this memory. | R   | 00H   |

### 11.1.6.55 Setupdat1

Setup data Register1

Offset = 0xbc

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | Setupdat1 | The setupdat is the 8 bytes SETUP from the latest CONTROL transfer. The microprocessor has read-only access to this memory. | R   | 00H   |

### 11.1.6.56 Setupdat2

Setup data Register2

Offset = 0xbd

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | Setupdat2 | The setupdat is the 8 bytes SETUP from the latest CONTROL transfer. The microprocessor has read-only access to this memory. | R   | 00H   |

### 11.1.6.57 Setupdat3

Setup data Register3

Offset = 0xbf

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | Setupdat3 | The setupdat is the 8 bytes SETUP from the latest CONTROL transfer. The microprocessor has read-only access to this memory. | R   | 00H   |

### 11.1.6.58 Setupdat4

Setup data Register4

Offset = 0xc1

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | Setupdat4 | The setupdat is the 8 bytes SETUP from the latest CONTROL transfer. The microprocessor has read-only access to this memory. | R   | 00H   |

### 11.1.6.59 Setupdat5

Setup data Register5

Offset = 0xc2

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | Setupdat5 | The setupdat is the 8 bytes SETUP from the latest CONTROL transfer. The microprocessor has read-only access to this memory. | R   | 00H   |

### 11.1.6.60 Setupdat6

Setup data Register6

Offset = 0xc3

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | Setupdat6 | The setupdat is the 8 bytes SETUP from the latest CONTROL transfer. The microprocessor has read-only access to this memory. | R   | 00H   |

### 11.1.6.61 Setupdat7

Setup data Register7

Offset = 0xc4

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | Setupdat7 | The setupdat is the 8 bytes SETUP from the latest CONTROL transfer. The microprocessor has read-only access to this memory. | R   | 00H   |

### 11.1.6.62 Fifoldat

FIFO 1 Data Register

Offset = 0xa7

| Bit(s) | Name     | Description          | R/W | Reset |
|--------|----------|----------------------|-----|-------|
| 7:0    | Fifoldat | FIFO 1 data register | R/W | -     |

### 11.1.6.63 Fifo2dat

FIFO 2 Data Register

Offset = 0xa9

| Bit(s) | Name     | Description          | R/W | Reset |
|--------|----------|----------------------|-----|-------|
| 7:0    | Fifo2dat | FIFO 2 data register | R/W | -     |

### 11.1.6.64 Fifo3dat

FIFO 3 Data Register

Offset = 0xaa

| Bit(s) | Name     | Description          | R/W | Reset |
|--------|----------|----------------------|-----|-------|
| 7:0    | Fifo3dat | FIFO 3 data register | R/W | -     |

### 11.1.6.65 Ep0indata

Ep0 in Data Register

Offset = 0xab

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |           |   |     |   |
|-----|-----------|---|-----|---|
| 7:0 | Ep0indata | Peripheral mode:<br>IN 0 endpoint data port<br><br>Host mode:<br>HC OUT 0 data port | R/W | - |
|-----|-----------|---|-----|---|

### 11.1.6.66 Ep0outdata

Ep0 out Data Register

Offset = 0xac

| Bit(s) | Name       | Description  | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | Ep0outdata | Peripheral mode:<br>OUT 0 endpoint data port<br><br>Host mode:<br>HC IN 0 endpoint data port | R/W | -     |

### 11.1.6.67 Ep03irq

Endpoint 0 to 3 Interrupt Request Register

Offset = 0xc5

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7:0    | Epxirq | Peripheral Mode operation –<br><br>For out endpoint<br><br>- The <b>AOTG</b> sets the epxirq bit to ‘1’ when it receives an error free OUT data packet to the endpoint x, and the “busy” bit (outxcs[1]) changes from ‘1’ to ‘0’ (all sub-buffers are empty and one sub-buffer is becoming full).<br><br>For In endpoint<br><br>- The <b>AOTG</b> sets the inxirq bit to ‘1’ when it transmits an IN x data packet, receives an ACK from the host, and the “busy” bit (inxcs[1]) changes from ‘1’ to ‘0’ (all sub-buffers are full and one sub-buffer is becoming empty).<br><br>Write a ‘1’ to this bit to clear the interrupt request<br><br>Host Mode operation – | R/W | 00H   |

|  |   |  |
|--|---|--|
|  | <p>For HC in endpoint</p> <ul style="list-style-type: none"> <li>- The HC sets the hcinxirq bit to ‘1’ when it receives an error free IN data packet to the endpoint x, and the “busy” bit (hcinxcs[1]) changes from ‘1’ to ‘0’ (all sub-buffers are empty and one sub-buffer is becoming full).</li> </ul> <p>For HC out endpoint</p> <ul style="list-style-type: none"> <li>- The HC sets the hcoutxirq bit to ‘1’ when it transmits an OUT x data packet, receives an ACK from peripheral device, and the “busy” bit (hcoutxcs[1]) changes from ‘1’ to ‘0’ (all sub-buffers are full and one sub-buffer is becoming empty).</li> <li>- Write a ‘1’ to this bit to clear the interrupt request</li> </ul> <p>Bit 0: ep0outirq</p> <p>Bit 1: reserved</p> <p>Bit 2: ep2outirq</p> <p>Bit 3: reserved</p> <p>Bit 4: ep0inirq</p> <p>Bit 5: ep1inirq</p> <p>Bit 6: reserved</p> <p>Bit 7: ep3inirq</p> |  |
|--|---|--|

11.1.6.68 Ep03ien

## Endpoint 0 to 3 Interrupt Enable Register

Offset = 0xc6

| Bit(s) | Name            | Description  | R/W | Reset |
|--------|-----------------|--|-----|-------|
| 7:0    | Epxien_hcepxien | <p>Peripheral Mode operation –</p> <p>x endpoint interrupt enable</p> <p>When the epxien bit is ‘0’, the interrupt request bit for the x endpoint is ignored.</p> <p>Host Mode operation –</p> <p>HC OUT x endpoint interrupt enable</p> | R/W | 00H   |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | <p>When the hcexpien bit is ‘0’, the interrupt request bit for the HC x endpoint is ignored.</p> <p>Bit 0: ep0outien</p> <p>Bit 1: reserved</p> <p>Bit 2: ep2outien</p> <p>Bit 3: reserved</p> <p>Bit 4: ep0inien</p> <p>Bit 5: ep1inien</p> <p>Bit 6: reserved</p> <p>Bit 7: ep3inien</p> |  |  |
|--|--|--|--|--|

### 11.1.6.69 Usbirq\_hcusbirq

USB Interrupt Request Register

Offset = 0xad

| Bit(s) | Name          | Description  | R/W | Reset |
|--------|---------------|--|-----|-------|
| 7      | Con_disconIrq | <p>Connect/disconnect interrupt</p> <p>When device connect to host or disconnect from host, this bit will be set, Write a ‘1’ to clear this bit</p>  | R/W | 0B    |
| 6      | NTRIRQ        | If the device cannot I SOF from the host 4times, IRQ is 1. Write 1 to this bit will clear it.  | R/W | 1     |
| 5      | Hsirq_hcHSirq | <p>Peripheral Mode operation –</p> <p>USB high-speed mode interrupt request.</p> <ul style="list-style-type: none"> <li>- The AOTG sets this bit to a ‘1’ when it switches into high speed mode</li> </ul> <p>Write a ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>USB high-speed mode interrupt request.</p> <ul style="list-style-type: none"> <li>- The HC sets this bit to a ‘1’ when it switches into high speed mode</li> </ul> <ul style="list-style-type: none"> <li>- Write a ‘1’ to this bit to clear the interrupt request</li> </ul> | R/W | 0B    |
| 4      | RstIrq        | <p>Peripheral Mode operation –</p> <p>USB reset interrupt request</p> <ul style="list-style-type: none"> <li>- The AOTG sets this bit to a ‘1’ when it detects</li> </ul>  | R/W | 0B    |

|   |          |  |     |    |
|---|----------|--|-----|----|
|   |          | <p>start of a USB bus reset.</p> <p>Write a ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>USB reset interrupt request</p> <ul style="list-style-type: none"> <li>- The AOTG sets this bit to a ‘1’ when HC stops a USB bus reset ffuse n.</li> <li>- Write a ‘1’ to this bit to clear the interrupt request</li> </ul>  |     |    |
| 3 | SuspIrq  | <p>Peripheral Mode operation –</p> <p>USB suspend interrupt request</p> <ul style="list-style-type: none"> <li>- The AOTG sets this bit to a ‘1’ when it detects USB SUSPEND signaling</li> </ul> <p>Write a ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Not used</p>   | R/W | 0B |
| 2 | SutokIrq | <p>Peripheral Mode operation –</p> <p>SETUP token interrupt request</p> <ul style="list-style-type: none"> <li>- The AOTG sets this bit to a ‘1’ when it receives a SETUP token</li> </ul> <p>Write a ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Not used</p>  | R/W | 0B |
| 1 | SOFIrq   | <p>Peripheral Mode operation –</p> <p>Start-of-frame interrupt request</p> <ul style="list-style-type: none"> <li>- The AOTG sets this bit to a ‘1’ when it receives a SOF packet</li> </ul> <p>Write a ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Start Of Frame interrupt request.</p> <ul style="list-style-type: none"> <li>- The HC sets this bit when SOF packet is sent on the USB</li> <li>- Write a ‘1’ to this bit to clear the interrupt request</li> </ul> | R/W | 0B |
| 0 | SudavIrq | <p>Peripheral Mode operation –</p> <p>SETUP data valid interrupt request</p>   | R/W | 0B |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | <ul style="list-style-type: none"> <li>- The AOTG sets this bit to a ‘1’ when it receives an error free SETUP data packet</li> </ul> <p>Write a ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Not used</p> |  |  |
|--|--|---|--|--|

### 11.1.6.70 Usbien\_hcusbien

USB Interrupt enable Register

Offset = 0xae

| Bit(s) | Name          | Description  | R/W | Reset |
|--------|---------------|--|-----|-------|
| 7      | Con_disconlen | Connect/disconnect interrupt enable  | R/W | 0B    |
| 6      | NTRIEN        | <b>NTRIRQ enable</b><br>1:enable 0:disable   | R/W | 0     |
| 5      | Hsien_hcHSien | Peripheral Mode operation –<br>USB high speed mode interrupt enable<br>When the hspie bit is ‘0’, the hspeedir interrupt request is ignored.<br>Host Mode operation –<br>USB high speed mode interrupt enable<br>- When the hspie bit is ‘0’, the hcbspeedir interrupt request is ignored. | R/W | 0B    |
| 4      | RstIen        | Peripheral Mode operation –<br>USB reset interrupt enable<br>When the uresie bit is ‘0’, the interrupt request bit for USB reset is ignored.<br>Host Mode operation –<br>USB reset interrupt enable<br>- When the uresie bit is ‘0’, the interrupt request bit for USB reset is ignored.   | R/W | 0B    |
| 3      | SusPten       | Peripheral Mode operation –<br>USB suspend interrupt enable<br>When the suspie bit is ‘0’, the interrupt request bit for USB suspend is ignored.   | R/W | 0B    |

|   |          |  |     |    |
|---|----------|--|-----|----|
|   |          | Host Mode operation –<br>Not used  |     |    |
| 2 | SutokIen | Peripheral Mode operation –<br>SETUP token interrupt enable<br><br>When the sutokie bit is ‘0’, the interrupt request bit for SETUP token is ignored.<br><br>Host Mode operation –<br>Not used   | R/W | 0B |
| 1 | SOFlen   | Peripheral Mode operation –<br>Start-of-frame interrupt enable<br><br>When the sofie bit is ‘0’, the interrupt request bit for start-of-frame is ignored.<br><br>Host Mode operation –<br><br>USB Start Of Frame interrupt enable<br><br>- When the hcfrmie bit is ‘0’, the hcfrmnrirq interrupt request is ignored. | R/W | 0B |
| 0 | Sudavlen | Peripheral Mode operation –<br>SETUP data valid interrupt enable<br><br>When the sudavie bit is ‘0’, the interrupt request bit for SETUP data valid is ignored.<br><br>Host Mode operation –<br>Not used   | R/W | 0B |

### 11.1.6.71 Ep03tokirq

Endpoint 0 to 3 Token Interrupt Request Register

Offset = 0x198

| Bit(s) | Name        | Description  | R/W | Reset |
|--------|-------------|--|-----|-------|
| 7      | Ep3intokirq | Peripheral Mode operation –<br>Ep3 in token interrupt request<br><br>- The <b>AOTG</b> sets the Ep3intokirq bit to ‘1’ when received an in token.<br><br>Write ‘1’ to this bit to clear the interrupt request<br><br>Host Mode operation – | R/W | 0B    |

|   |               |  |     |    |
|---|---------------|--|-----|----|
|   |               | Not used   |     |    |
| 6 | -             | reserved   | R   | 0B |
| 5 | Ep1intokirq   | <p>Peripheral Mode operation –</p> <p>Ep1 in token interrupt request</p> <ul style="list-style-type: none"> <li>- The <b>AOTG</b> sets the Ep1intokirq bit to ‘1’ when received a in token.</li> </ul> <p>Write ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Not used</p>        | R/W | 0B |
| 4 | Ep0intokirq   | <p>Peripheral Mode operation –</p> <p>Ep0 in token interrupt request</p> <ul style="list-style-type: none"> <li>- The <b>AOTG</b> sets the Ep0intokirq bit to ‘1’ when received an in token.</li> </ul> <p>Write ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Not used</p>       | R/W | 0B |
| 3 | Ep2pingtokirq | <p>Peripheral Mode operation –</p> <p>Ep2 ping token interrupt request</p> <ul style="list-style-type: none"> <li>- The <b>AOTG</b> sets the Ep2pingtokirq bit to ‘1’ when received an ping token.</li> </ul> <p>Write ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Not used</p> | R/W | 0B |
| 2 | Ep2outtokirq  | <p>Peripheral Mode operation –</p> <p>Ep2 out token interrupt request</p> <ul style="list-style-type: none"> <li>- The <b>AOTG</b> sets the Ep2outtokirq bit to ‘1’ when received a out token.</li> </ul> <p>Write ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Not used</p>     | R/W | 0B |
| 1 | Ep0pingirq    | <p>Peripheral Mode operation –</p> <p>Ep0 ping token interrupt request</p> <ul style="list-style-type: none"> <li>- The <b>AOTG</b> sets the Ep0pingirq bit to ‘1’ when</li> </ul>   | R/W | 0B |

|   |              |   |     |    |
|---|--------------|---|-----|----|
|   |              | <p>received an pingtoken.</p> <p>Write ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Not used</p>  |     |    |
| 0 | Ep0outtokirq | <p>Peripheral Mode operation –</p> <p>Ep0 out token interrupt request</p> <p>- The AOTG sets the Ep0outtokirq bit to ‘1’ when received an out token.</p> <p>Write ‘1’ to this bit to clear the interrupt request</p> <p>Host Mode operation –</p> <p>Not used</p> | R/W | 0B |

### 11.1.6.72 Ep03tokien

Endpoint 0 to 3 Token Interrupt Request Enable Register

Offset = 0xc8

| Bit(s) | Name          | Description  | R/W | Reset |
|--------|---------------|--|-----|-------|
| 7      | Ep3intokien   | Peripheral Mode operation –<br>Ep3 in token interrupt request enable   | R/W | 0B    |
| 6      | -             | reserved   | R   | 0B    |
| 5      | Ep1intokien   | Peripheral Mode operation –<br>Ep1 in token interrupt request enable   | R/W | 0B    |
| 4      | Ep0intokien   | Peripheral Mode operation –<br>Ep0 in token interrupt request enable   | R/W | 0B    |
| 3      | Ep2pingtokien | Peripheral Mode operation –<br>Ep2 ping token interrupt request enable | R/W | 0B    |
| 2      | Ep2outtokien  | Peripheral Mode operation –<br>Ep2 out token interrupt request enable  | R/W | 0B    |
| 1      | Ep0pingien    | Peripheral Mode operation –<br>Ep0 ping token interrupt request enable | R/W | 0B    |
| 0      | Ep0outtokien  | Peripheral Mode operation –  | R/W | 0B    |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | Ep0 out token interrupt request enable |  |  |
|--|--|--|--|--|

### 11.1.6.73 IVECT

Interrupt Vector Register

Offset = 0xc9

| Bit(s) | Name | Description  | R/W | Reset    |
|--------|------|--|-----|----------|
| 7:1    | lv   | The ivect register contains the interrupt vector. When the <b>AOTG</b> generates an interrupt request by setting the usbintreq signal high, the ivect register is updated to indicate the source of the interrupt..<br><br>Host Mode Operation –<br><br>This register is used also in Host Mode. The hcivect register contains the interrupt vector..When the HC generates an interrupt request by setting the usbintreq signal high, the hcivect register is updated to indicate the source of the interrupt. | R   | 0000000B |
| 0      | -    | This bit always is zero  | R   | 0B       |

### 11.1.6.74 EPRST

Endpoint Reset Register

Offset = 0xca

| Bit(s) | Name    | Description   | R/W | Reset |
|--------|---------|---|-----|-------|
| 7      | -       | reserved  | R   | 0B    |
| 6      | Fiforst | Fifo reset bit.<br><br>To reset the FIFO buffer of the appropriate endpoint, the microprocessor should perform the following steps:<br><br>- select the appropriate endpoint by writing to the ep1, ep0 bits<br><br>- reset the FIFO buffer by writing the “0,1,0,X,X, X, ep1, ep0” value to the endprst register<br><br>Resetting the FIFO buffer causes all associated byte counters to be reset. After reset, the FIFO buffer and associated busy bit is in the default state, which is the same as after a hardware | R/W | 0B    |

|     |        |   |     |      |
|-----|--------|---|-----|------|
|     |        | reset.  |     |      |
| 5   | togrst | <p>Toggle reset bit.</p> <p>To clear the toggle bit of the appropriate endpoint, the microprocessor should perform the following steps:</p> <ul style="list-style-type: none"> <li>- select the appropriate endpoint by writing to the ep1, ep0 bits</li> <li>- clear the toggle bit by writing the “0,0,1,X,X, X, ep1, ep0” value to the endprst register</li> </ul>                     | R/W | 0B   |
| 4:2 | -      | reserved  | R   | 000B |
| 1:0 | Ep_num | <p>Endpoint number.</p> <p>Valid values are 0 to 3. The microprocessor writes the ep1 and ep0 bits to select the appropriate endpoint address.</p> <p>(ep1, ep0):</p> <ul style="list-style-type: none"> <li>(0,0) – all endpoints from 1 to 3 are addressed</li> <li>(0,1) – endpoint 1 addressed</li> <li>(1,0) – endpoint 2 addressed</li> <li>(1,1) – endpoint 3 addressed</li> </ul> | R/W | 00B  |

### 11.1.6.75 UsbCTRL\_STUS

USB Control And Status Register

Offset = 0xcb

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7      | -      | reserved  | R   | 0B    |
| 6      | discon | <p>Peripheral Mode operation –</p> <p>Software disconnect bit.</p> <p>The microprocessor can write this bit (a ‘1’ value) to force the “disconnect” state. In the “disconnect” state, an external USB2.0 transceiver disconnects pull-up resistors from the D+ and D- lines and the USB host detects device disconnection.</p> <p>Host Mode operation –</p> <p>Bit is not used in Host Mode. To force software disconnect in Host Mode the microprocessor</p> | R/W | 1B    |

|     |          |  |     |       |
|-----|----------|--|-----|-------|
|     |          | uses otgctrl(1) (abusdrop) bit.  |     |       |
| 5   | sigrsume | <p>Peripheral Mode operation –</p> <p>Remote wakeup bit.</p> <p>When the microprocessor sets this bit to ‘1’, the <b>AOTG</b> initializes remote wakeup ffuse ng.</p> <p>Host Mode operation –</p> <p>Bit is not used in Host Mode. To force resume signalling in Host Mode the microprocessor uses otgctrl(0) (busreq) bit.</p> | R/W | 0B    |
| 4:1 | -        | reserved   | R   | 0000B |
| 0   | hclsmode | <p>Peripheral Mode operation –</p> <p>Not used.</p> <p>Host Mode operation –</p> <p>Read only bit.</p> <p>‘1’ – Peripheral device works in LS mode</p> <p>‘0’ – Peripheral device works in FS or HS mode</p>   | R   | 0B    |

### 11.1.6.76 Fnaddr

Function Address Register

Offset = 0xce

| Bit(s) | Name   | Description  | R/W | Reset    |
|--------|--------|--|-----|----------|
| 7      | -      | reserved   | R   | 0B       |
| 6:0    | Fnaddr | <p>The fnaddr “function address” register contains function address that was sent by the USB host during the most recent Set_Address request. The <b>AOTG</b> responds only with its assigned address. The fnaddr is a read-only register.</p> <p>Host Mode Operation –</p> <p>This register is used also in Host Mode. The microprocessor can write to the hcfnaddr at any time. The hcfnaddr value is used for all token packets that are sent by the HC. The hcfnaddr value is copied to “addr” field of token packets.</p> | R/W | 0000000B |

### 11.1.6.77 Clkgate

Clock Gate Register

Offset = 0xcf

| Bit(s) | Name      | Description  | R/W | Reset  |
|--------|-----------|--|-----|--------|
| 7:1    | -         | RESERVED   | R   | 00000B |
| 0      | gosuspend | Set 1: SIE will force ffuse to 0 ,the otg will enter the suspend state, when the resume signal is received, this bit will auto cleared | W   | 0B     |

### 11.1.6.78 Fifoctrl

FIFO Control Register

Offset = 0xd2

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:6    | reserved | reserved  | R   | 00B   |
| 5      | fifoauto | FIFO auto bit.<br><br>The microprocessor uses the fifoauto bit to set or clear the “autoout” or “autoin” bit of the appropriate endpoint.   | W   | 0B    |
| 4:2    | -        | reserved  | R   | 000B  |
| 1:0    | Ep_num   | Endpoint number.<br><br>Valid values are 1 to 3. The microprocessor writes the ep1 and ep0 bits to select the appropriate endpoint address. | W   | 00B   |

### 11.1.6.79 FrmCNTL

USB Frame Counter Low Register

Offset = 0xcd

| Bit(s) | Name | Description             | R/W | Reset  |
|--------|------|-------------------------|-----|--------|
| 7:3    | FrL  | USB Frame Counter Low   | R   | 00000B |
| 2:0    | Mfr  | The micro frame counter | R   | 000B   |

### 11.1.6.80 FrmCNTH

USB Frame Counter High Register

Offset = 0xcc

| Bit(s) | Name | Description            | R/W | Reset   |
|--------|------|------------------------|-----|---------|
| 7:6    | -    | reserved               | R   | 00B     |
| 5:0    | FrH  | USB Frame Counter High | R   | 000000B |

### 11.1.6.81 USBEIRQ

USB external Interrupt request register

Offset = 0xeb

| Bit(s) | Name        | Description   | R/W | Reset |
|--------|-------------|---|-----|-------|
| 7      | usbintirq   | External USB interrupt request.<br>Write 1 will clear this bit. | R/W | 0B    |
| 6      | wakirq      | USB wakeup interrupt request                                    | R/W | 0B    |
| 5:4    | -           | reserved  | R   | 00B   |
| 3      | usbintirqen | External USB interrupt request enable                           | R/W | 0B    |
| 2      | wakirqen    | USB wakeup interrupt enable                                     | R/W | 0B    |
| 1:0    | -           | reserved  | R   | 00B   |

### 11.1.6.82 AUTONAKCTRL

auto nak control register

Offset = 0xec

| Bit(s) | Name      | Description   | R/W | Reset   |
|--------|-----------|---|-----|---------|
| 7:2    | -         | reserved  | R   | 000000B |
| 1      | ep2nakout | This bit is used for ep2 out only in device mode<br><br>Set 1: when ep2 out received a short packet, the SIE will send NAK for next ping or out token, until clear the short packet interrupt.<br><br>Set 0: quit the auto NAK mode | R/W | 0B      |
| 0      | -         | reserved  | R   | 0B      |

### 11.1.6.83 HCINCTRL

Host in control register

Offset = 0xed

| Bit(s) | Name        | Description   | R/W | Reset |
|--------|-------------|---|-----|-------|
| 7      | -           | reserved  | R   | 0B    |
| 6      | hcinendirq  | Host in end interrupt request<br>When host in token stop ,the hcinendirq will be set, write 1 to clear this bit.  | R/W | 0B    |
| 5      | -           | reserved  | R   | 0B    |
| 4      | hcinendien  | Set 1: enable the hcinendirq<br>Set 0: disable the hcinendirq   | R/W | 0B    |
| 3:2    | -           | reserved  | R   | 00B   |
| 1      | hcinstart   | Set 1: take host to start send in token according to hcincounter<br>Set 0: take host to stop send in token  | R/W | 0B    |
| 0      | hcinshpctrl | Set 1: When the device send a short packet to host, host will stop send in token continue and the hcinstart bit will be clear, until the hcinstart bit Is set again.<br>Set 0: host send in token until the received packets are equal of hcincounter | R/W | 0B    |

### 11.1.6.84 SHORTPCKIRQ

Short packets Interrupt request and enable register

Offset = 0xaf

| Bit(s) | Name           | Description  | R/W | Reset |
|--------|----------------|--|-----|-------|
| 7:6    | -              | reserved   | R   | 00B   |
| 5      | ep2shortpckirq | If the ep2shortpckien is 1, this bit will be set 1 when ep2 out received a short packet.<br>If the ep2shortpckien is 0, this bit is disable<br>Write 1 to clear this bit | R/W | 0B    |
| 4:2    | -              | reserved   | R   | 000B  |
| 1      | ep2shortpckien | Set 1: enable the ep2 short packets interrupt  | R/W | 0B    |

|   |   |  |   |    |
|---|---|--|---|----|
|   |   | Clear 0: disable ep2 short packets interrupt |   |    |
| 0 | - | reserved                                     | R | 0B |

### 11.1.6.85 DBGMODE

debug mode register

Offset = 0xee

| Bit(s) | Name    | Description   | R/W | Reset |
|--------|---------|---|-----|-------|
| 7:5    | -       | reserved  | R   | 000B  |
| 4      | dbgout  | Debug mode output enable  | R/W | 0B    |
| 3:0    | dbgmode | <p>Debug mode control.</p> <p>Normally, don't write those bits with non zero value.</p> <p>Usbdbgo[31:0] output select.</p> <p>0000:{utmidata[15:0],vbusvalid_sie,utmitxvalid[0],utmrxvalid[0],utmivalidh,utmixtready,utmrxactive,utmrxerror,utmilinestate[1:0],utmiopmode[1:0],utmitemselect,utmixcvrselect[1:0],utmisuspendm,clkusb}</p> <p>0001:{utmixtvalid[0],utmrxvalid[0],utmivalidh,utmixtready,utmrxactive,utmrxerror,utmilinestate[1:0],utmispendm,clkusb,utmiopmode[1:0],utmitemselect,utmixcvrselect[1:0],utmppulldown,utmiddmpulldown,utmidischrgvbus,utmichrgvbus,utmidrvvbus,utmivalid,utmibvalid,utmisessend,utmidata[3:0],utmihostdiscon,vbusvalid_sie,utmivbusvalid,utmispendm,clkusb}</p> <p>0010:{ uint0b, sfroe, sfrwe, sfrdataout, sfrdatain, sfraddr[11:0], cpu_clk }</p> <p>0011:{usb_ack, usb_mreq, usb_dir, mrdata, maddr[9:0], mwdata, dmaclk, mrd, mwr}</p> <p>0100:{ramaccsizerd,ramaccsizerdnxt ,ramaccsizewr ,ramaddr, ramrd, ramwr ,ramdata[15:0]}</p> <p>0101:{indataval, rd_ram, wr_ram, usbaddr[11:2], usbdata[15:0]}</p> <p>0110:{ haddr, hrdata[21:0]}</p> | R/W | 0000B |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | 0111:{vstatus[7:0],usbivect[6:0],otgstate[3:0],downstrstate[3:0],upstrstate[4:0],utmisuspendm,utmilinestate[1:0],1'b0}<br><br>1000:{datain}<br><br>1001:{dataout}<br><br>1010:{uram_addr,ce,wen,oen,byte_en,usbaddr,vload,1'b0}<br><br>1011:{softid,softid_en,softvbus,softvbus_en,dmpuen,dppuen,dmpddis,dppddis,standby1,standby2,SIRCV,clk32k_scan,u_mode,linedeten,plugin,scancode,test_clk,waitsel,uram_addr[6:0],1'b0};<br><br>1100:{ramdata}<br><br>1101:{usbdta}<br><br>1110:{maddr,sfraddr}<br><br>1111:{vstatus[7:0],usbivect[6:0],otgstate[3:0],downstrstate[3:0],upstrstate[4:0],utmisuspendm,utmilinestate[1:0],1'b0}; |  |  |
|--|--|--|--|--|

PS: this register use the MCUCLK to accessd

### 11.1.6.86 VDCTRL

USB PHY vendor control register

Offset = 0xef

| Bit(s) | Name     | Description                         | R/W | Reset |
|--------|----------|-------------------------------------|-----|-------|
| 7:5    | -        | reserved                            | R   | 000B  |
| 4      | vload    | Write enable of PHY vendor control. | R/W | 1B    |
| 3:0    | vcontrol | address of PHY registers.           | R/W | 0000B |

PS: this register use the MCUCLK to accessd

### 11.1.6.87 VDSTAT

USB PHY vendor status register

Offset = 0xf1

| Bit(s) | Name    | Description  | R/W | Reset |
|--------|---------|--|-----|-------|
| 7:0    | vstatus | When vload is low, this is the value which will write to PHY register; when vload is high, this is the value of selected PHY register. | R/W | XXH   |

PS: this register use the MCUCLK to accessd

### 11. 1. 6. 88 BKDOOR

Test back door register

Offset = 0xf3

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7      | hsdisable | HS disable.<br>If 1, HS USB will be disabled, this bit used for debug purpose.  | R/W | 0B    |
| 6      | usbspeed  | Indicated the usb transfer speed, this bit is available in<br>Usb transaction only.<br>1: high speed<br>0: full speed | R   | 0B    |
| 5      | a_ndiscon | Set 1: the discon(USBCS bit 6) will not effect on duel role A-device FSM  | R/W | 0B    |
| 4:2    | -         | Reserved  | R   | 0B    |
| 1      | forcefs   | If 1, PHY will be set to FS mode; this bit is for test purpose only.<br>This bit must set to 0 in normal operation.   | R/W | 0B    |
| 0      | forcehs   | If 1, PHY will be set to HS mode; this bit is for test purpose only.<br>This bit must set to 0 in normal operation.   | R/W | 0B    |

### 11. 1. 6. 89 OTGTRIEN

OTR status machine interrupt enable register

Offset = 0xf5

| Bit(s) | Name       | Description  | R/W | Reset  |
|--------|------------|--|-----|--------|
| 7:2    | -          | reserved   | R   | 00000B |
| 1      | awtbconien | Set 1: enable the awaitbconnect irq<br>Clear0: disable the awaitbconnect irq | R/W | 0B     |
| 0      | asuspien   | Set 1: enable the asuspend irq<br>Clear0: disable the asuspend irq           | R/W | 0B     |

## 11.1.6.90 OTGTRIRQ

OTR status machine interrupt request register

Offset = 0xf6

| Bit(s) | Name       | Description   | R/W | Reset  |
|--------|------------|---|-----|--------|
| 7:2    | -          | reserved  | R   | 00000B |
| 1      | awtbconirq | When the OTG FSM enter awaitconnect, this bit will assert.<br>Write 1 to clear this bit | R/W | 0B     |
| 0      | asuspien   | When the OTG FSM enter asuspend, this bit will assert.<br>Write 1 to clear this bit     | R/W | 0B     |

## 11.1.6.91 USB\_Efuse\_Ref

Usb Access Efuse\_Ref register

Offset = 0xf7

| Bit(s) | Name         | Description   | R/W | Reset   |
|--------|--------------|---|-----|---------|
| 7      | Efacen       | If Efacen=0, the Efref (bit5:0) is the output of Efuse_REF[5:0], which is from the ChipID module to Usb module.<br>If Efacen=1, the Up sets Efuse_REF[5:0] in the Usb module by writing the Efref (bit5:0) and gets Efuse_REF[5:0] by reading the Efref (bit5:0). | R/W | 0B      |
| 6      | USB_Efuse_EN | Efuse standby control bit·control ENB Value :<br>0 : ENB=0 , operation mode<br>1 : ENB=1 , standby. Efuse is standby, and its standby current is 5Ua max.   | R/W | 1B      |
| 5:0    | awtbconirq   | Usb 6.25k refer value.<br>When Efacen = 0, the value is from effuse load output<br>When Efacen = 1, the value is from cpu configuration   | R/W | 011110B |

PS: this register use the MCUCLK to accesssd

PS：在 Efref[5:0]取默认值 011110 的情况下，假设实际测得电阻为 R1，那么 efuse 应烧入 Efref[5:0]= $6250 \times 100 / R1 - 70$ ，如果该数值小于 0 或者大于 62，则无法 efuse 到所需要的电阻值。  
如果要采用外挂电阻，则需要设置 Efref[5:0]=111111，从而断开内部电阻，采用外挂电阻。

### 11.1.6.92 FSMPRESTATE

FSM pre-state register

Offset = 0xfc

| Bit(s) | Name          | Description  | R/W | Reset |
|--------|---------------|--|-----|-------|
| 7:4    | awtbcon_state | The 4 bits designate former state before enter the awaitbconnect , the index of FSM refer to the otgstate register | R   | 0000B |
| 3:0    | reserved      | reserved   | R   | 0000B |

### 11.1.6.93 HCIN2CNTL

hcin2 packet counter low register

Offset = 0xfe

| Bit(s) | Name          | Description              | R/W | Reset |
|--------|---------------|--------------------------|-----|-------|
| 7:0    | hcin2counterl | hcin2 packet counter low | R/W | 00H   |

### 11.1.6.94 HCIN2CNTH

hcin2 packet counter high register

Offset = 0xfd

| Bit(s) | Name          | Description                 | R/W | Reset |
|--------|---------------|-----------------------------|-----|-------|
| 7:0    | hcin2counterh | hcin2 packet counter (high) | R/W | 00H   |

### 11.1.6.95 EP1STADDRL

EP1 FIFO start address low register

Offset = 0x8b

| Bit(s) | Name       | Description                  | R/W | Reset |
|--------|------------|------------------------------|-----|-------|
| 7:0    | EP1STADDRL | EP1 FIFO start address (low) | R/W | 00H   |

### 11.1.6.96 EP1STADDRH

EP1 FIFO start address high register

Offset = 0x8a

| Bit(s) | Name       | Description                   | R/W | Reset |
|--------|------------|-------------------------------|-----|-------|
| 7:0    | EP1STADDRH | EP1 FIFO start address (high) | R/W | 00H   |

### 11.1.6.97 EP2STADDRL

EP2 FIFO start address low register

Offset = 0xb8

| Bit(s) | Name       | Description                  | R/W | Reset |
|--------|------------|------------------------------|-----|-------|
| 7:0    | EP2STADDRL | EP2 FIFO start address (low) | R/W | 00H   |

### 11.1.6.98 EP2STADDRH

EP2 FIFO start address high register

Offset = 0xb7

| Bit(s) | Name       | Description                   | R/W | Reset |
|--------|------------|-------------------------------|-----|-------|
| 7:0    | EP2STADDRH | EP2 FIFO start address (high) | R/W | 00H   |

## 11.1.7 PHY Register Description

### 11.1.7.1 APHY Register Description

R30 (6'h30 – VCTR = 8'He0)

| Bit(s) | Name            | Description   | RW  | Reset |
|--------|-----------------|---|-----|-------|
| 7:5    | SEL_CDRTEST     | CDR test output signal select   | R/W | 101B  |
| 4:3    | REG_PLL_TESTSEL | PLL test select<br>2'b00: HV<br>2'b01: VC_RC<br>2'b10: GND<br>2'b11: VDD_VCO                        | R/W | 00B   |
| 2      | REG_FORCE_VC    | Force VC in PLL<br>1'b0 Force VC<br>1'b1 not force  | R/W | 0B    |
| 1:0    | REG_PLL_ICP     | Adjust bias current of charge pump<br>2'b00: 2.5Ua<br>2'b01: 5.0Ua<br>2'b10: 7.5Ua<br>2'b11: 10.0Ua | R   | 01B   |

R31 (6'h31 – VCTR = 8'He1)

| Bit(s) | Name           | Description  | RW  | Reset |
|--------|----------------|--|-----|-------|
| 7      | REG_SQLCHX2    | Set squelch threshold to double value during chirp process<br>1'b0: not double<br>1'b1: double               | R/W | 0B    |
| 6:5    | REG_RCALVREF   | Resistor Calibration reference voltage(0.4V)<br>2'b00: 380Mv<br>2'b01: 400Mv<br>2'b10: 420Mv<br>2'b11: 440Mv | R/W | 01B   |
| 4:3    | REG_TDSEL      | Adjust Ckusable and 480M output delay<br>2'b00: 11u<br>2'b01: 86u<br>2'b10: 340u<br>2'b11: 1360u             | R/W | 00B   |
| 2      | REG_PLL_TESTEN | PLL test enable  |     | 0B    |
| 1:0    | REG_PLL_RS     | Adjust RS of loop filter in pll<br>2'b00: 30K<br>2'b01: 40K<br>2'b10: 50K<br>2'b11: 60K                      | R   | 01B   |

R32 (6'h32 – VCTR = 8'He2)

| Bit(s) | Name                   | Description  | RW  | Reset     |
|--------|------------------------|--|-----|-----------|
| 7      | REG_PU_SQLCHVR<br>EF   | Force squelch threshold voltage on<br>1'b0: not force<br>1'b1: force   | R/W | 0B        |
| 6:4    | REG_DIFVREF_HSD<br>ISC | Adjust disconnection detection threshold voltage<br>000: 450Mv<br>001: 475Mv<br>010: 500Mv<br>011: 525Mv<br>100: 550Mv<br>101: 575Mv<br>110: 600Mv<br>111: 625Mv | R/W | 011B      |
| 3:0    | REG_HSTX_IDR           | Adjust HS TX bias current<br>$IB = 160u + 40u*REG\_HSTX\_IDR[3] + 20u*REG\_HSTX\_IDR[2] + 10u*REG\_HSTX\_IDR[1] + 5u*REG\_HSTX\_IDR[0]$                          | R/W | 0100<br>B |

## R33 (6'h33 – VCTR = 8'He3)

| Bit(s) | Name          | Description   | RW  | Reset |
|--------|---------------|---|-----|-------|
| 7      | REG_EN_HSTXPD | HSTX power down enable<br>1'b0: not enable<br>1'b1: enable  | R/W | 0B    |
| 6      | REG_HSTX_CAP  | Connect cap to DP/DM to eliminate overshoot<br>1'b0: not connect<br>1'b1: connnect  | R/W | 0B    |
| 5:3    | REG_BIASVREF  | Adjust internal reference voltage(1.25V)<br>3'b000: 1.125V<br>3'b 001:1.167V<br>3'b 010: 1.208V<br>3'b 011: 1.25V<br>3'b 100: 1.292V<br>3'b 101: 1.333V<br>3'b 110: 1.375V<br>3'b 111: 1.417V | R/W | 011B  |
| 2:0    | REG_LFSTX_IB  | adjust LFS TX bias current  | R/W | 011B  |

## R34 (6'h34 – VCTR = 8'He4)

| Bit(s) | Name                  | Description   | RW  | Reset |
|--------|-----------------------|---|-----|-------|
| 7:6    | REG_IB_HSD2S          | Adjust HSRCV bias current<br>IB= 10u+10u*REG_IB_HSD2S[1]<br>+5u* REG_IB_HSD2S[0]  | R/W | 01B   |
| 5:3    | REG_USBTESTSEL        | Digital output signal select in debug mode  | R/W | 000B  |
| 2:0    | REG_DIFVREF_SQL<br>CH | Adjust squelch threshold voltage<br>3'b000: 80Mv<br>3'b001: 90Mv<br>3'b010: 100Mv<br>3'b011:110Mv<br>3'b100: 120Mv<br>3'b101: 130Mv<br>3'b110: 140Mv<br>3'b111: 150Mv | R/W | 100B  |

## R35 (6'h35 – VCTR = 8'He5)

| Bit(s) | Name       | Description   | RW  | Reset |
|--------|------------|---|-----|-------|
| 7      | EDRSTEN    | Edge Detection Reset Enable<br>1'b0: Enable=HSEDEN<br>1'b1: Enable=1'b1 | R/W | 0B    |
| 6      | REG_DMTEST | 1'b0:<br>1'b1: DM=0   | R/W | 0B    |
| 5      | REG_DPTEST | 1'b0:<br>1'b1: DM=0   | R/W | 0B    |

|     |                |  |     |      |
|-----|----------------|--|-----|------|
| 4:2 | REG_HSTX_SLEW  | HSTX slew rate control<br>111: the highest<br>000: the lowest                        | R/W | 100B |
| 1   | REG_EN_USBTEST | USB debug test enable<br>1'b0: debug test disable<br>1'b1: debug test enable         | R/W | 0B   |
| 0   | REG_NSQDLY     | Shorten Squelch Detection time<br>1'b0: not shorten<br>1'b1: shorten by 1CLK(480MHz) | R/W | 0B   |

R36 (6'h36 – VCTR = 8'He6)

| Bit(s) | Name               | Description  | RW  | Reset |
|--------|--------------------|--|-----|-------|
| 7      | REG_THROUGH        | edg quantity control   | R/W | 0B    |
| 6:5    | REG_SET_VC         | Set VC of PLL<br>2'b00: VC=0.6V<br>2'b01: VC=1.2V<br>2'b10: VC=1.8V<br>2'b11: VC=2.4V                        | R/W | 00B   |
| 4:3    | REG_RESPULLUP      | Adjust 1.5K pull-up resister<br>2'b00: RES=1.18k<br>2'b01: RES=1.31k<br>2'b10: RES=1.49k<br>2'b11: RES=1.62k | R/W | 01B   |
| 2      | REG_BYPASSMODE_SEL | HSTX current bypass control<br>1'b0: bypassen=bypassmode1<br>1'b1: bypassen=POWBB&OEB                        | R/W | 1B    |
| 1:0    | REG_IB_HSDISC      | Adjust HS disconnection detector bias current<br>IB= 20u + 10u*REG_IB_HSDISC [1]<br>+ 5u*REG_IB_HSDISC [0]   | R/W | 01B   |

R37 (6'h37 – VCTR = 8'He7)

| Bit(s) | Name               | Description  | RW  | Reset |
|--------|--------------------|--|-----|-------|
| 7      | REG_IB_SQLCH[2]    | Adjust bias current of squelch detection   | R/W | 0B    |
| 6      | REG_PUB_MBIAS_TEST | Force bias power down<br>1'b0: not force<br>1'b1: force                                      | R/W | 0B    |
| 5      | REG_EDG_TYPE       | Control edge detection<br>1'b0: detect dual edge of data<br>1'b1: detect rising edge of data | R/W | 0B    |
| 4      | REG_LFSTX_IB[3]    | LDO test enable  | R/W | 0B    |
| 3      | REG_PU_HSDISC      | High speed disconnection enable, judge enable  | R/W | 1B    |
| 2      | REG_RCALEN         | 45ohm resistor calibration and hold enable<br>1'b0: disable<br>1'b1: enable                  | R/W | 0B    |

|   |            |   |     |    |
|---|------------|---|-----|----|
| 1 | REG_RALLEN | 45ohm resistor calibration and hold enable<br>1'b0: disable<br>1'b1: enable | R/W | 1B |
| 0 | OTGDETEN   | VBUS Detection enable<br>1'b0: disable<br>1'b1: enable                      | R/W | 1B |

R03 (6'h03 – VCTR = 8'h83)

| Bit(s) | Name            | Description   | RW  | Reset |
|--------|-----------------|---|-----|-------|
| 7      | REG_NSQTEST     | Generate a “NSQ” by reg for test  | R/W | 0B    |
| 6:5    | REG_RESPULLDN   | Adjust 15K pull down resistor<br>2'b00: 15.6K<br>2'b00: 18.3K<br>2'b00: 21.4K<br>2'b00: 24.2K | R/W | 01B   |
| 4      | REG_LFSRX_IB    | Adjust bias current of LFSRX  | R/W | 0B    |
| 3      | REG_HSTX_TEST   | HSTX test enable<br>1'b0: disable<br>1'b1: enable   | R/W | 0B    |
| 2:0    | REG_LFSTX_CROSS | Adjust cross point of DP/DM in LFSTX<br>3'b000: lowest<br>3'b111: highest                     | R/W | 011B  |

R04 (6'h04 – VCTR = 8'h84)

| Bit(s) | Name           | Description   | RW  | Reset |
|--------|----------------|---|-----|-------|
| 7:6    | Reserved       |   | R/W | 00B   |
| 5      | REG_DMOFF      | Turn off one of DDA in HSDISC<br>1'b0: not turn off<br>1'b1: turn off                         | R/W | 0B    |
| 4:3    | Reserved       |   | R/W | 00B   |
| 2      | REG_CDRRBSEL   | CDR reset select<br>1'b0: select normal reset<br>1'b1: select test reset signal               | R/W | 0B    |
| 1      | REG_CDRRBTTEST | CDR reset signal for test   | R/W | 1B    |
| 0      | REG_CDRINSEL   | CDR input select<br>1'b0: select normal input of CDR<br>1'b1: select test input signal of CDR | R/W | 0B    |

R05 (6'h05 – VCTR = 8'h85)

| Bit(s) | Name     | Description | RW  | Reset |
|--------|----------|-------------|-----|-------|
| 7:0    | Reserved |             | R/W | 0B    |

R06 (6'h06 – VCTR = 8'h86)

| Bit(s) | Name     | Description | RW  | Reset |
|--------|----------|-------------|-----|-------|
| 7:0    | Reserved |             | R/W | 0B    |

### 11.1.7.2 DPHY Register Description

R38 (6'h38– VCTR = 8'HF0)

| Bit(s) | Name            | Description   | RW  | Reset |
|--------|-----------------|---|-----|-------|
| 7:6    | REG_IB_SQLCH    | Adjust bias current of squelch detection  | R/W | 01B   |
| 5      | EN_ERR_UNDERRUN | The packet error mode due to EB under run will be disabled if this bit is cleared.  | R/W | 1B    |
| 4      | LATE_DLLEN      | HSDLLEN will be asserted to start RX 100ns after TX is done if LATE_DLLEN is set.<br>Otherwise, only 66ns duration will be set. This is to avoid the reception of ECHO following the TX issued by PHY itself. | R/W | 1B    |
| 3      | INTG            | 3ns shorter in latency in TX path will be achieved if INTG is set, however it required data should be stable more quickly   | R/W | 1B    |
| 2      | SOP_KK          | The first KK encountered in a packet is recognized as SOP if SOP_KK is set. Otherwise a 1's in the output of DE-NRZI will be recognized as SOP instead.   | R/W | 1B    |
| 1      | SLB_INNER       | This bit is valid only when SLB_EN is set. The loop-back exists within digital part only. Otherwise if SLB_INNER is cleared, analog transceiver will join the loop-back.                                      | R/W | 0B    |
| 0      | SLB_EN          | To enable self loop-back for PHY, with either digital loop-back only, or digital-analog loop-back.  | R/W | 0B    |

R39 (6'h39– VCTR = 8'HF1)

| Bit(s) | Name           | Description       | RW  | Reset |
|--------|----------------|-------------------|-----|-------|
| 7:0    | TEST_SEED[7:0] | bist initial seed | R/W | 0B    |

R3a (6'h3a– VCTR = 8'HF2)

| Bit(s) | Name         | Description  | RW  | Reset |
|--------|--------------|--|-----|-------|
| 7      | dbge_HSVO    | HSVO debug mux enable  | R/W | 0B    |
| 6      | dbge_HSVOEN  | HSVOEN debug mux enable  | R/W | 0B    |
| 5      | dbge_HSXMTPD | HSXMTPD debug mux enable   | R/W | 0B    |
| 4      | dbge_LFVO    | LFVO debug mux enable  | R/W | 0B    |
| 3:2    | TEST_PSL     | select tx data<br>2'b00: 16'h0000<br>2'b 01: {test_seed, test_seed}<br>2'b 10: {~tx_crc, tx_crc} | R/W | 00B   |

|   |          |                                      |     |    |
|---|----------|--------------------------------------|-----|----|
|   |          | 2'b 11: {tx_cnt, 1'b1, tx_cnt, 1'b0} |     |    |
| 1 | TEST_EN  | bist enable                          | R/W | 0B |
| 0 | TEST_RST | bist reset                           | R/W | 0B |

R3b (6'h3b– VCTR = 8'HF3)

| Bit(s) | Name          | Description               | RW  | Reset |
|--------|---------------|---------------------------|-----|-------|
| 7      | dbge_HSTXEN   | HSTXEN debug mux enable   | R/W | 0B    |
| 6      | dbge_HSRXEN   | HSRXEN debug mux enable   | R/W | 0B    |
| 5      | dbge_FSTERMEN | FSTERMEN debug mux enable | R/W | 0B    |
| 4      | dbge_LSTERMEN | LSTERMEN debug mux enable | R/W | 0B    |
| 3      | dbge_LFSPDSEL | LFSPDSEL debug mux enable | R/W | 0B    |
| 2      | dbge_LFXCVREN | LFXCVREN debug mux enable | R/W | 0B    |
| 1      | dbge_LFOEN    | LFOEN debug mux enable    | R/W | 0B    |
| 0      | dbge_LFFSE0   | LFFSE0 debug mux enable   | R/W | 0B    |

R3c (6'h3c– VCTR = 8'HF4)

| Bit(s) | Name         | Description                | RW  | Reset |
|--------|--------------|----------------------------|-----|-------|
| 7      | dbg_HSTXEN   | HSTXEN debug mux data in   | R/W | 0B    |
| 6      | dbg_HSRXEN   | HSRXEN debug mux data in   | R/W | 0B    |
| 5      | dbg_FSTERMEN | FSTERMEN debug mux data in | R/W | 0B    |
| 4      | dbg_LSTERMEN | LSTERMEN debug mux data in | R/W | 0B    |
| 3      | dbg_LFSPDSEL | LFSPDSEL debug mux data in | R/W | 0B    |
| 2      | dbg_LFXCVREN | LFXCVREN debug mux data in | R/W | 0B    |
| 1      | dbg_LFOEN    | LFOEN debug mux data in    | R/W | 0B    |
| 0      | dbg_LFFSE0   | LFFSE0 debug mux data in   | R/W | 0B    |

R3d (6'h3d– VCTR = 8'HF5)

| Bit(s) | Name          | Description               | RW  | Reset |
|--------|---------------|---------------------------|-----|-------|
| 7      | dbge_DPPULLDN | DPPULLDN debug mux enable | R/W | 0B    |
| 6      | dbge_DMPULLDN | DMPULLDN debug mux enable | R/W | 0B    |
| 5      | Reserved      |                           |     |       |
| 4      | dbge_OPMODE   | OPMODE debug mux enable   | R/W | 0B    |
| 3      | Reserved      |                           |     |       |
| 2      | dbge_XCVRSEL  | XCVRSEL debug mux enable  | R/W | 0B    |
| 1      | dbge_TERMSEL  | TERMSEL debug mux enable  | R/W | 0B    |
| 0      | dbge_SUSPENDM | SUSPENDM debug mux enable | R/W | 0B    |

R3e (6'h3e– VCTR = 8'HF6)

| Bit(s) | Name         | Description                | RW  | Reset |
|--------|--------------|----------------------------|-----|-------|
| 7      | dbg_DPPULLDN | DPPULLDN debug mux data in | R/W | 0B    |
| 6      | dbg_DMPULLDN | DMPULLDN debug mux data in | R/W | 0B    |
| 5:4    | dbg_OPMODE   | OPMODE debug mux data in   | R/W | 00B   |

|     |               |                             |     |     |
|-----|---------------|-----------------------------|-----|-----|
| 3:2 | dbg_XCVRSEL   | XCVRSEL debug mux data in   | R/W | 00B |
| 1   | dbg_TERMSEL   | TERMSEL debug mux data in   | R/W | 0B  |
| 0   | dbg_SUSPENDMD | SUSPENDMD debug mux data in | R/W | 0B  |

R3f (6'h3f– VCTR = 8'Hz7)

| Bit(s) | Name             | Description   | RW  | Reset |
|--------|------------------|---|-----|-------|
| 7:6    | Reserved         |   | R/W | 00B   |
| 5      | dbg_HSXMTPD      | HSXMTPD debug mux data in   | R/W | 0B    |
| 4      | dbg_LFVO         | LFVO debug mux data in  | R/W | 0B    |
| 3      | dbg_nsq_rxactive | Nsq rxactive delay  | R/W | 0B    |
| 2      | dbg_nsq_rxerror  | Nsq rxerror delay   | R/W | 0B    |
| 1      | SOP_JK           | KJKK encountered in a packet is recognized as SOP if this bit is set. | R/W | 0B    |
| 0      | NPKTERR_DIS      | Disable no packet bitstuff error                                      | R/W | 0B    |

R22 (6'h22– VCTR = 8'ha2)

| Bit(s) | Name      | Description            | RW  | Reset     |
|--------|-----------|------------------------|-----|-----------|
| 7:4    | dbg_HSOEN | HSOEN debug mux enable | R/W | 0000<br>B |
| 3:0    | dbg_HSVO  | HSVO debug mux data in | R/W | 0000<br>B |

### 11.1.7.3 Debug Register Description

| R/<br>W | Ad<br>dr | VC<br>TL | D7          | D6                     | D5                     | D4           | D3                   | D2           | D1                | D0                       |
|---------|----------|----------|-------------|------------------------|------------------------|--------------|----------------------|--------------|-------------------|--------------------------|
| R       | 'h0<br>0 | 'h8<br>0 | pu_pll      | CK120_ON               | clk120_sel<br>clk120M  | clk12        | clkusb12             | clk4x        | clk120            | clk_UTMI                 |
| R       | 'h0<br>1 | 'h8<br>1 |             |                        |                        |              | tcal[4:0]            | tpc          | tpb               | tpa                      |
| R       | 'h0<br>2 | 'h8<br>2 |             |                        |                        |              |                      |              |                   | cdr_test[7:0]            |
| R<br>W  | 'h0<br>3 | 'h8<br>3 | reg_nsqtest |                        | reg_respulldn[<br>1:0] | reg_lfsrx_ib | reg_hstx_test        |              |                   | reg_lfstx_cro<br>ss[2:0] |
| R<br>W  | 'h0<br>4 | 'h8<br>4 |             | reg_ana_resv<br>0[1:0] | reg_dmoff              |              | reg_svbus_0<br>[1:0] | reg_cdrrbsel | reg_cdrrbt<br>est | reg_cdrinsel             |
| R<br>W  | 'h0<br>5 | 'h8<br>5 |             |                        |                        |              |                      |              |                   | reg_ana_resv<br>1[7:0]   |
| R<br>W  | 'h0<br>6 | 'h8<br>6 |             |                        |                        |              |                      |              |                   | reg_ana_resv<br>2[7:0]   |
| R       | 'h0<br>7 | 'h8<br>7 |             |                        |                        |              |                      |              |                   |                          |

|        |          |                    |                    |                        |                    |                        |                    |                       |                        |
|--------|----------|--------------------|--------------------|------------------------|--------------------|------------------------|--------------------|-----------------------|------------------------|
| R<br>8 | 'h0<br>0 |                    |                    |                        |                    |                        |                    |                       |                        |
| R<br>9 | 'h0<br>1 |                    |                    |                        |                    |                        |                    |                       |                        |
| R<br>A | 'h0<br>2 |                    |                    |                        |                    |                        |                    |                       |                        |
| R<br>B | 'h0<br>3 |                    |                    |                        |                    |                        |                    |                       |                        |
| R<br>C | 'h0<br>4 |                    |                    |                        |                    |                        |                    |                       |                        |
| R<br>D | 'h0<br>5 | pub_mbias          | pu_pll             |                        | clk_usable         | en_fsterm              | en_lsterm          | lfspdsel              | pu_lfxcvr              |
| R<br>E | 'h0<br>6 | -                  | en_phdet           | force_cdrrb            | pu_hsrx            | clk_usable             | hsrv               | cdrvld                | nsq                    |
| R<br>F | 'h0<br>7 | 0                  | 0                  | 0                      | 0                  | 0                      | 0                  | 0                     | 0                      |
| R<br>0 | 'h1<br>0 |                    |                    | cdrout_120M<br>[3:0]   | cdrvld             | clkext_120<br>M        | EB_readrd<br>y     | EB_outvalid           |                        |
| R<br>1 | 'h1<br>1 |                    |                    | cdrout_120M<br>[3:0]   |                    |                        |                    | EB_readout[<br>3:0]   |                        |
| R<br>2 | 'h1<br>2 |                    |                    | wr_cnt1[2:0]           |                    |                        | rd_cnt1[2:0]       | EB_readrd<br>y        | EB_outvalid            |
| R<br>3 | 'h1<br>3 |                    |                    |                        | denrziout[3:0]     |                        |                    |                       | EB_readout[<br>3:0]    |
| R<br>4 | 'h1<br>4 |                    |                    |                        | denrziout[3:0]     |                        | rxhistory[5:<br>4] | bitstuff_er<br>r      | err_hseop_no<br>t_bond |
| R<br>5 | 'h1<br>5 |                    |                    |                        |                    |                        |                    |                       | buf8[7:0]              |
| R<br>6 | 'h1<br>6 |                    |                    |                        |                    |                        |                    |                       | byte_out[7:0]          |
| R<br>7 | 'h1<br>7 |                    |                    |                        | store_cnt[7:3<br>] | byte_out_vl<br>d       |                    |                       | holdcnt[1:0]           |
| R<br>8 | 'h1<br>0 |                    |                    |                        |                    |                        |                    |                       | holdreg[7:0]           |
| R<br>9 | 'h1<br>1 |                    |                    |                        |                    |                        |                    |                       | holdreg[15:8]          |
| R<br>A | 'h1<br>2 |                    |                    |                        |                    |                        |                    |                       | holdreg[23:1<br>6]     |
| R<br>B | 'h1<br>3 | err_ebunde<br>rrun | err_eboverflo<br>w | err_fseop_not<br>_bond | err_fsbs           | err_hseop_n<br>ot_bond |                    | FS_rxhist<br>ory[5:4] | FS_bitstuff_e<br>rr    |
| R<br>C | 'h1<br>4 | hsdisc             | sync_din           | lfse0                  | FS_denrzi_ou<br>t  |                        | FS_fs_cs[1:<br>0]  | FS_outvld             | FS_holdcnt             |
| R      | 'h1<br>- | -                  | -                  | -                      | lfse0              | sync_din               | LFVPO              | LFVMO                 | LFRCV                  |

|     | D         | 5        |                      |         |                       |                             |                            | lfvpo                | lfvmo                | lfrcv                 |
|-----|-----------|----------|----------------------|---------|-----------------------|-----------------------------|----------------------------|----------------------|----------------------|-----------------------|
| R   | 'h1<br>E  | 'Hb<br>6 | -                    | -       | -                     | -                           | -                          | test_fail            | test_done            |                       |
| R   | 'h1<br>F  | 'Hb<br>7 |                      |         |                       |                             |                            |                      |                      |                       |
| R   | 'h2<br>0  | 'Hc<br>0 |                      |         | hsvo[3:0]             | hstxclk                     | hstxpd                     | pu_hstx              | reg_en_hstxp<br>d    |                       |
| R   | 'h2<br>1  | 'Hc<br>1 |                      |         | hsoen[3:0]            | hstxclk                     | hstxpd                     | pu_hstx              | reg_en_hstxp<br>d    |                       |
| R/W | 'h2<br>2  | 'Hc<br>2 |                      |         | hsoen[3:0]            |                             |                            |                      | hsvo[3:0]            |                       |
| R   | 'h2<br>3  | 'Hc<br>3 |                      |         | stuff_out[3:0]        |                             |                            |                      | hsvo[3:0]            |                       |
| R   | 'h2<br>4  | 'Hc<br>4 |                      |         | stuff_out[3:0]        | readbyte                    |                            |                      | insert_cnt[2:<br>0]  |                       |
| R   | 'h2<br>5  | 'Hc<br>5 |                      |         |                       |                             |                            |                      | holdreg_out[<br>7:0] |                       |
| R   | 'h2<br>6  | 'Hc<br>6 | shift_en             | nrzi_en | rdbblk                |                             | txcs[1:0]                  | stuff_en             |                      | eopout[1:0]           |
| R   | 'h2<br>7  | 'Hc<br>7 | eop                  | eopflag |                       | eophdreg[1:0]               |                            | serv_cnt[1:0]<br>]   |                      | req_cnt[1:0]          |
| R   | 'h2<br>8  | 'Hd<br>0 |                      |         |                       |                             |                            |                      |                      | tx_dout[7:0]          |
| R   | 'h2<br>9  | 'Hd<br>1 |                      |         |                       |                             |                            |                      |                      | tx_dout[15:8]         |
| R   | 'h2<br>A  | 'Hd<br>2 | lfvo                 | lfoen   | lfse0                 | FS_stuff_out                | FS_shift_en                | FS_rise_txvl<br>d    | FS_nrzi_e<br>n       | FS_stuff_en           |
| R   | 'h2<br>B  | 'Hd<br>3 |                      |         |                       |                             |                            | FS_history[<br>5:0]  | FS_readyby<br>te     | FS_insert             |
| R   | 'h2<br>C  | 'Hd<br>4 |                      |         |                       |                             |                            |                      |                      | FS_tx_dout[<br>7:0]   |
| R   | 'h2<br>D  | 'Hd<br>5 |                      |         |                       |                             |                            |                      |                      | FS_shiftreg[7:<br>0]  |
| R   | 'h2<br>E  | 'Hd<br>6 | bypassmod<br>e1      | HSOE    | long_eop              | eop_end                     | nse_hsoen                  | nseop                | eop                  | stuff_en              |
| R   | 'h2<br>F  | 'Hd<br>7 | nseop                |         | stuen_out[1:0]        | stuff_en                    |                            |                      |                      | stuff_out[3:0]<br>]   |
| R/W | 'h3<br>W0 | 'He<br>0 | sel_cdrtest[<br>2:0] |         |                       |                             | reg_pll_tests<br>el_0[1:0] | reg_force_v<br>c_0   |                      | reg_pll_icp_0[1:0]    |
| R/W | 'h3<br>W1 | 'He<br>1 | reg_sqlchx<br>2      |         | reg_rcalvref[1:<br>0] |                             | reg_Tdsel[1:<br>0]         | reg_pll teste<br>n_0 |                      | reg_pll_rs_0[<br>1:0] |
| R/W | 'h3<br>W2 | 'He<br>2 | reg_pu_sql<br>chVref |         |                       | reg_difvref_h<br>sdisc[2:0] |                            |                      |                      | reg_hstx_Idr[<br>3:0] |

|     |          |          |                    |                          |                           |                     |                         |                         |                    |                            |
|-----|----------|----------|--------------------|--------------------------|---------------------------|---------------------|-------------------------|-------------------------|--------------------|----------------------------|
| R/W | 'h3<br>3 | 'He<br>3 | reg_en_hst<br>xpdp | reg_hstx_cap             |                           |                     | reg_biasref<br>_0[2:0]  |                         |                    | reg_lfstx_ib[<br>2:0]      |
| R/W | 'h3<br>4 | 'He<br>4 |                    | reg_lb_hsd2s[<br>1:0]    |                           |                     | reg_usbtests<br>el[2:0] |                         |                    | reg_difvref_s<br>qlch[2:0] |
| R/W | 'h3<br>5 | 'He<br>5 | edrstn             | reg_dmtest               | reg_dptest                |                     |                         | reg_hstx_sle<br>w[2:0]] | reg_en_us<br>btest | reg_nsqdlly                |
| R/W | 'h3<br>6 | 'He<br>6 | reg_throug<br>h    |                          | reg_set_vc_0[<br>1:0]     |                     | reg_respullu<br>p[1:0]  | reg_bypass<br>modesel   |                    | reg_lb_hsdis<br>c[1:0]     |
| R/W | 'h3<br>7 | 'He<br>7 | reg_phclk12<br>0M  | reg_pub_mbi<br>as_test_0 | reg_edg_type              | reg_efuse_off       | reg_pu_hsdis<br>c       | reg_rcalen              | reg_rallen         | otgdeten_0                 |
| R/W | 'h3<br>8 | 'Hf<br>0 |                    | reg_lb_sqlch             | EN_ERR_UN<br>DERRUN [1:0] | LATE_DLLE<br>N      | INTG<br>usbsoc          | SOP_KK                  | SLB_INN<br>ER      | SLB_EN                     |
| R/W | 'h3<br>9 | 'Hf<br>1 |                    |                          |                           |                     |                         |                         |                    | test_seed[7:0<br>]         |
| R/W | 'h3<br>A | 'Hf<br>2 | dbge_hsvo          | dbge_hsvoen              | dbge_hsxmtpd              | dbge_lfvo           |                         | test_psl[1:0]           | test_en            | test_RST                   |
| R/W | 'h3<br>B | 'Hf<br>3 | dbge_HST<br>XEN    | dbge_HSRX<br>EN          | dbge_FSTER<br>MEN         | dbge_LSTER<br>MEN   | dbge_LFSP<br>DSEL       | dbge_LFXC<br>VREN       | dbge_LFO<br>EN     | dbge_LFFSE<br>0            |
| R/W | 'h3<br>C | 'Hf<br>4 | dbg_HSTX<br>EN     | dbg_HSRXE<br>N           | dbg_FSTERM<br>EN          | dbg_LSTER<br>MEN    | dbg_LFSPD<br>SEL        | dbg_LFXC<br>VREN        | dbg_LFO<br>EN      | dbg_LFFSE0                 |
| R/W | 'h3<br>D | 'Hf<br>5 | dbge_DPP<br>ULLDN  | dbge_DMPU<br>LLDN        | -                         | dbge_OPMO<br>DE     | -                       | dbge_XCV<br>RSEL        | dbge_TER<br>MSEL   | dbge_SUSPE<br>NDM          |
| R/W | 'h3<br>E | 'Hf<br>6 | dbg_DPPU<br>LLDN   | dbg_DMPUL<br>LDN         |                           | dbg_OPMOD<br>E[1:0] |                         | dbg_XCVR<br>SEL[1:0]    | dbg_TER<br>MSEL    | dbg_SUSPE<br>NDM           |
| R/W | 'h3<br>F | 'Hf<br>7 | dppuen             | dmpuen                   | dbg_hsxmtpd               | dbg_lfvo            | dbg_nsq_rxactive        | dbg_nsq_rxerror         | sop_jk             | npkterr_dis                |

## 11.2 SPI 子模块（胡世飞、蔡瑞仁）

| 日期         | 版本     | 描述   | 修订人 |
|------------|--------|--|-----|
| 2012-07-24 | V1. 00 | initial  | 胡世飞 |
| 2012-09-12 | V1. 02 | SPI_RAND[5:4]用于设置 SPI 3 线模式，同时删除 SPI_BCH.2 功能，用于 reserved。 | 胡世飞 |
| 2012-12-12 | V2. 02 | 完善 opertaion manual 章节                                     | 胡世飞 |

### 11.2.1 Features

- Support dual I/O write and read mode
- Support SPI norflash boot mode

- ◆ Support IRQ and DMA mode to transmit data
- ◆ Support Randomizer

## 11.2.2 Function Description

The SPI module is designed according to Motorola serial peripheral interface protocols. It can be configured as either a master or slave device. It can generate a large range of SPI clock so as to communicate with different devices supporting SPI protocols. Especially, this module support three operation mode: write & read, write only, read only mode.

SPI write & read mode use the MOSI pin to serially write instructions, addresses or data to the device. It also uses the MISO pin to read data or status from the device synchronous. This mode is designed to meet normal SPI application.

The write only & read only mode support 1x I/O and 2x I/O mode. The 2x I/O mode allows data to be transfer to or from slave at twice the rate of the 1x I/O mode. This mode is designed to meet special application.

There are Randomizers to scramble / descramble data. Linear feedback shift register (LSFR) primitive polynomial:  $X^{16}+X^{12}+X^5+1$

### 11.2.3 Module Description

#### 11.2.3.1 Block Diagram

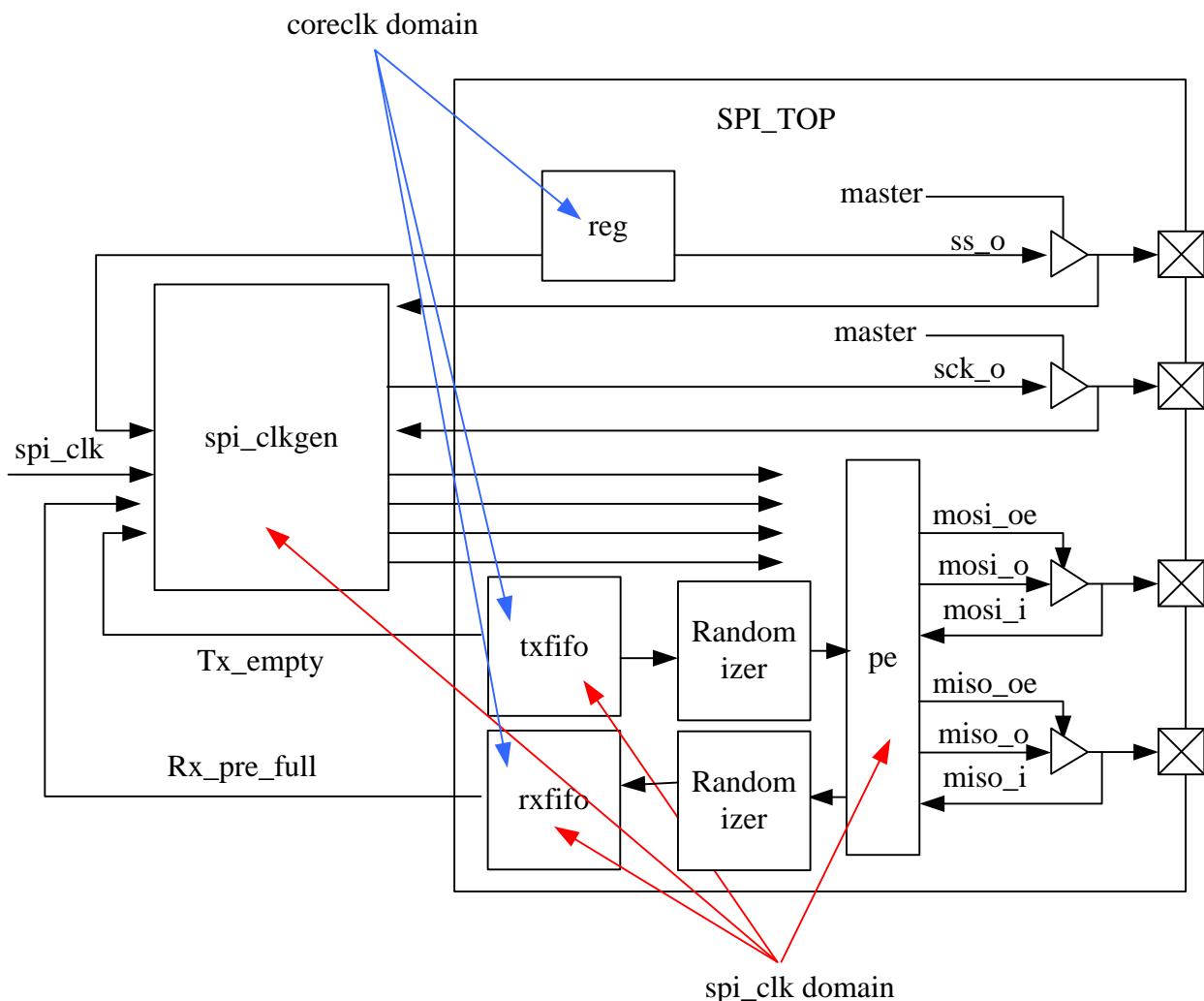


Figure 11-14 SPI Block Diagram

### 11.2.4 Operation Manual

#### 11.2.4.1 软件控制流程

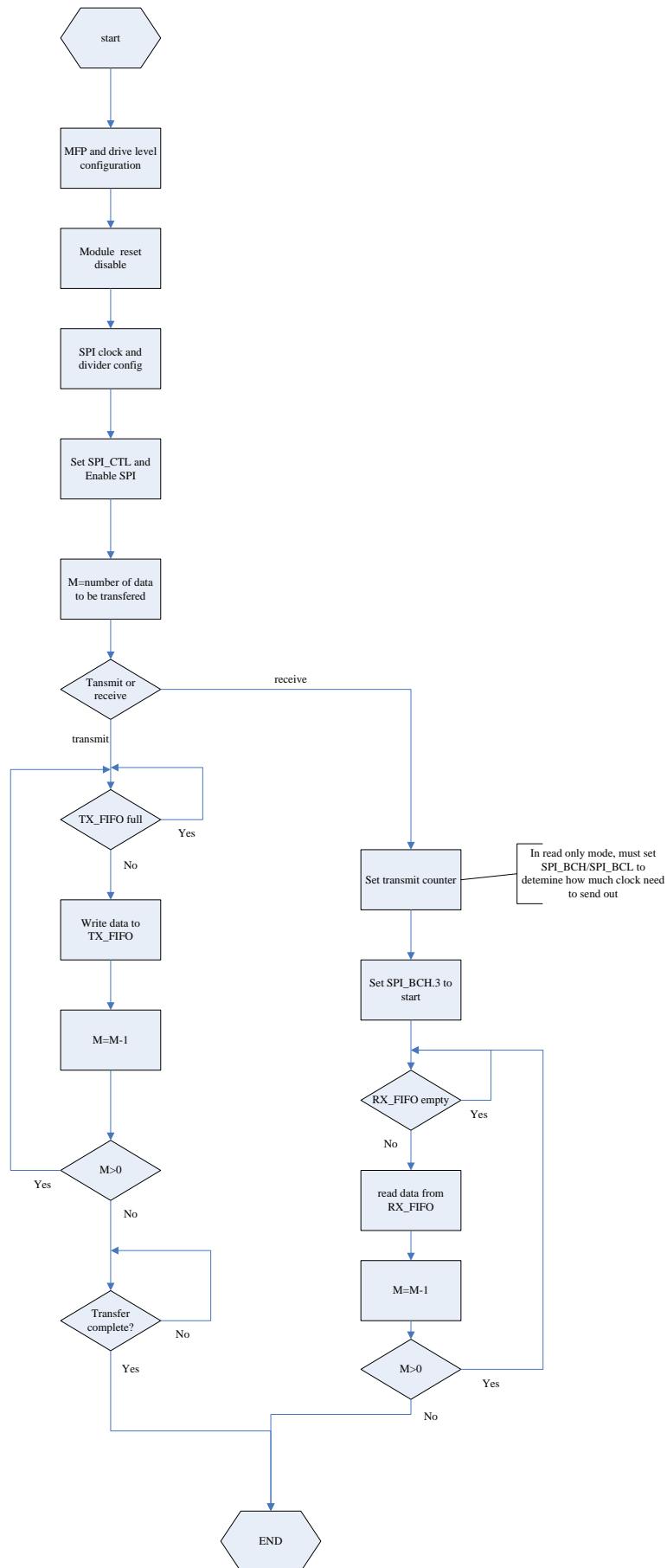


Figure 11-15 SPI 作 master 查询方式读写流程

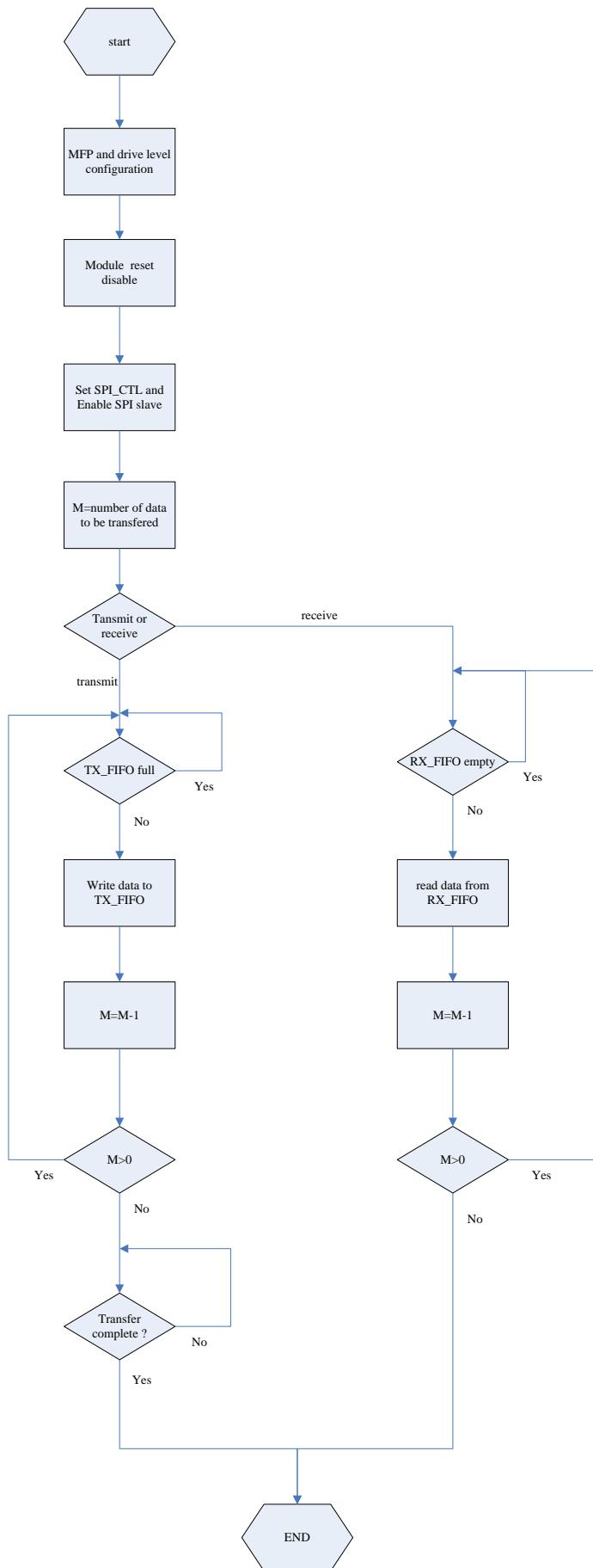
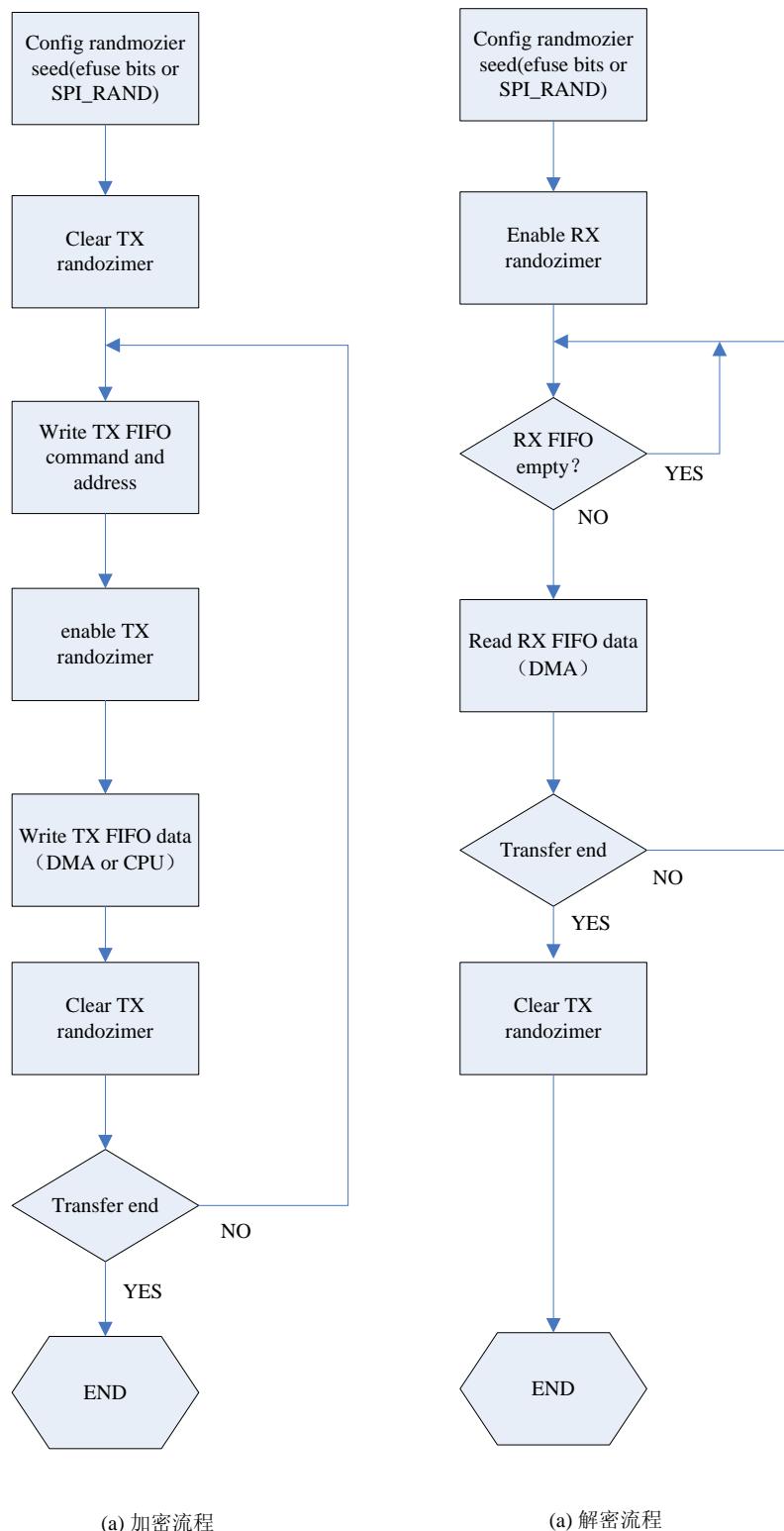


Figure 11-16 SPI 作 slave 查询方式读写流程



(a) 加密流程

(a) 解密流程

Figure 11-17 SPI 加密解密流程图

#### SPI 模块使用说明

1. SPI 可以工作在 master 或 slave 模式。在 slave 模式下时钟 SPI\_CLK 为输入信号, 由 master 产生; 在 master 模式下, SPI\_CLK 是输入信号, 它与 SPI 源时钟的关系是: SPI\_CLK =SPI source clock/(SPI\_CLKDIV[3:0]\*2), 其中 SPI\_CLKDIV 通过寄存器 SPI\_CLKDIV 配置, SPI source clock 可以是

COREClk、HOSC 或者 MCUPLL，通过 CMU 模块中的 SPICLKCTL 寄存器配置，如下图所示：

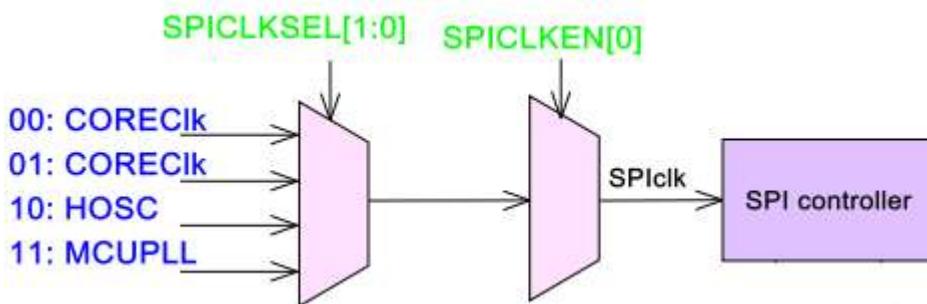


Figure 11-18 SPI clock

2. SPI 可以通过 DMA 收发数据。TX 触发 DMA 的条件是 FIFO 中的剩余空间大于等于 8 Bytes (因为 DMA 为 Burst8 模式，每次启动至少传递 8 Bytes)，RX 触发 DMA 的条件是 FIFO 中接收到的数据大于等于 8 Bytes。注意当数据少于 8 Bytes，不能用 DMA 收发，可以用查询或者 IRQ 方式。
3. SPI 可以使用 IRQ 方式收发数据。TX FIFO 为空或者 RX FIFO 非空都会触发 IRQ。
4. SPI master 条件下，如果是 read only 模式，需要通过配置 SPI\_BCH 和 SPI\_BCL 控制发送出去的 clock 个数，然后启动 read start (SPI\_BCH.3) 开始读取数据。注意实际发送出去的 clock number (以 Byte 为单位) = SPI\_BCH/L+1。
5. 相比较以前的项目，GL5115 的一个改动点是 SPI 作为 master 发送数据时与 SPI\_SS 无关(即使 SPI\_SS 为高，数据也能够发送)。这样改动的好处是一个 SPI master 接口可以挂几个 slave 分时复用：SPI\_MOSI、SPI\_MISO 和 SPI\_CLK 公用，用 SPI\_SS 和 GPIO 作为每个 slave 的片选信号。
6. 增加 SPI 3-wire 模式，通过 SPI\_RAND.3 使能。在 3-wire 模式下，SPI 作为 master 通过 SPI\_MOSI 读写数据(SPI\_MISO 不会用到)，需要将外部 nor 的 SPI\_MOSI 和 SPI\_MISO 短接，配置 SPI\_RAND.4 控制读写的方向。注意 3-wire 模式目前考虑到的应用场景仅限于 1 线模式读写 snor。
7. SPI 新加入加密功能。密钥可以用 16bits ffuse 值或者寄存器 SPI\_RAND 中的值，加密用 DMA 或者 CPU 的方式都可以，解密必须用 DMA 的方式，如果用 CPU 方式得到的是未解密的数据，这样主要是为了增强保密性。
8. 由于采用的是块模式加密解密数据包，所以要求这个过程对称，即一次加密过程与它对应的解密过程数据长度和起始位置都必须一致。
9. SPI master 设计最大工作频率 60MHz，实际测试中可能根据需要调节 delay chain，保证数据采样正确。
10. SPI 支持数据自发自收测试，可以用于 CP/FT 测试，内部 loop back 不需要外挂器件。

## 11.2.5 SPI Register List

SFR BANK: 0x0a

| Index | Mnemonic   | Description                       | Bank |
|-------|------------|-----------------------------------|------|
| 0x99  | SPI_CTL    | SPI Control Register              | 0x0a |
| 0x9a  | SPI_DRQ    | SPI DMA/IRQ control Register.     | 0x0a |
| 0x9b  | SPI_STA    | SPI Status Register               | 0x0a |
| 0x9c  | SPI_CLKDIV | SPI Clock Divide Control Register | 0x0a |
| 0x9d  | SPI_TXDAT  | SPI tx fifo register              | 0x0a |
| 0x9e  | SPI_RXDAT  | SPI rx fifo register              | 0x0a |

|      |           |                                    |      |
|------|-----------|------------------------------------|------|
| 0x9f | SPI_BCL   | SPI Bytes Count Low Register       | 0x0a |
| 0xa2 | SPI_BCH   | SPI Bytes Count high Register      | 0x0a |
| 0xa3 | SPI_DEBUG | SPI debug register                 | 0x0a |
| 0xc8 | SPI_RAND  | SPI Randomizer Control Register    | 0x0a |
| 0xc9 | SPI_SEED0 | SPI Randomizer test seed Register. | 0x0a |
| 0xca | SPI_SEED1 | SPI Randomizer test seed Register. | 0x0a |

## 11.2.6 Register Description

### 11.2.6.1 SPI\_CTL

SPI Control Register, This register is used for enabling SPI module, selecting SPI mode and SPI SS output voltage. (SFR address 0x99, SFR Bank = 0x0a)

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7      | SPI_EN   | <b>SPI Enable</b><br>0: Disable<br>1: Enable  | R/W | 0     |
| 6      | SPI_MS   | <b>SPI master/slave select</b><br>0: master<br>1: slave   | R/W | 0     |
| 5      | SPI_LM   | <b>LSB/MSB First Select</b><br>0: transmit and receive MSB first<br>1: transmit and receive LSB first   | R/W | 0     |
| 4      | SPI_SS   | <b>SPI SS pin control output</b> , this bit is valid only in master mode<br>0: output low<br>1: output high<br><b>修改点: master 模式不 care SPI_SS,默认值改为1.</b> | R/W | 1     |
| 3:2    | SPI_MODE | <b>SPI mode select</b><br>CPOL CPHA<br>00 : mode 0<br>01 : mode 1<br>10 : mode 2<br>11 : mode 3   | R/W | 11    |
| 1:0    | SPI_WR   | <b>SPI write/read select</b><br>00: write and read<br>01: write and read<br>10: write only<br>11: read only   | R/W | 00    |

### 11.2.6.2 SPI\_DRQ

SPI DMA/IRQ control Register. This register is used for enabling SPI DRQ/IRQ, and selecting SPI DRQ/IRQ trigger threshold. (SFR address 0x9a, SFR Bank = 0x0a)

| Bit(s) | Name        | Description   | R/W | Reset |
|--------|-------------|---|-----|-------|
| 7      | SPI_TDRQ_EN | <b>SPI TX DRQ Enable</b> , trigger DRQ when SPI TX FIFO at least 8 level empty; When DMA remain counter < 8, trigger DRQ until all data transfer completely;<br>0: disable<br>1: enable | R/W | 0     |
| 6      | SPI_RDRQ_EN | <b>SPI RX DRQ Enable</b> , trigger DRQ when SPI RX FIFO at least 8 level full.; When DMA remain counter < 8, trigger DRQ until all data received completely;<br>0: disable<br>1: enable | R/W | 0     |
| 5:4    | -           | <b>Reserved, be read as zero</b>  | -   | -     |
| 3      | SPI_TIRQ_EN | <b>SPI TX IRQ Enable</b> , trigger SPI TX IRQ when SPI TX FIFO is empty.<br>0: disable<br>1: enable   | R/W | 0     |
| 2      | SPI_RIRQ_EN | <b>SPI RX IRQ Enable</b> , trigger SPI RX IRQ when SPI RX FIFO is not empty.<br>0: disable<br>1: enable   | R/W | 0     |
| 1      | SPI_TIRQ_PD | <b>SPI TX IRQ Pending</b> , Write 1 to this bit will clear it.<br>0: No TX IRQ Pending<br>1: TX IRQ Pending.  | R/W | 0     |
| 0      | SPI_RIRQ_PD | <b>SPI RX IRQ Pending</b> , Write 1 to this bit will clear it.<br>0: No RX IRQ Pending<br>1: RX IRQ Pending.  | R/W | 0     |

### 11.2.6.3 SPI\_STA

SPI Status Register, This register is used for displaying current SPI FIFO status.  
(SFR address 0x9b, SFR Bank = 0x0a)

| Bit(s) | Name | Description                              | R/W | Reset |
|--------|------|--|-----|-------|
| 7      | TXEM | <b>SPI TX FIFO Empty</b><br>0: not empty | R   | 1     |

|   |          |   |     |   |
|---|----------|---|-----|---|
|   |          | 1: empty  |     |   |
| 6 | TXFU     | <b>SPI TX FIFO Full</b><br>0: not full<br>1: full   | R   | 0 |
| 5 | RXEM     | <b>SPI RX FIFO Empty</b><br>0: not empty<br>1: empty  | R   | 1 |
| 4 | RXFU     | <b>SPI RX FIFO Full</b><br>0: not full<br>1: full   | R   | 0 |
| 3 | SPI_BUSY | <b>SPI master mode busy status bit.</b><br>The bit is automatically clear when all data have been send out or received and SPISCK has finished; and automatically setup in transmitting/receiving status ; this bit is valid only in SPI master mode;<br>0: SPI idle status<br>1: SPI busy status | R   | 0 |
| 2 | TXER     | <b>SPI TX FIFO error Pending.</b> Writing 1 to this bit will clear it and reset the TX FIFO, otherwise unchanged.<br>This bit set when SPI TX FIFO is wrote overflow;   | R/W | 0 |
| 1 | RXER     | <b>SPI RX FIFO error Pending.</b> Writing 1 to this bit will clear it and reset the TX FIFO, otherwise unchanged.<br>This bit set when SPI RX FIFO is wrote or read overflow;   | R/W | 0 |
| 0 | -        | <b>Reserved, be read as zero</b>  | -   | - |

#### 11.2.6.4 SPI\_CLKDIV

SPI Clock Divide Control Register, This register is used for setting SPI source clock divide factor, and selecting SPI read mode. (SFR address 0x9c, SFR Bank = 0x0a)

| Bit(s) | Name       | Description  | R/W | Reset |
|--------|------------|--|-----|-------|
| 7      | -          | Reserve, be read as zero   | R   | 0     |
| 3:0    | SPI_CLKDIV | <b>SPI Clock Divide Factor [3:0]</b><br>0000 /1<br>0001 /2<br>0010 /4<br>...<br>1111 /30<br>SPI clock =<br>SPI source clock/ (SPI_CLKDIV[3:0]*2) | R/W | 1111  |

### 11.2.6.5 SPI\_TXDAT

SPI data register, this register is used for writing data to SPI TX FIFO.

(SFR address 0x9d, SFR Bank = 0x0a)

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | SPI_TXDAT | <b>SPI Data[7:0]</b><br>Writing this field will send 1 byte to 8bitx16 levels depth SPI TX FIFO, be read as zero. | W   | 0     |

### 11.2.6.6 SPI\_RXDAT

SPI data register, this register is used for reading data from SPI RX FIFO.

(SFR address 0x9e, SFR Bank = 0x0a)

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | SPI_RXDAT | <b>SPI Data[7:0]</b><br>Reading this field will fetch 1 byte from 8bitx16 levels depth SPI RX FIFO | R   | 0     |

### 11.2.6.7 SPI\_BCL

SPI Bytes Count Low Register, this register is used for setting SPI bytes counter low bits in the SPI read mode.

(SFR address 0x9f, SFR Bank = 0x0a)

| Bit(s) | Name    | Description                          | R/W | Reset |
|--------|---------|--------------------------------------|-----|-------|
| 7:0    | SPI_BCL | <b>Bytes Counter Low bits [7: 0]</b> | R/W | 0     |

### 11.2.6.8 SPI\_BCH

SPI Bytes Count High Register, This register is used to setting SPI bytes counter high bits, selecting SPI data I/O mode and delay chain. (SFR address 0xa2, SFR Bank = 0x0a)

| Bit(s) | Name         | Description   | R/W | Reset |
|--------|--------------|---|-----|-------|
| 7      | SPI_IO       | <b>SPI data I/O mode select</b> (valid when SPI select write or read only mode)<br>0: 1x I/O mode select<br>1: 2x I/O mode select | R/W | 0     |
| 6      | SPI_DELAY_EN | <b>SPI delay chain enable</b><br>0: Disable<br>1: Enable  | R/W | 0     |

|     |           |   |     |    |
|-----|-----------|---|-----|----|
| 5:4 | SPI_DELAY | SPI read clock delay time (valid when SPI select write/read and read mode)<br>00: delay 2 ns<br>01: delay 4 ns<br>10: delay 8 ns<br>11: delay 12 ns       | R/W | 00 |
| 3   | SPI_RS    | <b>Read Start Control</b> , write 1 to start read clock, valid when SPI select read only mode. (When transfer is finished, this bit will be auto cleared) | R/W | 0  |
| 2   | -         | <b>reserved</b>   | R   | 0  |
| 1:0 | SPI_BCH   | Bytes Counter High bits [1: 0]  | R/W | 00 |

### 11.2.6.9 SPI\_DEBUG

SPI debug register.

(SFR address 0xa3, SFR Bank = 0x0a)

| Bit(s) | Name      | Description   | R/<br>W | Rese<br>t |
|--------|-----------|---|---------|-----------|
| 7      | SPI_LB    | SPI_MOSI and SPI_MISO loopback enable<br>0: disable<br>1: enable  | R/W     | 0         |
| 6:0    | SPI_DEBUG | spi debug_signal output select<br>7'h0: {mosi_cpha0,miso_cpha0,mosi_cpha1,miso_cpha1,<br>sckt_cnt[2:0], rd_tff };//tx<br>7'h1: pe_din[7:0];<br>7'h2: rshift[7:0];<br>7'h3: pe_dout[7:0];<br>7'h4: {sckr_cnt[2:0],wr_rff,pe_inpro_r,nssi_sel,shifth,shiftl };//rx<br>7'h5: {spidrqr,spidrqt,drqte,tff_full,tff_empty,pe_empty_dd,<br>pe_inpro_dd, spi_inpro};<br>7'h6: {clr_tff,clr_rff,txrd_pulse,rxwr_pulse,tff_empty,tff_full,<br>rff_empty,rff_full} ;<br>7'h7: {tx_rptr[3:0],tx_sckt_wptr[3:0]}; //fifo<br>7'h8: {tx_bus_rptr[3:0],tx_wptr[3:0] };<br>7'h9: {rx_bus_wptr[3:0],rx_rptr[3:0] };<br>7'ha: {rx_wptr[3:0],rx_sckr_rptr[3:0]};<br>7'hb: rff_dout[7:0];<br>other: RESERVED | R/W     | 7'h0      |

### 11.2.6.10 SPI\_RAND

SPI Randomizer control register (SFR address 0xc8, SFR Bank = 0x0a)

| Bit(s) | Name         | Description  | R/W | Reset |
|--------|--------------|--|-----|-------|
| 7:5    | -            | Reserved   | R   | 0     |
| 4      | spi_3wire_rw | <b>SPI 3-wire read/write direction</b><br><b>0:write</b><br><b>1:read</b>  | R/W | 0     |
| 3      | spi_3wire_en | <b>SPI 3-wire mode enable</b><br><b>0:disable</b><br><b>1:enable</b>   | R/W | 0     |
| 2      | RAND_SEED    | Randomizer seed select<br>0: use effuse bits<br>1: use register SPI_RAND0/1 value<br>用于选择Randomizer使用的是efuse bits 还是 SPI_SEED0/1中的值。 | R/W | 0     |
| 1      | RAND_TXEN    | TX Randomizer enable<br>0 : disable<br>1 : enable  | R/W | 0     |
| 0      | RAND_RXEN    | RX Randomizer enable<br>0:disable<br>1:enable  | R/W | 0     |

### 11.2.6.11 SPI\_SEED0

SPI Randomizer test seed register (SFR address 0xc9, SFR Bank = 0x0a)

| Bit(s) | Name           | Description               | R/W | Reset |
|--------|----------------|---------------------------|-----|-------|
| 7:0    | RAND_TST_SEED0 | Randomizer seed0 for test | RW  | 0     |

### 11.2.6.12 SPI\_SEED1

SPI Randomizer test seed register (SFR address 0xca, SFR Bank = 0x0a)

| Bit(s) | Name           | Description               | R/W | Reset |
|--------|----------------|---------------------------|-----|-------|
| 7:0    | RAND_TST_SEED1 | Randomizer seed1 for test | RW  | 0     |

备注：

1. 加入 16 阶的 Randomizer 电路，初始化的 seed 可以选择 efuse 值或者 SPI\_SEED0/1 寄存器中的值。
2. 为了增强保密，如果 Randomizer 使能，禁止 cpu 访问 SPI RX-FIFO，只能用 DMA 的方式读取数据。

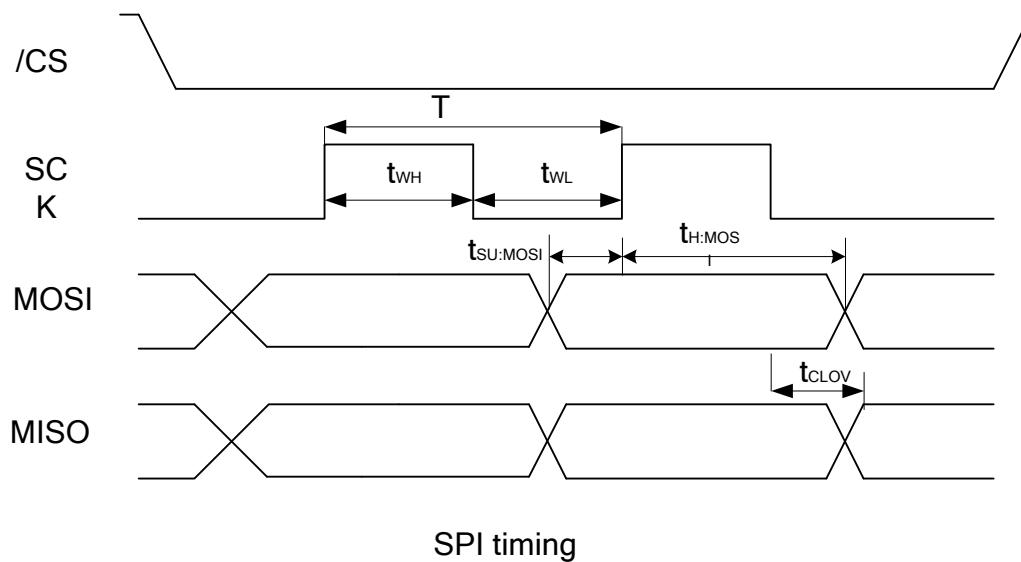
## 11.2.7 TESTMODE

### 11.2.7.1 BLOCK DIAGRAM

### 11.2.7.2 SIGNAL DESCRIPTION

### 11.2.7.3 TEST METHOD

## 11.2.8 Electronics parameter



SPI timing

Figure 11-19 SPI timing

| Parameter                 | Symbol               | MIN | TYP | MAX | Unit |
|---------------------------|----------------------|-----|-----|-----|------|
| SCK Clock                 | fclk                 | -   | 24  | 60  | MHz  |
| SCK High time             | t <sub>WH</sub>      | 8   | 21  | -   | ns   |
| SCK Low time              | t <sub>WL</sub>      | 8   | 21  | -   | ns   |
| SCK rise time             | t <sub>r</sub>       | -   | 0.5 | -   | ns   |
| SCK fall time             | t <sub>f</sub>       | -   | 0.6 | -   | ns   |
| Data output setup time    | t <sub>SU:MOSI</sub> | -   | 2   | -   | ns   |
| Data output hold time     | t <sub>H:MOSI</sub>  | -   | 24  | -   | ns   |
| Clock low to output valid | T <sub>clov</sub>    | -   | -   | 9   | ns   |

## 11.3 UART 子模块（胡世飞、孙睿）

| 日期         | 版本     | 描    述                 | 修订人 |
|------------|--------|------------------------|-----|
| 2012-07-24 | V1. 00 | initial                | 胡世飞 |
| 2012-09-12 | V1. 02 | UART_CLK 增加支持 6M 波特率   | 胡世飞 |
| 2012-10-20 | V2. 00 | 增加 UART 时序错误状态位        | 胡世飞 |
| 2010-12-10 | V2. 02 | 完善 operation manual 章节 | 胡世飞 |

### 11.3.1 Features

- support BaudRate up to 1.5Mbps;
- support UART auto flow mode;

### 11.3.2 Function Description

The UART module is designed according to UART protocols. It can generate a large range of standard baud rate so as to communicate with different devices.

UART module support MCU access to UART data. The command instructions, addresses or data you write to UART TX FIFO will be transfer to UART\_TX pin immediately. When you read a byte from UART RX FIFO, the RX FIFO pointer will decrease one until it point to the bottom of the RX FIFO.

UART module also support DMA access to UART data.

Especially, UART module support auto flow mode to control the data transiting sequence.

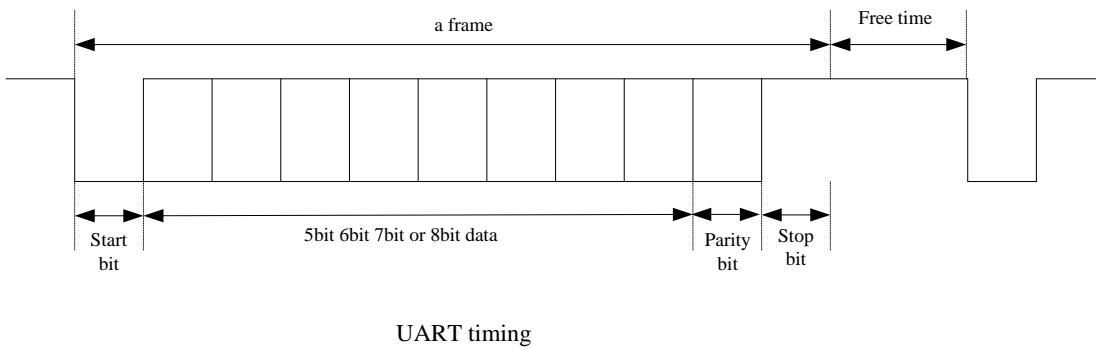


Figure 11-20 Uart Timing

RTS/CTS hardware Autoflow control flow description:

When the receiver FIFO level reaches the trigger level of 14 bytes, RTS- will be pulled up to invalid state. The sending UART may send an additional byte after the trigger level is reached (in case the sending UART has

another byte to send) because it may not recognize the invalid state of RTS- until after it has begun sending the additional byte. RTS- is automatically reasserted once the receiver FIFO is emptied by reading the receiver buffer register. The reassertion signals the sending UART to continue transmitting data.

The transmitter checks CTS- before sending the next data byte. When CTS- is active, the transmitter sends the next byte. To stop the transmitter from sending the following byte, CTS- must be released before the middle of the last stop bit that is currently being sent.

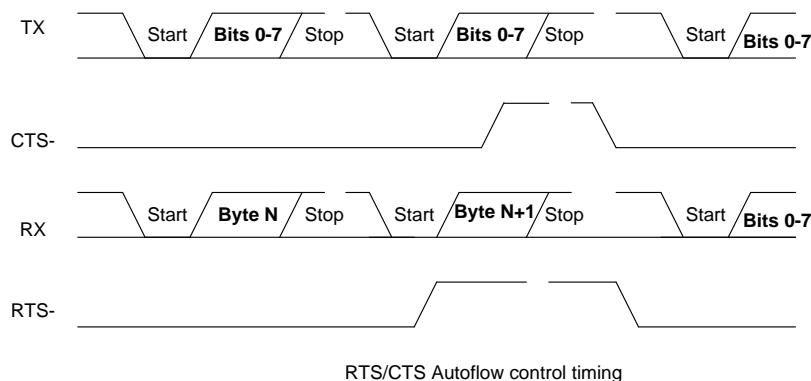


Figure 11-21 RTS/CTS Autoflow control Timing

#### UARTBaudRate

The clock to drive UART controller is HOSC. The UART Baud Rate can be selected by UART\_CTL register, Special Note:

~~When receive data by IRQ mode, if the FIFO number received not meet the trigger level, wait until the time is out of the trigger level data bytes.~~

Trigger level is 1 (RX FIFO not empty) .

### 11.3.3 Module Description

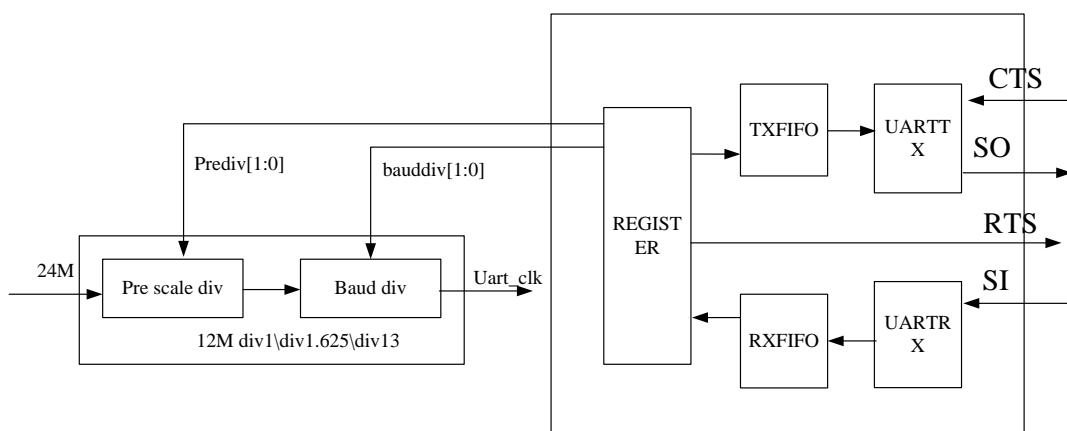


Figure 11-22 Uart Block Diagram

## 11.3.4 Operation Manual

### 11.3.4.1 软件控制流程

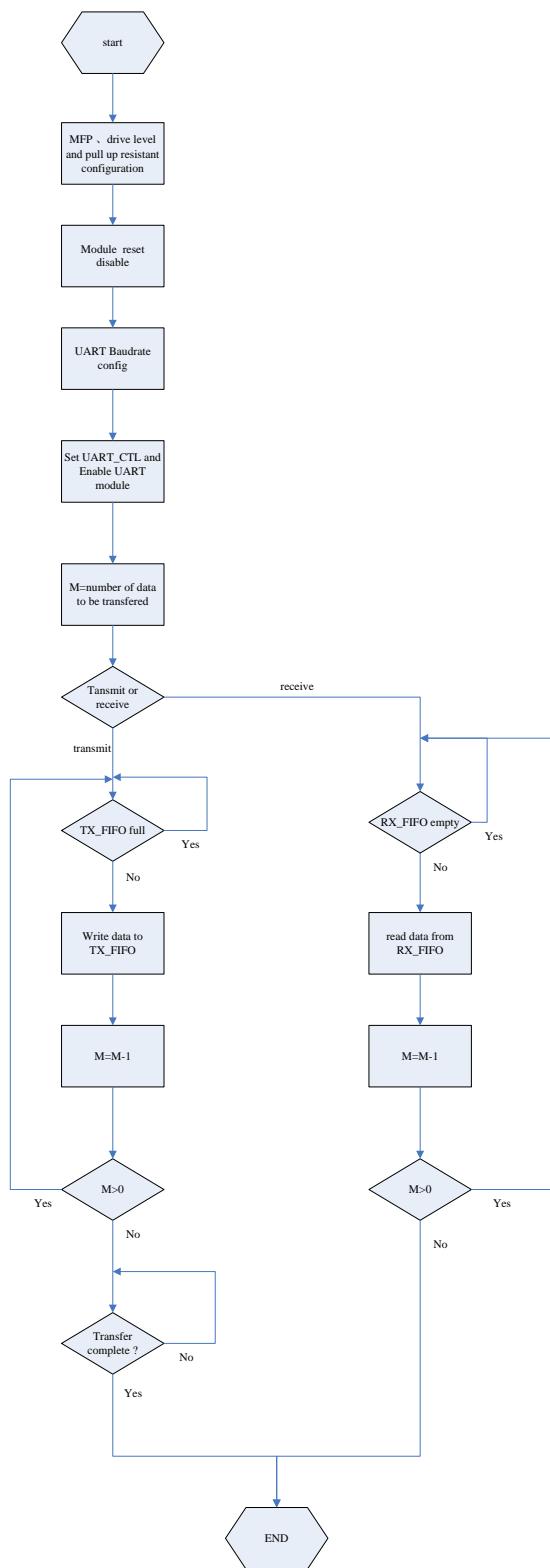


Figure 11-23 UART transmit and receive flow

UART 模块使用说明:

2. UART 支持 normal 模式 (UART\_TX、UART\_RX) 和 auto flow control mode 模式 (UART\_TX、UART\_RX、UART\_RTS、UART\_CTS)。在 auto flow control mode 下, 当 RX FIFO 中的数据达到 14 个时, RTS 自动拉高, 当小于 14 时再次拉低, 这样可以控制对方数据的发送, 使得 RX FIFO 不至于被写爆。同理, 在 auto flow control mode 下当 CTS 被拉高后, UART TX 不再发送数据。
3. UART baudrate 通过 UART\_MODE [7:6]和 UART\_BR 配置:  $BaudRate = \text{standard BaudRate}/(\text{BaudRate divider} + 1)$ , 最高支持到 6M baudrate。
4. UART 可以通过 DMA 收发数据。TX 触发 DMA 的条件是 FIFO 中的剩余空间大于等于 8 Bytes (因为 DMA 为 Burst8 模式, 每次启动至少传递 8 Bytes), RX 触发 DMA 的条件是 FIFO 中接收到的数据大于等于 8 Bytes。注意当数据少于 8 Bytes, 不能用 DMA 收发, 可以用查询或者 IRQ 方式。
5. UART 可以使用 IRQ 方式收发数据。TX FIFO 为空或者 RX FIFO 非空都会触发 IRQ。
6. 通过查询 UART\_BUSY (UART\_STA.3) 可以检测当前的 TX 数据是否已经发送完毕。
7. UART 支持数据自发自收测试, 可以用于 CP/FT 测试, 内部 loop back 不需要外挂器件。

### 11.3.5 UART Register List

SFR BANK: 0x0a

| Index | Mnemonic   | Description               | Bank |
|-------|------------|---------------------------|------|
| 0x90  | UART_BR    | UART BAUDRATE Register.   | 0x0a |
| 0x91  | UART_MODE  | UART mode setup Register. | 0x0a |
| 0x92  | UART_CTL   | UART Control Register.    | 0x0a |
| 0x93  | UART_DRQ   | UART DRQ/IRQ register     | 0x0a |
| 0x94  | UART_STA   | UART Status Register      | 0x0a |
| 0x95  | UART_TXDAT | UART TX FIFO register     | 0x0a |
| 0x97  | UART_RXDAT | UART RX FIFO register     | 0x0a |
| 0x98  | UART_DEBUG | UART debug register.      | 0x0a |

#### 11.3.1 Register Description

##### 11.3.1.1 UART\_BR

UART BAUDRATE Register.

(SFR address 0x90, SFR Bank = 0x0a)

| Bit(s) | Name    | Description   | R/W | Reset |
|--------|---------|---|-----|-------|
| 7:0    | UART_BR | UART BAUDRATE divider<br>BaudRate<br>= standard BaudRate/(BaudRate divider + 1) | R/W | 0     |

##### 11.3.1.2 UART\_MODE

UART mode setup Register.

(SFR address 0x91, SFR Bank = 0x0a)

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:6    | UART_CLK  | <b>UART standard BaudRate select</b><br>0 0 /select 115200 standard BaudRate<br>0 1 /select 921600 standard BaudRate<br>1 0 /select 1.5M standard BaudRate<br>1 1 /select 6M standard BaudRate | R/W | 00    |
| 5:4    | UART_PA   | <b>UART parity select</b><br>00: no parity<br>01: no parity<br>10: odd parity<br>11: even parity   | R/W | 00    |
| 3      | UART_STOP | <b>UART TX stop bits select</b><br>0: 1 stop bit is generated.<br>1: 2 stop bit is generated.<br>UART RX always just check 1 stop bit in receiving process.                                    | R/W | 0     |
| 2      | -         | <b>Reserved, be read as zero.</b>  | -   | -     |
| 1:0    | UART_BW   | <b>UART bit width select</b><br>0 0 8 bits<br>0 1 7 bits<br>1 0 6 bits<br>1 1 5 bits   | R/W | 00    |

### 11. 3. 1. 3 UART\_CTL

UART Control Register.

(SFR address 0x92, SFR Bank = 0x0a)

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7      | UART_TX_EN | <b>UART TX enable</b><br>0: disable<br>1: enable  | R/W | 0     |
| 6      | UART_RX_EN | <b>UART RX enable</b><br>0: disable<br>1: enable  | R/W | 0     |
| 5      | U_FLOW_CTL | <b>UART flow control mode select</b><br>0: normal mode<br>1: auto flow control mode enable<br><b>UART TX/RX don't care CTS/RTS pin status in normal mode;</b> | R/W | 0     |

|     |          |   |   |   |
|-----|----------|---|---|---|
|     |          | RTS will be controlled by hardware and UART TX will be controlled by CTS in auto flow control mode; |   |   |
| 4:2 | -        | <b>Reserved, be read as zero.</b>   | - | - |
| 1   | UART_RTS | <b>UART RTS pin status.</b><br>0: RTS pin low status<br>1: RTS pin high status                      | R | 0 |
| 0   | UART_CTS | <b>UART CTS pin status.</b><br>0: CTS pin low status<br>1: CTS pin high status                      | R | 0 |

### 11. 3. 1. 4 UART\_DRQ

UART DRQ/IRQ control Register.

(SFR address 0x93, SFR Bank = 0x0a)

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7      | UT_DRQ_EN | <b>UART TX DRQ enable</b> , trigger DRQ when UART TX FIFO at least 8 level empty; When DMA remain counter < 8, trigger DRQ until all data transfer completely;<br>0: disable DRQ<br>1: enable DRQ | R/W | 0     |
| 6      | UR_DRQ_DN | <b>UART RX DRQ enable</b> , trigger DRQ when UART RX FIFO at least 8 level full; When DMA remain counter < 8, trigger DRQ until all data received completely;<br>0: disable DRQ<br>1: enable DRQ  | R/W | 0     |
| 5      | -         | <b>Reserved</b>   | -   | -     |
| 4      | UR_TMERR  | <b>UART received timing error pending bit</b><br>0: no err<br>1: timing err<br>This bit set when RX data disobey UART standard baudrate. Write 1 to clear it, otherwise unchanged.                | R/W | 0     |
| 3      | UT_IRQ_EN | <b>UART TX IRQ enable</b> , trigger IRQ when UART TX FIFO is empty.<br>0: disable<br>1: enable  | R/W | 0     |
| 2      | UR_IRQ_EN | <b>UART RX IRQ enable</b> , trigger IRQ when UART RX FIFO is not empty.<br>0: disable<br>1: enable  | R/W | 0     |

|   |           |   |     |   |
|---|-----------|---|-----|---|
| 1 | UT_IRQ_PD | UART TX IRQ pending bit. Write 1 to clear it.<br>0: not IRQ pending<br>1: IRQ pending | R/W | 0 |
| 0 | UR_IRQ_PD | UART RX IRQ pending bit. Write 1 to clear it.<br>0: not IRQ pending<br>1: IRQ pending | R/W | 0 |

### 11. 3. 1. 5 UART\_STA

UART Status Register, This register is used for displaying current UART FIFO status.

(SFR address 0x94, SFR Bank = 0x0a)

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7      | TXEM      | <b>UART TX FIFO Empty</b><br>0: not empty<br>1: empty   | R   | 1     |
| 6      | TXFU      | <b>UART TX FIFO Full</b><br>0: not full<br>1: full  | R   | 0     |
| 5      | RXEM      | <b>UART RX FIFO Empty</b><br>0: not empty<br>1: empty   | R   | 1     |
| 4      | RXFU      | <b>UART RX FIFO Full</b><br>0: not full<br>1: full  | R   | 0     |
| 3      | UART_BUSY | <b>UART transmitting status bit.</b><br>The bit is automatically clear when uart finish transmitted data; and set in transmitting status automatically;<br>0: uart idle status<br>1: uart transmitting status | R   | 0     |
| 2      | TXER      | <b>UART TX FIFO error Pending.</b> Writing 1 to this bit will clear it and reset the TX FIFO, otherwise unchanged.<br>This bit set when SPI TX FIFO is wrote overflow;  | R/W | 0     |
| 1      | RXER      | <b>UART RX FIFO error Pending.</b> Writing 1 to this bit will clear it and reset the RX FIFO, otherwise unchanged.<br>This bit set when SPI RX FIFO is wrote or read overflow;                                | R/W | 0     |
| 0      | UART_ERR  | <b>UART received error pending bit</b><br>This bit set when RX data disobey UART parity   | R/W | 0     |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | checkout. Write 1 to clear it, otherwise unchanged. |  |  |
|--|--|---|--|--|

### 11.3.1.6 UART\_TXDAT

UART tx fifo register, this register is used for writing data to UART TX FIFO.

(SFR address 0x95, SFR Bank = 0x0a)

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | UART_TXDAT | <b>UART Data[7:0]</b><br>Writing this field will send 1 byte to 8bitx8 levels depth UART TX FIFO. | W   | 0     |

### 11.3.1.7 UART\_RXDAT

UART rx fifo register, this register is used for reading data from UART RX FIFO.

(SFR address 0x97, SFR Bank = 0x0a)

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | UART_RXDAT | <b>UART Data[7:0]</b><br>Writing this field will send 1 byte to 8bitx8 levels depth UART TX FIFO. | R   | 0     |

### 11.3.1.8 UART\_DEBUG

UART debug register.

(SFR address 0x98, SFR Bank = 0x0a)

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7      | U_TXRX_LB  | <b>UART TX/RX loopback enable</b><br>0: disable<br>1: enable  | R/W | 0     |
| 6      | U_RCTS_LB  | <b>UART RTS/CTS loopback enable</b><br>0: disable<br>1: enable  | R/W | 0     |
| 5:0    | UART_DEBUG | uart debug_signal output select:<br>0: disable debug signal output<br>1: tx_std_cnt;<br>2: tx_div_cnt;<br>3:{4'd0,tx_start, tx_enable, tx_std_end, tx_div_end}<br>4: { 2'd0, tx_cs, tx_bit_cnt }<br>5:{ 2'd0, rx_stop_cnt_ctrl, rx_std_cnt_clr, rx_std_end, rx_bit_sample, rx_rise_pulse, | R/W | 6'h0  |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | <pre> rx_fall_pulse} 6: rx_div_cnt 7:{      3'd0,      rx_div_end,      rx_div_cnt_clr, bitbound_forbid, rx_pulse_error, rx_parity_error} 8: { 2'd0, rx_cs, rx_bit_cnt} 9: {3'd0, tx_fifo_level} 10: {3'd0, rx_fifo_level} other: RESERVED </pre> |  |  |
|--|--|---|--|--|

## 11.3.2 TESTMODE

### 11.3.2.1 BLOCK DIAGRAM

### 11.3.2.2 SIGNAL DESCRIPTION

### 11.3.2.3 TEST METHOD

## 11.3.3 Timing fault-tolerant parameter

UART 三种标准波特率，误差率测量如下：

| 115200 标准波特率 |              |          |       |
|--------------|--------------|----------|-------|
| 波特率          | 理论位宽<br>(us) | 实测位宽(us) | 误差率   |
| 1200         | 833          | 832      | 0.12% |
| 2400         | 416. 67      | 416      | 0.16% |

|        |         |        |       |
|--------|---------|--------|-------|
| 3600   | 277. 8  | 277. 5 | 0.11% |
| 4800   | 208. 33 | 208    | 0.15% |
| 7200   | 138. 8  | 138. 8 | 0     |
| 9600   | 104. 17 | 104    | 0.16% |
| 14400  | 69. 4   | 69. 3  | 0.14% |
| 19200  | 52. 08  | 52     | 0.15% |
| 28800  | 34. 7   | 34. 65 | 0.14% |
| 38400  | 26. 04  | 26     | 0.15% |
| 57600  | 17. 36  | 17. 35 | 0.05% |
| 115200 | 8. 68   | 8. 66  | 0.02% |

| 921600 标准波特率 |              |          |       |
|--------------|--------------|----------|-------|
| 波特率          | 理论位宽<br>(us) | 实测位宽(us) | 误差率   |
| 3600         | 277. 8       | 277. 5   | 0.11% |
| 4800         | 208. 33      | 208      | 0.16% |
| 7200         | 138. 8       | 138. 8   | 0     |
| 9600         | 104. 17      | 104      | 0.16% |
| 14400        | 69. 4        | 69. 3    | 0.14% |
| 19200        | 52. 08       | 52       | 0.15% |
| 28800        | 34. 7        | 34. 7    | 0     |
| 38400        | 26. 04       | 26       | 0.15% |
| 57600        | 17. 36       | 17. 32   | 0.23% |
| 115200       | 8. 68        | 8. 66    | 0.23% |
| 230400       | 4. 34        | 4. 33    | 0.23% |
| 460800       | 2. 17        | 2. 165   | 0.23% |
| 921600       | 1. 085       | 1. 084   | 0.1%  |

| 1.5M 波特率 |                 |          |       |
|----------|-----------------|----------|-------|
| 波特率      | 理 论 位 宽<br>(us) | 实测位宽(us) | 备注    |
| 7200     | 138. 8          | 138. 8   | 0     |
| 9600     | 104. 17         | 104      | 0.16% |
| 14400    | 69. 4           | 69. 3    | 0.14% |
| 19200    | 52. 08          | 52       | 0.15% |
| 28800    | 34. 7           | 34. 65   | 0.14% |
| 38400    | 26. 04          | 26       | 0.15% |
| 57600    | 17. 36          | 17. 34   | 0.12% |
| 115200   | 8. 68           | 8. 68    | 0     |
| 750000   | 1. 33           | 1. 334   | 0.3%  |
| 1500000  | 0. 667          | 0. 666   | 0.15% |

## 11.4 IR 子模块 (胡世飞、蔡瑞仁)

| 日期         | 版本     | 描述                     | 修订人 |
|------------|--------|------------------------|-----|
| 2012-07-24 | V1. 00 | initial                | 胡世飞 |
| 2012-10-20 | V2. 00 | 删除 IR 模拟部分相关寄存器        | 胡世飞 |
| 2012-12-12 | V2. 02 | 完善 operation manual 章节 | 胡世飞 |

### 11.4.1 Features

- Infrared remote control hardware decoder.
- Support three infrared remote control decode mode: Toshiba 9012 code、8 bits NEC code、Philips RC5 code.

### 11.4.2 Function Description

IRC interface is designed according to Toshiba 9012 code timing、8 bits NEC code timing、Philips RC5 code timing.

IRC interface support MCU access to IRC data register.

#### 2013、 9012 Protocol

The 9012 protocol uses a pulse distance encoding of the bits. Each pulse is one Tm ( $560\mu s$ ) long 38kHz carrier burst. A logical “1” takes 4Tm (2.25ms) to transmit, while a logical “0” is only 2Tm (1.12ms). The recommended carrier duty-cycle is 1/4 or 1/3.

Tm=256/Fosc=0.56ms (Fosc=455kHz)

Repetition time=192Tm=108ms

Carrier frequency = Fosc/12

- 8 bit customer code and 8 bit command code length
- customer and command are transmitted twice for reliability
- Pulse distance modulation
- Bit time of 2Tm(1.12ms) for logic “0” or 4Tm (2.25ms) for logic “1”

With this protocol the LSB is transmitted first. In this case Customer code and Command is transmitted. A message is started by 8Tm (4.5ms) AGC burst, which was used to set the gain of the earlier IR receivers. This AGC burst is then followed by 8Tm (4.5ms) space, which is then followed by the Customer code and Command.

Customer code and Command are transmitted twice. The second time the command bits are inverted and can be used for verification of the received message.

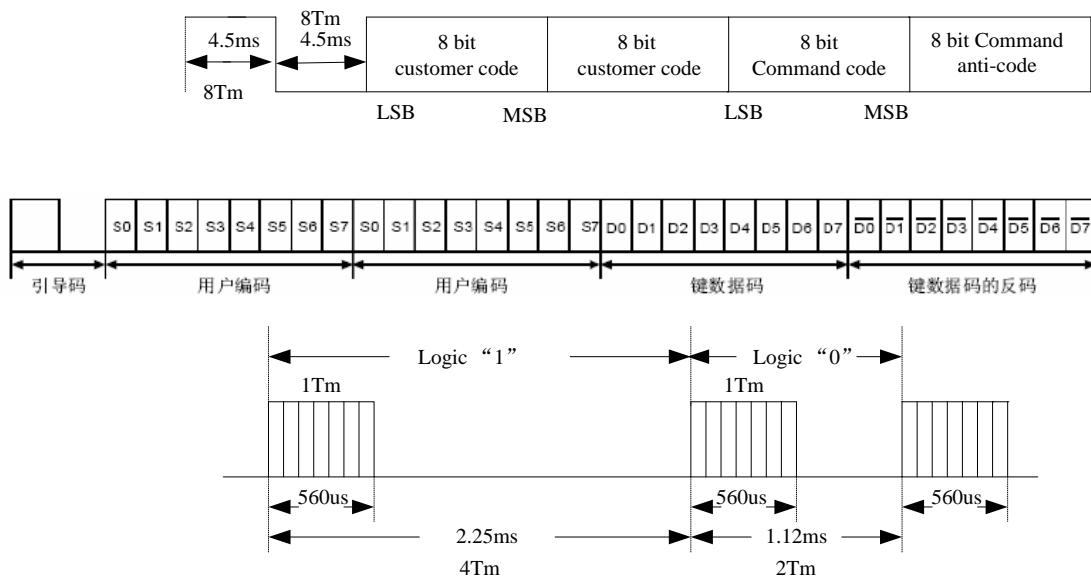


Figure 11-24 9012 data format

A command is transmitted only once, even when the key on the remote control remains pressed. Every 192Tm a repeat code is transmitted for as long as the key remains down. This repeat code is simply one 8Tm (4.5ms) AGC pulse followed by one 8Tm (4.5ms) space and a logic "1" +1Tm (560μs) burst.

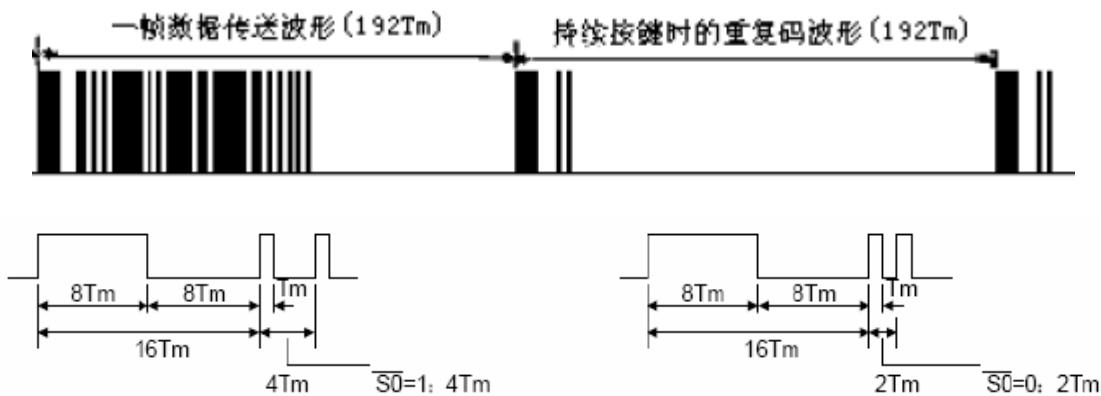


Figure 11-25 9012 repeat code timing

## 2014、NEC Protocol

The NEC protocol uses a pulse distance encoding of the bits. Each pulse is one Tm (560μs) long 38kHz carrier burst. A logical "1" takes 4Tm (2.25ms) to transmit, while a logical "0" is only 2Tm (1.12ms). The recommended carrier duty-cycle is 1/4 or 1/3.

$$Tm = 256/Fosc = 0.56ms \quad (Fosc = 455kHz)$$

$$\text{Repetition time} = 192Tm = 108ms$$

Carrier frequency = Fosc/12

- 8 bit customer and 8 bit command length
- customer and command are transmitted twice for reliability
- Pulse distance modulation
- Bit time of 2Tm(1.12ms) for logic “0” or 4Tm (2.25ms) for logic “1”

With this protocol the LSB is transmitted first. In this case Customer code and Command is transmitted. A message is started by 16Tm (9ms) AGC burst, which was used to set the gain of the earlier IR receivers. This AGC burst is then followed by 8Tm (4.5ms) space, which is then followed by the Customer code and Command. Customer code and Command are transmitted twice. The second time all bits are inverted and can be used for verification of the received message. The total transmission time is constant because every bit is repeated with its inverted length.

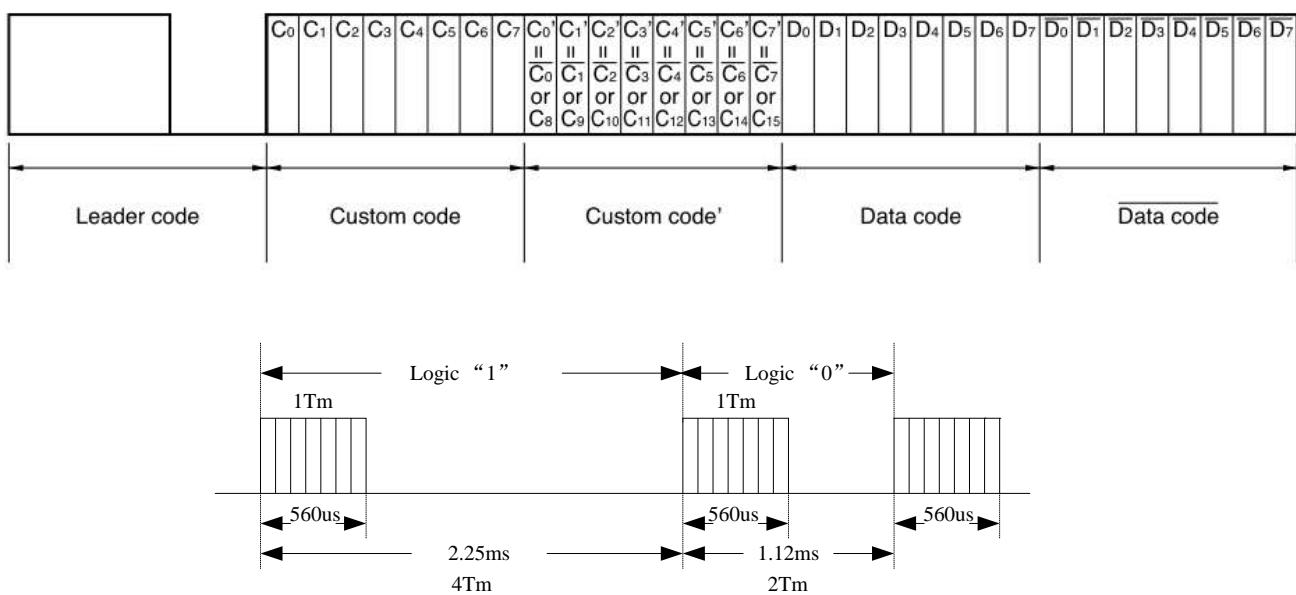
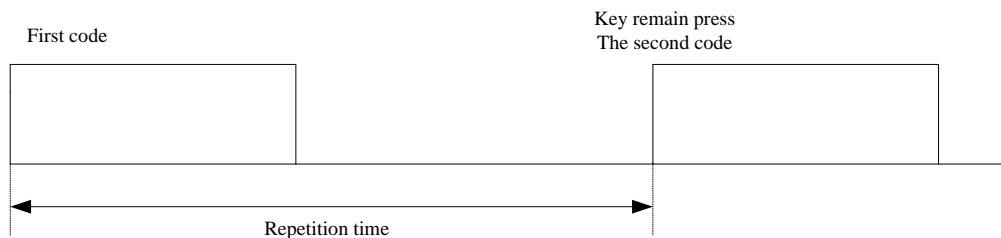


Figure 11-26 NEC data format

A command is transmitted only once, even when the key on the remote control remains pressed. Every 192Tm a repeat code is transmitted for as long as the key remains down. This repeat code is simply a 16Tm (9ms) AGC pulse followed by a 4Tm (2.25ms) space and one Tm (560  $\mu$ s) burst.



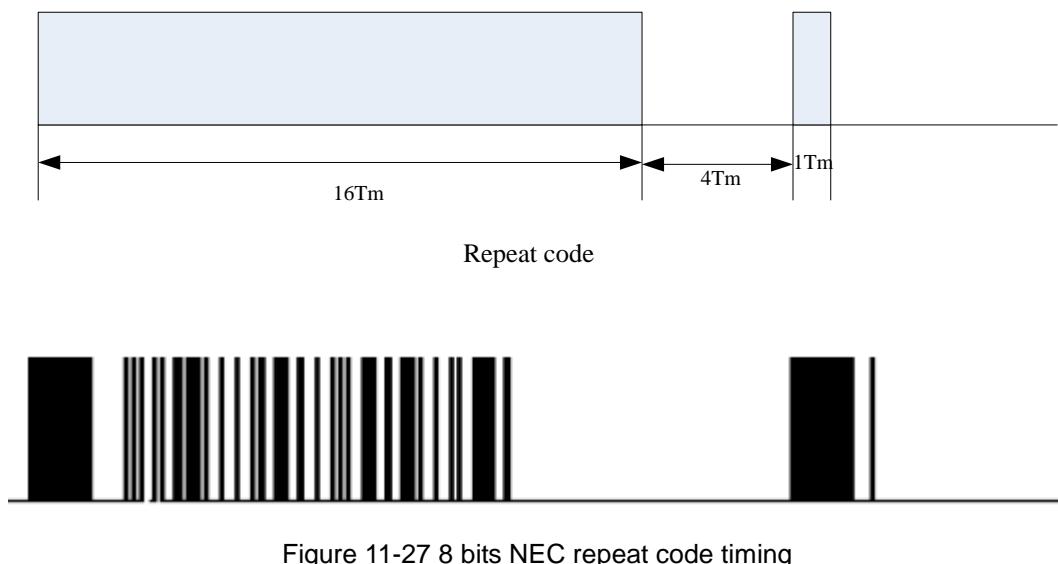


Figure 11-27 8 bits NEC repeat code timing

## 2015、 RC5 Protocol

The protocol uses bi-phase modulation (or so-called Manchester coding) of a 38kHz IR carrier frequency. All bits are of equal length of 1.8ms in this protocol, with half of the bit time filled with a burst of the 38kHz carrier and the other half being idle. A logical zero is represented by a burst in the first half of the bit time. A logical one is represented by a burst in the second half of the bit time. The pulse/pause ratio of the 38kHz carrier frequency is 1/3 or 1/4, to reduce power consumption.

$$1 \text{ bit-time} = 3 \times 256 / \text{Fosc} = 1.688\text{ms} (\text{Fosc}=455\text{kHz})$$

$$\text{Tm} = 1 \text{ bit-time}/2 = 0.844\text{ms}$$

$$\text{Repetition time} = 4 \times 16 \times 2\text{Tm} = 108\text{ms}$$

$$\text{Carrier frequency} = \text{Fosc}/12$$

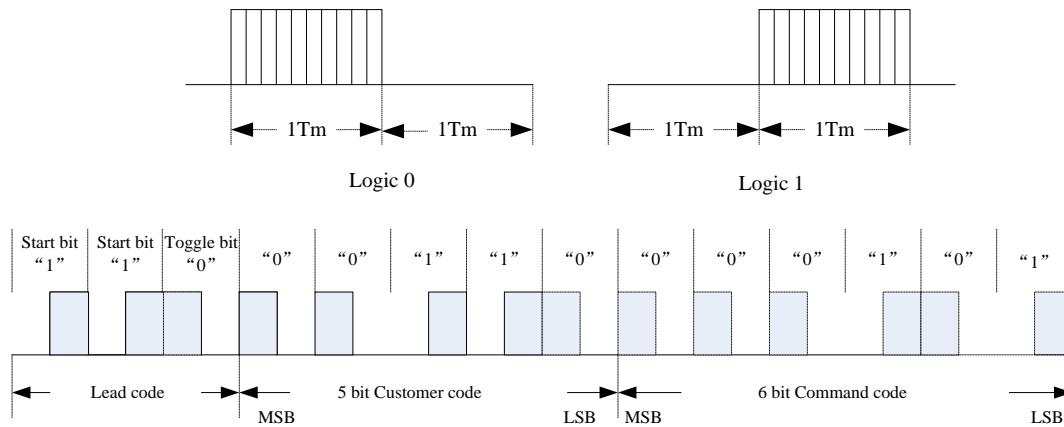


Figure 11-28 RC5 data format

The first two pulses are the start pulses, and are both logical “1”. Please note that half a bit time is elapsed before the receiver will notice the real start of the message.

The 3d bit is a toggle bit. This bit is inverted every time a key is released and pressed again. This way the

receiver can distinguish between a key that remains down, or is pressed repeatedly.

The next 5 bits represent the IR device address, which is sent with MSB first. The address is followed by a 6 bit command, again sent with MSB first.

A message consists of a total of 14 bits, which adds up to a total duration of 28Tm. Sometimes a message may appear to be shorter because the first half of the start bit S1 remains idle. And if the last bit of the message is a logic “0” the last half bit of the message is idle too.

As long as a key remains down the message will be repeated every 128Tm(108ms). The toggle bit will retain the same logical level during all of these repeated messages. It is up to the receiver software to interpret this auto repeat feature.

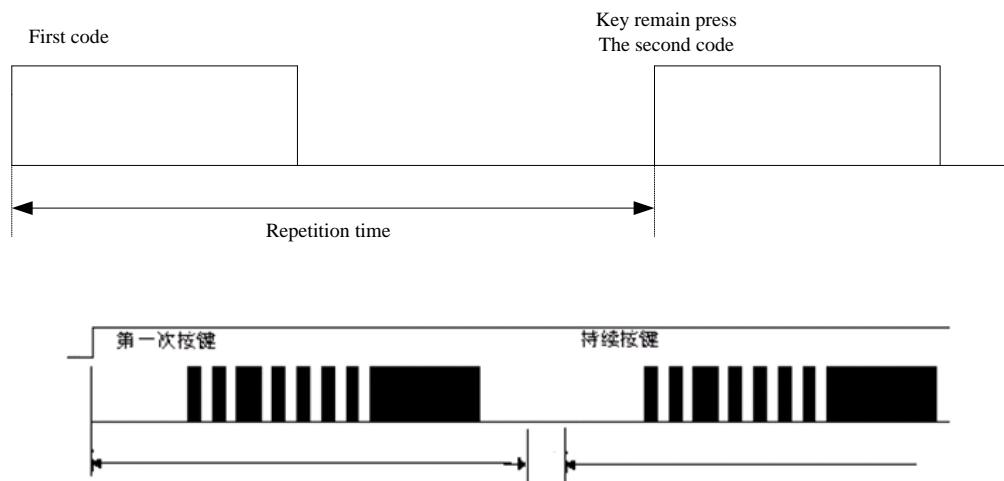


Figure 11-29 Philips RC5 repeat code timing

## 2016、RC6 Protocol

Only support RC6 mode 0.

RC-6 signals are modulated on a 36 kHz Infra Red carrier. The duty cycle of this carrier has to be between 25% and 50%.

Data is modulated using Manchester coding. This means that each bit (or symbol) will have both a mark and space in the output signal. If the symbol is a “1” the first half of the bit time is a mark and the second half is a space. If the symbol is a “0” the first half of the bit time is a space and the second half is a mark.

The main timing unit is 1t, which is 16 times the carrier period ( $1/36k * 16 = 444\mu s$ )

$$1T = 1 \times 16 / 36K = 444\mu s$$

$$1Bit = 2T = 888\mu s$$

Transmission time= Total 22 Bits= 23.1 ms (message) + 2.7 ms (no signal)

Repetition time=240T= 106.7ms

|        |    |     |     |         |             |         |             |
|--------|----|-----|-----|---------|-------------|---------|-------------|
| LS     | SB | mb2 | mb0 | TR      | a7...a0     | c7...c0 | signal free |
| Header |    |     |     | Control | information |         |             |

The signal frame as below:

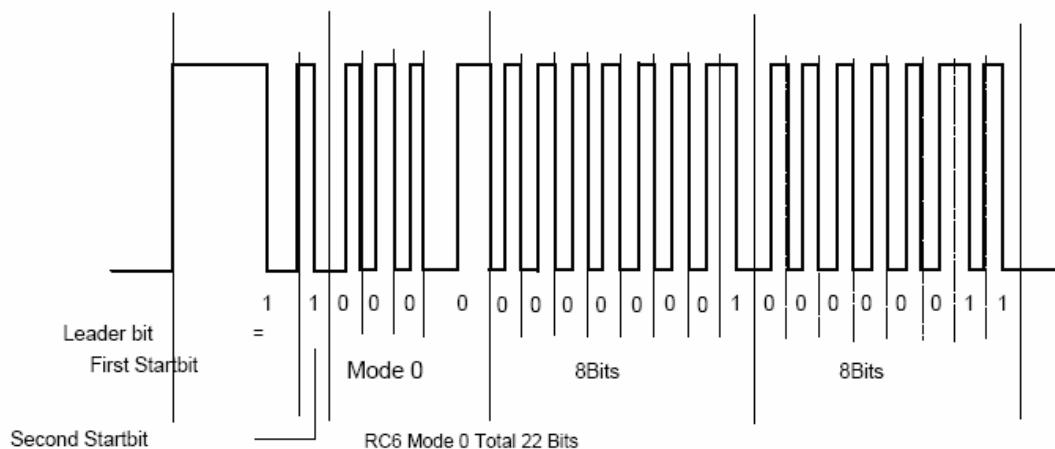
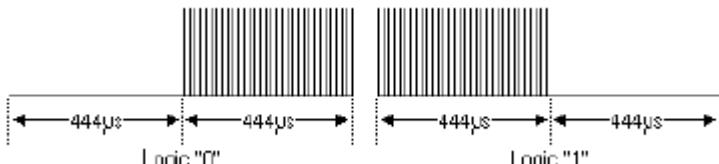


Figure 11-30 RC6 data format

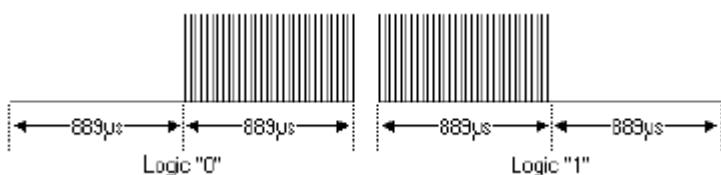
The leader pulse, which has a mark time of  $6t$  (2.666ms) and a space time of  $2t$  (0.889ms). This leader pulse is normally used to set the gain of the IR receiver unit.



Normal bits, which have a mark time of 1t (0.444ms) and space time of 1t (0.444ms). A “0” and “1” are encoded by the position of the mark and space in the bit time.



Trailer bits TR, which have a mark time of  $2t$  (0.889ms) and a space time of  $2t$  (0.889ms). Again a “0” and “1” are encoded by the position of the mark and space in the bit time. This bit serves as the traditional toggle bit, which will be inverted whenever a key is released. This allows the receiver to distinguish between a new key or a repeated key.



Control field:

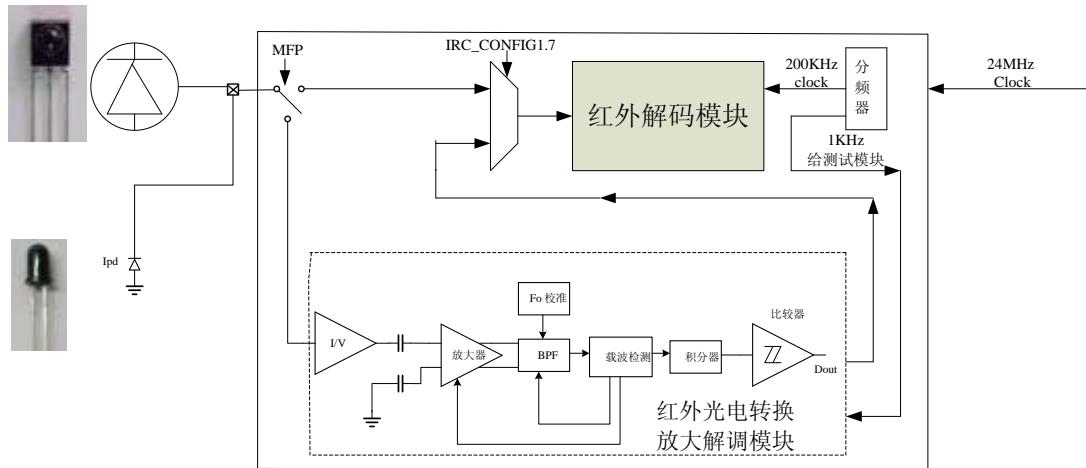
This field holds 8 bits which are used as address byte. This means that a total of 256 different devices can be controlled using mode 0 of RC-6. The msb is transmitted first.

Information field:

The information field holds 8 bits which are used as command byte. This means that each device can have up to 256 different commands. The msb is transmitted first.

## 2017、 红外放大解调模块

GL5110 增加红外接收二极管接收红外遥控的处理电路，方框图如下：



红外接收模块方框图

Figure 11-31 红外接收模块方框图

它工作原理是：红外接收二极管的电流(1Ua-10Ma)首先经过 I/V 转换为电压值，放大器再将电压放大到适当的范围内，BPF 以载波频率(36KHz、38KHz 或者 40KHz)为中心，滤除带外信号的干扰。通过检测信号的幅度和频率载波检测模块可以自动调节前端放大器的增益和 BPF 的中心频率。最后经过积分器（？）和比较器输出红外时序。

红外接收二极管固定使用 GPIO\_A3 引脚。数字红外模块通过 IRC\_CONFIG1.7 IRSS 选择 IRC 的信号源为外部的红外接收二极管或者是内部的红外接收一体头。

需要注意的是，红外一体头出来的波形和内嵌红外放大解调模块出来的的信号是同相的，如下图：

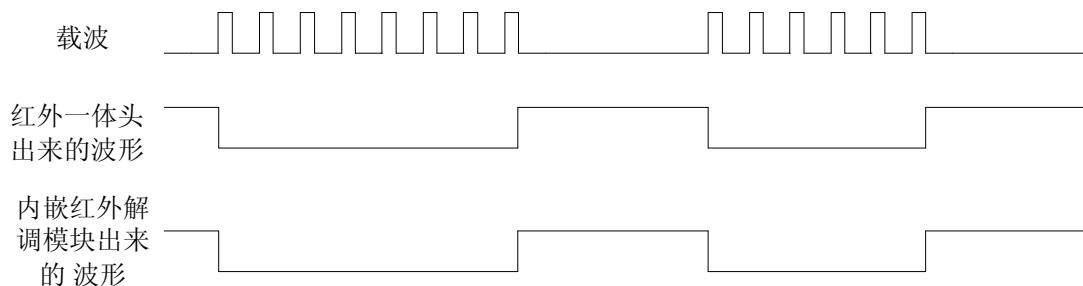


Figure 11-32 红外接收波形图

### 11.4.3 Module Description

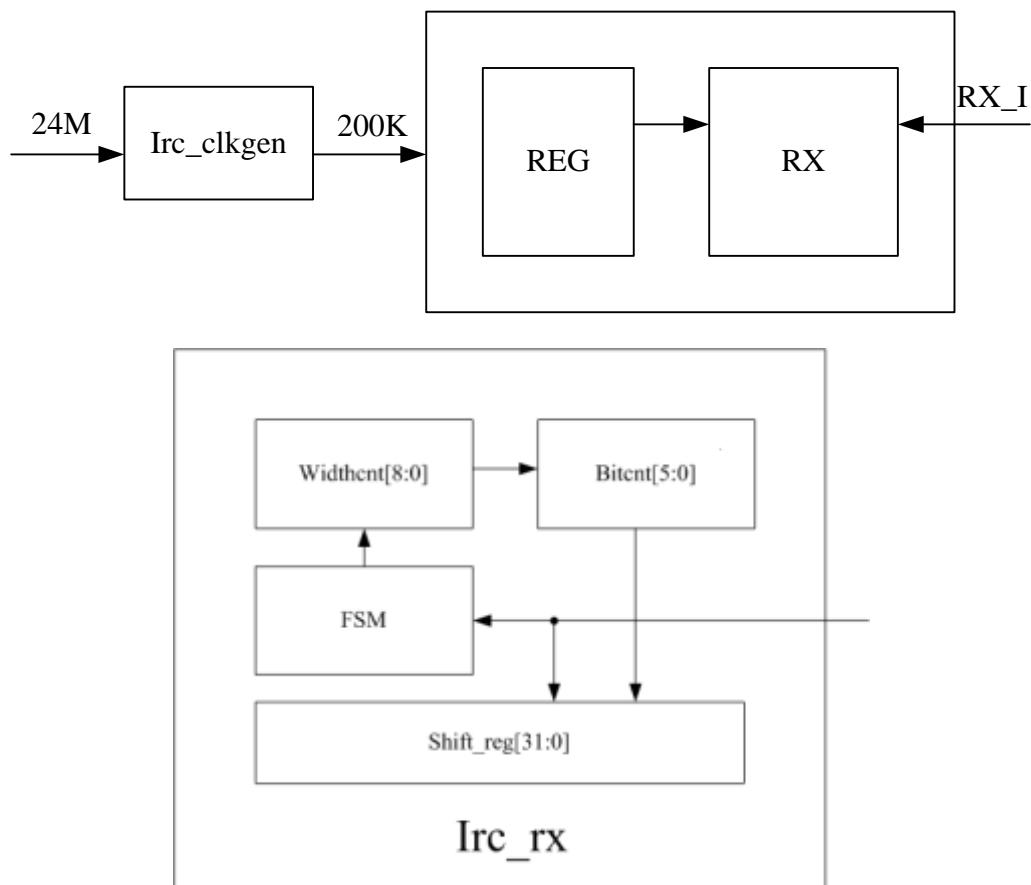


Figure 11-33 红外接收模块结构图

## 11.4.4 Operation Manual

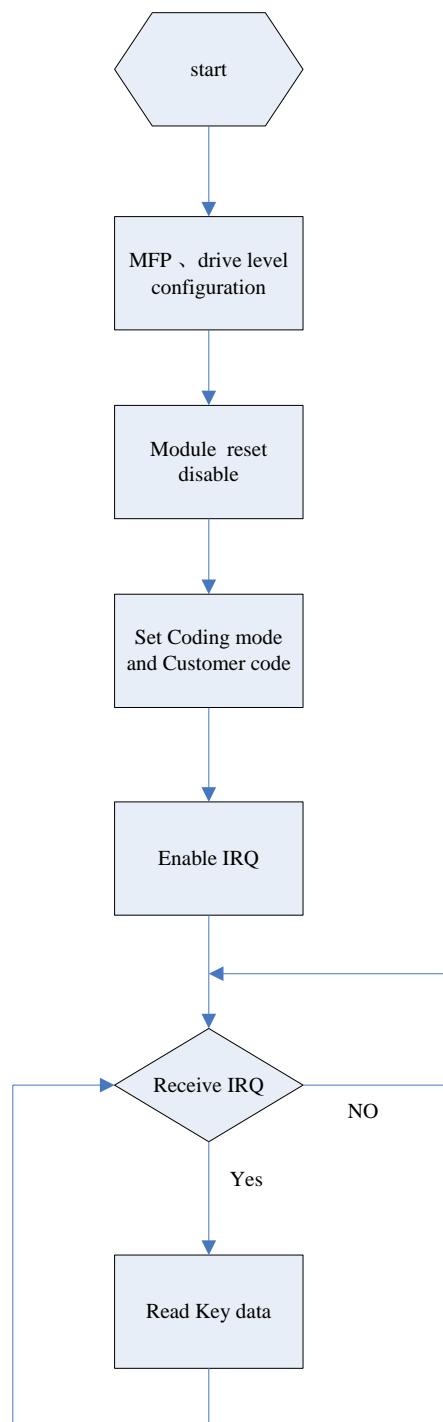


Figure 11-34 IRC receive flow

IRC 模块使用说明：

2. 在红外发射端高电平表示有载波信号，低电平表示无载波信号。经过红外接收头后刚好相反，即输入到 IC pin 上的信号低电平表示有载波信号，高电平表示无载波信号。
3. 写 IR\_LUC/IR\_HUC 是配置 customer code，读 IR\_LUC/IR\_HUC 得到的是 IRC 解析到的实际 IR 信号的 customer code，两者不一定相同。即：如果没有接收到 IR 信号，或者接收到相同 customer code 的信号，读到的 IR\_LUC/IR\_HUC 与写入的值保持一致；如果接收到的 IR 信号 customer code

与写入 IR\_LUC/IR\_HUC 的不一致, 再次读 IR\_LUC/IR\_HUC 时得到的是接收 IR 信号的 customer code。利用这个特点, 只要知道当前信号的码型, 先随意设一个 customer code, 接受一次 IR 波形, 然后再读 IR\_LUC/IR\_HUC 就可以得到 customer code, 免去用示波器分析的麻烦, 接下来把读到的 customer code 值配入 IR\_LUC/IR\_HUC 就可以正确接收 IR。

4. 只有当 customer code 和 key code 都接收正确才会产生 IRQ 信号。
5. IR\_STA 寄存器中的 error pending 位在接收到正确的 IR 信号后会自动清零。
6. IRC 设计具有一定的容错能力, 可以满足正常的要求。在特殊条件下可以通过 **IR\_TOL\_SEL** (**IR\_ONFIG8.0**) 进一步增强容错范围。
7. IRC 没有自收自发功能, CP/FT 测试中需要通过 GPIO 模拟 IR 信号, 验证模块是否工作正常。

## 11.4.5IRC Register List

SFR BANK: 0x0a

| Index | Mnemonic   | Description                  | Bank |
|-------|------------|------------------------------|------|
| 0xa5  | IR_CTL     | IR Control Register          | 0x0a |
| 0xa6  | IR_STA     | IR Status Register           | 0x0a |
| 0xa7  | IR_LUC     | IR low user code register.   | 0x0a |
| 0xa9  | IR_HUC     | IR high user code register.  | 0x0a |
| 0xaa  | IR_KDC     | IR key data code register.   | 0x0a |
| 0xb5  | IR_CONFIG8 | IR inner demodulator config8 | 0x0a |

## 11.4.6 Register Description

### 11.4.6.1 IR\_CTL

IR Control Register. This register is used for enabling IR interface, selecting IR coding mode.

(SFR address 0xa5, SFR Bank = 0x0a)

| Bit(s) | Name    | Description  | R/W | Reset |
|--------|---------|--|-----|-------|
| 7      | IR_EN   | <b>IR enable</b><br>0: IR disable.<br>1: IR enable.  | R/W | 0     |
| 6:5    | IR_CODE | <b>IR coding mode select.</b><br>00 : 9012 code<br>01 : 8 bits NEC code<br>10 : RC5 code<br>11: RC6 mode0 code | R/W | 0     |
| 4      | IR_IRQ  | <b>IR IRQ enable</b><br>0: disable<br>1: enable<br>产生中断的条件是, 所有码都接收正确, 包括用                                     | R/W | 0     |

|     |   |  |   |   |
|-----|---|--|---|---|
|     |   | 户码和键值都接收正确才产生中断，而且，如果用户码和键值如果不正确，接收到跟着该帧的repeat code也不能产生中断。 |   |   |
| 3:0 | - | Reserved, be read as zero.                                   | - | - |

### 11.4.6.2 IR\_STA

IR Status Register, This register is used for displaying current IR status.

(SFR address 0xa6, SFR Bank = 0x0a)

| Bit(s) | Name       | Description  | R/W | Reset |
|--------|------------|--|-----|-------|
| 7      | -          | Reserved, be read as zero.   | -   | -     |
| 6      | IR_USER    | <b>User code don't match pending bit.</b><br>This bit is set when IR user code don't match.<br>Automatically clear when new user code match,<br>0: user code match.<br>1: user code don't match.                                       | R   | 0     |
| 5      | IR_KEY     | <b>Key data code verify error pending bit.</b><br>This bit is set when IR key data code verify error. Automatically clear when new key data code verify ok.<br>0: key data code verify ok.<br>1: key data code verify error.           | R   | 0     |
| 4      | IR_RC_OV   | <b>IR receive overflow pending bit.</b> Write 1 to this bit will clear it, otherwise don't change.<br>0: IR receive not overflow.<br>1: IR receive overflow.   | R/W | 0     |
| 3      | IR_IRQ_PD  | <b>IR IRQ pending bit.</b> Write 1 to this bit will clear it, otherwise don't change.<br>0: not IRQ pending<br>1: IRQ pending  | R/W | 0     |
| 2      | -          | Reserved   | -   | -     |
| 1      | IR_ERR     | <b>IRC receive error pending.</b><br>0: receive ok<br>1: receive error occurs if not match the protocol.<br>Writing 1 to this bit will clear this bit, or auto clear if receive the correct user code and key data code the next time. | RW  | 0     |
| 0      | IR REP_DET | <b>IR repeat flag detect bit.</b><br>Write 1 to this bit will clear it, otherwise don't change.<br>0: repeat code is not detected.<br>1: repeat code is detected.<br><b>NOTE: 9012 repeat code检测需要支持customer</b>                       | R/W | 0     |

|  |  |                 |  |  |
|--|--|-----------------|--|--|
|  |  | code bit0为1的情况。 |  |  |
|--|--|-----------------|--|--|

### 11.4.6.3 IR\_LUC

IR low user code register.

(SFR address 0xa7, SFR Bank = 0x0a)

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7:0    | IR_LUC | <b>IR user code [7:0]</b><br>In RC5 mode, Bit 4:0 is the customer code.<br>In RC6 mode, Bit 7:0 is the customer code.<br>写这个寄存器是配置 IR 的用户码，读这个寄存器将获得当前遥控器的用户码； | R/W | 0     |

### 11.4.6.4 IR\_HUC

IR high user code register.

(SFR address 0xa9, SFR Bank = 0x0a)

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | IR_HUC | <b>IR user code [15:8]</b><br>写这个寄存器是配置 IR 的用户码，读这个寄存器将获得当前遥控器的用户码； | R/W | 0     |

### 11.4.6.5 IR\_KDC

IR key data code register.

(SFR address 0xaa, SFR Bank = 0x0a)

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7:0    | IR_KDC | IR key data code [7:0]<br>如果收到键值就更新寄存器，如果收到 repeat code，则不更新寄存器。 | R   | 0     |

### 11.4.6.6 IR\_CONFIG8

IR config8 register.

(SFR address 0xb5, SFR Bank = 0x0a)

| Bit(s) | Name     | Description                                     | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:2    | -        | Reserved, be read as zero.                      | -   | -     |
| 1      | IR_DEBUG | 0: {1'b0,state[2:0],level_flag,sample_decoded}, | R/W | 0     |

|   |            |  |     |   |
|---|------------|--|-----|---|
|   |            | irc_data_in_dd,irc_clk}<br>1: {state[2:0],pulse_width_error,irc_data_in_dd,<br>jump_to_low,jump_to_high,irc_clk}                 |     |   |
| 0 | IR_TOL_SEL | <b>IR pulse width toleration range select</b><br>0: normal IR receiver toleration range.<br>1: bad IR receiver toleration range. | R/W | 0 |

## 11.4.7 TESTMODE

### 11.4.7.1 BLOCK DIAGRAM

### 11.4.7.2 SIGNAL DESCRIPTION

### 11.4.7.3 TEST METHOD

## 11.4.8 Timing fault-tolerant parameter

|                                 | Min | TYP | Max | Unit |
|---------------------------------|-----|-----|-----|------|
| Infrared wave carrier frequency | 36  | 38  | 40  | Khz  |

## 12 Audio

### 12.1 Audio DAC (林立, 彭洪, 何积军, 丁家平)

| 日期         | 版本     | 描述  | 修订人 |
|------------|--------|---|-----|
| 2012-07-24 | V1. 00 | initial   | 林立  |
| 2012-08-09 | V1. 01 | 1、去掉 PAEN 注释，PA 不需加 MIC boost。<br>2、更改 PA_VOLUME 描述，pa volume total 41 level<br>改为 8 level。<br>3、DAC clock description 框图删除，改为描述参考<br>CMU digital SPEC. | 林立  |
| 2012-09-12 | V1. 02 | 1、更改 bank。Register List 与 Description 不对应。<br>2、设计改为三种半空半满产生方式，且内部产生方<br>式 counter 可配。  | 林立  |

|            |        |  |    |
|------------|--------|--|----|
| 2012-10-20 | V2. 00 | 1、因去掉内部 FM，修改主框图<br>2、修改以下寄存器：<br>DAC_CH0_SRFT_CTL0、<br>DAC_CH1_SRFT_CTL1、<br>DAC_CH0_SRFT_CTL5、<br>DAC_CH1_SRFT_CTL5、<br>DAC_CH0_SR_GAIN、<br>DAC_CH1_SR_GAIN、<br>DAC_CH1_PCMH、<br>AINOP_CTL、<br>DAC_CH0_FIFO_CTL、<br>DAC_CH1_FIFO_CTL | 林立 |
| 2012-12-12 | V2. 02 | 1、删除 SRC Fine Tune 硬件 counter 相关寄存器及配置<br>2、增加 operation manual 章节   | 林立 |
| 2013-01-25 | V2. 03 | 1、ASRC 模块新增 DA0DE2EN, DA1DE2EN, ADDE2EN 控制位。   | 林立 |
| 2013-07-05 | V2. 04 | 1、增加比特位 I2SDAOSEN 控制 I2S 和 DAC 是否同时输出。<br>2、更改 PA_VOLUME.  | 林立 |

## 12.1.1 Feature

- ◆ Built-in stereo Sigma-delta DAC of 92Db SNR, SNR(A-WEIGHTING)>95Db, THD<-75Db
- ◆ Built-in stereo Power Amplifier for 16/32 ohm Headphones, output power : 2×20Mw@16ohm
- ◆ Built-in anti-pop circuit for ‘click and pop’ suppressing
- ◆ Built-in TTS circuit to add speech to music
- ◆ Support Multi-Sample-Rate of 8k/12k/11.025k/16k/22.05k/24k/32k/44.1/48/96kHz
- ◆ Support Sample Rate Convert
- ◆ Support Sample Rate Fine Tune
- ◆ 8\*2Level\*18bit FIFO for DAC Channel0, accessed by MCU and Audio IP, when 6\*2level empty a drq or an irq is send
- ◆ 8\*2Level\*16bit FIFO for DAC Channel1, accessed by MCU and Audio IP, when 6\*2level empty a drq or an irq is send
- ◆ Support I2S Transmitter and Receiver
- ◆ Power Consumption: DAC 1.5Ma@AVCC + 1 Ma@VDD ; PA (static) 1Ma@VCC
- ◆ PA output supports traditional mode and direct drive mode. Direct drive mode PIN vro and vros can be used as GPIO.
- ◆ PA supports Plug-In detect
- ◆ PA output supports 1.6Vpp and 2.8Vpp swing selection

## 12.1.2 Function Description

### 12.1.2.1 Audio DAC/PA

There are a built-in DAC, also a headphone driver for music playback.

The audio DAC with two stereo channel input, which can mix together. Channel 0 has a FIFO of 8\*2Levels will accept 18-bit linear PCM data from either Audio IP, MCU or DMA. Channel 1 has a FIFO of 8\*2Levels will accept 16-bit linear PCM data from either Audio IP, MCU or DMA. The two FIFO data can mix together and send to DAC.

After Digital-to-Analog Conversion, there is a Power Amplifier (PA) for driving a headphone of 16/32ohm, output power will mainly dependent on VCC, with VCC=3.1V, output power will approximately be  $2 \times 20\text{mW}$ @16ohm. Headphone with impedance less than 16ohm is not recommended.

An anti-pop circuit for eliminating the ‘click and pop’ caused by powering-up and –down the PA is also included.

Audio-Line Plug-In or Plug-Off Detection supports direct drive and AC-couple PA output. The state can be read from the state bit.

In order to prevent the click noise when volume changed, a Zero Cross Detection function is provided. Digital Volume Change supports Zero Cross Detection.

Sample Rate Convert supports input and output with different sample rate. It supports convert 8k,12k,16k,24k,32k,48k,96k,192k to 48k, and supports convert 11.025k,22.05k,44.1k to 44.1k.

Sample Rate Fine Tune supports to fine tune the sample rate difference between different clock when use in USB boombox or other external clock.

## 12.1.3 Module Description

### 12.1.3.1 Block Diagram

### 12.1.3.2 Signal Flow

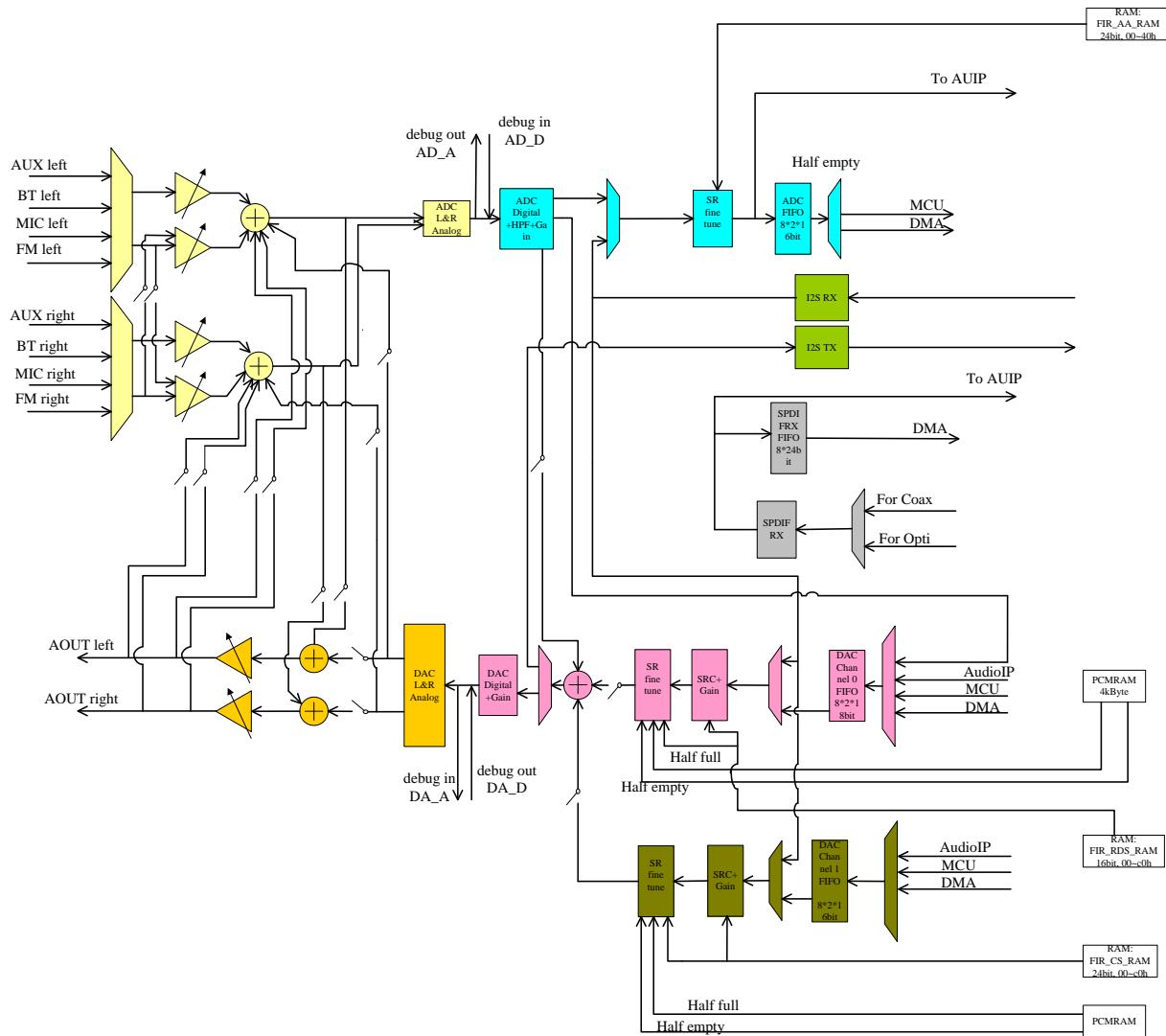


Figure 12-1 Audio ADC / DAC/I2S /SPDIF Module Signal Flow

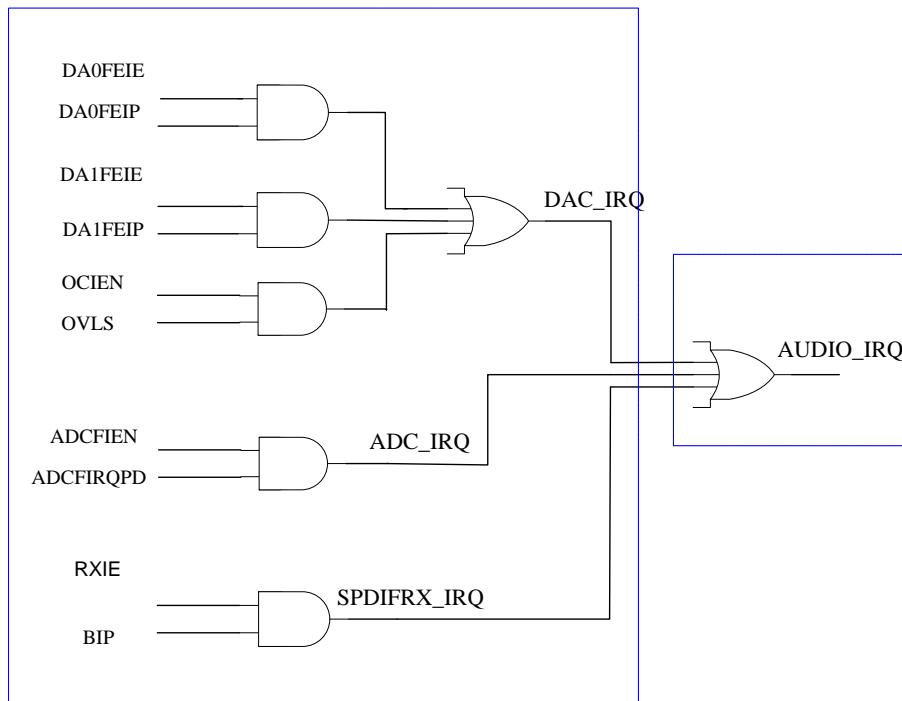


Figure 12-2 DAC/ADC/SPDIF IRQ

DAC 模块, ADC 模块, SPDIFRX 模块分别产生 DAC\_IRQ, ADC\_IRQ, SPDIFRX\_IRQ 信号给系统中断模块，系统中断模块再将这三个中断合成 AUDIO\_IRQ。

### 12.1.3.3 Clock Description

reference to CMU Digital SPEC DAC/IIS-TX controller clock.

### 12.1.3.4 PIN Description and Electrical Parameters

Audio DAC/PA relative pins are as follows:

| Pin Name | Description                  | Electrical parameters   | Attribute | Notes   |
|----------|------------------------------|---|-----------|---|
| AVCC     | Analog power                 | Adjustable(refer to PMU register list), normally 2.95V when VCC is 3.1V | PO        | Normally not for external power supply                  |
| AGND     | Analog ground                |   | GND       |   |
| PAVCC    | Power supply for internal PA | Approximately equal to VCC  | PO        | Not for external power supply                           |
| PAGND    | Ground for PA                |   | GND       | Tied the headphone ground as short as possible to PAGND |

|           |                              |                             |     |  |
|-----------|------------------------------|-----------------------------|-----|--|
|           |                              |                             |     | to minimize crosstalk  |
| AOUTL     | Left output of PA            | Biased to VREFI when enable | AO  |  |
| AOUTR     | Right output of PA           | Biased to VREFI when enable | AO  | when use AOUTL/R & AGND as audio output, each channel will need a coupling capacitor to block the DC component |
| VRO       | Virtual Ground for PA        | Biased to VREFI             | AIO | connect AOUTL / R / VRO to headphone L / R / GND, and no coupling capacitors will be needed                    |
| VROS      | VRO Sense for PA             | Sensing VRO voltage         | AI  |  |
| I2S_BCLK  | Bit Clock                    |                             | IO  |  |
| I2S_LRCLK | Left and Right Channel Clock |                             | IO  |  |
| I2S_MCLK  | Main Clock for Codec         |                             | IO  |  |
| I2S_DO    | Data Out                     |                             | O   |  |
| I2S_DI    | Data In                      |                             | I   |  |

### 12.1.3.5 AC Parameters

#### DAC+PA Characteristics

| Symbol             | Parameter                         | Test Conditions  | Value | Unit  |
|--------------------|-----------------------------------|--|-------|-------|
| THD+N              | Total Harmonic Distortion + Noise | VCC = 3.1V, AVCC = 2.95V,<br>0Dbfs, f = 1kHz, AES17 filter                           | -80   | Db    |
|                    |                                   |  | 0.01  | %     |
| SNR                | Signal to Noise Ratio             | VCC= 3.1V, AVCC = 2.95V, P <sub>O</sub> = 20Mw@16ohm, f = 20Hz – 20kHz, No Weighting | 92    | Db    |
| V <sub>NOISE</sub> | Output Noise                      |  | 15    | Mvrms |

### 12.1.1 Operation Manual

### 12.1.1.1 软件控制流程

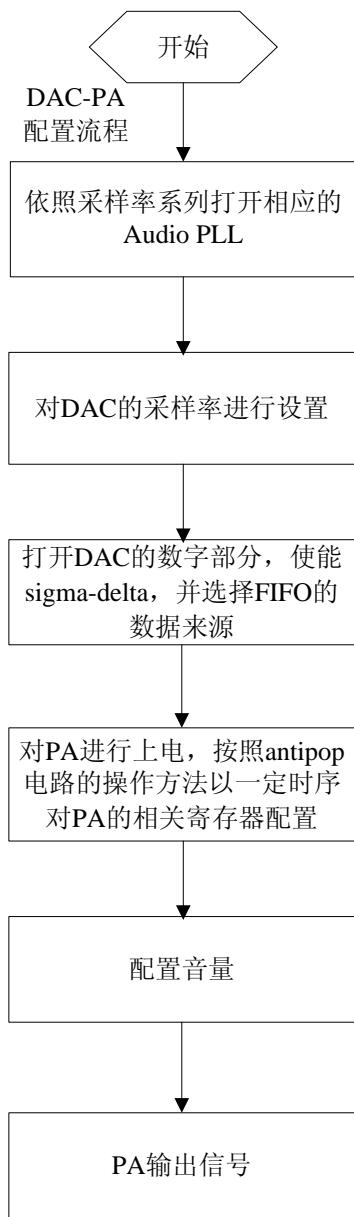


Figure 12-21 DAC&PA programming flow

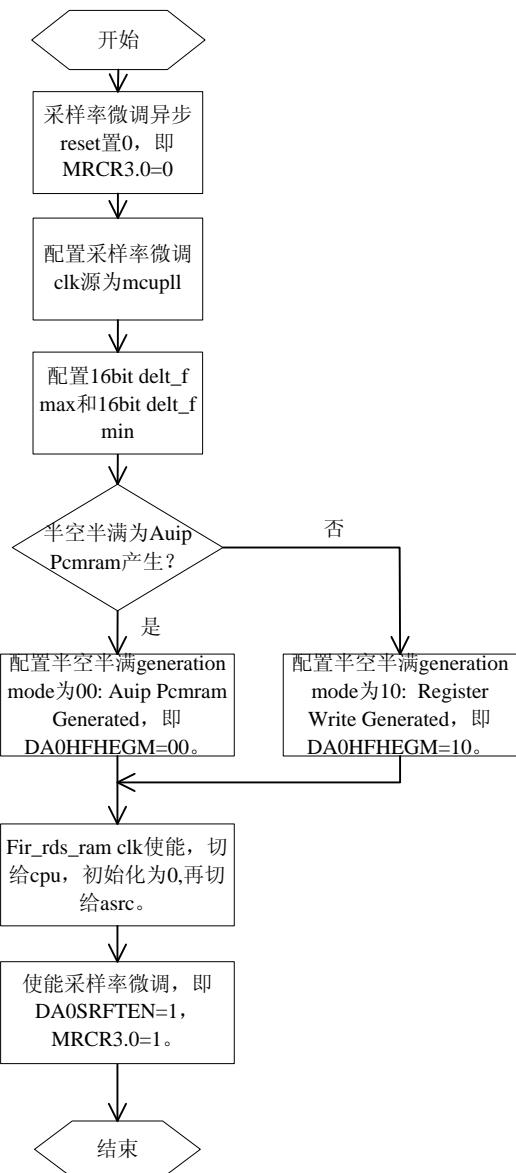


Figure 12-22 Dac Channel 0 采样率微调配置流程

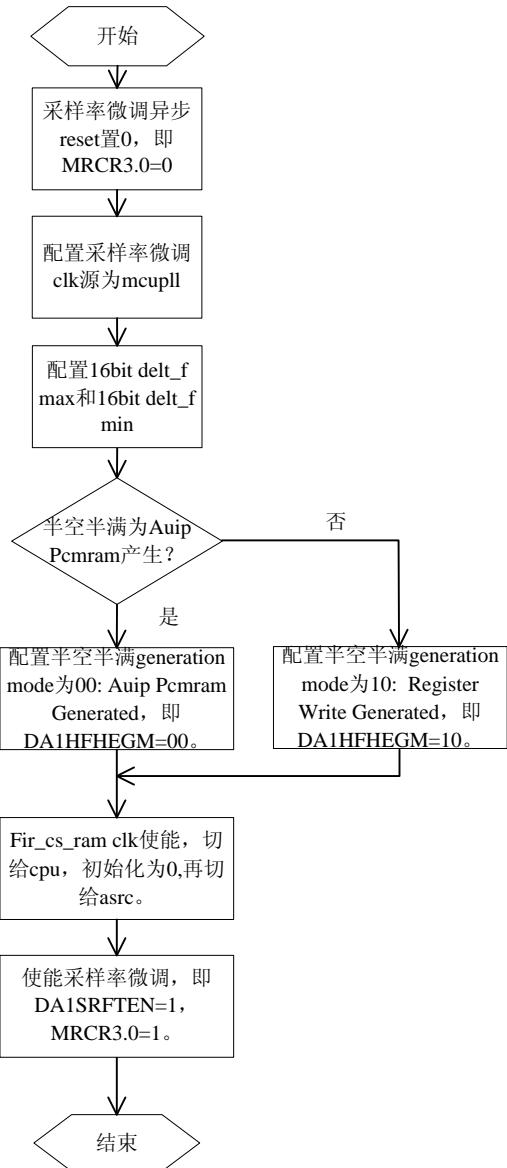


Figure 12-23 Dac Channel 1 采样率微调配置流程

注:

- 2013、采样率微调dac ch0,ch1做在dac里，配置需先对dac的异步复位置1和使能clock。采样率微调adc做在adc里，配置需先对adc的异步复位置1和使能clock。
- 2, 配置采样率微调需要将asrc的异步复位在整个配置最后才置为normal。
- 2014、配置采样率微调clk为mcupll, mcupll需要是dac/adc clk的最少2倍，如dac采样率为48k，则dac clk为 $512 \times 48\text{k} = 24.576\text{MHz}$ ，如adc采样率为48k，则adc clk为 $256 \times 48\text{k} = 12.288\text{MHz}$ 。
- 2015、半空半满产生方式如果选auip，则由pcmram硬件自动产生。如选register write，则需由软件根据ram的空满状态去配置采样率微调的半空或半满。
- 2016、dac采样率微调寄存器配置半空，硬件选择delt\_f min，输出频率会偏快；寄存器配置半满，硬件选择delt\_f max，输出频率会偏慢。Adc采样率微调寄存器配置

半空，硬件选择delt\_f max，输出频率会偏慢；寄存器配置半满，硬件选择delt\_f min，输出频率会偏快。

2017、delt\_f为Q4.20无符号格式。0x200000为1，0x400000为2，0x100000为0.5。表示采样采样时间为标准采样时间的倍数。

7，需要在整个adc采样率微调配置完成之后，才使能adc左右声道，才能保证左右声道不反。

2018、采样率微调模块只适合双声道，adc/dac配置时需要左右声道都打开。

## 12.1.2 DAC Register List

| Index | Mnemonic                          | Description  | BANK      |
|-------|-----------------------------------|--|-----------|
| 0xd5  | <a href="#">DAC_CTL</a>           | DAC Control Register                                   | 0x13/0x04 |
| 0xd6  | <a href="#">DAC_VOLUME0</a>       | DAC Volume Control register 0                          | 0x13/0x04 |
| 0xd7  | <a href="#">DAC_VOLUME1</a>       | DAC Volume Control register 1                          | 0x13/0x04 |
| 0xd8  | <a href="#">DAC_CH0_FIFO_CTL0</a> | DAC Channel 0 FIFO Control Register 0                  | 0x13/0x04 |
| 0xd9  | <a href="#">DAC_CH0_FIFO_CTL1</a> | DAC Channel 0 FIFO Control Register 1                  | 0x13/0x04 |
| 0xda  | <a href="#">DAC_CH0_PCML</a>      | DAC Channel 0 PCM DATA Low byte                        | 0x13/0x04 |
| 0xdb  | <a href="#">DAC_CH0_PCMM</a>      | DAC Channel 0 PCM DATA Middle byte                     | 0x13/0x04 |
| 0xdc  | <a href="#">DAC_CH0_PCMH</a>      | DAC Channel 0 PCM DATA High byte                       | 0x13/0x04 |
| 0xdd  | <a href="#">DAC_CH1_FIFO_CTL0</a> | DAC Channel 1 FIFO Control Register 0                  | 0x13/0x04 |
| 0xde  | <a href="#">DAC_CH1_FIFO_CTL1</a> | DAC Channel 1 FIFO Control Register 1                  | 0x13/0x04 |
| 0xdf  | <a href="#">DAC_CH1_PCML</a>      | DAC Channel 1 PCM DATA Low byte                        | 0x13/0x04 |
| 0xe1  | <a href="#">DAC_CH1_PCMH</a>      | DAC Channel 1 PCM DATA High byte                       | 0x13/0x04 |
| 0xe2  | <a href="#">I2S_CTL0</a>          | I2S Control Register 0                                 | 0x13/0x04 |
| 0xe3  | <a href="#">I2S_CTL1</a>          | I2S Control Register 1                                 | 0x13/0x04 |
| 0xe4  | <a href="#">DAC_ANALOG0</a>       | DAC Analog Register 0                                  | 0x13/0x04 |
| 0xe5  | <a href="#">DAC_ANALOG1</a>       | DAC Analog Register 1                                  | 0x13/0x04 |
| 0xe6  | <a href="#">DAC_TUNE0</a>         | DAC tuning Control Register 0                          | 0x13/0x04 |
| 0xe7  | <a href="#">DAC_TUNE1</a>         | DAC tuning Control Register 1                          | 0x13/0x04 |
| 0xe9  | <a href="#">PA_VOLUME</a>         | PA VOLUME Control Register                             | 0x13/0x04 |
| 0xea  | <a href="#">PA_CTL</a>            | PA Control Register                                    | 0x13/0x04 |
| 0xeb  | <a href="#">PA_APCTL</a>          | PA anti-pop Control Register                           | 0x13/0x04 |
| 0xec  | <a href="#">DDV_CTL0</a>          | Direct drive Control Register 0                        | 0x13/0x04 |
| 0x90  | <a href="#">DAC_CH0_SR_CTL</a>    | DAC Channel 0 Sample Rate Control Register             | 0x13      |
| 0x91  | <a href="#">DAC_CH0_SR_GAIN</a>   | DAC Channel 0 Sample Rate Gain Control Register        | 0x13      |
| 0x92  | <a href="#">DAC_CH0_SRFT_CTL0</a> | DAC Channel 0 Sample Rate Fine Tune Control Register 0 | 0x13      |
| 0x93  | <a href="#">DAC_CH0_SRFT_CTL1</a> | DAC Channel 0 Sample Rate Fine Tune Control Register 1 | 0x13      |
| 0x94  | <a href="#">DAC_CH0_SRFT_CTL2</a> | DAC Channel 0 Sample Rate Fine Tune Control Register 2 | 0x13      |
| 0x95  | <a href="#">DAC_CH0_SRFT_CTL3</a> | DAC Channel 0 Sample Rate Fine Tune Control Register 3 | 0x13      |
| 0x97  | <a href="#">DAC_CH0_SRFT_CTL4</a> | DAC Channel 0 Sample Rate Fine Tune Control Register 4 | 0x13      |
| 0x98  | <a href="#">DAC_CH1_SR_CTL</a>    | DAC Channel 1 Sample Rate Control Register             | 0x13      |
| 0x99  | <a href="#">DAC_CH1_SR_GAIN</a>   | DAC Channel 1 Sample Rate Gain Control Register        | 0x13      |

|      |                   |  |      |
|------|-------------------|--|------|
| 0x9a | DAC_CH1_SRFT_CTL0 | DAC Channel 1 Sample Rate Fine Tune Control Register 0 | 0x13 |
| 0x9b | DAC_CH1_SRFT_CTL1 | DAC Channel 1 Sample Rate Fine Tune Control Register 1 | 0x13 |
| 0x9c | DAC_CH1_SRFT_CTL2 | DAC Channel 1 Sample Rate Fine Tune Control Register 2 | 0x13 |
| 0x9d | DAC_CH1_SRFT_CTL3 | DAC Channel 1 Sample Rate Fine Tune Control Register 3 | 0x13 |
| 0x9e | DAC_CH1_SRFT_CTL4 | DAC Channel 1 Sample Rate Fine Tune Control Register 4 | 0x13 |

## 12.1.1 Register Description

### 12.1.1.1 DAC\_CTL

Bank:0x13, 0x04. Address: 0xd5. DAC Control Register 0

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:6    | DAOSR     | DAC Over Sample Rate:<br>00: 32X<br>01: 64X<br>10: 128X<br>11: 256X                           | RW  | 01    |
| 5      | DMDA1MDEN | Digital Mixer, DAC Channel 1 Digital Mix to Dac<br>Digital Enable:<br>0: disable<br>1: enable | RW  | 0     |
| 4      | DMDA0MDEN | Digital Mixer, DAC Channel 0 Digital Mix to Dac<br>Digital Enable:<br>0: disable<br>1: enable | RW  | 0     |
| 3      | DMAMDEN   | Digital Mixer, ADC Digital Mix to Dac Digital<br>Enable:<br>0: disable<br>1: enable           | RW  | 0     |
| 2      | DITHEN    | DAC Dither Enable:<br>0: disable<br>1: enable   | RW  | 0     |
| 1      | SDEN      | On-Chip Sigma-Delta DAC Enable.<br>0: Disable<br>1: Enable                                    | RW  | 0     |
| 0      | DAIFEN    | DAC_IF Enable.<br>0: Disable<br>1: Enable   | RW  | 0     |

### 12.1.1.2 DAC\_VOLUME0

Bank:0x13, 0x04. Address: 0xd6. DAC Left Channel Volume Register

| Bit(s) | Name   | Description  | R/W | Reset     |
|--------|--------|--|-----|-----------|
| 7:0    | VOLDAL | <p>DAC Left Channel Volume( step:0.5dB):</p> <p>0xcb~0xcf: 18dB</p> <p>.</p> <p>.</p> <p>0xbb~0xbf: 12dB</p> <p>.</p> <p>.</p> <p>0xab~0xaf: 6dB</p> <p>.</p> <p>.</p> <p>0x9b~0x9f: 0dB</p> <p>.</p> <p>.</p> <p>0x8b~0x8f: -6dB</p> <p>.</p> <p>.</p> <p>0x7b~0x7f: -12dB</p> <p>.</p> <p>.</p> <p>0x6b~0x6f: -18dB</p> <p>.</p> <p>.</p> <p>0x5b~0x5f: -24dB</p> <p>.</p> <p>.</p> <p>0x4b~0x4f: -30dB</p> <p>.</p> <p>.</p> <p>0x3b~0x3f: -36dB</p> <p>.</p> <p>.</p> <p>0x2b~0x2f: -42dB</p> <p>.</p> <p>.</p> <p>0x1b~0x1f: -48dB</p> <p>.</p> <p>.</p> <p>0x0b~0x0f: -54dB</p> <p>.</p> | RW  | 1001_1011 |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | .<br>0x01: -59dB<br>0x00: -59.5dB<br>Others: reserved |  |  |
|--|--|---|--|--|

**Note:**

| 音量等级配置 | 音量衰减   |
|--------|--------|
| 9f     | 0dB    |
| 9e     | 0dB    |
| 9d     | 0dB    |
| 9c     | 0dB    |
| 9b     | 0dB    |
| 9a     | -0.5dB |
| 99     | -1.0dB |
| 98     | -1.5dB |
| 97     | -2.0dB |
| 96     | -2.5dB |
| 95     | -3.0dB |
| 94     | -3.5dB |
| 93     | -4.0dB |
| 92     | -4.5dB |
| 91     | -5.0dB |
| 90     | -5.5dB |

| 音量等级配置 | 音量衰减      |
|--------|-----------|
| cx     | Gain+18dB |
| bx     | Gain+12dB |
| ax     | Gain+6dB  |
| 9x     | Gain      |
| 8x     | Gain-6dB  |
| 7x     | Gain-12dB |
| 6x     | Gain-18dB |
| 5x     | Gain-24dB |
| 4x     | Gain-30dB |
| 3x     | Gain-36dB |
| 2x     | Gain-42dB |
| 1x     | Gain-48dB |
| 0x     | Gain-54dB |

**12.1.1.3 DAC\_VOLUME1**

Bank:0x13, 0x04. Address: 0xd7. DAC Right Channel Volume Register

| Bit(s) | Name   | Description   | R/W | Reset     |
|--------|--------|---|-----|-----------|
| 7:0    | VOLDAR | DAC Right Channel Volume( step:0.5dB):<br>同 DAC Left Channel Volume 配置。 | RW  | 1001_1011 |

#### 12.1.1.4 DAC\_CH0\_FIFO\_CTL0

Bank:0x13, 0x04. Address: 0xd8. DAC Channel 0 FIFO Control Register 0

| Bit(s) | Name        | Description   | R/W | Reset |
|--------|-------------|---|-----|-------|
| 7      | DAVOLZCTOEN | DAC Volume Control& Ch0 SRC Gain Control &Ch1 SRC Gain Control Zero Cross Detection Time Out Enable:<br><br>0: Disable<br>1: Enable<br><br>Note:<br><br>When DAVOLZCEN ==1 & DAVOLZCTOEN ==1, input data >=0x83 , input data is 18bit.<br>After 100ms, Volume change will be set. | R/W | 0     |
| 6      | DAVOLZCEN   | DAC Volume Control& Ch0 SRC Gain Control &Ch1 SRC Gain Control Zero Cross Detection Enable:<br><br>0: Disable<br>1: Enable<br><br>Note:<br><br>When abs(input data)<0x83, input data is 18bit.<br><br>0x83/2^17<br><br>*560mVrms=0.56mVrms=-60dBFS.                               | R/W | 0     |
| 5      | DA0FEZR     | DAC Channel 0 FIFO Empty Output Zero:<br><br>0: disable<br>1: enable  | RW  | 0     |
| 4      | DADOUTS     | DAC Digital Output Select.<br>0: to On-Chip sigma-delta DAC<br>1: to I2S  | RW  | 0     |
| 3:1    | DA0FINS     | DAC Channel 0 FIFO Input Select.<br>000: from MCU<br>001: from AUIP<br>010: reserved<br>011: from DMA<br>100: from ADC_OUT (ADDA 通路, 经过 DA FIFO)<br>Others: reserved  | RW  | 000   |

|   |         |  |    |   |
|---|---------|--|----|---|
|   |         | Note:<br>dac fifo0 和 fifo1 同一时间里只能有一个给 auip 访问, 当 DA0FINS 和 DA1FINS 同时选到 auip 访问时, fifo0 给 auip 访问是有效的, fifo1 给 auip 访问是无效的。 |    |   |
| 0 | DA0FIRT | DAC Channel 0 FIFO Reset.<br>0: Reset FIFO<br>1: Enable FIFO   | RW | 0 |

### 12.1.1.5 DAC\_CH0\_FIFO\_CTL1

Bank:0x13, 0x04. Address: 0xd9. DAC Channel 0 FIFO Control Register 1(0x08)

| Bit(s) | Name    | Description   | R/W | Reset |
|--------|---------|---|-----|-------|
| 7:4    | -       | Reserved for digital future use   | RW  | 0000  |
| 3      | DA0FUF  | DAC Channel 0 FIFO Full Flag.<br>0: Not Full<br>1: Full   | R   | 0     |
| 2      | DA0FEDE | DAC Channel 0 FIFO Empty DRQ Enable.<br>0: Disable<br>1: Enable   | RW  | 0     |
| 1      | DA0FEIP | DAC Channel 0 FIFO Empty IRQ Pending Bit.<br>0: No IRQ<br>1: IRQ,<br>Writing 1 to the bit to clear it.<br>2*2level 空时产生 IRQ/DRQ<br>在 IRQ disable 时, pending bit 仍然可以置 1 | RW  | 0     |
| 0      | DA0FEIE | DAC Channel 0 FIFO Empty IRQ Enable.<br>0: Disable<br>1: Enable   | RW  | 0     |

### 12.1.1.6 DAC\_CH0\_PCML

Bank:0x13, 0x04. Address: 0xda. DAC Channel 0 PCM DATA Low byte

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | DA0PCMDATL | Internal D/A PCM[7:0]<br>Note:<br>Dac is 18bit, so low 6bit of data set zero to fifo. | W   | XX    |

### 12.1.1.7 DAC\_CH0\_PCMM

Bank:0x13, 0x04. Address: 0xdb. DAC Channel 0 PCM DATA Middle byte

| Bit(s) | Name       | Description            | R/W | Reset |
|--------|------------|------------------------|-----|-------|
| 7:0    | DA0PCMDATM | Internal D/A PCM[15:8] | W   | XX    |

### 12.1.1.8 DAC\_CH0\_PCMH

Bank:0x13, 0x04. Address: 0xdc. DAC Channel 0 PCM DATA High byte

| Bit(s) | Name       | Description  | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | DA0PCMDATH | Internal D/A PCM[23:16]<br>PCMDATH /M/L ports are mapped to DAC FIFO.<br>Writing to this register will transfer 24-bit PCM data into DAC FIFO. | W   | XX    |

### 12.1.1.9 DAC\_CH1\_FIFO\_CTL0

Bank:0x13, 0x04. Address: 0xdd. DAC Channel 1 FIFO Control Register 0(0xc0)

| Bit(s) | Name    | Description  | R/W | Reset |
|--------|---------|--|-----|-------|
| 7:6    | -       | Reserved for digital future use  | RW  | 11    |
| 5      | DA1FEZR | DAC Channel 1 FIFO Empty Output Zero:<br>0: disable<br>1: enable   | RW  | 0     |
| 4:3    | -       | Reserved for digital future use  | RW  | 0     |
| 2:1    | DA1FINS | DAC Channel 1 FIFO Input Select.<br>00: from MCU<br>01: from AUIP<br><b>10: reserved</b><br>11: from DMA<br>Note:<br><b>dac fifo0 和 fifo1 同一时间里只能有一个给 auiip 访问，当 DA0FINS 和 DA1FINS 同时选到 auiip 访问时，fifo0 给 auiip 访问是有效的，fifo1 给 auiip 访问是无效的。</b> | RW  | 00    |
| 0      | DA1FIRT | DAC Channel 0 FIFO Reset.<br>0: Reset FIFO<br>1: Enable FIFO   | RW  | 0     |

### 12.1.1.10 DAC\_CH1\_FIFO\_CTL1

Bank:0x13, 0x04. Address: 0xde. DAC Channel 1 FIFO Control Register 1(0x08)

| Bit(s) | Name    | Description  | R/W | Reset |
|--------|---------|--|-----|-------|
| 7:4    | -       | Reserved for digital future use  | RW  | 0000  |
| 3      | DA1FUF  | DAC Channel 1 FIFO Full Flag.<br>0: Not Full<br>1: Full  | R   | 0     |
| 2      | DA1FEDE | DAC Channel 1 FIFO Empty DRQ Enable.<br>0: Disable<br>1: Enable  | RW  | 0     |
| 1      | DA1FEIP | DAC Channel 1 FIFO Empty IRQ Pending Bit.<br>0: No IRQ<br>1: IRQ,<br>Writing 1 to the bit to clear it.<br><b>2*2level</b> 空时产生 IRQ/DRQ<br>在 IRQ disable 时, pending bit 仍然可以置 1 | RW  | 0     |
| 0      | DA1FEIE | DAC Channel 1 FIFO Empty IRQ Enable.<br>0: Disable<br>1: Enable  | RW  | 0     |

### 12.1.1.11 DAC\_CH1\_PCML

Bank:0x13, 0x04. Address: 0xdf. DAC Channel 1 PCM DATA Low byte

| Bit(s) | Name       | Description           | R/W | Reset |
|--------|------------|-----------------------|-----|-------|
| 7:0    | DA1PCMDATL | Internal D/A PCM[7:0] | W   | XX    |

### 12.1.1.12 DAC\_CH1\_PCMH

Bank:0x13, 0x04. Address: 0xE1. DAC Channel 1 PCM DATA High byte

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | DA1PCMDATH | Internal D/A PCM[15:8]<br>PCMDATH /M/L ports are mapped to DAC FIFO.<br>Writing to this register will transfer 16-bit PCM data into DAC FIFO. | W   | XX    |

### 12.1.1.13 I2S\_CTL0

Bank:0x13, 0x04. Address: 0xe2. I2S Control Register 0

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7      | I2SSRS | I2SIN Internal DAC Sample Rate Select.<br>0: Sample Rate Internal Set<br>1: External Digital Audio Rate, for avoiding I2S input clock mismatch the internal Fs clock. | RW  | 0     |
| 6:5    | IBND   | I2S Input Bit Number Detect.<br>00: 24bit<br>01: 20bit<br>10: 18bit<br>11: 16bit<br>If detected bit is out of range, this bit is equal to default value.              | R   | 00    |
| 4      | FSRF   | I2S Fs Ready Flag.<br>0: Not Ready<br>1: Ready<br>Writing 1 to the bit is clear the bit.  | RW  | 0     |
| 3      | RMS    | I2SRX Mode Select.<br>0: Slave<br>1: Master<br>Note:<br>I2STX only supports Master mode.  | RW  | 0     |
| 2:0    | I2SISR | IIS input sample rate.<br>000: Reserved<br>001: 32KHz<br>010: 44.1KHz<br>011: 48KHz<br>100: Reserved<br>101: Reserved<br>110: 88.2kHz<br>111: 96KHz                   | R   | 000   |

### 12.1.1.14 I2S\_CTL1

Bank:0x13, 0x04. Address: 0xe3. I2S Control Register 1

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7      | I2SDAOSEN | I2S Ouput & DAC Ouput Synchronously Enable:<br>0: Disable<br>1: Enable | RW  | 0     |

|   |      |  |    |   |
|---|------|--|----|---|
|   |      | Note: when enable this bit, i2s uses dac clock/2 as clock, and uses the same data as dac uses. |    |   |
| 6 | INEN | I2S Input Enable.<br>0: Disable<br>1: Enable   | RW | 0 |
| 5 | OEN  | I2S Output Enable.<br>0: Disable<br>1: Enable  | RW | 0 |
| 4 | LB   | I2S Loop Back Enable:<br>0: Disable<br>1: Enable   | RW | 0 |
| 3 | DAEN | DAC Analog part debug enable:<br>0: Disable<br>1: Enable                                       | RW | 0 |
| 2 | DDEN | DAC Digital part debug enable:<br>0: Disable<br>1: Enable                                      | RW | 0 |
| 1 | DDCS | DAC Digital part debug channel select:<br>0: Left<br>1: Right                                  | RW | 0 |
| 0 | AAEN | ADC Analog part debug enable:<br>0: Disable<br>1: Enable                                       | RW | 0 |

### 12.1.1.15 DAC\_ANALOG0

Bank:0x13, 0x04. Address: 0xe4. DAC Analog Control Register

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7      | DACLEN | Analog circuit of the internal Left DAC enable,<br>0: Left channel disable<br>1: Left channel Enable    | RW  | 0     |
| 6      | DACREN | Analog circuit of the internal Right DAC enable,<br>0: Right channel disable<br>1: Right channel Enable | RW  | 0     |
| 5      | ZERODT | Zero data for DAC analog part,<br>0:disable,<br>1:enable  | RW  | 0     |
| 4      | PAEN   | MIXEN, analog mixer and PA enable,<br>0: Disable<br>1: Enable   | RW  | 0     |
| 3      | AINM   | AnalogIN Mute:<br>0:Mute,   | RW  | 0     |

|     |          |   |    |   |
|-----|----------|---|----|---|
|     |          | 1:Dismute,connect                         |    |   |
| 2:1 | -        | Reserved for analog future use            | RW | 0 |
| 0   | DAC_MUTE | DAC MUTE:<br>0:Mute,<br>1:Dismute,connect | RW | 0 |

### 12.1.1.16 DAC\_ANALOG1

Bank:0x13, 0x04. Address: 0xe5. DAC Analog Register 1

| Bit(s) | Name    | Description  | R/W | Reset |
|--------|---------|--|-----|-------|
| 7      | -       | Reserved for analog future use   | RW  | 0     |
| 6      | DDOVV   | Direct Drive overload protect and recover<br>0: Overload protect and recover is valid<br>1: Overload protect and recover is invalid  | RW  | 0     |
| 5      | ALPIDEN | Audio output jet Plug-In Detect enable<br>0: disable<br>1: enable<br>detect by VROS  | RW  | 0     |
| 4      | ALPIDS  | Audio output jet Plug-In Detect state:<br>0: not in<br>1: in   | R   | 0     |
| 3      | OVLS    | DAC VRO overload state:<br>1: VRO overload<br>0: normal work state   | R   | 0     |
| 2:1    | -       | Reserved for analog future use   | R   | 01    |
| 0      | OCIEN   | Direct drive output over-current status IRQ enable<br>0: disable<br>1: enable<br>This bit when enable will enable an interrupt signal to the interrupt controller, while DDOVV=0 and OVLS=1. | RW  | 0     |

### 12.1.1.17 DAC\_TUNE0

Bank:0x13, 0x04. Address: 0xe6. DAC TUNE Register 0

| Bit(s) | Name | Description                                | R/W | Reset |
|--------|------|--|-----|-------|
| 7:6    | PAIQ | PA output stage IQ control.<br>00:smallest | RW  | 00    |

|     |        |  |    |    |
|-----|--------|--|----|----|
|     |        | ...  |    |    |
| 5:4 | PAIB   | PA bias current control.<br>00:smallest<br>...<br>11:biggest     | RW | 01 |
| 3:2 | OPDAVB | OPDA bias voltage control.<br>00: smallest<br>...<br>11: biggest | RW | 01 |
| 1:0 | OPVBIB | OPVB bias current control.<br>00:smallest<br>...<br>11:biggest   | RW | 01 |

### 12.1.1.18 DAC\_TUNE1

Bank:0x13, 0x04. Address: 0xe7. DAC TUNE Register 1

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7:6    | -      | Reserved for analog future use                                   | RW  | 01    |
| 5:3    | OPDAIB | OPDA bias current control.<br>000:smallest<br>...<br>111:biggest | RW  | 011   |
| 2:0    | OPGIB  | OPG bias current control.<br>000:smallest<br>...<br>111:biggest  | RW  | 101   |

### 12.1.1.19 PA\_VOLUME

Bank:0x13, 0x04. Address: 0xe9. PA VOLUME Control Register

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:3    | -      | Reserved for analog future use  | RW  | 00000 |
| 2:0    | Volume | Headphone Amp Volume control, total 8 levels<br>111, 4.9dB step up<br>110, 0dB step down<br>101, -4dB step down<br>100, -9dB step down, rough | RW  | 000   |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | 011, -14dB step down<br>010, -20dB step down, rough<br>001, -29dB step down<br>000, -41dB step down |  |  |
|--|--|---|--|--|

### 12.1.1.20 PA\_CTL

Bank:0x13, 0x04. Address: 0xea. PA CTL Register

| Bit(s) | Name    | Description   | R/W | Reset |
|--------|---------|---|-----|-------|
| 7:4    | -       | Reserved for analog future use  | RW  | 0100  |
| 3:2    | OPCM1IB | OPCM1 bias current control.<br>00:smallest<br>...<br>11:biggest       | RW  | 01    |
| 1:0    | ICTRL   | All current control:<br>00: × 0.5<br>01: × 0.75<br>10: × 1<br>11: × 2 | RW  | 10    |

### 12.1.1.21 PA\_APCTL

Bank:0x13, 0x04. Address: 0xeb. PA anti-pop Control Register

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:6    | -      | Reserved for analog future use  | RW  | 00    |
| 5      | ATPLP2 | Antipop loop2 enable.<br>0:disable<br>1:enable                                  | RW  | 0     |
| 4      | CCH    | Antipop function enable:<br>0: disable<br>1: enable, ramp wave connect          | RW  | 0     |
| 3:2    | -      | Reserved for analog future use  | RW  | 00    |
| 1      | OSEN   | Output stage enable, 0: Disable, 1: Enable                                      | RW  | 0     |
| 0      | CDISCH | Antipop for blocking C discharge control:<br>0: disable<br>1: enable, discharge | RW  | 0     |

Antipop:

PAVCC 前级控制由 AVCC 改为 PAVCC 供电。

~~AOUT 到 PAVCC 二极管增加一级。~~

### 12.1.1.22 DDV\_CTL0

Bank:0x13, 0x04. Address: 0xec. Direct drive Control Register 0

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:5    | OPVROIB   | OPVRO bias current control.<br>000:smallest<br>...<br>111:biggest                                       | RW  | 011   |
| 4      | OPVROEN   | Analog circuit of the internal DAC_OPVRO enable,<br>0: Disable<br>1: Enable                             | RW  | 0     |
| 3      | DDATPR    | Direct Drive antipop_VRO Resistant Connect<br>Enable:<br>0: disconnect<br>1: connect                    | RW  | 0     |
| 2:0    | OPVROOSIB | Analog circuit of the internal DAC_OPVRO output stage IQ control.<br>000:smallest<br>...<br>111:biggest | RW  | 000   |

This register can only be changed by system software.

There are two pins for direct drive circuit, with independent PADs named VRO and VRO\_SENSE.

When VRO is enabled, VRO-pin will output a voltage  $\sim=1.5V$ (tracing VREF), together with AOUTL/R to support a stereo audio output without coupling capacitors to block DC component.

With VRO\_SENSE enabled, VRO\_SENSE will trace the virtual ground of headphone, acting as a feedback of load, to minimize distortion and crosstalk.

### 12.1.1.23 DAC\_CH0\_SR\_CTL

Bank:0x13. Address: 0x90. DAC Channel 0 Sample Rate Control Register

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7:3    | -      | Reserved for digital future use  | RW  | 000   |
| 2:0    | DA0SRR | DAC Channel 0 SRC Ratio (interpolation, decimation):<br>000: (1,1)<br>001: (6,1)<br>010: (4,1) | R/W | 0     |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | 011: (3,1)<br>100: (2,1)<br>101: (3,2)<br>110: (1,2)<br>111: (1,4) |  |  |
|--|--|--|--|--|

### 12.1.1.24 DAC\_CH0\_SR\_GAIN

Bank:0x13. Address: 0x91. DAC Channel 0 Sample Rate Control Register

| Bit(s) | Name   | Description   | R/W | Reset     |
|--------|--------|---|-----|-----------|
| 7:0    | DA0SRG | DAC Channel 0 SRC Gain, step -0.5dB:<br>同 DAC Left Channel Volume 定义。 | RW  | 1001_1011 |

### 12.1.1.25 DAC\_CH0\_SRFT\_CTL0

Bank:0x13. Address: 0x92. DAC Channel 0 Sample Rate Fine Tune Control Register 0

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7      | ADDE2EN   | ADC Decimation 2X Enable:<br>0: Disable<br>1: Enable. Configure the data out of ASRC module for decimation by 2.<br><br>Note:<br>Should be configure before ASRC enable.           | R/W | 0     |
| 6      | DA1DE2EN  | DAC Channel 1 Decimation 2X Enable:<br>0: Disable<br>1: Enable. Configure the data out of ASRC module for decimation by 2.<br><br>Note:<br>Should be configure before ASRC enable. | R/W | 0     |
| 5      | DA0DE2EN  | DAC Channel 0 Decimation 2X Enable:<br>0: Disable<br>1: Enable. Configure the data out of ASRC module for decimation by 2.<br><br>Note:<br>Should be configure before ASRC enable. | R/W | 0     |
| 4:3    | DA0HFHEGM | DAC Channel 0 Half Full & Half Empty Generation Mode:  | R/W | 00    |

|   |           |   |     |   |
|---|-----------|---|-----|---|
|   |           | 00: Auip Pcmram Generated,<br>01: Reserved<br>10: Register Write Generated for Debug,<br>11: Reserved   |     |   |
| 2 | DA0FHF    | DAC Channel 0 FIFO Half Full:<br>0: not half full<br>1: half full<br>Writing 1 to this bit will set Half Full signal to DAC SR Fine Tune.     | R/W | 0 |
| 1 | DA0FHE    | DAC Channel 0 FIFO Half Empty:<br>0: not half empty<br>1: half empty<br>Writing 1 to this bit will set Half Empty signal to DAC SR Fine Tune. | R/W | 0 |
| 0 | DA0SRFTEN | DAC Channel 0 SR Fine Tune enable:<br>0: disable and bypass<br>1: enable  | R/W | 0 |

### 12.1.1.26 DAC\_CH0\_SRFT\_CTL1

Bank:0x13. Address: 0x93. DAC Channel 0 Sample Rate Control Register 1

| Bit(s) | Name       | Description                                      | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | DA0SFTFMAH | DAC Channel 0 SR Fine Tune delta_f max high byte | R/W | 0     |

### 12.1.1.27 DAC\_CH0\_SRFT\_CTL2

Bank:0x13. Address: 0x94. DAC Channel 0 Sample Rate Control Register 2

| Bit(s) | Name       | Description                                     | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | DA0SFTFMAL | DAC Channel 0 SR Fine Tune delta_f max low byte | R/W | 0     |

### 12.1.1.28 DAC\_CH0\_SRFT\_CTL3

Bank:0x13. Address: 0x95. DAC Channel 0 Sample Rate Control Register 3

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |            |  |     |   |
|-----|------------|--|-----|---|
| 7:0 | DA0SFTFMIH | DAC Channel 0 SR Fine Tune delta_f min high byte | R/W | 0 |
|-----|------------|--|-----|---|

### 12.1.1.29 DAC\_CH0\_SRFT\_CTL4

Bank:0x13. Address: 0x97. DAC Channel 0 Sample Rate Control Register 4

| Bit(s) | Name       | Description                                     | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | DA0SFTFMIL | DAC Channel 0 SR Fine Tune delta_f min low byte | R/W | 0     |

### 12.1.1.30 DAC\_CH1\_SR\_CTL

Bank:0x13. Address: 0x98. DAC Channel 1 Sample Rate Control Register

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7:3    | -      | Reserved for digital future use  | RW  | 00000 |
| 2:0    | DA1SRR | DAC Channel 1 SRC Ratio (interpolation, decimation):<br>000: (1,1)<br>001: (6,1)<br>010: (4,1)<br>011: (3,1)<br>100: (2,1)<br>101: (3,2)<br>110: (1,2)<br>111: (1,4) | R/W | 0     |

### 12.1.1.31 DAC\_CH1\_SR\_GAIN

Bank:0x13. Address: 0x99. DAC Channel 1 Sample Rate Control Register

| Bit(s) | Name   | Description   | R/W | Reset     |
|--------|--------|---|-----|-----------|
| 7:0    | DA1SRG | DAC Channel 1 SRC Gain, step -0.5dB:<br>同 DAC Left Channel Volume 定义。 | RW  | 1001_1011 |

### 12.1.1.32 DAC\_CH1\_SRFT\_CTL0

Bank:0x13. Address: 0x9a. DAC Channel 1 Sample Rate Fine Tune Control Register 0

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:5    | -         | Reserved for analog future use  | RW  | 000   |
| 4:3    | DA1HFHEGM | DAC Channel 1 Half Full & Half Empty Generation Mode:<br>00: Auip Pcmram Generated,<br><b>01: Reserved</b><br>10: Register Write Generated for Debug,<br>11: Reserved | R/W | 00    |
| 2      | DA1FHF    | DAC Channel 1 FIFO Half Full:<br>0: not half full<br>1: half full<br>Writing 1 to this bit will set Half Full signal to DAC SR Fine Tune.                             | R/W | 0     |
| 1      | DA1FHE    | DAC Channel 1 FIFO Half Empty:<br>0: not half empty<br>1: half empty<br>Writing 1 to this bit will set Half Empty signal to DAC SR Fine Tune.                         | R/W | 0     |
| 0      | DA1SRFTEN | DAC Channel 1 SR Fine Tune enable:<br>0: disable and bypass<br>1: enable  | R/W | 0     |

### 12.1.1.33 DAC\_CH1\_SRFT\_CTL1

Bank:0x13. Address: 0x9b. DAC Channel 1 Sample Rate Control Register 1

| Bit(s) | Name       | Description                            | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | DA1SFTFMAH | DAC Channel 1 SR Fine Tune delta_f max | R/W | 0     |

|  |  |           |  |  |
|--|--|-----------|--|--|
|  |  | high byte |  |  |
|--|--|-----------|--|--|

### 12.1.1.34 DAC\_CH1\_SRFT\_CTL2

Bank:0x13. Address: 0x9c. DAC Channel 1 Sample Rate Control Register 2

| Bit(s) | Name       | Description                                     | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | DA1SFTFMAL | DAC Channel 1 SR Fine Tune delta_f max low byte | R/W | 0     |

### 12.1.1.35 DAC\_CH1\_SRFT\_CTL3

Bank:0x13. Address: 0x9d. DAC Channel 1 Sample Rate Control Register 3

| Bit(s) | Name       | Description                                      | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | DA1SFTFMIH | DAC Channel 1 SR Fine Tune delta_f min high byte | R/W | 0     |

### 12.1.1.36 DAC\_CH1\_SRFT\_CTL4

Bank:0x13. Address: 0x9e. DAC Channel 1 Sample Rate Control Register 4

| Bit(s) | Name        | Description                                     | R/W | Reset |
|--------|-------------|---|-----|-------|
| 7:0    | DA1SFTFAMIL | DAC Channel 1 SR Fine Tune delta_f min low byte | R/W | 0     |

**注:** 在播放歌曲的时候，FIFO 的数据源选择的是 AHC。正确的使用方法是，先使能好 DAC 之后让 FIFO 处于 **reset** 状态，之后等 AHC 初始化好了，再使能 DAC FIFO，保证数据的正确顺序。

## 12.1.2 TESTMODE

### 12.1.2.1 BLOCK DIAGRAM

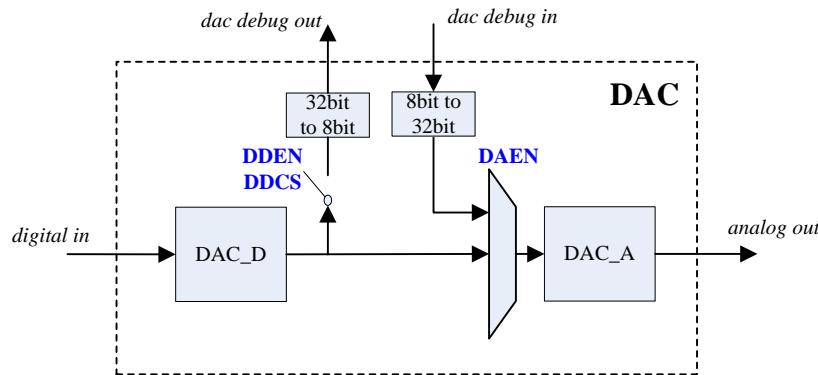


Figure 12-6 DAC debug data

### 12.1.2.2 SIGNAL DESCRIPTION

*dac debug out* 包括 1x clock out, 4x clock out, 8bit data out. 其中 32bit to 8bit 模块将 DAC\_D 送给 DAC\_A 的数据作了串行转换. 即 4x clock out 为 4 倍 DAC\_D 送给 DAC\_A 的 clock. 8bit 数据将 32bit 数据转换为[31:24] [23:16] [15:8] [7:0] 分 4 个 clock out 周期依次送出. 其中 1x clock out 用作同步信号.

*Dac debug in* 包括 1x clock in, 4x clock in, 8bit data in. 其中 8bit to 32bit 模块将外部灌入的 8 bit data in 作了并行处理再送给 DAC\_A. 4 个 4x clock in 周期里 T0,T1,T2,T3 对应的 4 个 8bit 数据 data0,data1,data2,data3 拼在一起[data0,data1,data2,data3]送给 DAC\_A. 其中 1x clock in 用作同步信号.

由于 CP,FT 测试时需要测试到 *dac debug out*, *dac debug out* MFP 需考虑各个封装能包出来.

### 12.1.2.3 TEST METHOD

TESTMODE 下, DAC CLK 切到 HOSC, 将 DAC digital 的 8bit 数据和 1x Clock, 4x clock 线引出来, 测试数字部分的电路。

Normal mode 下, 直接测试转换后的模拟信号的 THD+N 指标, 经由片外辅助测试的 IC 进行 ADC 数据采集并由 DSP 做 FFT 分析。

## 12.2 Audio ADC (林立, 彭洪, 何积军, 唐晓)

| 日期 | 版本 | 描述 | 修订人 |
|----|----|----|-----|
|----|----|----|-----|

|            |        |  |    |
|------------|--------|--|----|
| 2012-07-24 | V1. 00 | initial  | 林立 |
| 2012-08-09 | V1. 01 | 1、更改 ADCFSS 默认寄存器配置，由 10 改为 0<br>2、ADC clock description 框图删除，改为描述参考<br>CMU digital SPEC | 林立 |
| 2012-09-12 | V1. 02 | 1、设计改为二种半空半满产生方式，且内部产生方式 counter 可配。<br>2、增加内部 FM 从 DAC 输出不经过 PA 直接到到 ADC                | 林立 |
| 2012-10-20 | V2. 00 | 1、删除 FM，相应修改主框图<br>2、修改以下寄存器：<br>ADC_SRFT_CTL0、<br>ADC_SRFT_CTL5                         | 林立 |
| 2012-12-12 | V2. 02 | 3、删除 SRC Fine Tune 硬件 counter 相关寄存器及<br>配置<br>4、增加 operation manual 章节                   | 林立 |

## 12.2.1 Features

- Built-in stereo Sigma-delta ADC of 84Db SNR
- Support Multi-Sample-Rate of 8k/12k/11.025k/16k/22.05k/24k/32k/44.1/48k/96kHz
- 8\*2Level\*16bit FIFO for ADC, accessed by MCU and Audio IP, when 2\*2level empty a drq or an irq is send
- Support Microphone/FM/AUX/BT input ,which can mix two of four to ADC. This input PIN can be used as GPIO
- Built-in power supply for electret Microphone
- Power Consumption: ADC+OP(stereo) 3.5Ma (VCC=3.10V, AVCC=2.95V); Digital filter 1Ma(VDD=1.7V)
- Highpass Filter to remove op offset, also can be bypassed
- Support Sample Rate Convert
- Support Sample Rate Fine Tune
- 

## 12.2.2 Function Description

### 12.2.2.1 Audio ADC

Analog input has one mux, which can select two analog input from four analog input. The two analog input has two independent gain control. After gain control it send to a mixer , which can mix the two input and Paout. The audio ADC can record the signal after mixer.

A PLL sourced from Ext. HOSC will generates sampling clock for conversion, 24.576MHz for 48kHz series and 22.5792MHz for 44.1kHz series.

To remove offset of ADC OP, a high-pass filter (HPF) is designed.

Audio processor can access FIFO directly with 16bit-width data path, while MCU can access it with 8bit-width IO port which can output 16 bit precision data.

Sample Rate Fine Tune supports to fine tune the sample rate difference between different clock when use in USB boombox or other external clock.

## 12.2.3Module Description

### 12.2.3.1 Block Diagram

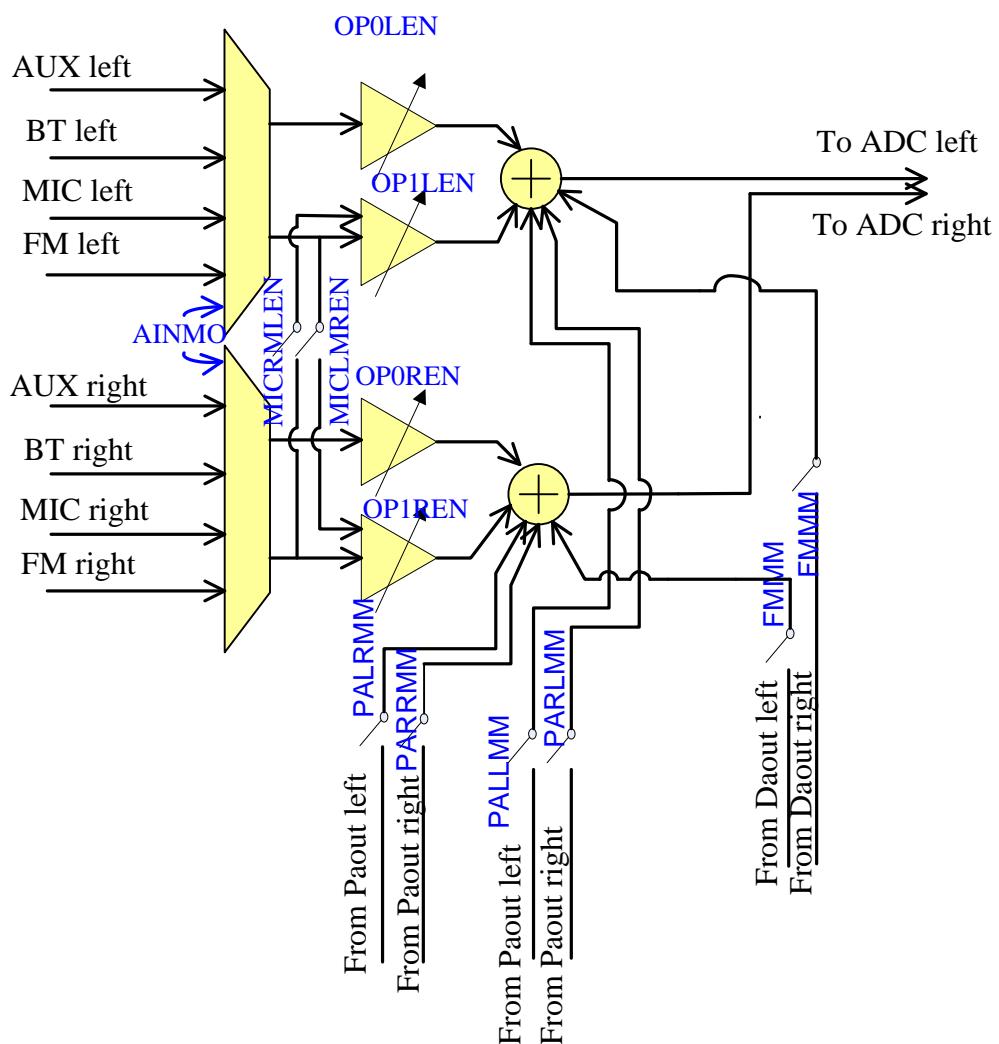


Figure 12-7 ADC Input

### 12.2.3.2 Signal Flow

Refer to [Audio ADC / DAC/I2S /SPDIF Module Signal Flow](#)

### 12.2.3.3 Clock Description

reference to CMU Digital SPEC DAC/IIS-TX controller clock.

### 12.2.3.4 PIN Description and Electrical Parameters

Audio DAC/ADC relative pins are as follows:

| Pin Name | Description                | Electrical parameters   | Attribute | Notes   |
|----------|----------------------------|---|-----------|---|
| AVCC     | Analog power               | Normally 2.9V when VCC3.1V, A 0.22Uf capacitor needed, adjustable.  | PO        | Normally not for external power supply                |
| AGND     | Analog ground              |   | AGND      |   |
| VMIC     | Power supply for Mic       | ~=2.6V, need a normally 2.2k(refer to microphone's specification) load resistor to electret microphone which Standard Operation Voltage is 2V | AO        | Not for other power supply except electret Microphone |
| AUXL     | AUX left channel input     | Biased to VREFI when enable   | AI        |   |
| AUXR     | AUX right channel input    | Biased to VREFI when enable   | AI        |   |
| MICINL   | MIC left channel input     | Biased to VREFI when enable   | AI        |   |
| MICINR   | MIC right channel input    | Biased to VREFI when enable   | AI        |   |
| FMINL    | FM left channel input      | Biased to VREFI when enable   | AI        |   |
| FMINR    | FM right channel input     | Biased to VREFI when enable   | AI        |   |
| BTL      | For BT left channel input  | Biased to VREFI when enable   | AI        |   |
| BTR      | For BT right channel input | Biased to VREFI when enable   | AI        |   |

### 12.2.3.5 AC Parameters

#### ADC Characteristics

| Symbol | Parameter                         | Test Conditions  | Value | Unit |
|--------|-----------------------------------|--|-------|------|
| THD+N  | Total Harmonic Distortion + Noise | VCC = 3.1V, AVCC = 2.95V,<br>0Dbfs, f = 1kHz, AES17 filter                             | -75   | Db   |
|        |                                   |  | 0.01  | %    |
| DR     | Dynamic Range                     | VCC= 3.1V, AVCC = 2.95V,<br>$V_{IN} = 2.6Vpp@1kHz$ , f = 20Hz –<br>20kHz, No Weighting | 84    | Db   |

### 12.2.1 Operation Manual

### 12.2.1.1 软件控制流程

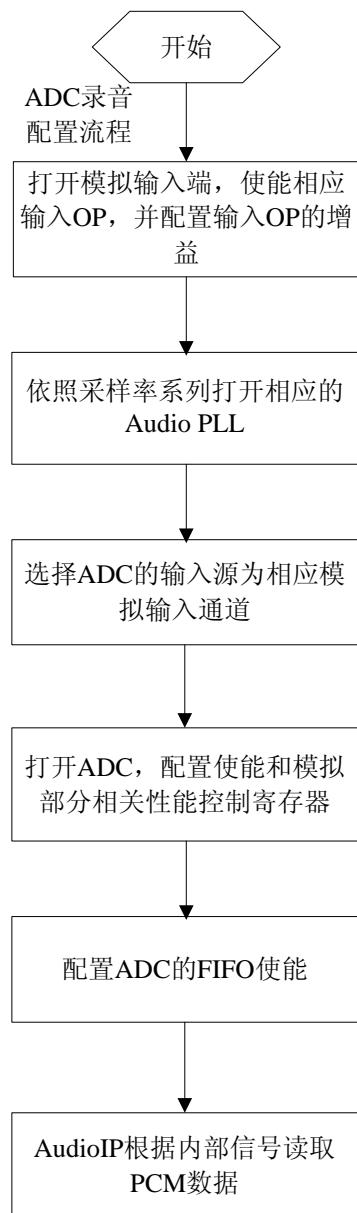


Figure 12-24 ADC Configure

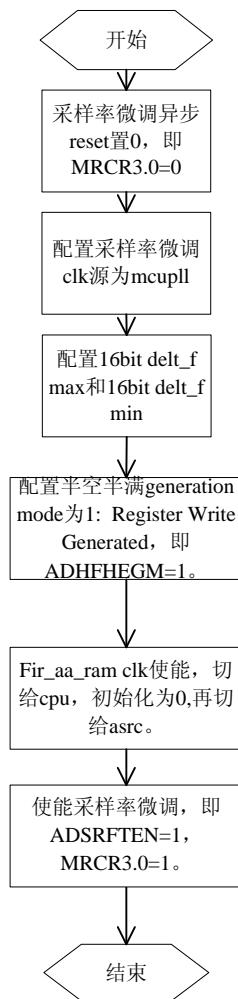


Figure 12-25 Adc 采样率微调配置流程

注:

2019、采样率微调dac ch0,ch1做在dac里，配置需先对dac的异步复位置1和使能clock。采样率微调adc做在adc里，配置需先对adc的异步复位置1和使能clock。

2, 配置采样率微调需要将asrc的异步复位在整个配置最后才置为normal。

2020、配置采样率微调clk为mcupll, mcupll需要是dac/adc clk的最少2倍，如dac采样率为48k，则dac clk为 $512 \times 48\text{kHz} = 24.576\text{MHz}$ ，如adc采样率为48k，则adc clk为 $256 \times 48\text{kHz} = 12.288\text{MHz}$ 。

2021、半空半满产生方式如果选auip，则由pcmram硬件自动产生。如选register write，则需由软件根据ram的空满状态去配置采样率微调的半空或半满。

2022、dac采样率微调寄存器配置半空，硬件选择delt\_f min，输出频率会偏快；寄存器配置半满，硬件选择delt\_f max，输出频率会偏慢。Adc采样率微调寄存器配置半空，硬件选择delt\_f max，输出频率会偏慢；寄存器配置半满，硬件选择delt\_f min，输出频率会偏快。

2023、delt\_f为Q4.20无符号格式。0x200000为1，0x400000为2，0x100000为0.5。表示采样时间为标准采样时间的倍数。

7, 需要在整个adc采样率微调配置完成之后，才使能adc左右声道，才能保证左右声道不反。

2024、采样率微调模块只适合双声道，adc/dac 配置时需要左右声道都打开。

## 12.2.2 ADC Register List

| Index | Mnemonic                      | Description                                  | BANK      |
|-------|-------------------------------|--|-----------|
| 0xf1  | <a href="#">ADC_CTL0</a>      | ADC Control Register 0                       | 0x13/0x04 |
| 0xf2  | <a href="#">AINOP_CTL</a>     | AnalogIN OP Control Register                 | 0x13/0x04 |
| 0xf3  | <a href="#">ADC_GAIN0</a>     | ADC gain Control Register0                   | 0x13/0x04 |
| 0xf4  | <a href="#">ADC_GAIN1</a>     | ADC gain Control Register1                   | 0x13/0x04 |
| 0xf5  | <a href="#">ADC_TUNE0</a>     | ADC tuning control Register 0                | 0x13/0x04 |
| 0xf6  | <a href="#">ADC_TUNE1</a>     | ADC tuning control Register 1                | 0x13/0x04 |
| 0xf7  | <a href="#">ADC_FIFO_DAT</a>  | ADC FIFO data register                       | 0x13/0x04 |
| 0xf8  | <a href="#">ADC_FIFOCTL0</a>  | ADC FIFO control register 0                  | 0x13/0x04 |
| 0xfc  | <a href="#">ADC_FIFOCTL1</a>  | ADC FIFO control register 1                  | 0x13/0x04 |
| 0xfd  | <a href="#">ADC_CTL1</a>      | ADC Control Register 1                       | 0x13/0x04 |
| 0xa2  | <a href="#">ADC_SRFT_CTL0</a> | ADC Sample Rate Fine Tune Control Register 0 | 0x13      |
| 0xa3  | <a href="#">ADC_SRFT_CTL1</a> | ADC Sample Rate Fine Tune Control Register 1 | 0x13      |
| 0xa4  | <a href="#">ADC_SRFT_CTL2</a> | ADC Sample Rate Fine Tune Control Register 2 | 0x13      |
| 0xa5  | <a href="#">ADC_SRFT_CTL3</a> | ADC Sample Rate Fine Tune Control Register 3 | 0x13      |
| 0xa6  | <a href="#">ADC_SRFT_CTL4</a> | ADC Sample Rate Fine Tune Control Register 4 | 0x13      |

## 12.2.3 Register Description

### 12.2.3.1 ADC\_CTL0

Bank:0x13, 0x04. Address: 0xf1.ADC Control Register

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7      | -      | Reversed for analog future use  | R/W | 00    |
| 6      | FMMM   | FM onchip Left & Right Channel to ADC Left & Right Channel Mixer Mute:<br>0: Mute<br>1: Dismute,connect | R/W | 0     |
| 5      | PARLMM | Paout Right Channel to ADC Left Channel Mixer Mute:<br>0: Mute<br>1: Dismute,connect                    | R/W | 0     |
| 4      | PALRMM | Paout Left Channel to ADC Right Channel Mixer Mute:   | R/W | 0     |

|   |        |  |     |   |
|---|--------|--|-----|---|
|   |        | 0: Mute<br>1: Dismute,connect  |     |   |
| 3 | PARRMM | Paout Right Channel to ADC Right Channel<br>Mixer Mute:<br>0: Mute<br>1: Dismute,connect | R/W | 0 |
| 2 | PALLMM | Paout Left Channel to ADC Left Channel Mixer<br>Mute:<br>0: Mute<br>1: Dismute,connect   | R/W | 0 |
| 1 | ADCLEN | A/D left channel enable; 0: Disable, 1: Enable   | R/W | 0 |
| 0 | ADCREN | A/D right channel enable; 0: Disable, 1: Enable  | R/W | 0 |

### 12.2.3.2 AINOP\_CTL

Bank:0x13, 0x04. Address: 0xf2, AIN OP Control Register

| Bit(s) | Name    | Description   | R/W | Reset |
|--------|---------|---|-----|-------|
| 7      | VRDAEN  | AD Reference VRDA Enable<br>0: Disable,<br>1: Enable  | R/W | 0     |
| 6      | VMICEN  | VMIC Control<br>0: Disable,<br>1: Enable  | R/W | 0     |
| 5      | OP1ZCEN | AnalogIN OP1 change at zero-cross enable<br>0: Disable,<br>1: Enable  | R/W | 0     |
| 4      | OP0ZCEN | AnalogIN OP0 change at zero-cross enable<br>0: Disable,<br>1: Enable  | R/W | 0     |
| 3      | OP1GB   | AnalogIN OP1 Gain Boost Enable:<br>0: Disable<br>1: Enable. When enable, OP1 Gain adds 10Db.<br>Note:<br>电路实际做法为，使能该位则 adc mixer, pa mixer 选用另一个衰减了 10Db 的电阻。 | R/W | 0     |
| 2:0    | AINMO   | AnalogIN Mux Output Select:<br>000: Aux & BT<br>001: Aux & Mic<br>010: Aux & FM<br>011: BT & Mic<br>100: BT & FM<br>101: FM & Mic                             | R/W | 110   |

|  |  |                   |  |  |
|--|--|-------------------|--|--|
|  |  | Others : reserved |  |  |
|--|--|-------------------|--|--|

### 12.2.3.3 ADC\_GAIN0

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:5    | -      | Reversed for Analog future use  | R/W | 0     |
| 4      | OP0REN | AnalogIN OP0 Right Channel Enable:<br>0: Disable<br>1: Enable   | R/W | 0     |
| 3      | OP0LEN | AnalogIN OP0 Left Channel Enable:<br>0: Disable<br>1: Enable  | R/W | 0     |
| 2:0    | OP0G   | AnalogIN OP0 Gain:<br>000: -12Db<br>001: -3Db<br>010: 0Db<br>011: 1.5Db<br>100: 3.0Db<br>101: 4.5Db<br>110: 6.0Db<br>111: 7.5db | R/W | 0010  |

Bank:0x13, 0x04. Address: 0xf3, ADC Gain 0 Control Register

### 12.2.3.4 ADC\_GAIN1

Bank:0x13, 0x04. Address: 0xf4, ADC Gain 1 Control Register

| Bit(s) | Name     | Description  | R/W | Reset |
|--------|----------|--|-----|-------|
| 7      | MICRMLEN | Micin Right Channel Mix to Left Channel Enable:<br>0: Disable<br>1: Enable | R/W | 0     |
| 6      | MICLMREN | Micin Left Channel Mix to Right Channel Enable:<br>0: Disable<br>1: Enable | R/W | 0     |
| 5      | OP1REN   | AnalogIN OP1 Right Channel Enable:<br>0: Disable<br>1: Enable              | R/W | 0     |

|     |        |   |     |      |
|-----|--------|---|-----|------|
| 4   | OP1LEN | AnalogIN OP1 Left Channel Enable:<br>0: Disable<br>1: Enable  | R/W | 0    |
| 3:0 | OP1G   | AnalogIN OP1 Gain:<br>0000: -12Db<br>0001: -3Db<br>0010: 0Db<br>0011: 1.5Db<br>0100: 3.0Db<br>0101: 4.5Db<br>0110: 6.0Db<br>0111: 7.5db<br>1000: 26Db (x20)<br>1001: 30Db (x30)<br>1010: 31.5Db (x38)<br>1011: 33Db (x45)<br>1100: 34.5Db (x53)<br>1101: 36Db (x63)<br>1110: 37.5Db (x75)<br>1111: 39Db (x89) | R/W | 0010 |

### 12.2.3.5 ADC\_TUNE0

Bank:0x13, 0x04. Address: 0xf5, ADC tuning control Register 0

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:5    | OPBC1  | The bias current select for OPAD1 in A/D:<br>2025、 4Ua<br>2026、 5Ua<br>...<br>111 11Ua          | R/W | 011   |
| 4      | TADCBC | Total ADC Bias current select<br>0 normal(4Ua)<br>1 +50%  | R/W | 0     |
| 3:2    | OPBC23 | The bias current select for OPAD2/3 in A/D:<br>2027、 Lower<br>2028、 Low<br>10 High<br>11 Higher | R/W | 01    |
| 1:0    | PABC   | Audio A/D Pre-amplifiers(AIN/ LINEIN OP) bias current select:<br>2029、 Lower                    | R/W | 01    |

|  |  |                                   |  |  |
|--|--|-----------------------------------|--|--|
|  |  | 2030、 Low<br>10 High<br>11 Higher |  |  |
|--|--|-----------------------------------|--|--|

This register can only be set by system.

### 12.2.3.6 ADC\_TUNE1

**Bank:0x13, 0x04. Address: 0xf6, ADC tuning control Register 1**

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:6    | LPFBC  | Audio A/D LPF bias current select:<br>2031、 Lower<br>2032、 Low<br>10 High<br>11 Higher        | R/W | 10    |
| 5:3    | VRDABC | Audio A/D Voltage Reference bias current select:<br>2033、 3Ua<br>2034、 4Ua<br>...<br>111 10Ua | R/W | 010   |
| 2:0    | -      | Reserved for analog future use  | R/W | 001   |

### 12.2.3.7 ADC\_FIFO\_DAT

**Bank:0x13, 0x04. Address: 0xf7, ADC FIFO DATA Register**

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:0    | ADCDAT | Audio A/D data<br>A/D in 16-bit mode: Firstly read the left channel high byte, low byte; and then read the right channel high byte, low byte. | R   | XX    |

### 12.2.3.8 ADC\_FIFOCCTL0

**Bank:0x13, 0x04. Address: 0xf8, ADC FIFO control register**

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7      | ADCFIEN   | Audio A/D FIFO Data Ready IRQ Enable.<br>0: Disable 1: Enable | R/W | 0     |
| 6      | ADCFDRQEN | Audio A/D FIFO Data Ready DRQ Enable.<br>0: Disable 1: Enable | R/W | 0     |

|     |           |   |     |   |
|-----|-----------|---|-----|---|
| 5   | ADCFEF    | Audio A/D FIFO EMPTY flag, Read Only,<br>0: empty, 1: not empty                           | R   | 0 |
| 4   | -         | Reversed for digital future use   | R/W | 0 |
| 3   | ADCFIRQPD | Audio A/D FIFO IRQ pending bit.<br>Writing 1 to this bit to clear it, otherwise unchanged | R/W | 0 |
| 2   | ADCFRST   | Audio A/D FIFO reset<br>0: reset A/D FIFO & byte counter,<br>1: enable FIFO               | R/W | 0 |
| 1:0 | -         | Reversed for digital future use   | R/W | 0 |

A/D FIFO 规格为 8level x 2channel x 16bit/sample,由软件控制以支持 MCU IRQ 和 DMA DRQ 数据传输。数据从 ADC 中产生，以固定频率轮流先左后右写入 ADC FIFO, FIFO 中数据达到一定 6\*2level 时即 2\*2level 空时将分别向 MCU 和 DMA 发出中断请求(分别有控制 bit 选择是否向两者发出中断)。当 ADC 选择单声道的时候仅仅向 ADC FIFO 写入选中通道的数据。

对于 I2S 的输入信号也是通过 ADC 的 FIFO 保存到内部 memory 中的。

A/D FIFO 在双声道对 **ADC\_DAT** 连续 4 次读取可获得左右声道的高低字节，顺序为先左后右，先高后低；单声道 16bit 模式下只需 2 次即可。

因为 MCU 和 DMA 在读取 ADC\_FIFO 数据时候是通过多次访问一个 8bit 的寄存器（ADC\_DAT）来实现的，所以电路上需要将 16bit 的 ADC 原始 PCM 数据分割成 2 bytes，同时利用了一个计数器来判断当前读取的是哪一部分。

在每次使用 ADC\_FIFO 前都需要对这个计数器进行复位（A/D Fifo Reset），这样才能保证 MCU/DMA 读取的第一个 byte 是正常 PCM 数据的最高 8 位。

如果中间出现意外打断的话，双声道条件下可能出现不确定前一次读取哪一个声道，也不知道读到哪一笔（高 /低）数据，也需要重新复位一次 FIFO。

DMA 和 MCU 时钟的源头是一致的。MCU 搬移 A/D 数据时，也应该对 **ADC\_DAT** 连续读取 4 次，即可获取左右声道的数据。

DMA 打开后，硬件将自动将 FIFO 中数据搬空为止（不存在成对的有效数据）；MCU 搬取数据时，应不断对 A/D FIFO 的 EMPTY FLAG 进行查询，由软件保证将数据搬空，且读取数据完整。在双声道模式和单声道模式间切换时需先将 A/D FIFO reset。

AHC 访问 ADC 时候自动控制与 ADC 的握手信号。AHC enable 后，ADC FIFO 发出 AD REQ 信号（AD REQ 拉高），AHC 硬件电路响应后开始搬数据，一直到 REQ 信号被拉低，AHC 响应停止搬数据。无需软件干预此过程。

### 12.2.3.9 ADC\_FIFOCCTL1

Bank:0x13, 0x04. Address: 0xfc, ADC FIFO control register

| Bit(s) | Name  | Description                              | R/W | Reset |
|--------|-------|--|-----|-------|
| 7:6    | DATGN | ADC digital gain control<br>00: 0Db (X1) | R/W | 00    |

|     |         |   |     |   |
|-----|---------|---|-----|---|
|     |         | 01: +6Db (X2)<br>10: +12Db (X4)<br>11: reserved<br><br>It will shift PCM data to left before writing them to ADC FIFO to achieve gain effect, and should be set before you receive ADC data. Please refer to DAC signal flow diagram. |     |   |
| 5   | -       | Reserved for digital future use   | R/W | 0 |
| 4   | ADCFSS  | ADC FIFO access MUX select.<br>0: MCU<br>1: DMA   | R/W | 0 |
| 3:1 | -       | Reserved for digital future use   | R   | 0 |
| 0   | ADCTHRH | ADC SR Fine Tune SOURCE select<br>0: ADC<br>1: I2S RX   | R/W | 0 |

### 12.2.3.10 ADC\_CTL1

Bank:0x13, 0x04. Address: 0xfd, ADC CONTROL register 1

| Bit(s) | Name  | Description   | R/W | Reset |
|--------|-------|---|-----|-------|
| 7:2    | -     | Reversed for digital future use                         | R/W | 0000  |
| 1      | HPFEN | High Pass Filter Enable:<br>0x0: Disable<br>0x1: Enable | R/W | 0     |
| 0      | ADDD  | ADC digital debug enable:<br>0: disable 1: enable       | R/W | 0     |

### 12.2.3.11 ADC\_SRFT\_CTL0

Bank:0x13. Address: 0xa2, ADC Sample Rate Fine Tune Control Register 0

| Bit(s) | Name     | Description  | R/W | Reset |
|--------|----------|--|-----|-------|
| 7:4    | -        | Reserved for analog future use   | RW  | 0000  |
| 3      | ADHFHEGM | ADC Half Full & Half Empty Generation Mode:<br>0: Reserved<br>1: Register Write Generated for Debug, | R/W | 0     |
| 2      | ADFHF    | ADC FIFO Half Full:<br>0: not half full  | R/W | 0     |

|   |          |   |     |   |
|---|----------|---|-----|---|
|   |          | 1: half full<br>Writing 1 to this bit will set Half Full signal to ADC SR Fine Tune.  |     |   |
| 1 | ADFHE    | ADC FIFO Half Empty:<br>0: not half empty<br>1: half empty<br>Writing 1 to this bit will set Half Empty signal to ADC SR Fine Tune. | R/W | 0 |
| 0 | ADSRFTEN | ADC SR Fine Tune enable:<br>0: disable and bypass<br>1: enable  | R/W | 0 |

### 12.2.3.12 ADC\_SRFT\_CTL1

Bank:0x13. Address: 0xa3, ADC Sample Rate Control Register 1

| Bit(s) | Name      | Description                            | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | ADSFTFMAH | ADC SR Fine Tune delta_f max high byte | R/W | 0     |

### 12.2.3.13 ADC\_SRFT\_CTL2

Bank:0x13.Address: 0xa4, ADC Sample Rate Control Register 2

| Bit(s) | Name      | Description                           | R/W | Reset |
|--------|-----------|---------------------------------------|-----|-------|
| 7:0    | ADSFTFMAL | ADC SR Fine Tune delta_f max low byte | R/W | 0     |

### 12.2.3.14 ADC\_SRFT\_CTL3

Bank:0x13.Address: 0xa5, ADC Sample Rate Control Register 3

| Bit(s) | Name      | Description                            | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | ADSFTFMIH | ADC SR Fine Tune delta_f min high byte | R/W | 0     |

### 12.2.3.15 ADC\_SRFT\_CTL4

Bank:0x13.Address: 0xa6, ADC Sample Rate Control Register 4

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |           |                                       |     |   |
|-----|-----------|---------------------------------------|-----|---|
| 7:0 | ADSFTFMIL | ADC SR Fine Tune delta_f min low byte | R/W | 0 |
|-----|-----------|---------------------------------------|-----|---|

## 12.2.4 TESTMODE

### 12.2.4.1 BLOCK DIAGRAM

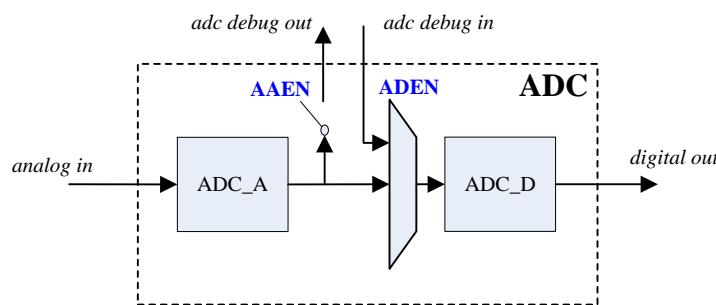


Figure 12-26 ADC debug data

*adc debug out* 包括 128fs clock out, 256fs clock out, data\_l\_0 out, data\_l\_1 out, data\_r\_0 out, data\_r\_1 out.

*Adc debug in* 包括 256fs clock in, data\_l\_0 in, data\_l\_1 in, data\_r\_0 in, data\_r\_1 in.

由于 CP,FT 测试时需要从 *adc debug in* 灌 clock 和 data 测试, *adc debug in* MFP 需考虑各个封装能包出来.

### 12.2.4.2 SIGNAL DESCRIPTION

TESTMODE 下, 由测试机台外灌 pattern, 从 GPIO 灌入, 经过 ADC 的 debug 端口进入内部 ADC 电路。

Normal mode 下, 由片外 IC 输入正弦波给 IC 的 AUX/FM/MIC/BT, 经由 ADC 采样获取数据后直接送给 DAC, 最后由 AOUTL/R 输出, 片外 IC 再用 ADC 采样并由 DSP 做 FFT 分析 THD+N 指标。

### 12.2.4.3 TEST METHOD

TESTMODE 下, ADC CLK 切到 HOSC, 程序内部控制 ADC 的数字接口电路接受数据并最终分析, ADC 收取数据后将数据送到外部 memory, 由测试机台将数据与标准数据进行比对。

Normal mode 下, 直接测试转换后的模拟信号的 THD+N 指标, 经由片外辅助测试的 IC 进行 ADC 数据采集并由 DSP 做 FFT 分析。

## 12.3 SPDIF (林立, 彭洪, 孙睿)

| 日期         | 版本     | 描述   | 修订人 |
|------------|--------|--|-----|
| 2012-07-24 | V1. 00 | initial  | 林立  |
| 2012-08-09 | V1. 01 | 1、更改 SPDIFRX 寄存器位 RXFR 的描述。<br>2、增加描述 SPDIFRX_CSTAT 同时作传送脉宽 counter 用。同时删除 SPDIFRX_CNT_RD, 及同时更改后面寄存器的地址。<br>3、更改描述 SRDIFRX Block IRQ 为 Channel Statue IRQ, 同时更改 IRQ 框图。<br>4、更改 PA_VOLUME 描述, pa volume total 41 level 改为 8 level。<br>5、SPDIFRX clock description 框图删除, 改为描述参考 CMU digital SPEC.<br>6、由于 SPDIFRX 设计更改, 删除采样率检测 SAMRD 及采样率 error pending。增加 CSFES 状态位。 | 林立  |
| 2012-09-12 | V1. 02 | 1、根据设计结果将 R/W Reserve 寄存器改为 R;<br>2、增加 3 个 bit 作 debug 用<br>3、SPDIFRX_CSTAT 和 SPDIFRX_DAT:描述   | 林立  |
| 2012-11-06 | V2. 01 | <u>1、增加 SPDIFRX 电路, 增加寄存器 RXPBDC, CSFU, RXCP.</u><br><u>2、增加 PWCTOP, 以支持 SPDIF 拔线 time out 场景.</u>   | 林立  |
| 2012-12-12 | V2. 02 | 1、增加完善 opertaion manual 章节   | 林立  |
| 2013-07-05 | V2. 04 | 1、增加状态比特位 CSPCM 和 CSCSM 软件可以查询, 硬件自动检测非支持源且丢 0.  | 林立  |

### 12.3.1 Features

- SPDIF supports receiver mode only1342770609
- Supports sampling rate 192k,96k,48k,44.1k,32k
- 8\*2level\*24bit fifo, when 8level empty a drq is send

### **12.3.2 Function Description**

SPDIF is used to receive digital signals of a number of formats. It supports sample rate up to 192k. The data is sent using Biphase mark code, which has either one or two transitions for every bit, allowing the original sample clock to be extracted from the signal itself.

SPDIF is meant to be used for transmitting 20 bit audio data streams plus other related information. To transmit sources with less than 20 bits of sample accuracy, the superfluous bits will be set to zero. SPDIF can also transport 24 bit samples by way of four extra bits, but not all equipment supports this, and might ignore these extra bits.

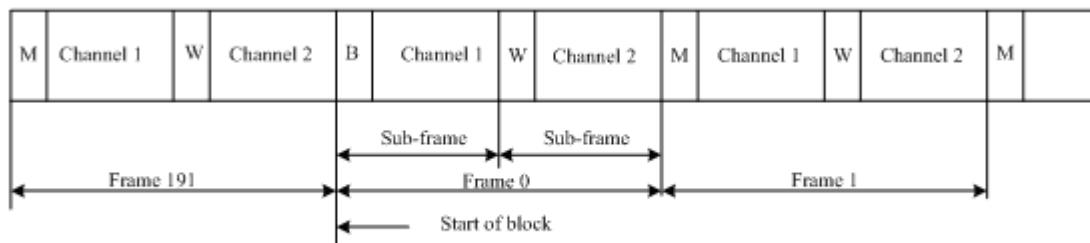


Figure 12-11 Frame format

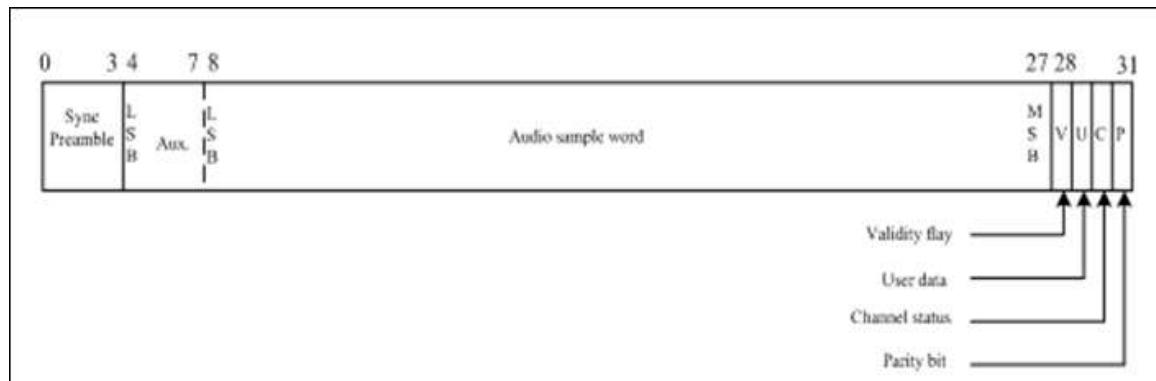
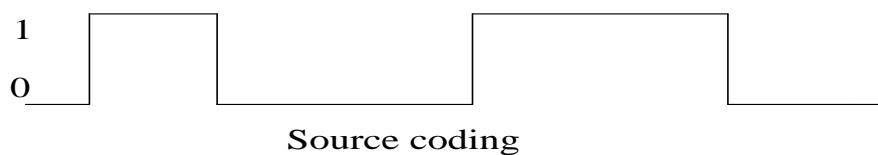


Figure 12-12 Sub-frame format



### Channel coding(biphase mark)

Figure 12-13 Coding

## 12.3.3 Module Description

### 12.3.3.1 Block Diagram

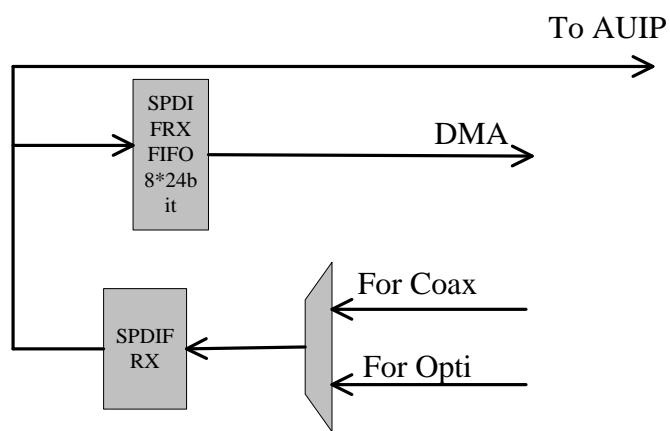


Figure 12-14 SPDIFRX

### 12.3.3.2 Signal Flow

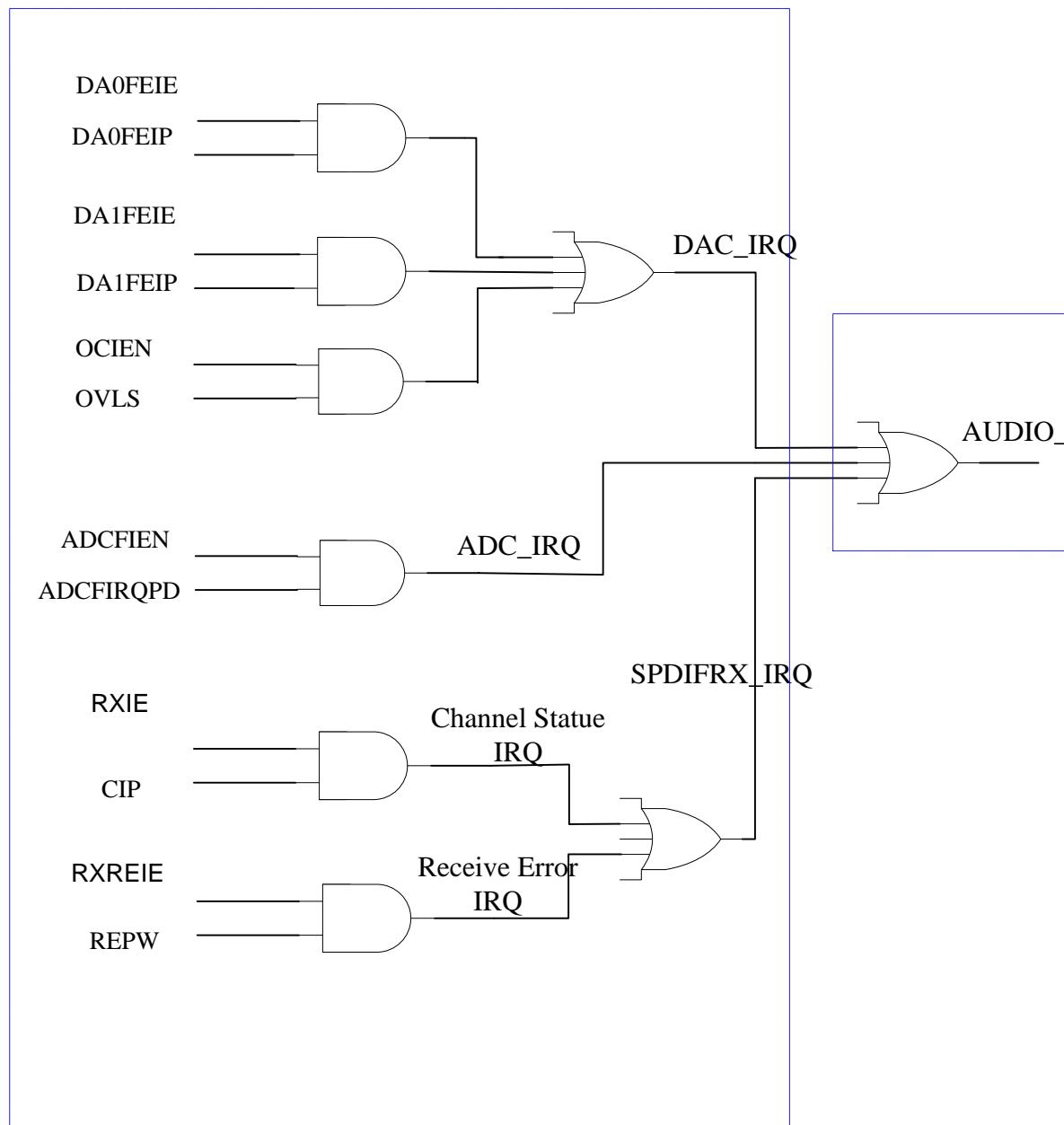


Figure 12-15 DAC/ADC/SPDIF IRQ

DAC 模块, ADC 模块, SPDIFRX 模块分别产生 DAC\_IRQ, ADC\_IRQ, SPDIFRX\_IRQ 信号给系统中断模块, 系统中断模块再将这三个中断合成 AUDIO\_IRQ。

其中 SPDIFRX IRQ 由 Channel Status IRQ 和 Receive Error IRQ 取或。

### 12.3.3.3 Clock Description

reference to CMU Digital SPEC ADC and SPDIF rx controller clock.

### 12.3.3.4 PIN Description and Electrical Parameters

| Pin Name    | Description                       | Electrical parameters | Attribute | Notes |
|-------------|-----------------------------------|-----------------------|-----------|-------|
| SPDIFRX_CH0 | SPDIF Receiver<br>Input Channel 0 | Max 24.576Mbps        | I         |       |
| SPDIFRX_CH1 | SPDIF Receiver<br>Input Channel 1 | Max 24.576Mbps        | I         |       |

### 12.3.1 Operation Manual

#### 12.3.1.1 软件控制流程

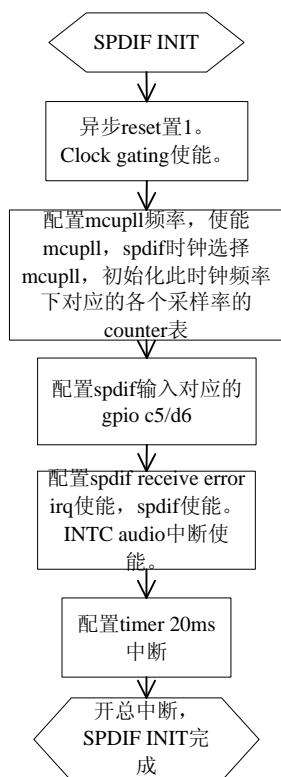


Figure 12-27 Spdif 初始化配置

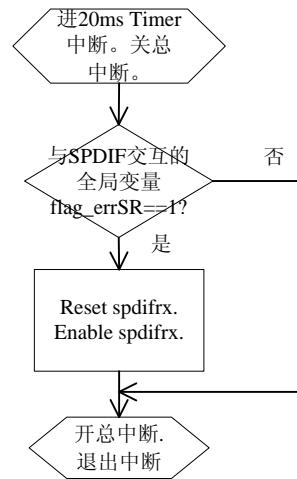


Figure 12-28 Timer 中断配置

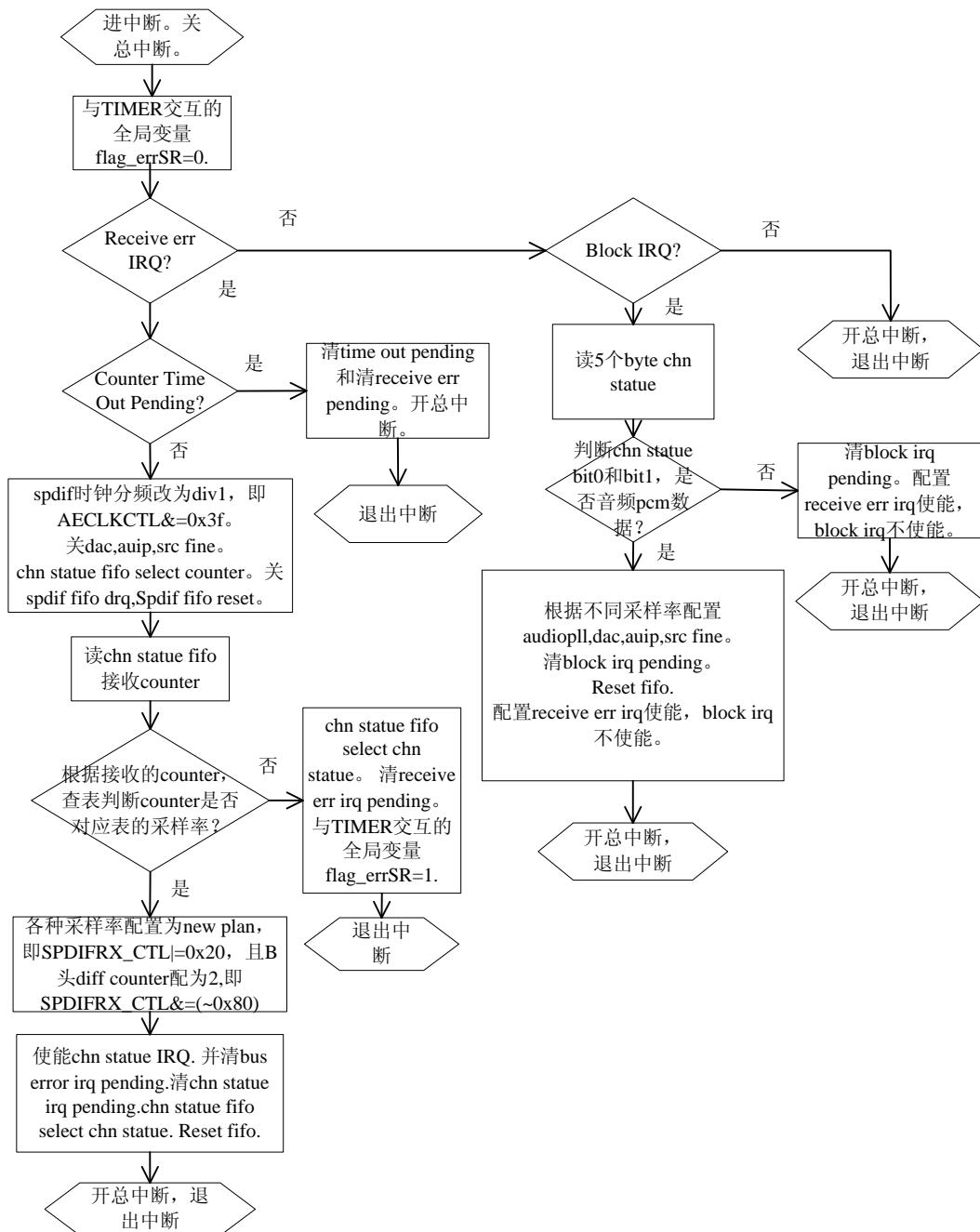


Figure 12-29 Spdif 播歌软件流程

注：

1, Spdif软件配置包括初始化配置， Timer中断， 播歌配置三部分。

2, spdif时钟mcupll不一样， Counter表也不一样。

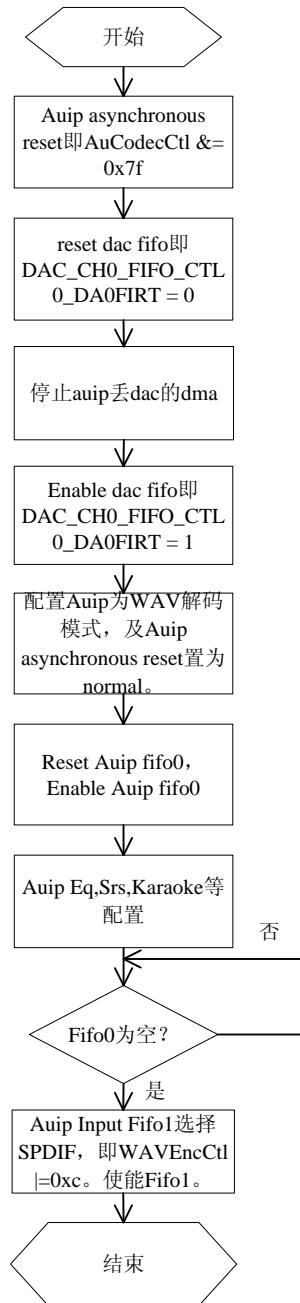


Figure 12-30 Spdif-Auip 配置

注：

1, 需要按照reset auip,reset dac fifo, abort dma, enable dac fifo,enable auip顺序才能保证左右声道不反。

2, Auip配置先传fifo0数据，再传fifo1数据。

### 12.3.2 SPDIFRX Register List

| Index | Mnemonic                    | Description              | BANK |
|-------|-----------------------------|--------------------------|------|
| 0xb0  | <a href="#">SPDIFRX_CTL</a> | SPDIFRX Control Register | 0x13 |

|      |                                 |                                      |      |
|------|---------------------------------|--------------------------------------|------|
| 0xb1 | <a href="#">SPDIFRX_STAT</a>    | SPDIFRX Statue Register              | 0x13 |
| 0xb2 | <a href="#">SPDIFRX_CSTAT</a>   | SPDIFRX Channel Statue Register      | 0x13 |
| 0xb3 | <a href="#">SPDIFRX_DEBUG</a>   | SPDIFRX Debug Register               | 0x13 |
| 0xb4 | <a href="#">SPDIFRX_DAT</a>     | SPDIFRX Data Register                | 0x13 |
| 0xb5 | <a href="#">SPDIFRX_CNT_WRO</a> | SPDIFRX Counter for Write Register 0 | 0x13 |
| 0xb6 | <a href="#">SPDIFRX_CNT_WR1</a> | SPDIFRX Counter for Write Register 1 | 0x13 |
| 0xb7 | <a href="#">SPDIFRX_CNT_WR2</a> | SPDIFRX Counter for Write Register 2 | 0x13 |

## 12.3.1 Register Description

### 12.3.1.1 SPDIFRX\_CTL

Bank:0x13. Address: 0xb0, SPDIFRX Control Register 0

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7      | RXPBDC | SPDIFRX Preamble B diff Counter:<br>0: counter =2<br>1: counter =3  | R/W | 0     |
| 6      | CSFU   | Channel Statue Fifo Usage:<br>0: for Receive Pulse Width Counter<br>1: for Receive Channel Statue                         | R/W | 0     |
| 5      | RXCP   | SPDIFRX Circuit Plan:<br>0: Plan 0 (for 192k)<br>1: Plan 1  | R/W | 0     |
| 4      | RXREIE | SPDIFRX Receive Error IRQ Enable:<br>0: Disable<br>1: Enable  | R/W | 0     |
| 3      | RXIE   | SPDIFRX Block IRQ Enable:<br>0: Disable<br>1: Enable  | R/W | 0     |
| 2      | RXDE   | SPDIFRX Fifo Full DRQ Enable:<br>0: Disable<br>1: Enable<br>Note:<br>12level data send DRQ.                               | R/W | 0     |
| 1      | RXFR   | SPDIFRX Fifo Reset:<br>1: Fifo Reset Valid<br>0: Fifo Reset Invalid ( normal work)<br>Note:<br>根据 designer 设计更改。应用时需要写 1。 | R/W | 0     |
| 0      | EN     | SPDIFRX Enable:<br>0: Disable (will reset RX state machine)<br>1: Enable  | R/W | 0     |

### 12.3.1.2 SPDIFRX\_STAT

Bank:0x13. Address: 0xb1, SPDIFRX Statue Register

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7      | T1SEL  | T1_sel:<br>0: normal<br>1: expand  | R/W | 0     |
| 6      | T2SEL  | T2_sel:<br>0: normal<br>1: expand  | R/W | 0     |
| 5      | T3SEL  | T3_sel:<br>0: normal<br>1: expand  | R/W | 0     |
| 4      | PWCTOP | Pulse Width Counter Timer Out Pending:<br>0: Not Time Out Pending<br>1: Time Out Pending             | R/W | 0     |
| 3      | CSFES  | Channel Statue Fifo Empty statue:<br>1: Empty<br>0: No Empty   | R   | 1     |
| 2      | RFFS   | RX FIFO full statue<br>0: no full<br>1: full   | R   | 0     |
| 1      | RFEM   | RX FIFO Empty.<br>1: Empty<br>0: No Empty  | R   | 1     |
| 0      | CIP    | SPDIF Channel Statue IRQ Pending Bit.<br>0: No IRQ<br>1: IRQ<br>Writing 1 to this bit will clear it. | R/W | 0     |

### 12.3.1.3 SPDIFRX\_CSTAT

Bank:0x13. Address: 0xb2, SPDIFRX Channel Statue Register

| Bit(s) | Name    | Description              | R/W | Reset |
|--------|---------|--------------------------|-----|-------|
| 7:0    | RXCSTAT | SPDIF RX Channel Status. | R   | 0     |

Notes:

There is 40bits channel statuses per one block (192 frames) transfer, only the left channel statuses. All this 5 bytes status bits are mapped into this register, and can be read as 0~7bits, 8~15bits, 16~23bit, 24~31bit, 32~39bits at five continuous reading. The channel statuses will be update when one new block is received and a block IRQ will be sent.

同时此 fifo 作为更改采样率时 Rx 模块将脉宽 counter 传送给。

### 12.3.1.4 SPDIFRX\_DEBUG

Bank:0x13. Address: 0xb3, SPDIFRX Debug Register

| Bit(s) | Name  | Description  | R/W | Reset |
|--------|-------|--|-----|-------|
| 7      | CSCSM | Channel Statue Detect Consumer Use Statue Bit.<br>0: Professional Use<br>1: Consumer Use                         | R   | 0     |
| 6:4    | -     | For debug  | R/W | 000   |
| 3      | RFEP  | RX FIFO Error : Pending Bit.<br>0: No Error<br>1: Error<br>Writing 1 to this bit will clear it.                  | R/W | 0     |
| 2      | CSPCM | Channel Statue Detect PCM Statue Bit.<br>0: Not PCM<br>1: PCM  | R   | 0     |
| 1      | REPA  | Receive Error: Parity Error Pending Bit.<br>0: No Error<br>1: Error<br>Writing 1 to this bit will clear it.      | R/W | 0     |
| 0      | REPW  | Receive Error: Pulse Width Error Pending Bit.<br>0: No Error<br>1: Error<br>Writing 1 to this bit will clear it. | R/W | 0     |

### 12.3.1.5 SPDIFRX\_DAT

Bank:0x13. Address: 0xb4, SPDIFRX Data Register

| Bit(s) | Name  | Description                             | R/W | Reset |
|--------|-------|---|-----|-------|
| 7:0    | RXDAT | SPDIFRX Data, configure for Dma Access. | R   | 0     |

### 12.3.1.6 SPDIFRX\_CNT\_WR0

Bank:0x13. Address: 0xb5, SPDIFRX Counter for Write Register 0

| Bit(s) | Name     | Description                           | R/W | Reset |
|--------|----------|---------------------------------------|-----|-------|
| 7:0    | RXCNTWR0 | SPDIFRX Counter for Write Register 0. | R/W | 0     |

### 12.3.1.7 SPDIFRX\_CNT\_WR1

Bank:0x13. Address: 0xb6, SPDIFRX Counter for Write Register 1

| Bit(s) | Name     | Description                           | R/W | Reset |
|--------|----------|---------------------------------------|-----|-------|
| 7:0    | RXCNTWR1 | SPDIFRX Counter for Write Register 1. | R/W | 0     |

### 12.3.1.8 SPDIFRX\_CNT\_WR2

Bank:0x13. Address: 0xb7, SPDIFRX Counter for Write Register 2

| Bit(s) | Name     | Description                           | R/W | Reset |
|--------|----------|---------------------------------------|-----|-------|
| 7:0    | RXCNTWR2 | SPDIFRX Counter for Write Register 2. | R/W | 0     |

## 12.3.2 TESTMODE

### 12.3.2.1 BLOCK DIAGRAM

### 12.3.2.2 SIGNAL DESCRIPTION

### 12.3.2.3 TEST METHOD

SPDIFRX CLK 切到 HOSC，外灌采样率小于或等于 44.1k 的信号到 SPDIFRX PIN，测试 SPDIFRX FIFO 收到的数据是否正确。

## 13 UI

### 13.1 LCD & LED (赵天亮、黄俏)

| 日期         | 版本     | 描述  | 修订人 |
|------------|--------|---|-----|
| 2012-07-24 | V1. 00 | initial   | 赵天亮 |
| 2012-09-12 | V1. 02 | 寄存器 LCD_mode 添加 bit4: <b>LCD_OUT_EN</b>   | 赵天亮 |
| 2012-12-12 | V2. 02 | 完善 operation manual 章节  | 赵天亮 |
| 2013-07-05 | V2. 04 | 1、模块使用指导部分加上 IC 验证后的说明: clk distribution 部分修改图片和频率设置说明。<br>2、增加模拟部分的设置说明: Seg_lcd 的 1/3 , 2/3 电压使能, Led 模块的恒流源使能。 | 赵天亮 |

#### 13.1.1 Features

- Support 3com or 4com Segment and Com Driving Timing
- Support 4com or 8com Digit-LED Driving Timing
- Support PWM Power Saving Mode
- Support SEG/COM LCD and Digit-LED Blinking
- Support 8bit dot matrix LCD through EX\_memory interface

#### 13.1.2 Function Description

##### 13.1.2.1 Seg & Com Driving Timing

#### 4 Com Row Inverting:

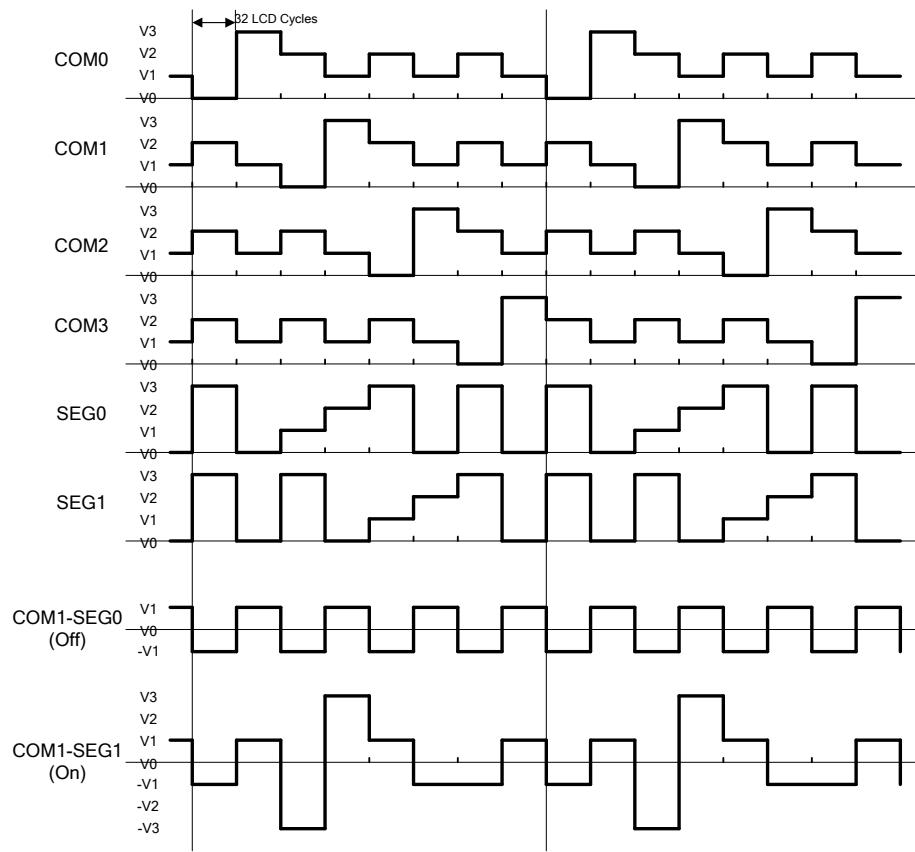


Figure 13-1 4 Com Row Inverting

#### 4 Com Frame Inverting:

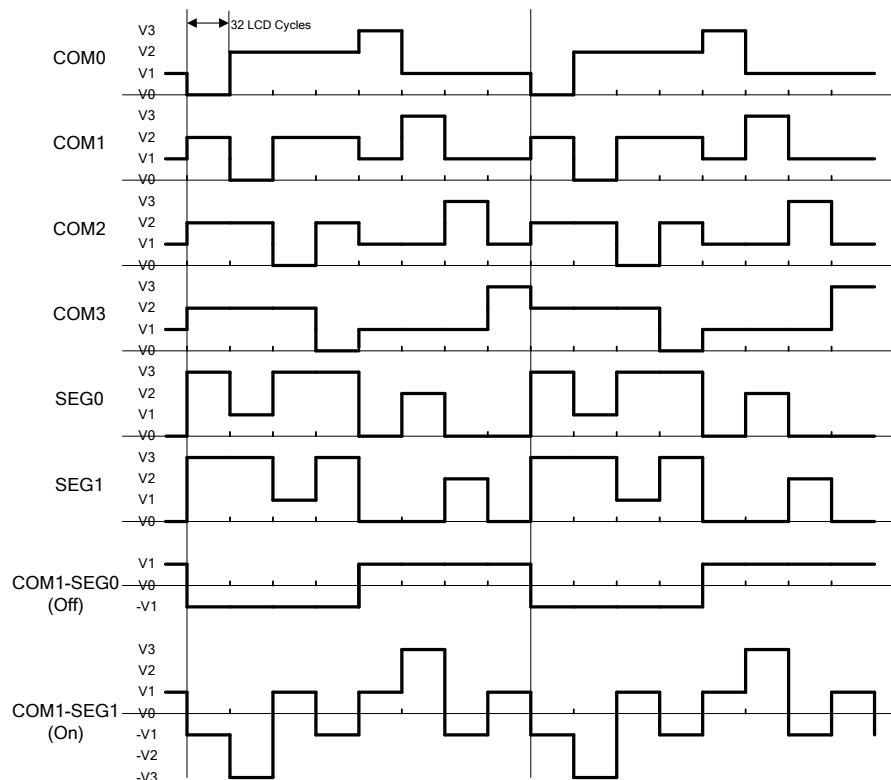


Figure 13-2 4 Com Frame Inverting

### 3 Com Row Inverting:

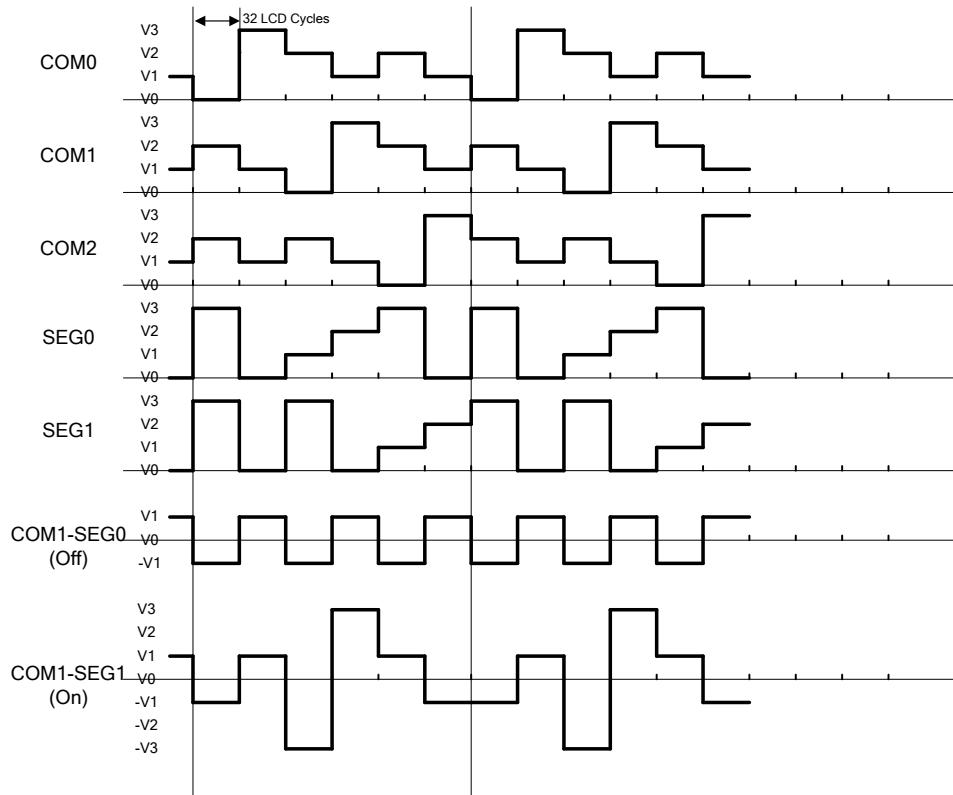


Figure 13-3 3 Com Frame Inverting

### 3 Com Frame Inverting:

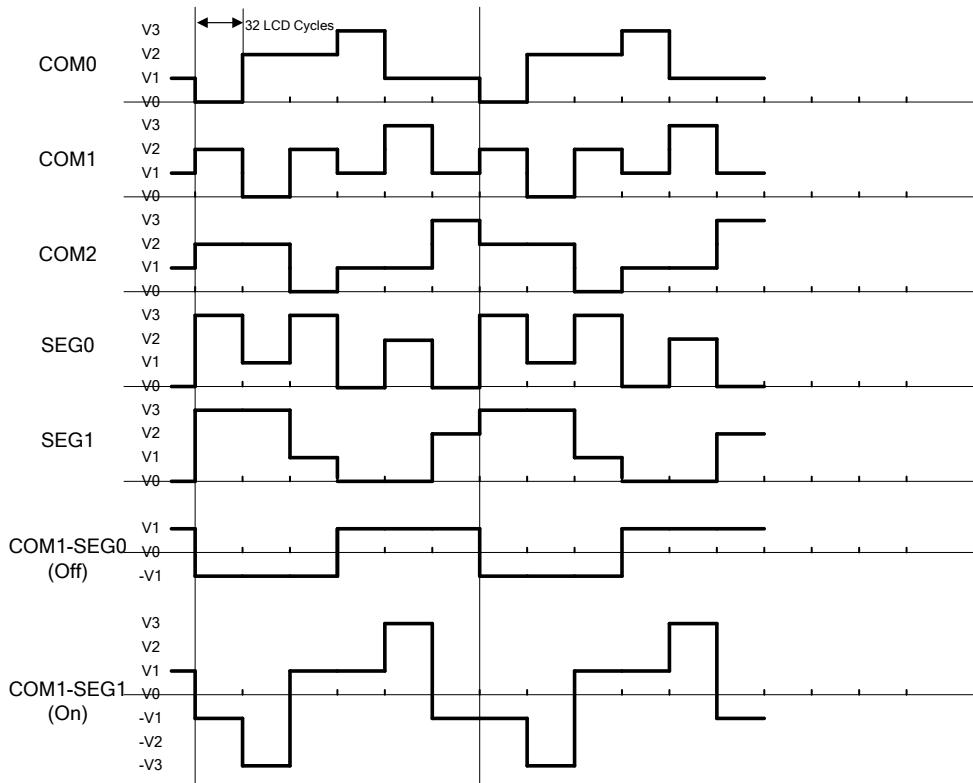


Figure 13-4 3 Com Frame Inverting

#### 4 Com Digit-LED Common-Cathode Driving Timing:

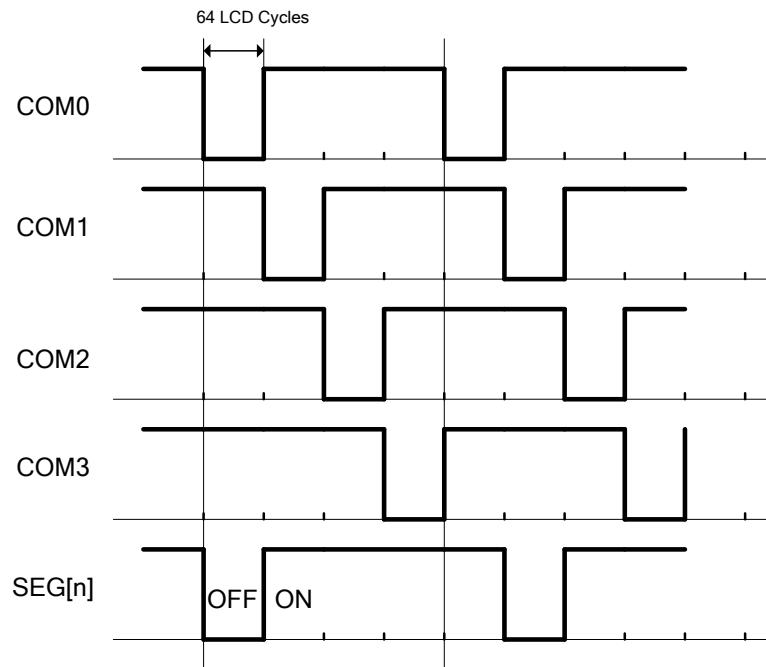


Figure 13-5 4 Com Digit-LED Common-Cathode Driving Timing

|        | COM0    | COM1   | COM2   | COM3   |
|--------|---------|--------|--------|--------|
| SEG[n] | 0 (OFF) | 1 (ON) | 1 (ON) | 1 (ON) |

#### 4 Com Digit-LED Common-Anode Driving Timing:

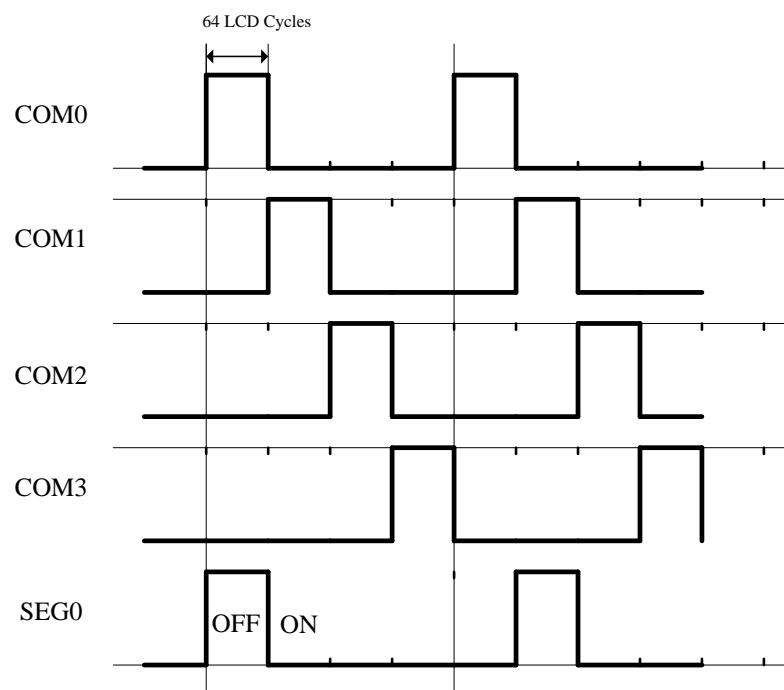


Figure 13-6 4 Com Digit-LED Common-Anode Driving Timing

|  | COM0 | COM1 | COM2 | COM3 |
|--|------|------|------|------|
|--|------|------|------|------|

|        |         |       |        |        |
|--------|---------|-------|--------|--------|
| SEG[n] | 0 (OFF) | 1(ON) | 1 (ON) | 1 (ON) |
|--------|---------|-------|--------|--------|

## 8 Com Digit-LED Common-Cathode Driving Timing:

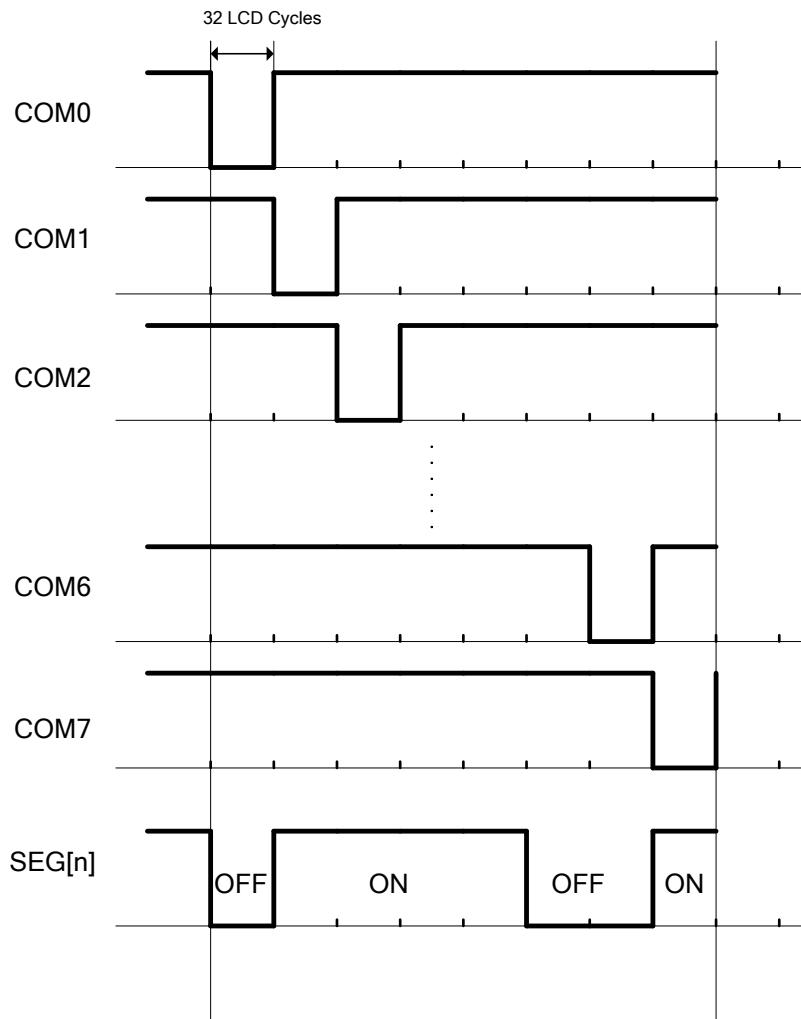


Figure 13-7 8 Com Digit-LED Common-Cathode Driving Timing

|        | COM0       | COM1   | COM2   | COM3   | COM4   | COM5       | COM6       | COM7   |
|--------|------------|--------|--------|--------|--------|------------|------------|--------|
| SEG[n] | 0<br>(OFF) | 1 (ON) | 1 (ON) | 1 (ON) | 1 (ON) | 0<br>(OFF) | 0<br>(OFF) | 1 (ON) |

### 13.1.2.2 SEG/COM LCD and Digit-LED Blinking

The LCD controller supports blinking. The LCD\_MODE[7] is AND with each segment's memory bit. When LCD\_MODE[7] = 1, each segment is on or off according to LCD\_DATA value. When LCD\_MODE[7] = 0, each LCD segment is off. Through writing this bit to "0" and "1" alternately, the LCD or LED is blinking.

### 13.1.3 Module Description

#### 13.1.3.1 Block Diagram

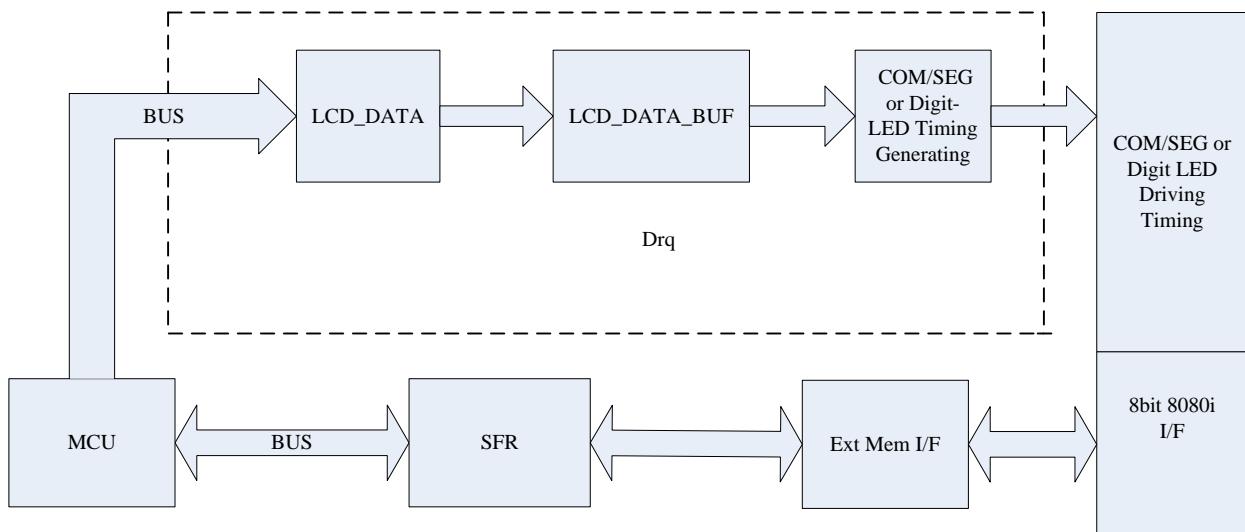


Figure 13-8 LCD Controller Block Diagram

#### 13.1.3.2 Clock Description

**SEGLCDCLKSEL[0] SEGLCDCLKEN[0]**

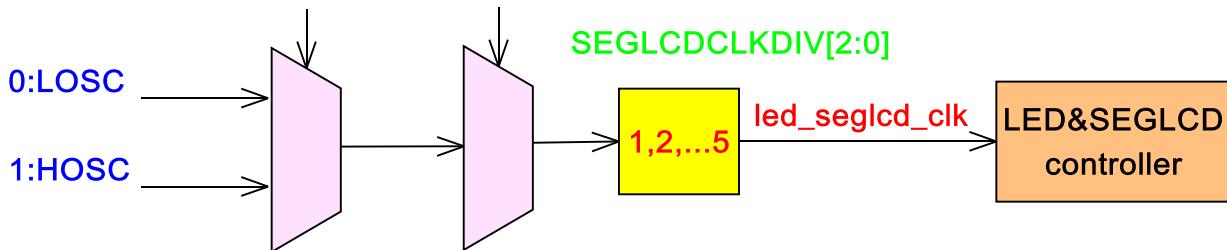


Figure 13-8 clock description

时钟设置总结：

Led 模块的 com 端的频率计算：

$$F_{REN\_OF\_LED\_COMX} = source\_clk / led\_clk\_div / 32 / 8$$

说明：

Source : 24MHz/32.768KHz

Clk\_div 为寄存器分频比

32 led 模块模块内嵌固定分频比

8       一个 com 周期有 8 个节拍。(如果是 4com 的 led, 则每个节拍的时间加倍, 所以时间周期和 8com 的周期一致)

Seg\_lcd 的 com 端的时钟频率计算:

$$F_{\text{ren\_of\_led\_comx}} = \text{source\_clk} / \text{led\_clk\_div} / 32 / \text{com\_number}$$

除了 com\_number 分别为 3, 4 之外, 和上面的叙述一致。

实际中频率设置推荐设置:

LED:

- 1, 如果 led 采用低频, 则不要设置 1/2 或者更低的分频比, 会导致显示闪烁。
- 2, 如果采用高频, 设置 1/4 可以得到 23K 以上的频率(按照 com 端计算)。推荐使用, 可以减少对音频段的干扰。

Seg\_lcd:

- 1,某些型号的 lcd 无法支持高频显示, 只能用低频, 使用上请注意调整。

### 13.1.3.3 Analog setting

Seg\_lcd 1/3 ,2/3 电压使能:

如果要点亮 seg\_lcd 则需要在 PMU 的寄存器中, 使能 MULT\_USED 寄存器 (bank 0x5, 0x8d) 的 bit3, 使能模拟电压, 才能使得 seg\_lcd 的电压正常。

Led 的恒流源的使能:

- 1,使能 PMU 模块寄存器 test\_ctl (bank 0x05, 0x9a) 寄存器的 bit4
- 2,使能 GPIO 模块寄存器 LED\_SEG\_LED\_RC\_EN (bank 0x06, 0xeb)
- 3,设置 GPIO 模块寄存器 LED\_SEG\_BIAS\_EN (bank 0x06, 0xec)。保证极性和 led 模块设定一致。并可以设置恒流源的电流大小。

### 13.1.3.4 PAD Driving Capacity

For driving the Digit-LED, the segments need maximum 15mA static driving current. The common port is to drive the external bipolar in order to increase the static driving current to sink the sum current of the eight segment.

| PAD              | COM0~COM7 | SEG0~SEG7 |
|------------------|-----------|-----------|
| Driving Capacity | 15mA      | 15mA      |

### 13.1.3.5 The LCD Debug Signals

## 13.1.4 Operation Manual

### 13.1.4.1 GL5115 新增内容

- 1, GL5115 的 LCD&LED 模块新增加了模块的 OUTPUT\_EN。在 MFP 中，只要是默认分配给 LCD 的 PADs 为了防止其再 brom 或者进系统之前的阶段输出高低电平，在 LCD 模块都做了门控，只有在 LCD OUTPUT 使能的情形下，LCD 模块才能向外输出高低电平。
- 2, LED 的横流源设计。为保证 LED 点亮的时候，相邻的 LED 之间不会因为点亮的字段数目不一致而导致亮度不一致的问题，GL5115 的 LED 模块在每个 LED 的 SEG 端都做了横流源的设计，以恒定的电流来驱动没一个 seg 端，使得每个 COM 的亮度和其点亮的 SEG 数目无关。请查看 GPIO 模块的寄存器 LED\_SEG\_RC\_EN 和 LED\_SEG\_BIAS\_EN，其定义了每个 LED 的 SEG 端的恒流功能的使能，电流方向，和电流的大小，相应的 LED COM 端的驱动能力也大大的增强了。为了兼容以前的方式，我们做了如下的优先级别设置：如果不开启 LED 的 SEG 的恒流源，则 LED 的 COM 和 SEG 是数字功能，能正常的输出高低电平，驱动 LED，只是会出现亮度不均匀的问题；一旦使能了相关 SEG 的恒流源，则此 PAD 切换到模拟功能，比数字部分优先级别高，此 PAD 自动屏蔽数字部分的输入和输出，只有恒流设置起作用。

### 13.1.4.2 Operation Flow for Software

### SEG/COM LCD and Digit-LED Software Flow:

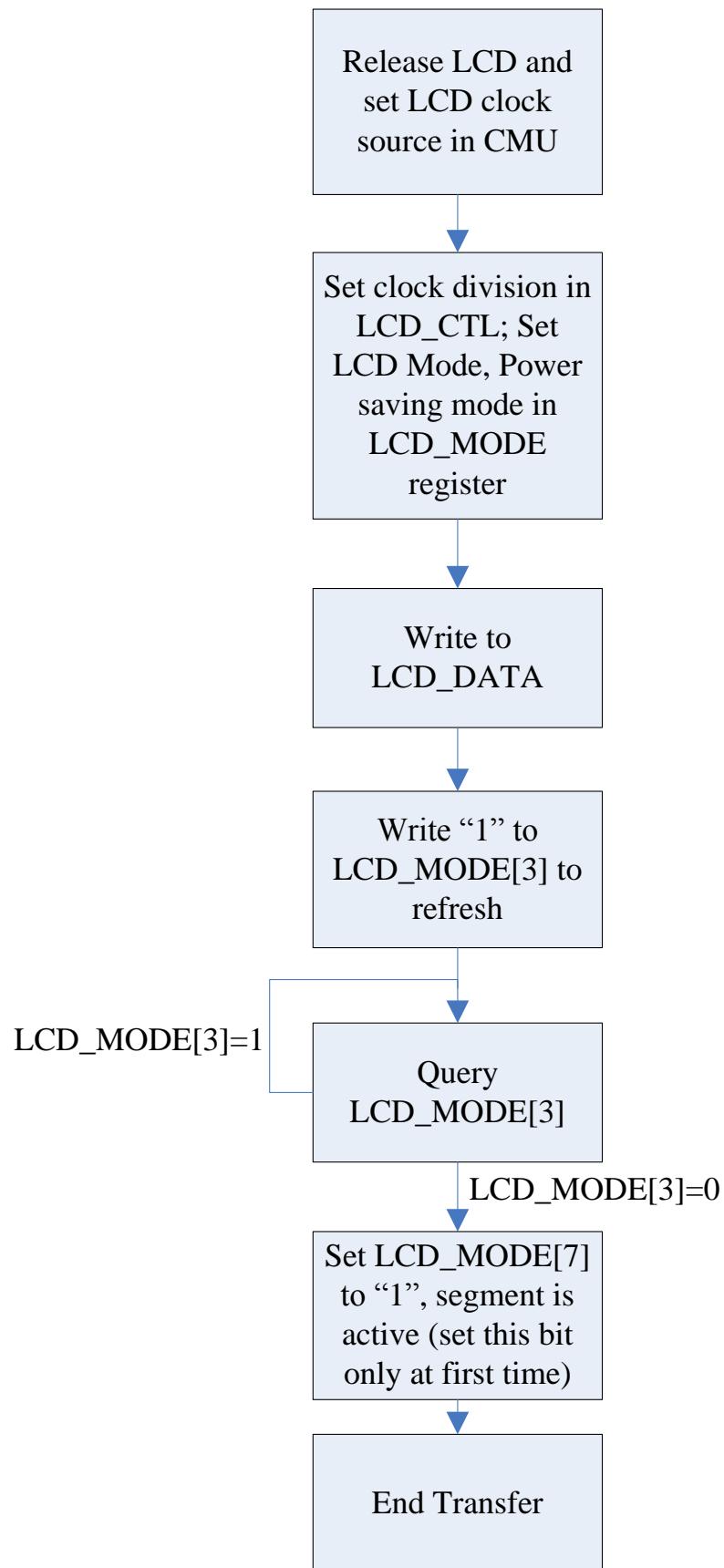


Figure 13-9 Seg/Com LCD&amp;digit-LED software flow

### 13.1.5 LCD Register List

| Index | Mnemonic   | Description                                   | BANK |
|-------|------------|---|------|
| 0x8a  | LCD_MODE   | LCD Mode Control Register                     | 0x06 |
| 0x8b  | LCD_DATA0  | COM[3:0] of SEG0 and SEG1; SEG[7:0] of COM0   | 0x06 |
| 0x8c  | LCD_DATA1  | COM[3:0] of SEG2 and SEG3; SEG[7:0] of COM1   | 0x06 |
| 0x8d  | LCD_DATA2  | COM[3:0] of SEG4 and SEG5; SEG[7:0] of COM2   | 0x06 |
| 0x90  | LCD_DATA3  | COM[3:0] of SEG6 and SEG7; SEG[7:0] of COM3   | 0x06 |
| 0x91  | LCD_DATA4  | COM[3:0] of SEG8 and SEG9; SEG[7:0] of COM4   | 0x06 |
| 0x92  | LCD_DATA5  | COM[3:0] of SEG10 and SEG11; SEG[7:0] of COM5 | 0x06 |
| 0x93  | LCD_DATA6  | COM[3:0] of SEG12 and SEG13; SEG[7:0] of COM6 | 0x06 |
| 0x94  | LCD_DATA7  | COM[3:0] of SEG14 and SEG15; SEG[7:0] of COM7 | 0x06 |
| 0x95  | LCD_DATA8  | COM[3:0] of SEG16 and SEG17                   | 0x06 |
| 0x97  | LCD_DATA9  | COM[3:0] of SEG18 and SEG19                   | 0x06 |
| 0x98  | LCD_DATA10 | COM[3:0] of SEG20 and SEG21                   | 0x06 |
| 0x99  | LCD_DATA11 | COM[3:0] of SEG22 and SEG23                   | 0x06 |
| 0x9a  | LCD_DATA12 | COM[3:0] of SEG24 and SEG25                   | 0x06 |
| 0x9b  | LCD_DATA13 | COM[3:0] of SEG26 and SEG27                   | 0x06 |
| 0x9c  | LCD_DATA14 | COM[3:0] of SEG28 and SEG29                   | 0x06 |
| 0x9d  | LCD_DATA15 | COM[3:0] of SEG30 and SEG31                   | 0x06 |

### 13.1.6 Register Description

#### 13.1.6.1 LCD\_MODE

LCD\_MODE (0x06: 0x8a)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7          | SEGOFF       | Segment Off<br>0:Segment is always off<br>1:Segment value is according to LCD_DATA<br>P.S. Only active in COM/SEG or Digit-LED Mode                       | R/W    | 0     |
| 6:5        | -            | Reserved for future use   | R      | 0     |
| 4          | LCD_OUT_EN   | LCD&LED pad output Enable select:<br>0: the pads of seg_LCD and LED will output “high_Z”.<br>1: the pads of seg_LCD and LED output signal as it's timing. | R/W    | 0     |
| 3          | REFRSH       | Refresh LCD/LED Data<br>0:Hold LCD_DATA<br>Refresh LCD/LED panel according to the LCD_DATA buffer value<br>1:Update LCD_DATA                              | R/W    | 0     |

|     |          |   |     |     |
|-----|----------|---|-----|-----|
|     |          | Refresh the LCD_DATA buffer value from LCD_DATA register<br><br>P.S. Only active in COM/SEG or Digit-LED Mode; When updating the value of LCD_DATA register, write “1” to this bit, the hardware will clear this bit when the LCD_DATA has been updated.  |     |     |
| 2:0 | MODE_SEL | Mode Select<br>000: 3Com,1/3 Bias SEG/COM LCD Frame-Invert<br>001: 4Com,1/3 Bias SEG/COM LCD Frame-Invert<br>010: 3Com,1/3 Bias SEG/COM LCD Row-Invert<br>011: 4Com,1/3 Bias SEG/COM LCD Row-Invert<br>100: 4Com Digit-LED Common-Cathode Mode<br>101: 4Com Digit-LED Common-Anode Mode<br>110: 8Com Digit-LED Common-Cathode Mode<br>111: 8Com Digit-LED Common-Anode Mode | R/W | 000 |

### 13.1.6.2 LCD\_DATA0

LCD\_DATA0 (0x06: 0x8b)

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | SEG1_COM0    | SEG/COM Mode:<br>COM [3:0] Of SEG1.<br><br>Digit-LED Mode:<br>SEG[7:4] of COM0<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF. | R/W    | 0x0   |
| 3:0        | SEG0_COM0    | SEG/COM Mode:<br>COM[3:0] of SEG0<br><br>Digit-LED Mode:<br>SEG[3:0] of COM0<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF.   | R/W    | 0x0   |

### 13.1.6.3 LCD\_DATA1

LCD\_DATA1 (0x06: 0x8c)

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | SEG3_COM1    | SEG/COM Mode:<br>COM [3:0] Of SEG3.<br><br>Digit-LED Mode:<br>SEG[7:4] of COM1<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF. | R/W    | 0x0   |
| 3:0        | SEG2_COM1    | SEG/COM Mode:<br>COM[3:0] of SEG2<br><br>Digit-LED Mode:<br>SEG[3:0] of COM1<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF.   | R/W    | 0x0   |

### 13.1.6.4 LCD\_DATA2

LCD\_DATA2 (0x06: 0x8d)

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | SEG5_COM2    | SEG/COM Mode:<br>COM [3:0] Of SEG5.<br><br>Digit-LED Mode:<br>SEG[7:4] of COM2<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF. | R/W    | 0x0   |
| 3:0        | SEG4_COM2    | SEG/COM Mode:<br>COM[3:0] of SEG4<br><br>Digit-LED Mode:<br>SEG[3:0] of COM2<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF.   | R/W    | 0x0   |

### 13.1.6.5 LCD\_DATA3

LCD\_DATA3 (0x06: 0x90)

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | SEG7_COM3    | SEG/COM Mode:<br>COM [3:0] Of SEG7.<br><br>Digit-LED Mode:<br>SEG[7:4] of COM3<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF. | R/W    | 0x0   |
| 3:0        | SEG6_COM3    | SEG/COM Mode:<br>COM[3:0] of SEG6<br><br>Digit-LED Mode:<br>SEG[3:0] of COM3<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF.   | R/W    | 0x0   |

### 13.1.6.6 LCD\_DATA4

LCD\_DATA4 (0x06: 0x91)

| Bit Number | Bit Mnemonic | Function   | Access | Reset |
|------------|--------------|--|--------|-------|
| 7:4        | SEG9_COM4    | SEG/COM Mode:<br>COM [3:0] Of SEG9.<br><br>Digit-LED Mode:<br>SEG[7:4] of COM4<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF. | R/W    | 0x0   |
| 3:0        | SEG8_COM4    | SEG/COM Mode:<br>COM[3:0] of SEG8<br><br>Digit-LED Mode:<br>SEG[3:0] of COM4   | R/W    | 0x0   |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF. |  |  |
|--|--|--|--|--|

### 13.1.6.7 LCD\_DATA5

LCD\_DATA5 (0x06: 0x92)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | SEG11_COM5   | SEG/COM Mode:<br>COM [3:0] Of SEG11.<br><br>Digit-LED Mode:<br>SEG[7:4] of COM5<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF. | R/W    | 0x0   |
| 3:0        | SEG10_COM5   | SEG/COM Mode:<br>COM[3:0] of SEG10<br><br>Digit-LED Mode:<br>SEG[3:0] of COM5<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF.   | R/W    | 0x0   |

### 13.1.6.8 LCD\_DATA6

LCD\_DATA6 (0x06: 0x93)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | SEG13_COM6   | SEG/COM Mode:<br>COM [3:0] Of SEG13.<br><br>Digit-LED Mode:<br>SEG[7:4] of COM6<br><br>When set to “1”, the cross of COM and SEG is ON;<br>Else is OFF. | R/W    | 0x0   |
| 3:0        | SEG12_COM6   | SEG/COM Mode:<br>COM[3:0] of SEG12  | R/W    | 0x0   |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  | <p>Digit-LED Mode:<br/>SEG[3:0] of COM6</p> <p>When set to “1”, the cross of COM and SEG is ON;<br/>Else is OFF.</p> |  |  |
|--|--|--|--|--|

### 13.1.6.9 LCD\_DATA7

LCD\_DATA7 (0x06: 0x94)

| Bit Number | Bit Mnemonic | Function  | Access | Reset |
|------------|--------------|---|--------|-------|
| 7:4        | SEG15_COM7   | <p>SEG/COM Mode:<br/>COM [3:0] Of SEG15.</p> <p>Digit-LED Mode:<br/>SEG[7:4] of COM7</p> <p>When set to “1”, the cross of COM and SEG is ON;<br/>Else is OFF.</p> | R/W    | 0x0   |
| 3:0        | SEG14_COM7   | <p>SEG/COM Mode:<br/>COM[3:0] of SEG14</p> <p>Digit-LED Mode:<br/>SEG[3:0] of COM7</p> <p>When set to “1”, the cross of COM and SEG is ON;<br/>Else is OFF.</p>   | R/W    | 0x0   |

### 13.1.6.10 LCD\_DATA8

LCD\_DATA8 (0x06: 0x95)

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:4        | SEG17        | COM[3:0] Of SEG17 | R/W    | 0x0   |
| 3:0        | SEG16        | COM[3:0] Of SEG16 | R/W    | 0x0   |

### 13.1.6.11 LCD\_DATA9

LCD\_DATA9 (0x06: 0x97)

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:4        | SEG19        | COM[3:0] Of SEG19 | R/W    | 0x0   |
| 3:0        | SEG18        | COM[3:0] Of SEG18 | R/W    | 0x0   |

### 13.1.6.12 LCD\_DATA10

LCD\_DATA10 (0x06: 0x98)

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:4        | SEG21        | COM[3:0] Of SEG21 | R/W    | 0x0   |
| 3:0        | SEG20        | COM[3:0] Of SEG20 | R/W    | 0x0   |

### 13.1.6.13 LCD\_DATA11

LCD\_DATA11 (0x06: 0x99)

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:4        | SEG23        | COM[3:0] Of SEG23 | R/W    | 0x0   |
| 3:0        | SEG22        | COM[3:0] Of SEG22 | R/W    | 0x0   |

### 13.1.6.14 LCD\_DATA12

LCD\_DATA12 (0x06: 0x9a)

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:4        | SEG25        | COM[3:0] Of SEG25 | R/W    | 0x0   |
| 3:0        | SEG24        | COM[3:0] Of SEG24 | R/W    | 0x0   |

### 13.1.6.15 LCD\_DATA13

LCD\_DATA13 (0x06: 0x9b)

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:4        | SEG27        | COM[3:0] Of SEG27 | R/W    | 0x0   |

|     |       |                   |     |     |
|-----|-------|-------------------|-----|-----|
| 3:0 | SEG26 | COM[3:0] Of SEG26 | R/W | 0x0 |
|-----|-------|-------------------|-----|-----|

### 13.1.6.16 LCD\_DATA14

LCD\_DATA14 (0x06: 0x9c)

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:4        | SEG29        | COM[3:0] Of SEG29 | R/W    | 0x0   |
| 3:0        | SEG28        | COM[3:0] Of SEG28 | R/W    | 0x0   |

### 13.1.6.17 LCD\_DATA15

LCD\_DATA14 (0x06: 0x9d)

| Bit Number | Bit Mnemonic | Function          | Access | Reset |
|------------|--------------|-------------------|--------|-------|
| 7:4        | SEG31        | COM[3:0] Of SEG31 | R/W    | 0x0   |
| 3:0        | SEG30        | COM[3:0] Of SEG30 | R/W    | 0x0   |

## 13.1.7 TESTMODE

### 13.1.7.1 BLOCK DIAGRAM

## 14 GPIO and I/O Multiplexer

### 14.1 GPIO (李克伟、蔡瑞仁)

| 日期         | 版本     | 描述   | 修订人 |
|------------|--------|--|-----|
| 2012-07-24 | V1. 00 | initial  | 李克伟 |
| 2012-09-12 | V1. 02 | 1. MFP_CTL2 增加 SPIBT 模式切换;<br>2. 增加 DBGAOE, DBGAIE, DBGBOE, DBGBIE, DBGCOE, DBGCIE, DBGDOE, DBGDIE 寄存器;<br>3. SPDIF0, SPDIF1 增加施密特触发器; | 李克伟 |

|            |        |   |  |
|------------|--------|---|--|
|            |        | <p>4. 在 AD_Select 增加一个 VDD 输出控制 bit;</p> <p>5. reserved 掉 MFP 寄存器中模拟复用的 bit;</p> <p>6. 将 DBGSEL 中的 pmu_debug 信号改为一组;</p> <p>7. 修改 PADDRV0—PADDRV4 寄存器;</p> <p>8. 修改 PAD_SMIT0—PAD_SMIT2 寄存器复位默认值;</p> <p>9. 去掉 SPI BOOT 说明;</p> <p>10. MFP 复用 GPIOA3, GPIOA4 改动, 去掉 EM_CEB1, 增加 PWM, MFP_CTL1 做相应改动;</p> <p>11. MFP 复用 GPIOE 增加 EM_[D8:D15], MFP_CTL 做相应改动;</p> <p>12. 各寄存器地址重新排序。</p> <p>13. PAD 增加一个 FM_VCC;</p> <p>14. 增加 LED SEG RC_EN, LED SEG BIAS_EN 寄存器;</p> <p>15. 增加 GPIOA0, GPIOA1, GPIOA2, GPIOA3, GPIOA4, GPIOA5, GPIOA6, GPIOA7 驱动档位; 增加 SPIBOOT 驱动能力寄存器设置。</p> <p>16. MFP_CTL8 BIT5, 4 改为:</p> <p>00: I2S0_MCLK</p> <p>01: LCD_SEG26</p> <p>10: EM_D10</p> <p>11: Reserved</p> <p>17. PAGE13 “URAT” 改为 “UART” ;</p> <p>18. PAGE13 GPIOE3 复用改为 I2S_D0;</p> <p>19. PADDRV2 bit[5:0] 默认值改为 0x01, PADDRV3 bit[7:2] 默认值改为 0x01;</p> <p>20. 更改 2.13.2 PAD 名称, 增加 CPAGND PIN;</p> <p>21. LED SEG BIAS EN 寄存器 bit2 改为 LED_Cathode/Anode Mode.</p> <p>22. 更改 2.1.32 PAD, 把一个 UVCC 去掉, 同时把 UVCC 改为 VCC</p> <p>23. 增加两个 FM_AGND_LNA</p> <p>24. 把 SPIBT_SCLK 改为 SPIBT_CLK; 把 FM_RF_MATCH 改为 FM_RF_MATCHB.</p> <p>25. LED SEG BIAS EN bit2 寄存器 LED_Cathode/Anode Mode 改为 LED_Cathode_Anode_Mode;</p> <p>26. 所有的 Reserved 位定义为 R/W;</p> <p>27. 去掉 PAD_SMIT0, PAD_SMIT1, PAD_SMIT2 寄存器;</p> <p>28. 去掉 DBGBIE, DBGDIE 寄存器。</p> <p>29. 去掉 DBGCIE 寄存器, 增加 DBGBIN 寄存器。</p> <p>30. 所有 Reserved 位的 name 改为 “Reserved”</p> |  |
| 2012-10-20 | V2. 00 | <p>1. 更改 GPIOAPUEN, GPIOAPDEN, GPIOCPUEN, GPIOCPDEN 描述:GPIOA6, GPIOC6 下拉电阻改为 100K;</p> <p>2. 更改 2.1.3.2 PAD, 去掉 FM, SPIBOOT, IR_Analog,</p>   |  |

|  |  |  |
|--|--|--|
|  | <p>I2C;</p> <p>3. 去掉 2.1.3.3 SPIBOOT 的描述;</p> <p>4. 更改 Analog and Digital PAD 表格, 修改 MFP;</p> <p>5. 更改 Multi-Function Bounding option 表格, 更改 MFP;</p> <p>6. 更改 2.1.3.9 PAD 的施密特触发器的描述;</p> <p>7. MFP_CTL6 寄存器 bit5, bit6 改为 Reserved;</p> <p>8. MFP_CTL8 寄存器 bit3:2 增加 11: UART_TX;<br/>Bit1:0 增加 11: UART_RX;</p> <p>9. AD_Select 寄存器 bit5 IR_RX_Analog 改为 Reserved; 同时 GPIOC4 复用的 AVCC 去掉, 增加 GPIOA5 和 GPIOA6 的 AVCC 复用, bit 位排序改变;</p> <p>10. PADPUPD 寄存器 bit1 改为 Reserved;</p> <p>DBGSEL 寄存器 bit4:0: 00100 I2C Debug Signals 改为 SPDIF Debug Signals; 00111:AUIP Debug Signals 改为 Reserved; 01011: MJPEG Debug Signals 改为 Reserved; 10000:B1B2 Debug Signals 改为 Reserved;</p> <p>10100:SPIBOOT Debug Signals 改为 Reserved;</p> <p>10101: FM Digital Debug Signals 改为 Reserved;</p> <p>10110: FM Analog Debug Signals 改为 Reserved.</p> <p>11. 更改 Multi-Function Bounding option 表格, 更改 MFP; 在 Priority 栏, 去掉 EJ_TRST;</p> <p>12. 更改 2.1.3.2 PAD 中的 MFP, 和 GPIOA5 复用的 AVCC 改为 AVCC0, 和 GPIOC6 复用的 AVCC 改为 AVCC1;</p> <p>13. AD_Select 寄存器 bit2 名称改为 AVCC0_OUT, bit1 名称改为 AVCC1_OUT;</p> <p>14. P_RESET 改为 P_RESETB;</p> <p>15. 增加各模块 DEBUG GPIO 列表;</p> <p>16. MFP_CTL2 bit1 改为 Reserved;</p> <p>17. PADDRV4 bit3, bit2 改为 Reserved.</p> <p>18. 更改 2.1.3.2 PAD, REM_CON 增加 GPIOG2 复用; GPIOG2, GPIOC4, GPIOC3, GPIOA6, GPIOA5, GPIOC7 分别增加 TK0, TK1, TK2, TK3, TK4, TK5 复用;</p> <p>19. 更改 2.1.3.3 表格, REM_CON 增加 GPIOG2 复用; GPIOG2, GPIOC4, GPIOC3, GPIOA6, GPIOA5, GPIOC7 分别增加 TK0, TK1, TK2, TK3, TK4, TK5 复用;</p> <p>20. 更改 2.1.3.4 表格, REM_CON 增加 GPIOG2 复用; GPIOG2, GPIOC4, GPIOC3, GPIOA6, GPIOA5, GPIOC7 分别增加 TK0, TK1, TK2, TK3, TK4, TK5 复用;</p> <p>21. GPIOGOUTEN, GPIOGINEN, GPIOGDAT, GPIOGPUEN, GPIOGPDEN 寄存器增加 bit2, GPIOG2;</p> <p>22. AD_Select 寄存器名称改为 AD_Select0;</p> <p>23. AD_Select0 寄存器增加 bit6, 用于 REM_CON;</p> <p>24. 增加 AD_Select1 寄存器, 用于选择 Touch Key;</p> |  |
|--|--|--|

|            |        |   |     |
|------------|--------|---|-----|
|            |        | 25. PADDRV4 寄存器增加 bit3, bit2, 用于调节 GPIOG2 驱动档位。<br>26. 更改 2.1.3.2 PAD, GPIOC6 增加 TK6 复用;<br>27. 更改 2.1.3.3 表格, GPIOC6 增加 TK6 复用;<br>28. 更改 2.1.3.4 表格, GPIOC6 增加 TK6 复用;<br>29. AD_Select1 寄存器, 增加 bit6, 用于选择 TK6。<br>30. LED_SEG_BIAS_EN 寄存器增加说明;<br>31. AD_Select0, AD_Select1 寄存器重新合并修改。<br>32. DBGSEL 寄存器 bit[4:0] 01010: Reserved 改为: 01010:TK analog debug;<br>33. 各模块使用的 DEBUG 信号, 增加 TK 信号。<br>34. 所有寄存器 SFR address 和 Bank 地址互换。 |     |
| 2012-11-07 | V2. 01 | 1. CMU debug 信号改为 P_GPIOD0<br>2. DBGSEL 寄存器 bit[4:0] 10000: Reserved 改为: EFUSE Debug Signals;<br>3. 各模块使用的 DEBUG 信号增加 EFUSE GPIOA[5:0]。   | 李克伟 |
| 2012-12-12 | V2. 02 | 1. 触摸按键 TK6 名称改为 SHIELD;<br>2. 增加 Operation Manual 的补充说明。   | 李克伟 |
| 2013-01-18 | V2. 03 | 1、FMINL 改成和 GPIOF2 复用, FMINR 改成和 GPIOF3 复用, BTINL 改成和 GPIOF4 复用, BTINR 改成和 GPIOF5 复用。   | 李克伟 |

### 14.1.1 Features

- 支持 50 个 GPIO
- 某些 PAD 内置上下拉电阻
- 驱动能力可调
- 动态切换 PAD 的功能

### 14.1.2 Function Description

GPIO 是通用输入输出端口, 可以输出 0 或 1, 也可以检测外部电路输入的电平信号, 通过内部寄存器读出是 1 或 0, 每一个 GPIO 都有对应的 ENABLE 位控制, 有相应的 DATA 寄存器. 不管用作输入还是输出, GPIO 的优先级高于 Normal Digital 功能, 低于 EJTAG 功能。

由于PAD资源有限, 必须采用复用的方式来解决这个矛盾, MFP模块就是为此而设计的。

GPIO 作为 PAD 的一个功能, 也和其他功能复用, 但具有最高优先级。只要 ENABLE 相应的 GPIO, 该 PAD 就切换到 GPIO 功能。

### 14.1.3 Module Description

#### 14.1.3.1 Block Diagram

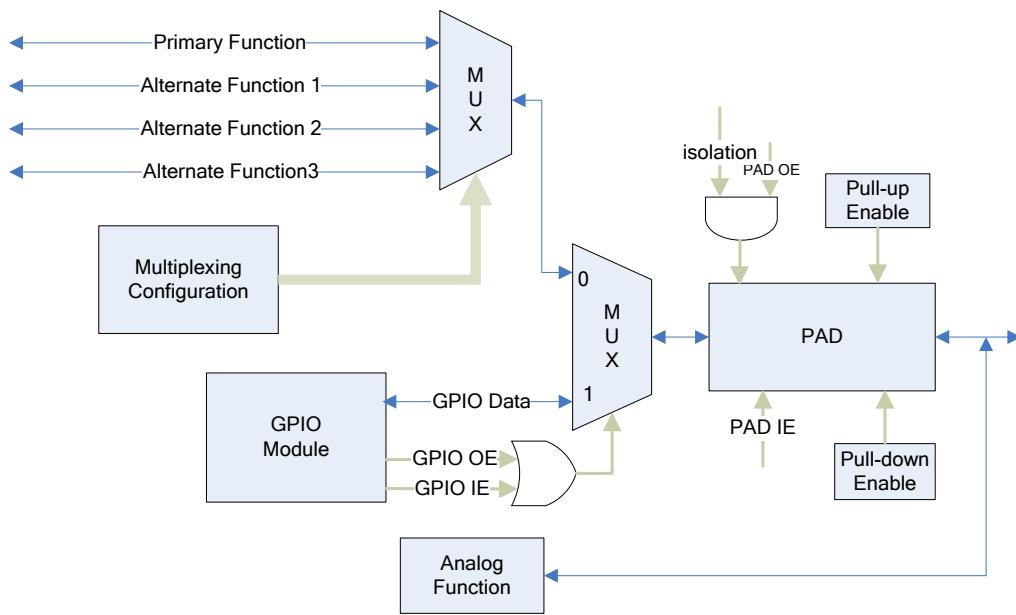


Figure 14-1 Block Diagram of GPIO Controller

#### 14.1.3.2 PAD

| MODULE    | PIN NAME | MFP |
|-----------|----------|-----|
| MODE      | P_TEST   |     |
|           | P_RESETB |     |
| USB       | HSDM     |     |
|           | HSDP     |     |
|           | UGND     |     |
|           | FSGND    |     |
|           | URES     |     |
|           | VCC      |     |
| HEADPHONE | AOUTL    |     |
|           | AOUTR    |     |
|           | PAGND    |     |
|           | VCC      |     |

|         |           |          |
|---------|-----------|----------|
|         | VRO       | P_GPIOG0 |
|         | VRO_SENSE | P_GPIOG1 |
| PMU     | AVCC      |          |
|         | DVCC      |          |
|         | AGND      |          |
|         | AGND1     |          |
|         | DGND      |          |
|         | VREFI     |          |
|         | VCC       |          |
|         | FSOURCE   |          |
|         | DC5V      |          |
|         | BAT       |          |
|         | ON_OFF    |          |
|         | RTCVDD    |          |
|         | GND       |          |
|         | VDD       |          |
|         | AVDD      |          |
|         | VCC       |          |
|         | VCC       |          |
|         | GND       |          |
|         | GND       |          |
|         | GND       |          |
| CP      | VCP       |          |
|         | CN        |          |
|         | CPBAT     |          |
|         | CP        |          |
|         | NGND      |          |
|         | CPAGND    |          |
| RMU/CMU | LOSCI     |          |

|          |          |  |
|----------|----------|--|
|          | LOSCO    |  |
|          | HOSCO    |  |
|          | HOSCI    |  |
| GPIO/MFP | P_GPIOA0 | EM_WRB/LED_COM0/LCD_COM0                           |
|          | P_GPIOA1 | EM_RS/LED_COM1/LCD_COM1                            |
|          | P_GPIOA2 | EM_RDB/LED_COM2/LCD_COM2                           |
|          | P_GPIOA3 | EM_CEB0/LED_COM3/LCD_SEG0                          |
|          | P_GPIOA4 | LED_COM4/LCD_COM3/PWM                              |
|          | P_GPIOA5 | LED_COM5/LCD_SEG1/UART_CTS/LRADC3/EJ_TMS/AVCC0/TK4 |
|          | P_GPIOA6 | LED_COM6/LCD_SEG2/UART RTS/LRADC4/SIRQ1/EJ_TCK/TK3 |
|          | P_GPIOA7 | LED_COM7/LCD_SEG3/FMCLKOUT                         |
|          | P_GPIOB0 | EM_D0/LED_SEG0/LCD_SEG4                            |
|          | P_GPIOB1 | EM_D1/LED_SEG1/LCD_SEG5                            |
|          | P_GPIOB2 | EM_D2/LED_SEG2/LCD_SEG6                            |
|          | P_GPIOB3 | EM_D3/LED_SEG3/LCD_SEG7                            |
|          | P_GPIOB4 | EM_D4/LED_SEG4/LCD_SEG8                            |
|          | P_GPIOB5 | EM_D5/LED_SEG5/LCD_SEG9                            |
|          | P_GPIOB6 | EM_D6/LED_SEG6/LCD_SEG10                           |
|          | P_GPIOB7 | EM_D7/LED_SEG7/LCD_SEG11                           |
|          | P_GPIOC0 | MMC_CMD  |
|          | P_GPIOC1 | MMC_CLK0   |
|          | P_GPIOC2 | MMC_DAT0   |
|          | P_GPIOC3 | MMC_DAT1/UART_RX/MMC_CLK1/LCD_SEG12/EJ_TDI/VDD/TK2 |
|          | P_GPIOC4 | MMC_DAT2/UART_TX/LCD_SEG13/EJ_TRST/TK1             |
|          | P_GPIOC5 | MMC_DAT3/IR_RX/EJ_TDO                              |
|          | P_GPIOC6 | SIRQ0/SPDIF0/LCD_SEG14/AVCC1/SHIELD                |
|          | P_GPIOC7 | LCD_SEG15/LRADC5/TK5                               |
|          | P_GPIOD0 | SPI_SS/LCD_SEG16                                   |
|          | P_GPIOD1 | SPI_MISO/LCD_SEG17                                 |
|          | P_GPIOD2 | SPI_MOSI/LCD_SEG18                                 |
|          | P_GPIOD3 | SPI_SCLK/LCD_SEG19                                 |
|          | P_GPIOD4 | EJ_TMS/LCD_SEG20                                   |
|          | P_GPIOD5 | EJ_TCK/SPDIF1/LCD_SEG21                            |
|          | P_GPIOD6 | EJ_TDI/LCD_SEG22                                   |
|          | P_GPIOD7 | EJ_TDO/LCD_SEG23                                   |
|          | P_GPIOE0 | I2S0_BCLK/LCD_SEG24/EM_D8/UART_RX                  |
|          | P_GPIOE1 | I2S0_LRCLK/LCD_SEG25/EM_D9/UART_TX                 |
|          | P_GPIOE2 | I2S0_MCLK/LCD_SEG26/EM_D10                         |
|          | P_GPIOE3 | I2S0_DO/LCD_SEG27/EM_D11                           |
|          | P_GPIOE4 | I2S1_BCLK/LCD_SEG28/EM_D12/EJ_TRST                 |
|          | P_GPIOE5 | I2S1_LRCLK/LCD_SEG29/EM_D13                        |
|          | P_GPIOE6 | I2S1_MCLK/LCD_SEG30/EM_D14                         |

|  |          |                          |
|--|----------|--------------------------|
|  | P_GPIOE7 | I2S1_DI/LCD_SEG31/EM_D15 |
|  | P_GPIOG2 | REM_COM/TK0              |
|  | MICINL   | GPIO_F0                  |
|  | MICINR   | GPIO_F1                  |
|  | BTINL    | GPIO_F4                  |
|  | BTINR    | GPIO_F5                  |
|  | FMINL    | GPIO_F2                  |
|  | FMINR    | GPIO_F3                  |
|  | LINEINL  | GPIO_F6                  |
|  | LINEINR  | GPIO_F7                  |

#### 14.1.3.3 Analog and Digital PAD

GL5115 有些 PAD 是 Analog 和 Digital 功能复用的，这些 PAD 如下：

| Analog           | Digital Function |  |
|------------------|------------------|--|
| Priority0        | Priority3        | Priority4                                  |
| VDD/TK2          | GPIOC3           | MMC_DAT1/UART_RX/MMC_CLK1/LCD_SEG12/EJ_TDI |
| TK1              | GPIOC4           | MMC_DAT2/UART_TX/LCD_SEG13/EJ_TRST         |
| AVCC0/LRADC3/TK4 | GPIOA5           | LED_COM5/LCD_SEG1/UART_CTS/EJ_TMS          |
| AVCC1/SHIELD     | GPIOC6           | SIRQ0/SPDIF0/LCD_SEG14                     |
| LRADC4/TK3       | GPIOA6           | LED_COM6/LCD_SEG2/UART_RTS/SIRQ1/EJ_TCK    |
| LRADC5/TK5       | GPIOC7           | LCD_SEG15                                  |
| MICINL           | GPIOF0           |  |
| MICINR           | GPIOF1           |  |
| FNINL            | GPIOF2           |  |
| FMINR            | GPIOF3           |  |
| BTINL            | GPIOF4           |  |
| BTINR            | GPIOF5           |  |
| LINEINL          | GPIOF6           |  |
| LINEINR          | GPIOF7           |  |
| VRO              | GPIOG0           |  |
| VRO_S            | GPIOG1           |  |
| REM_CON/TK0      | GPIOG2           |  |

对

于数模混用 GPIO，软件需要注意不能配置错，如果打开了模拟功能，GPIO 模块的寄存器（MFP\_CTL）又配置成数字 PAD 的话，功耗会非常大。

#### 14.1.3.4 Multi-Function Bonding Option

Priority0>Priority1>Priority2>Priority3>Priority4

| Analog               |           | Digital Function |           |                                     |
|----------------------|-----------|------------------|-----------|-------------------------------------|
| Priority0            | Priority1 | Priority2        | Priority3 | Priority4                           |
|                      |           | DEBUGA0          | GPIO_A0   | EM_WRB/LED_COM0/LCD_COM0            |
|                      |           | DEBUGA1          | GPIO_A1   | EM_RS/LED_COM1/LCD_COM1             |
|                      |           | DEBUGA2          | GPIO_A2   | EM_RDB/LED_COM2/LCD_COM2            |
|                      |           | DEBUGA3          | GPIO_A3   | EM_CEB0/LED_COM3/LCD_SEG0           |
|                      |           | DEBUGA4          | GPIO_A4   | LED_COM4/LCD_COM3/PWM               |
| LRADC3/<br>AVCC0/TK4 | EJ_TMS    | DEBUGA5          | GPIO_A5   | LED_COM5/LCD_SEG1/UART_CTS          |
| LRADC4/TK3           | EJ_TCK    | DEBUGA6          | GPIO_A6   | LED_COM6/LCD_SEG2/UART RTS/SIRQ1    |
|                      |           | DEBUGA7          | GPIO_A7   | LED_COM7/LCD_SEG3/FMCLKOUT          |
|                      |           | DEBUGB0          | GPIO_B0   | EM_D0/LED_SEG0/LCD_SEG4             |
|                      |           | DEBUGB1          | GPIO_B1   | EM_D1/LED_SEG1/LCD_SEG5             |
|                      |           | DEBUGB2          | GPIO_B2   | EM_D2/LED_SEG2//LCD_SEG6            |
|                      |           | DEBUGB3          | GPIO_B3   | EM_D3/LED_SEG3/LCD_SEG7             |
|                      |           | DEBUGB4          | GPIO_B4   | EM_D4/LED_SEG4/LCD_SEG8             |
|                      |           | DEBUGB5          | GPIO_B5   | EM_D5/LED_SEG5/LCD_SEG9             |
|                      |           | DEBUGB6          | GPIO_B6   | EM_D6/LED_SEG6/LCD_SEG10            |
|                      |           | DEBUGB7          | GPIO_B7   | EM_D7/LED_SEG7/LCD_SEG11            |
|                      |           | DEBUGC0          | GPIO_C0   | MMC_CMD                             |
|                      |           | DEBUGC1          | GPIO_C1   | MMC_CLK0                            |
|                      |           | DEBUGC2          | GPIO_C2   | MMC_DAT0                            |
| VDD/TK2              | EJ_TDI    | DEBUGC3          | GPIO_C3   | MMC_DAT1/UART_RX/MMC_CLK1/LCD_SEG12 |
| TK1                  | EJ_TRST   | DEBUGC4          | GPIO_C4   | MMC_DAT2/UART_TX/LCD_SEG13          |
|                      | EJ_TDO    | DEBUGC5          | GPIO_C5   | MMC_DAT3/IR_RX                      |
| AVCC1/SHIELD         |           | DEBUGC6          | GPIO_C6   | SIRQ0/SPDIF0/LCD_SEG14              |
| LRADC5/TK5           |           | DEBUGC7          | GPIO_C7   | LCD_SEG15                           |
|                      |           | DEBUGD0          | GPIO_D0   | SPI_SS/LCD_SEG16                    |
|                      |           | DEBUGD1          | GPIO_D1   | SPI_MISO /LCD_SEG17                 |
|                      |           | DEBUGD2          | GPIO_D2   | SPI莫斯/LCD_SEG18                     |
|                      |           | DEBUGD3          | GPIO_D3   | SPI_SCLK/LCD_SEG19                  |
|                      | EJ_TMS    | DEBUGD4          | GPIO_D4   | LCD_SEG20                           |
|                      | EJ_TCK    | DEBUGD5          | GPIO_D5   | SPDIF1/LCD_SEG21                    |
|                      | EJ_TDI    | DEBUGD6          | GPIO_D6   | LCD_SEG22                           |
|                      | EJ_TDO    | DEBUGD7          | GPIO_D7   | LCD_SEG23                           |
|                      |           |                  | GPIO_E0   | I2S0_BCLK /LCD_SEG24/EM_D8/UART_RX  |
|                      |           |                  | GPIO_E1   | I2S0_LRCLK/LCD_SEG25/EM_D9/UART_TX  |
|                      |           |                  | GPIO_E2   | I2S0_MCLK/LCD_SEG26/EM_D10          |
|                      |           |                  | GPIO_E3   | I2S0_DO/LCD_SEG27/EM_D11            |
|                      | EJ_TRST   |                  | GPIO_E4   | I2S1_BCLK/LCD_SEG28/EM_D12          |
|                      |           |                  | GPIO_E5   | I2S1_LRCLK/LCD_SEG29/EM_D13         |
|                      |           |                  | GPIO_E6   | I2S1_MCLK/LCD_SEG30/EM_D14          |
|                      |           |                  | GPIO_E7   | I2S1_DI/LCD_SEG31/EM_D15            |

|             |  |  |         |  |
|-------------|--|--|---------|--|
| MICINL      |  |  | GPIO_F0 |  |
| MICINR      |  |  | GPIO_F1 |  |
| BTINL       |  |  | GPIO_F4 |  |
| BTINR       |  |  | GPIO_F5 |  |
| FMINL       |  |  | GPIO_F2 |  |
| FMINR       |  |  | GPIO_F3 |  |
| LINEINL     |  |  | GPIO_F6 |  |
| LINEINR     |  |  | GPIO_F7 |  |
| VRO         |  |  | GPIO_G0 |  |
| VRO_S       |  |  | GPIO_G1 |  |
| REM_CON/TK0 |  |  | GPIO_G2 |  |

#### 14.1.3.5 EJTAG Multi-Function

共有两种 EJTAG 的 Mapping 关系。这两种 EJTAG 的优先级高于普通 digital 功能，只要 EJTAG Enable 位有效，结合 EJTAG 的计时选择机制，对应的 PAD 就切换给 EJTAG。

| EJ_MAP |   |
|--------|---|
| 0      | EJTAG is mapping to GPIOD[7:4], GPIOE4                |
| 1      | EJTAG is mapping to GPIOA[6:5] , GPIOC3,GPIOC4,GPIOC5 |

上电默认选择第一组 EJTAG 且 EJTAG Enable 位是使能的；

有一个计时 counter 用于控制 EJTAG 的 Mapping 切换；

当系统复位从有效变成无效时，这个 counter 被复位成 0；

counter 从系统复位信号有效开始计时，计到 16 秒时，硬件自动将 EJTAG 选择第二组 Mapping 方式；

除非重新上电，EJTAGMAP 才能恢复成 0，选择第一组 Mapping 方式；

注意：因复位是按键 8s (6s 到 14s 可设) 所以当持续计数时不要将 counter 计数的值也复位这样一旦按键 16s 就固定切换 EJTAG 到第二组。

#### 14.1.3.6 PAD 的上下拉电阻

每一个 GPIO 都有一个可以分别 enable 和 disable 的 50K 上拉和 50K 下拉电阻，由相应的寄存器控制，默认关闭；其他功能模块则根据需要，设置有可以寄存器控制 Enable 的上下拉电阻。

#### 14.1.3.7 P\_TEST、P\_RESETB

以下上下拉电阻一直使能，不能关闭：

P\_TEST 内置 100K 下拉电阻；

P\_RESETB 内置 100K 上拉电阻；

### 14.1.3.8 模块上下拉电阻

1. 当 MFP 把相应的 PAD 切换成 EJTAG 功能且 EJTAG Enable 时, EJ\_TCK、EJ\_TMS、EJ\_TDI、EJ\_TRST 100K 上拉电阻有效;
2. 当 MFP 把相应的 PAD 切换成 UART\_RX 功能时, 相应的 PAD 的 10K 上拉电阻, 可以用寄存器控制打开或关闭;
3. 当 MFP 把相应的 PAD 切换成 SIRQ0 或 SIRQ1 时, 相应的 PAD 有 100K 上拉电阻和 100K 下拉电阻, 可以用寄存器控制分别打开或关闭; 当 SIRQ 设置成上升沿或高电平有效时, 软件应将下拉电阻打开; 当 SIRQ 设置成下降沿或低电平有效时, 软件应将上拉电阻打开;
4. 当 MFP 把相应的 PAD 切换成 SD/MMC 功能时, MMC\_CMD、MMC\_DATA 的 50K 上拉电阻, 可以用寄存器控制打开或关闭; designer 需要保证在 SD2.0 3.3V 下该电阻的范围在 10K 到 50K 之间。

### 14.1.3.9 PAD 的施密特触发器

当 MFP 把单独 PAD 中 P\_TEST, P\_RESETB 及复用 PAD 切换到功能 EJTAG、SIRQ0/1、SPI\_SS, I2S0\_BCLK、I2S0\_LRCLK、I2S0\_MCLK、I2S1\_BCLK、I2S1\_LRCLK、I2S1\_MCLK、I2S1\_DI、IR\_RX 时, 相应的 PAD 的施密特触发器功能会打开。

### 14.1.3.10 PWM

The PWM module divides the source frequency and adjust the duty occupancy according to the active polarity (High level active or Low level active)、frequency dividing and duty setting. Note that 16 level duty adjustment needs a 16 times frequency as a reference. E.g. A 2K PWM output with 16 level duty adjustment needs 32Khz as a reference. There are totally 16 adjusting level for each frequency, which can meet most PWM backlight IC application.

## 14.1.4 Operation Manual

### 14.1.4.1 Operation Manual

1. 一个 PAD 可以复用成模块功能, GPIO 功能和模拟功能, 分别由 MFP\_CTL, GPIOINEN/GPIOOUTEN 和 AD\_Select 控制, 注意其优先级关系; Analog 的优先级最高, 其次是 EJTAG, 然后是 DEBUG, 最后才是各个模块的 MFP 功能。
2. Analog 和 GPIO 及 MFP 复用的 PIN 要特别注意; 当 PIN 使用为 GPIO 或 MFP 时, 一定要注意其和 Analog 复用的 AD\_Select0 和 AD\_Select1 寄存器是否已经指定成 digital function。
3. 有些 MFP 模块是有上下拉电阻的, 详见寄存器 PADPUPD; 当使用做 MFP 功能且把功能上下拉使能

时，一定要把 GPIO 的上下拉 Disable，否则后出现电平错乱，功能不正常。

4. LED\_SEG\_BIAS\_EN 和 LED\_SEG\_BIAS\_EN 寄存器用于数码管的恒流源开启和关闭，电流大小的选择；当对应的 GPIOB[7:0]用做非数码管功能时，一定要把相应的 bit Disable，否则功能不正常。

#### 14.1.4.2 GPIO Output

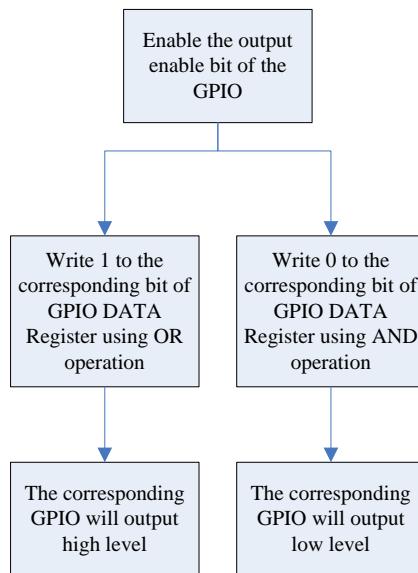


Figure 14-2 GPIO Output flow

#### 14.1.4.3 GPIO Input

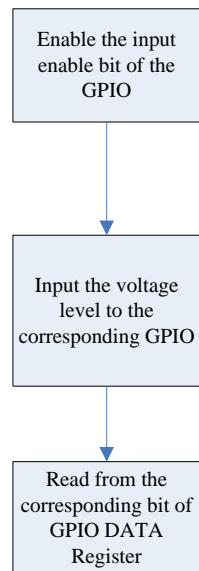


Figure 14-3 GPIO Input flow

#### 14.1.4.4 GPIO Output/Input Loop Test

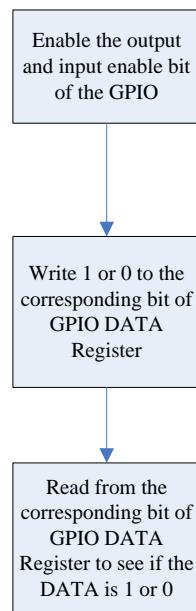


Figure 14-4GPIO Output/Input Loop Test flow

#### 14.1.5 GPIO Register List

| Index | Mnemonic          | Description  | BANK |
|-------|-------------------|--|------|
| 0xa2  | <b>GPIOAOUTEN</b> | General Purpose Input Output Group A Output Enable | 0x06 |
| 0xa3  | <b>GPIOAINEN</b>  | General Purpose Input Output Group A Input Enable  | 0x06 |
| 0xa4  | <b>GPIOADAT</b>   | General Purpose Input Output Group A Data          | 0x06 |
| 0xa5  | <b>GPIOAPUEN</b>  | General Purpose Input Output Group A PU Enable     | 0x06 |
| 0xa6  | <b>GPIOAPDEN</b>  | General Purpose Input Output Group A PD Enable     | 0x06 |
| 0xa7  | <b>GPIOBOUTEN</b> | General Purpose Input Output Group B Output Enable | 0x06 |
| 0xa9  | <b>GPIOBINEN</b>  | General Purpose Input Output Group B Input Enable  | 0x06 |
| 0xaa  | <b>GPIOBDAT</b>   | General Purpose Input Output Group B Data          | 0x06 |
| 0xab  | <b>GPIOBPUEN</b>  | General Purpose Input Output Group B PU Enable     | 0x06 |
| 0xac  | <b>GPIOBPDEN</b>  | General Purpose Input Output Group B PD Enable     | 0x06 |
| 0xad  | <b>GPIOCOUTEN</b> | General Purpose Input Output Group C Output Enable | 0x06 |
| 0xae  | <b>GPIOCINEN</b>  | General Purpose Input Output Group C Input Enable  | 0x06 |
| 0xaf  | <b>GPIOCDAT</b>   | General Purpose Input Output Group C Data          | 0x06 |
| 0xb0  | <b>GPIOCPUEN</b>  | General Purpose Input Output Group C PU Enable     | 0x06 |
| 0xb1  | <b>GPIOCPDEN</b>  | General Purpose Input Output Group C PD Enable     | 0x06 |
| 0xb2  | <b>GPIODOUTEN</b> | General Purpose Input Output Group D Output Enable | 0x06 |
| 0xb3  | <b>GPIODINEN</b>  | General Purpose Input Output Group D Input Enable  | 0x06 |
| 0xb4  | <b>GPIODDAT</b>   | General Purpose Input Output Group D Data          | 0x06 |
| 0xb5  | <b>GPIODPUEN</b>  | General Purpose Input Output Group D PU Enable     | 0x06 |
| 0xb6  | <b>GPIODPDEN</b>  | General Purpose Input Output Group D PD Enable     | 0x06 |

|      |                        |  |      |
|------|------------------------|--|------|
| 0xb7 | <b>GPIOEOUTEN</b>      | General Purpose Input Output Group E Output Enable | 0x06 |
| 0xb8 | <b>GPIOEINEN</b>       | General Purpose Input Output Group E Input Enable  | 0x06 |
| 0xb9 | <b>GPIOEDAT</b>        | General Purpose Input Output Group E Data          | 0x06 |
| 0xba | <b>GPIOEPUE</b>        | General Purpose Input Output Group E PU Enable     | 0x06 |
| 0xbb | <b>GPIOEPDEN</b>       | General Purpose Input Output Group E PD Enable     | 0x06 |
| 0xbc | <b>GPIOFOUTEN</b>      | General Purpose Input Output Group F Output Enable | 0x06 |
| 0xbd | <b>GPIOFINEN</b>       | General Purpose Input Output Group F Input Enable  | 0x06 |
| 0xbf | <b>GPIOFDAT</b>        | General Purpose Input Output Group F Data          | 0x06 |
| 0xc1 | <b>GPIOFPUE</b>        | General Purpose Input Output Group F PU Enable     | 0x06 |
| 0xc2 | <b>GPIOFPDEN</b>       | General Purpose Input Output Group F PD Enable     | 0x06 |
| 0xc3 | <b>GPIOGOUTEN</b>      | General Purpose Input Output Group G Output Enable | 0x06 |
| 0xc4 | <b>GPIOGINEN</b>       | General Purpose Input Output Group G Input Enable  | 0x06 |
| 0xc5 | <b>GPIOGDAT</b>        | General Purpose Input Output Group G Data          | 0x06 |
| 0xc6 | <b>GPIOGPUEN</b>       | General Purpose Input Output Group G PU Enable     | 0x06 |
| 0xc7 | <b>GPIOGPDEN</b>       | General Purpose Input Output Group G PD Enable     | 0x06 |
| 0xc8 | <b>MFP_CTL0</b>        | Multi-Function PAD Control Register 0              | 0x06 |
| 0xc9 | <b>MFP_CTL1</b>        | Multi-Function PAD Control Register 1              | 0x06 |
| 0xca | <b>MFP_CTL2</b>        | Multi-Function PAD Control Register 2              | 0x06 |
| 0xcb | <b>MFP_CTL3</b>        | Multi-Function PAD Control Register3               | 0x06 |
| 0xcc | <b>MFP_CTL4</b>        | Multi-Function PAD Control Register4               | 0x06 |
| 0xcd | <b>MFP_CTL5</b>        | Multi-Function PAD Control Register5               | 0x06 |
| 0xce | <b>MFP_CTL6</b>        | Multi-Function PAD Control Register6               | 0x06 |
| 0xcf | <b>MFP_CTL7</b>        | Multi-Function PAD Control Register7               | 0x06 |
| 0xd2 | <b>MFP_CTL8</b>        | Multi-Function PAD Control Register8               | 0x06 |
| 0xd3 | <b>AD_Select0</b>      | ANALOG/DIGITAL Select0                             | 0x06 |
| 0xdc | <b>AD_Select1</b>      | ANALOG/DIGITAL Select1                             | 0x06 |
| 0xd4 | <b>PADPUPD</b>         | PADPUPD  | 0x06 |
| 0xd5 | <b>PADDRV0</b>         | Pad Drive CTL0                                     | 0x06 |
| 0xd6 | <b>PADDRV1</b>         | Pad Drive CTL1                                     | 0x06 |
| 0xd7 | <b>PADDRV2</b>         | Pad Drive CTL2                                     | 0x06 |
| 0xd9 | <b>PADDRV3</b>         | Pad Drive CTL3                                     | 0x06 |
| 0xda | <b>PADDRV4</b>         | Pad Drive CTL4                                     | 0x06 |
| 0xdb | <b>DBGSEL</b>          | Debug Select Register                              | 0x06 |
| 0xe2 | <b>DBGAOE</b>          | DEBUGAOUTEN  | 0x06 |
| 0xe3 | <b>DBGAIIE</b>         | DEBUGAINEN   | 0x06 |
| 0xe4 | <b>DBGBOE</b>          | DEBUGBOUTEN  | 0x06 |
| 0xe5 | <b>DBGBIE</b>          | DEBUGBINEN   | 0x06 |
| 0xe6 | <b>DBGCOE</b>          | DEBUGCOUTEN  | 0x06 |
| 0xe9 | <b>DBGDOE</b>          | DEBUGDOUTEN  | 0x06 |
| 0xeb | <b>LED_SEG_RC_EN</b>   | LED SEG Restrict Current                           | 0x06 |
| 0xec | <b>LED_SEG_BIAS_EN</b> | LED SEG Restrict Current                           | 0x06 |

## 14.1.6 Register Description

### 14.1.6.1 GPIOAOUTEN

**GPIOAOUTEN** (General Purpose Input Output Group A Output Enable, SFR address 0xa2, bank:0x06)

| Bit(s) | Name       | Description  | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | GPIOAOUTEN | GPIOA[7:0] Output enable<br>0: Disable, 1: Enable. | R/W | 00    |

### 14.1.6.2 GPIOAINEN

**GPIOAINEN** (General Purpose Input Output Group A Input Enable, SFR address 0xa3, bank:0x06)

| Bit(s) | Name      | Description                                       | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | GPIOAINEN | GPIOA[7:0] Input enable<br>0: Disable, 1: Enable. | R/W | 00    |

### 14.1.6.3 GPIOADAT

**GPIOADAT** (General Purpose Input Output Group A Data, SFR address 0xa4, bank:0x06)

| Bit(s) | Name     | Description                  | R/W | Reset |
|--------|----------|------------------------------|-----|-------|
| 7:0    | GPIOADAT | GPIOA[7:0] Output/Input Data | R/W | 00    |

### 14.1.6.4 GPIOAPUEN

**GPIOAPUEN** (General Purpose Input Output Group A PU Enable, SFR address 0xa5, bank:0x06)

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | GPIOAPUEN | GPIOA[7:0] 50K PU Enable.<br>0: Disable , 1: Enable | R/W | 00    |

### 14.1.6.5 GPIOAPDEN

**GPIOAPDEN** (General Purpose Input Output Group A PD Enable, SFR address 0xa6, bank:0x06)

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOAPDEN | GPIOA[7:0] 50K PD Enable(GPIOA6 100K).<br>0: Disable , 1: Enable | R/W | 00    |

### 14. 1. 6. 6 GPIOBOUTEN

**GPIOBOUTEN (General Purpose Input Output Group B Output Enable, SFR address 0xa7, bank:0x06)**

| Bit(s) | Name       | Description  | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | GPIOBOUTEN | GPIOB[7:0] Output enable<br>0: Disable, 1: Enable. | R/W | 00    |

### 14. 1. 6. 7 GPIOBINEN

**GPIOBINEN (General Purpose Input Output Group B Input Enable, SFR address 0xa9, bank:0x06)**

| Bit(s) | Name      | Description                                       | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | GPIOBINEN | GPIOB[7:0] Input enable<br>0: Disable, 1: Enable. | R/W | 00    |

### 14. 1. 6. 8 GPIOBDAT

**GPIOBDAT (General Purpose Input Output Group B Data, SFR address 0xaa, bank:0x06)**

| Bit(s) | Name     | Description                  | R/W | Reset |
|--------|----------|------------------------------|-----|-------|
| 7:0    | GPIOBDAT | GPIOB[7:0] Output/Input Data | R/W | 00    |

### 14. 1. 6. 9 GPIOBPUEN

**GPIOBPUEN (General Purpose Input Output Group B PU Enable, SFR address 0xab, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOBPUEN | GPIOB[7:0] 50K PU Enable.<br>0: Disable, 1: Enable | R/W | 00    |

### 14. 1. 6. 10 GPIOBPDEN

**GPIOBPDEN (General Purpose Input Output Group B PD Enable, SFR address 0xac, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOBPDEN | GPIOB[7:0] 50K PD Enable.<br>0: Disable, 1: Enable | R/W | 00    |

### 14. 1. 6. 11 GPIOCOUTEN

**GPIOCOUTEN (General Purpose Input Output Group C Output Enable, SFR address 0xad, bank:0x06)**

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |            |   |     |    |
|-----|------------|---|-----|----|
| 7:0 | GPIOCOUTEN | GPIOC[7:0] Output enable,<br>0: Disable, 1: Enable. | R/W | 00 |
|-----|------------|---|-----|----|

#### 14. 1. 6. 12 GPIOCINEN

**GPIOCINEN (General Purpose Input Output Group C Input Enable, SFR address 0xae, bank:0x06)**

| Bit(s) | Name      | Description                                       | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | GPIOCINEN | GPIOC[7:0] Input enable<br>0: Disable, 1: Enable. | R/W | 00    |

#### 14. 1. 6. 13 GPIOCDAT

**GPIOCDAT (General Purpose Input Output Group C Data, SFR address 0xaf, bank:0x06)**

| Bit(s) | Name     | Description                  | R/W | Reset |
|--------|----------|------------------------------|-----|-------|
| 7:0    | GPIOCDAT | GPIOC[7:0] Output/Input Data | R/W | 00    |

#### 14. 1. 6. 14 GPIOCPUEN

**GPIOCPUEN (General Purpose Input Output Group C PU Enable, SFR address 0xb0, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOCPUEN | GPIOC[7:0] 50K PU Enable.<br>0: Disable, 1: Enable | R/W | 00    |

#### 14. 1. 6. 15 GPIOCPDEN

**GPIOCPDEN (General Purpose Input Output Group C PD Enable, SFR address 0xb1, bank:0x06)**

| Bit(s) | Name      | Description   | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | GPIOCPDEN | GPIOC[7:0] 50K PD Enable(GPIOC6 100K).<br>0: Disable, 1: Enable | R/W | 00    |

#### 14. 1. 6. 16 GPIODOUTEN

**GPIODOUTEN (General Purpose Input Output Group D Output Enable, SFR address 0xb2, bank:0x06)**

| Bit(s) | Name       | Description  | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:0    | GPIODOUTEN | GPIOD[7:0] Output enable<br>0: Disable, 1: Enable. | R/W | 00    |

### 14.1.6.17 GPIODINEN

**GPIODINEN (General Purpose Input Output Group D Input Enable, SFR address 0xb3, bank:0x06)**

| Bit(s) | Name      | Description                                       | R/W | Reset |
|--------|-----------|---|-----|-------|
| 7:0    | GPIODINEN | GPIOD[7:0] Input enable<br>0: Disable, 1: Enable. | R/W | 00    |

### 14.1.6.18 GPIODDAT

**GPIODDAT (General Purpose Input Output Group D Data, SFR address 0xb4, bank:0x06)**

| Bit(s) | Name     | Description                  | R/W | Reset |
|--------|----------|------------------------------|-----|-------|
| 7:0    | GPIODDAT | GPIOD[7:0] Output/Input Data | R/W | 00    |

### 14.1.6.19 GPIODPUE

**GPIODPUE (General Purpose Input Output Group D PU Enable, SFR address 0xb5, bank:0x06)**

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:0    | GPIODPUE | GPIOD[7:0] 50K PU Enable.<br>0: Disable , 1: Enable | R/W | 00    |

### 14.1.6.20 GPIODPDEN

**GPIODPDEN (General Purpose Input Output Group D PD Enable, SFR address 0xb6, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIODPDEN | GPIOD[7:0] 50K PD Enable.<br>0: Disable, 1: Enable | R/W | 00    |

### 14.1.6.21 GPIOEOUTEN

**GPIOEOUTEN (General Purpose Input Output Group E Output Enable, SFR address 0xb7, bank:0x06)**

| Bit(s) | Name       | Description                                       | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | GPIOEOUTEN | GPIOE[7:0] Output Enable<br>0: Disable, 1: Enable | R/W | 00    |

## 14. 1. 6. 22 GPIOEINEN

**GPIOEINEN (General Purpose Input Output Group E Input Enable, SFR address 0xb86, bank:0x06)**

| Bit(s) | Name      | Description                                      | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOEINEN | GPIOE[7:0] Input Enable<br>0: Disable, 1: Enable | R/W | 00    |

## 14. 1. 6. 23 GPIOEDAT

**GPIOEDAT (General Purpose Input Output Group E Data, SFR address 0xb9, bank:0x06)**

| Bit(s) | Name     | Description                  | R/W | Reset |
|--------|----------|------------------------------|-----|-------|
| 7:0    | GPIOEDAT | GPIOE[7:0] Output/Input Data | R/W | 00    |

## 14. 1. 6. 24 GPIOEPUEN

**GPIOEPUEN (General Purpose Input Output Group E PU Enable, SFR address 0xba, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOEPUEN | GPIOE[7:0] 50K PU Enable.<br>0: Disable, 1: Enable | R/W | 00    |

## 14. 1. 6. 25 GPIOEPDEN

**GPIOEPDEN (General Purpose Input Output Group E PD Enable, SFR address 0xbb, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOEPDEN | GPIOE[7:0] 50K PD Enable.<br>0: Disable, 1: Enable | R/W | 00    |

## 14. 1. 6. 26 GPIOFOUTEN

**GPIOFOUTEN (General Purpose Input Output Group F Output Enable, SFR address 0xbc, bank:0x06)**

| Bit(s) | Name       | Description                                       | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:0    | GPIOFOUTEN | GPIOF[7:0] Output Enable<br>0: Disable, 1: Enable | R/W | 00    |

## 14. 1. 6. 27 GPIOFINEN

**GPIOFINEN (General Purpose Input Output Group F Input Enable, SFR address 0xbd, bank:0x06)**

| Bit(s) | Name      | Description                                      | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOFINEN | GPIOF[7:0] Input Enable<br>0: Disable, 1: Enable | R/W | 00    |

## 14. 1. 6. 28 GPIOFDAT

**GPIOFDAT (General Purpose Input Output Group F Data, SFR address 0xbf, bank:0x06)**

| Bit(s) | Name     | Description                  | R/W | Reset |
|--------|----------|------------------------------|-----|-------|
| 7:0    | GPIOFDAT | GPIOF[7:0] Output/Input Data | R/W | 00    |

## 14. 1. 6. 29 GPIOFPUEN

**GPIOFPUEN (General Purpose Input Output Group F PU Enable, SFR address 0xc1, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOFPUEN | GPIOF[7:0] 50K PU Enable.<br>0: Disable, 1: Enable | R/W | 00    |

## 14. 1. 6. 30 GPIOFPDEN

**GPIOFPDEN (General Purpose Input Output Group F PD Enable, SFR address 0xc2, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:0    | GPIOFPDEN | GPIOF[7:0] 50K PD Enable.<br>0: Disable, 1: Enable | R/W | 00    |

## 14. 1. 6. 31 GPIOGOUTEN

**GPIOGOUTEN (General Purpose Input Output Group G Output Enable, SFR address 0xc3, bank:0x06)**

| Bit(s) | Name       | Description                                       | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:3    | Reserved   | Reserved  | R/W | 00    |
| 2:0    | GPIOGOUTEN | GPIOG[2:0] Output Enable<br>0: Disable, 1: Enable | R/W | 00    |

### 14.1.6.32 GPIOGINEN

**GPIOGINEN (General Purpose Input Output Group G Input Enable, SFR address 0xc4, bank:0x06)**

| Bit(s) | Name      | Description                                      | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:3    | Reserved  | Reserved   | R/W | 00    |
| 2:0    | GPIOGINEN | GPIOG[2:0] Input Enable<br>0: Disable, 1: Enable | R/W | 00    |

### 14.1.6.33 GPIOGDAT

**GPIOGDAT (General Purpose Input Output Group G Data, SFR address 0xc5, bank:0x06)**

| Bit(s) | Name     | Description                  | R/W | Reset |
|--------|----------|------------------------------|-----|-------|
| 7:3    | Reserved | Reserved                     | R/W | 00    |
| 2:0    | GPIOGDAT | GPIOG[2:0] Output/Input Data | R/W | 00    |

### 14.1.6.34 GPIOGPUEN

**GPIOGPUEN (General Purpose Input Output Group G PU Enable, SFR address 0xc6, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:3    | Reserved  | Reserved   | R/W | 00    |
| 2:0    | GPIOGPUEN | GPIOG[2:0] 50K PU Enable.<br>0: Disable, 1: Enable | R/W | 00    |

### 14.1.6.35 GPIOGPDEN

**GPIOGPDEN (General Purpose Input Output Group G PD Enable, SFR address 0xc7, bank:0x06)**

| Bit(s) | Name      | Description  | R/W | Reset |
|--------|-----------|--|-----|-------|
| 7:3    | Reserved  | Reserved   | R/W | 00    |
| 2:0    | GPIOGPDEN | GPIOG[2:0] 50K PD Enable.<br>0: Disable, 1: Enable | R/W | 00    |

### 14.1.6.36 MFP\_CTL0

**MFP\_CTL0 (Multi-Function PAD Control Register 0, SFR address 0xc8, bank:0x06)**

| Bit(s) | Name | Description | R/W | Reset |
|--------|------|-------------|-----|-------|
|        |      |             |     |       |

|     |          |  |     |    |
|-----|----------|--|-----|----|
| 7   | Reserved | Reserved   | R/W | 0  |
| 6:5 | GPIOA7   | 00: LED_COM7<br>01: LCD_SEG3<br>10: FMCLKOUT<br>11: Reserved | R/W | 00 |
| 4:3 | GPIOA6   | 00: LED_COM6<br>01: LCD_SEG2<br>10: UART_RTS<br>11: SIRQ1    | R/W | 00 |
| 2:1 | GPIOA5   | 00: LED_COM5<br>01: LCD_SEG1<br>10: UART_CTS<br>11: Reserved | R/W | 00 |
| 0   | Reserved | Reserved   | R/W | 0  |

#### 14.1.6.37 MFP\_CTL1

**MFP\_CTL1** (Multi-Function PAD Control Register 1, SFR address 0xc9, bank:0x06)

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:6    | GPIOA4 | 00: LED_COM4<br>01: LCD_COM3<br>10: PWM<br>11: Reserved     | R/W | 00    |
| 5:4    | GPIOA3 | 00: EM_CEB0<br>01: LED_COM3<br>10: LCD_SEG0<br>11: Reserved | R/W | 00    |
| 3:2    | GPIOA2 | 00: EM_RDB<br>01: LED_COM2<br>10: LCD_COM2<br>11: Reserved  | R/W | 00    |
| 1:0    | GPIOA1 | 00: EM_RS<br>01: LED_COM1<br>10: LCD_COM1<br>11: Reserved   | R/W | 00    |

### 14.1.6.38 MFP\_CTL2

**MFP\_CTL2 (Multi-Function PAD Control Register 2, SFR address 0xca, bank:0x06)**

| Bit(s) | Name     | Description  | R/W | Reset |
|--------|----------|--|-----|-------|
| 7:4    | Reserved | Reserved   | R/W | 0     |
| 3:2    | GPIOA0   | 00: EM_WRB<br>01: LED_COM0<br>10: LCD_COM0<br>11: Reserved                     | R/W | 00    |
| 1      | Reserved | Reserved   | R/W | 0     |
| 0      | EJTAGEN  | Not:上电使能只控制第一组 EJTAG, 其他组通过长按复位键的时间来控制选择。<br>0:EJTAG Disable<br>1:EJTAG Enable | R/W | 1     |

### 14.1.6.39 MFP\_CTL3

**MFP\_CTL3 (Multi-Function PAD Control Register 3, SFR address 0xcb, bank:0x06)**

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7:6    | GPIOB7 | 00: EM_D7<br>01: LED_SEG7<br>10: LCD_SEG11<br>11: Reserved | R/W | 00    |
| 5:4    | GPIOB6 | 00: EM_D6<br>01: LED_SEG6<br>10: LCD_SEG10<br>11: Reserved | R/W | 00    |
| 3:2    | GPIOB5 | 00: EM_D5<br>01: LED_SEG5<br>10: LCD_SEG9<br>11: Reserved  | R/W | 00    |
| 1:0    | GPIOB4 | 00: EM_D4<br>01: LED_SEG4<br>10: LCD_SEG8<br>11: Reserved  | R/W | 00    |

### 14.1.6.40 MFP\_CTL4

**MFP\_CTL4 (Multi-Function PAD Control Register 4, SFR address 0xcc, bank:0x06)**

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:6    | GPIOB3 | 00: EM_D3<br>01: LED SEG3<br>10: LCD SEG7<br>11: Reserved | R/W | 00    |
| 5:4    | GPIOB2 | 00: EM_D2<br>01: LED SEG2<br>10: LCD SEG6<br>11: Reserved | R/W | 00    |
| 3:2    | GPIOB1 | 00: EM_D1<br>01: LED SEG1<br>10: LCD SEG5<br>11: Reserved | R/W | 00    |
| 1:0    | GPIOB0 | 00: EM_D0<br>01: LED SEG0<br>10: LCD SEG4<br>11: Reserved | R/W | 00    |

#### 14.1.6.41 MFP\_CTL5

**MFP\_CTL5 (Multi-Function PAD Control Register 5, SFR address 0xcd, bank:0x06)**

| Bit(s) | Name     | Description  | R/W | Reset |
|--------|----------|--|-----|-------|
| 7      | Reserved | Reserved   | R/W | 0     |
| 6:5    | GPIOC6   | 00: SIRQ0<br>01: SPDIF0<br>10: LCD SEG14<br>11: Reserved     | R/W | 00    |
| 4      | GPIOC5   | 0: MMC_DAT3<br>1: IR_RX                                      | R/W | 0     |
| 3:2    | GPIOC4   | 00: MMC_DAT2<br>01: UART_TX<br>10: LCD SEG13<br>11: Reserved | R/W | 00    |
| 1:0    | GPIOC3   | 00: MMC_DAT1<br>01: UART_RX<br>10: MMC_CLK1<br>11: LCD SEG12 | R/W | 00    |

### 14.1.6.42 MFP\_CTL6

**MFP\_CTL6 (Multi-Function PAD Control Register 6, SFR address 0xce, bank:0x06)**

| Bit(s) | Name     | Description                 | R/W | Reset |
|--------|----------|-----------------------------|-----|-------|
| 7:5    | Reserved | Reserved                    | R/W | 0     |
| 4      | GPIOD5   | 0: SPDIF1<br>1: LCD_SEG21   | R/W | 0     |
| 3      | GPIOD3   | 0: SPI_SCLK<br>1: LCD_SEG19 | R/W | 0     |
| 2      | GPIOD2   | 0: SPI_MOSI<br>1: LCD_SEG18 | R/W | 0     |
| 1      | GPIOD1   | 0: SPI_MISO<br>1: LCD_SEG17 | R/W | 0     |
| 0      | GPIOD0   | 0: SPI_SS<br>1: LCD_SEG16   | R/W | 0     |

### 14.1.6.43 MFP\_CTL7

**MFP\_CTL7 (Multi-Function PAD Control Register 7, SFR address 0xcf, bank:0x06)**

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:6    | GPIOE7 | 00: I2S_DI<br>01: LCD_SEG31<br>10: EM_D15<br>11: Reserved     | R/W | 00    |
| 5:4    | GPIOE6 | 00: I2S1_MCLK<br>01: LCD_SEG30<br>10: EM_D14<br>11: Reserved  | R/W | 00    |
| 3:2    | GPIOE5 | 00: I2S1_LRCLK<br>01: LCD_SEG29<br>10: EM_D13<br>11: Reserved | R/W | 00    |
| 1:0    | GPIOE4 | 00: I2S1_BCLK<br>01: LCD_SEG28<br>10: EM_D12<br>11: Reserved  | R/W | 00    |

### 14.1.6.44 MFP\_CTL8

**MFP\_CTL8 (Multi-Function PAD Control Register 8, SFR address 0xd2, bank:0x06)**

| Bit(s) | Name   | Description  | R/W | Reset |
|--------|--------|--|-----|-------|
| 7:6    | GPIOE3 | 00: I2S0_DO<br>01: LCD SEG27<br>10: EM_D11<br>11: Reserved   | R/W | 00    |
| 5:4    | GPIOE2 | 00: I2S0_MCLK<br>01: LCD SEG26<br>10: EM_D10<br>11: Reserved | R/W | 00    |
| 3:2    | GPIOE1 | 00: I2S0_LRCLK<br>01: LCD SEG25<br>10: EM_D9<br>11: UART_TX  | R/W | 00    |
| 1:0    | GPIOE0 | 00: I2S0_BCLK<br>01: LCD SEG24<br>10: EM_D8<br>11: UART_RX   | R/W | 00    |

### 14.1.6.45 AD\_Select0

**AD\_Select0 (ANALOG/DIGITAL Select, SFR address 0xd3, bank:0x06)**

| Bit(s) | Name   | Description   | R/W | Reset |
|--------|--------|---|-----|-------|
| 7:6    | GPIOC6 | 00: GPIOC6 is used as digital function,<br>01: GPIOC6 is used as AVCC1 (Analog Function),<br>10: GPIOC6 is used as TK6 (Analog Function),<br>11: Reserved                                   | R/W | 0     |
| 5:4    | GPIOC7 | 00: GPIOC7 is used as digital function,<br>01: GPIOC7 is used as LRADC5 (Analog Function),<br>10: GPIOC7 is used as TK5 (Analog Function),<br>11: Reserved                                  | R/W | 0     |
| 3:2    | GPIOA5 | 00: GPIOA5 is used as digital function,<br>01: GPIOA5 is used as LRADC3 (Analog Function),<br>10: GPIOA5 is used as AVCC0 (Analog Function),<br>11: GPIOA5 is used as TK4 (Analog Function) | R/W | 0     |
| 1:0    | GPIOA6 | 00: GPIOA6 is used as digital function,<br>01: GPIOA6 is used as LRADC4 (Analog Function),<br>10: GPIOA6 is used as TK3 (Analog Function),<br>11: Reserved                                  | R/W | 0     |

### 14. 1. 6. 46 AD\_Select1

**AD\_Select1 (ANALOG/DIGITAL Select, SFR address 0xdc, bank:0x06)**

| Bit(s) | Name     | Description   | R/W | Reset |
|--------|----------|---|-----|-------|
| 7:5    | Reserved | Reserved  | R/W | 0     |
| 4      | GPIOC4   | 0:GPIOC4 is used as digital function,<br>1:GPIOC4 is used as TK1 (Analog Function)  | R/W | 0     |
| 3:2    | GPIOC3   | 00: GPIOC3 is used as digital function,<br>01: GPIOC3 is used as VDD (Analog Function),<br>10: GPIOC3 is used as TK2 (Analog Function),<br>11: Reserved     | R/W | 0     |
| 1:0    | GPIOG2   | 00: GPIOG2 is used as digital function,<br>01: GPIOG2 is used as REM_CON (Analog Function),<br>10: GPIOG2 is used as TK0 (Analog Function),<br>11: Reserved | R/W | 0     |

### 14. 1. 6. 47 PADPUPD

**PADPUPD (PADPUPD, SFR address 0xd4, bank:0x06)**

| Bit(s) | Name         | Description  | R/W | Reset |
|--------|--------------|--|-----|-------|
| 7      | SIRQ0PUPDEN  | SIRQ0 100k PU or PD Enable<br>0:Disable<br>1:Enable                            | R/W | 0     |
| 6      | SIRQ0PUPDSEL | SIRQ0 100k PU or PD Select<br>0:100k Pull-up Enable<br>1:100k Pull-down Enable | R/W | 0     |
| 5      | SIRQ1PUPDEN  | SIRQ1 100k PU or PD Enable<br>0:Disable<br>1:Enable                            | R/W | 0     |
| 4      | SIRQ1PUPDSEL | SIRQ1 100k PU or PD Select<br>0:100k Pull-up Enable<br>1:100k Pull-down Enable | R/W | 0     |
| 3      | MMCCMDPU     | MMC/SD CMD 50k Pull-up Enable<br>0:Disable<br>1:Enable                         | R/W | 0     |
| 2      | MMCDATPU     | MMC/SD Data 50k Pull-up Enable<br>0:Disable<br>1:Enable                        | R/W | 0     |
| 1      | Reserved     | Reserved   | R/W | 00    |

|   |        |  |     |   |
|---|--------|--|-----|---|
| 0 | UARTPU | UART 10k Pull-up Enable<br>0:Disable<br>1:Enable | R/W | 0 |
|---|--------|--|-----|---|

#### 14. 1. 6. 48 PADDRV0

**PADDRV0 (Pad Drive CTL0, SFR address 0xd5, bank:0x06)**

| Bit(s) | Name       | Description  | R/W | Reset |
|--------|------------|--|-----|-------|
| 7:6    | GPIOA7DRV  | GPIOA7 Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Level 16      | R/W | 0     |
| 5:4    | GPIOA65DRV | GPIOA[6:5] Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Level 16  | R/W | 0     |
| 3:2    | GPIOA43DRV | GPIOA[4:3] Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11: Level 16 | R/W | 00    |
| 1:0    | GPIOA20DRV | GPIOA[2:0] Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11: Level 16 | R/W | 00    |

#### 14. 1. 6. 49 PADDRV1

**PADDRV1 (Pad Drive CTL1, SFR address 0xd6, bank:0x06)**

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:6    | GPIOB70DRV | GPIOB Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11: Reserved | R/W | 00    |
| 5:0    | Reserved   | Reserved  | R/W | 00    |

### 14.1.6.50 PADDRV2

**PADDRV2 (Pad Drive CTL2, SFR address 0xd7, bank:0x06)**

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7      | GPIOC7DRV  | GPIOC7 Pad Drive Control<br>0:Level 1<br>1:Level 2                                    | R/W | 0     |
| 6      | GPIOC6DRV  | GPIOC6 Pad Drive Control<br>0:Level 1<br>1:Level 2                                    | R/W | 0     |
| 5:4    | GPIOC52DRV | GPIOC[5:2] Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Reserved | R/W | 01    |
| 3:2    | GPIOC1DRV  | GPIOC1 Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Reserved     | R/W | 01    |
| 1:0    | GPIOC0DRV  | GPIOC0 Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Reserved     | R/W | 01    |

### 14.1.6.51 PADDRV3

**PADDRV3 (Pad Drive CTL3, SFR address 0xd9, bank:0x06)**

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:6    | GPIOD76DRV | GPIOD[7:6] Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Reserved | R/W | 01    |
| 5:4    | GPIOD54DRV | GPIOD[5:4] Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Reserved | R/W | 01    |
| 3:2    | GPIOD30DRV | GPIOD[3:0] Pad Drive Control  | R/W | 01    |

|     |          |   |     |    |
|-----|----------|---|-----|----|
|     |          | 00:Level 1<br>01:Level 2<br>10:Level 4<br>11:Reserved |     |    |
| 1:0 | Reserved | Reserved  | R/W | 00 |

### 14. 1. 6. 52 PADDRV4

**PADDRV4 (Pad Drive CTL4, SFR address 0xda, bank:0x06)**

| Bit(s) | Name       | Description   | R/W | Reset |
|--------|------------|---|-----|-------|
| 7:6    | GPIOE74DRV | GPIOE[7:4] Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Reserved | R/W | 00    |
| 5:4    | GPIOE30DRV | GPIOE[3:0] Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Reserved | R/W | 00    |
| 3:2    | GPIOG2DRV  | GPIOG2 Pad Drive Control<br>00:Level 1<br>01:Level 2<br>10:Level 3<br>11:Reserved     | R/W | 00    |
| 1:0    | Reserved   | Reserved  | R/W | 0000  |

### 14. 1. 6. 53 DBGSEL

**DBGSEL (Debug Select Register, SFR address 0xdb, bank:0x06)**

| Bit(s) | Name     | Description  | R/W | Reset |
|--------|----------|--|-----|-------|
| 7      | BDREN    | CMU Analog Back Door Enable<br>1:Enable<br>0:Disable   | R/W | 0     |
| 6:5    | Reserved | Reserved   | R/W | 00    |
| 4:0    | DGBSE    | 00000:Disable<br>00001:SD Debug Signals<br>00010:LCD Debug Signals<br>00011:SPI and SPI_CK Debug Signals | R/W | 00    |

|  |  |  |  |
|--|--|--|--|
|  | 00100:SPDIF Debug Signals<br>00101:UART Debug Signals<br>00110:IR Debug Signals<br>00111: Reserved<br>01000:USB Debug Signals<br>01001:PMU Debug Signals<br>01010:TK analog debug<br>01011: Reserved<br>01100: DMA0/1/2 Debug Signals<br>01101: RTC Debug Signals<br>01110: AD Debug Signals<br>01111: DA Debug Signals<br>10000: EFUSE Debug Signals<br>10001:Output 0x55<br>10010:Output 0xAA<br>10011:CMU Analog Debug Signals<br>10100: Reserved<br>10101: Reserved<br>10110: Reserved<br>Others: Reserved |  |  |
|--|--|--|--|

各模块使用的 DEBUG 信号如下：

| Module  | Debug GPIO                                  |
|---------|---|
| PMU     | GPIOA[7:0];GPIOD[3:0]                       |
| SPI     | GPIOA[7:0]                                  |
| SPI_CK  | GPIOB[7:0]                                  |
| UART    | GPIOB[7:0]                                  |
| CARD    | GPIOA[7:0];GPIOB[7:0]                       |
| IR      | GPIOB[7:0]                                  |
| USB     | GPIOA[7:0];GPIOB[7:0];GPIOC[7:0];GPIOD[7:0] |
| LCD/LED | GPIOA[7:4];GPIOC[7:0];GPIOD[3:0]            |
| DMA     | GPIOA[6:0];GPIOC[3:0]                       |
| RTC     | GPIOD[7:0]                                  |
| AD      | GPIOA[5:0]                                  |
| DA      | GPIOA[7:0];GPIOB[1:0]                       |
| CMU     | GPIOD0                                      |
| SPDIF   | GPIOA[7:0]                                  |
| TK      | GPIOA[4:0];GPIOA7;GPIOB[7:0];GPIOD0         |
| EFUSE   | GPIOA[5:0]                                  |

#### 14.1.6.54 DBGAOE

**DBGAOE (DEBUGAOUTEN, SFR address 0xe2, bank:0x06)**

| Bit(s) | Name    | Description                | R/W | Reset |
|--------|---------|----------------------------|-----|-------|
| 7      | DBGA70E | Debug GPIOA7 Output Enable | R/W | 0     |
| 6      | DBGA60E | Debug GPIOA6 Output Enable | R/W | 0     |
| 5      | DBGA50E | Debug GPIOA5 Output Enable | R/W | 0     |
| 4      | DBGA40E | Debug GPIOA4 Output Enable | R/W | 0     |
| 3      | DBGA30E | Debug GPIOA3 Output Enable | R/W | 0     |
| 2      | DBGA20E | Debug GPIOA2 Output Enable | R/W | 0     |
| 1      | DBGA10E | Debug GPIOA1 Output Enable | R/W | 0     |
| 0      | DBGA00E | Debug GPIOA0 Output Enable | R/W | 0     |

#### 14.1.6.55 DBGAIE

**DBGAIE (DEBUGAINEN, SFR address 0xe3, bank:0x06)**

| Bit(s) | Name    | Description               | R/W | Reset |
|--------|---------|---------------------------|-----|-------|
| 7      | DBGA7IE | Debug GPIOA7 Input Enable | R/W | 0     |
| 6      | DBGA6IE | Debug GPIOA6 Input Enable | R/W | 0     |
| 5      | DBGA5IE | Debug GPIOA5 Input Enable | R/W | 0     |
| 4      | DBGA4IE | Debug GPIOA4 Input Enable | R/W | 0     |
| 3      | DBGA3IE | Debug GPIOA3 Input Enable | R/W | 0     |
| 2      | DBGA2IE | Debug GPIOA2 Input Enable | R/W | 0     |
| 1      | DBGA1IE | Debug GPIOA1 Input Enable | R/W | 0     |
| 0      | DBGA0IE | Debug GPIOA0 Input Enable | R/W | 0     |

#### 14.1.6.56 DBGBOE

**DBGBOE (DEBUGBOUTEN, SFR address 0xe4, bank:0x06)**

| Bit(s) | Name    | Description                | R/W | Reset |
|--------|---------|----------------------------|-----|-------|
| 7      | DBGB70E | Debug GPIOB7 Output Enable | R/W | 0     |
| 6      | DBGB60E | Debug GPIOB6 Output Enable | R/W | 0     |
| 5      | DBGB50E | Debug GPIOB5 Output Enable | R/W | 0     |
| 4      | DBGB40E | Debug GPIOB4 Output Enable | R/W | 0     |
| 3      | DBGB30E | Debug GPIOB3 Output Enable | R/W | 0     |
| 2      | DBGB20E | Debug GPIOB2 Output Enable | R/W | 0     |
| 1      | DBGB10E | Debug GPIOB1 Output Enable | R/W | 0     |
| 0      | DBGB00E | Debug GPIOB0 Output Enable | R/W | 0     |

### 14.1.6.57 DBGBIE

**DBGBIE (DEBUGBINEN, SFR address 0xe5, bank:0x06)**

| Bit(s) | Name    | Description               | R/W | Reset |
|--------|---------|---------------------------|-----|-------|
| 7      | DBGB7IE | Debug GPIOB7 Input Enable | R/W | 0     |
| 6      | DBGB6IE | Debug GPIOB6 Input Enable | R/W | 0     |
| 5      | DBGB5IE | Debug GPIOB5 Input Enable | R/W | 0     |
| 4      | DBGB4IE | Debug GPIOB4 Input Enable | R/W | 0     |
| 3      | DBGB3IE | Debug GPIOB3 Input Enable | R/W | 0     |
| 2      | DBGB2IE | Debug GPIOB2 Input Enable | R/W | 0     |
| 1      | DBGB1IE | Debug GPIOB1 Input Enable | R/W | 0     |
| 0      | DBGB0IE | Debug GPIOB0 Input Enable | R/W | 0     |

### 14.1.6.58 DBGCOE

**DBGCOE (DEBUGCOUTEN, SFR address 0xe6, bank:0x06)**

| Bit(s) | Name    | Description                | R/W | Reset |
|--------|---------|----------------------------|-----|-------|
| 7      | DBGC70E | Debug GPIOC7 Output Enable | R/W | 0     |
| 6      | DBGC60E | Debug GPIOC6 Output Enable | R/W | 0     |
| 5      | DBGC50E | Debug GPIOC5 Output Enable | R/W | 0     |
| 4      | DBGC40E | Debug GPIOC4 Output Enable | R/W | 0     |
| 3      | DBGC30E | Debug GPIOC3 Output Enable | R/W | 0     |
| 2      | DBGC20E | Debug GPIOC2 Output Enable | R/W | 0     |
| 1      | DBGC10E | Debug GPIOC1 Output Enable | R/W | 0     |
| 0      | DBGC00E | Debug GPIOC0 Output Enable | R/W | 0     |

### 14.1.6.59 DBGDOE

**DBGDOE (DEBUGDOUTEN, SFR address 0xe9, bank:0x06)**

| Bit(s) | Name    | Description                | R/W | Reset |
|--------|---------|----------------------------|-----|-------|
| 7      | DBGD70E | Debug GPIOD7 Output Enable | R/W | 0     |
| 6      | DBGD60E | Debug GPIOD6 Output Enable | R/W | 0     |
| 5      | DBGD50E | Debug GPIOD5 Output Enable | R/W | 0     |
| 4      | DBGD40E | Debug GPIOD4 Output Enable | R/W | 0     |
| 3      | DBGD30E | Debug GPIOD3 Output Enable | R/W | 0     |

|   |         |                            |     |   |
|---|---------|----------------------------|-----|---|
| 2 | DBGD20E | Debug GPIOD2 Output Enable | R/W | 0 |
| 1 | DBGD10E | Debug GPIOD1 Output Enable | R/W | 0 |
| 0 | DBGD00E | Debug GPIOD0 Output Enable | R/W | 0 |

#### 14. 1. 6. 60 LED\_SEG\_RC\_EN

**LED\_SEG\_RC\_EN (LED SEG Restrict Current Enable, SFR address 0xeb, bank:0x06)**

| Bit(s) | Name     | Description                      | R/W | Reset |
|--------|----------|----------------------------------|-----|-------|
| 7      | LED_SEG7 | LED SEG7 Restrict Current Enable | R/W | 0     |
| 6      | LED_SEG6 | LED SEG6 Restrict Current Enable | R/W | 0     |
| 5      | LED_SEG5 | LED SEG5 Restrict Current Enable | R/W | 0     |
| 4      | LED_SEG4 | LED SEG4 Restrict Current Enable | R/W | 0     |
| 3      | LED_SEG3 | LED SEG3 Restrict Current Enable | R/W | 0     |
| 2      | LED_SEG2 | LED SEG2 Restrict Current Enable | R/W | 0     |
| 1      | LED_SEG1 | LED SEG1 Restrict Current Enable | R/W | 0     |
| 0      | LED_SEG0 | LED SEG0 Restrict Current Enable | R/W | 0     |

#### 14. 1. 6. 61 LED\_SEG\_BIAS\_EN

**LED\_SEG\_BIAS\_EN (LED SEG BIAS Current Enable, SFR address 0xec, bank:0x06)**

| Bit(s) | Name                   | Description  | R/W | Reset |
|--------|------------------------|--|-----|-------|
| 7:4    | Reserved               | Reserved   | R/W | 00    |
| 3      | LED_SEG_ALL_EN         | LED SEG Restrict Current ALL Enable<br>0:Disable<br>1:Enable | R/W | 0     |
| 2      | LED_Cathode_Anode_Mode | LED Cathode/Anode Mode<br>0: Cathode Mode<br>1: Anode Mode   | R/W | 0     |
| 1:0    | LED_SEG_BIAS           | LED SEG BIAS:<br>00:2mA<br>01:3mA<br>10:3.5mA<br>11:4mA      | R/W | 01    |

注：当把 LED\_SEG\_ALL\_EN Enable 后，和 LED 复用的数字 PAD 及其上下拉电阻都 Disable。

# 15 BROM (黄晓佳)

| 日期         | 版本     | 描述           | 修订人 |
|------------|--------|--------------|-----|
| 2012-12-12 | V2. 02 | Initial (新增) | 黄晓佳 |

## 15.1 Features

- ◆ BROM 启动(Bootloader)功能主要包括:
  - Power On (上电初始化)
  - Mbrec Launcher (存储介质上 Mbrc 加载)
  - Storage Driver(存储介质的驱动)
  - ADFU Launcher (USB ADFU 固件升级)
- ◆ BROM 空间大小为 5KB
- ◆ BROM 可以支持存储介质包括 NorFLASH、SD/MMC

## 15.2 Function Description

BROM 的起始地址是 0xff9c00，内部各模块程序存放的位置如 Figure2.1 所示，包括 PowerOn, BRECLauncher, Storage Driver(包括 NorFLASH、SD/MMC)和 ADFULauncher。

BROM 的总体流程图如所示。上电后，PC 到 0xffc000 地址执行的第一条跳转到 0xff9c00 的指令，开始运行 PowerOn，进行必要的寄存器初始化，然后将控制权交给 BRECLauncher。BRECLauncher 通过调用 StorageDrivers 提供的各种介质驱动，读取 MBRC 后进行校验，如果正确则跳转到对应的 MBRC 地址，否则继续下一个引导介质。如果没有任何一种存储介质有正确的 MBRC，系统进入 ADFULauncher。

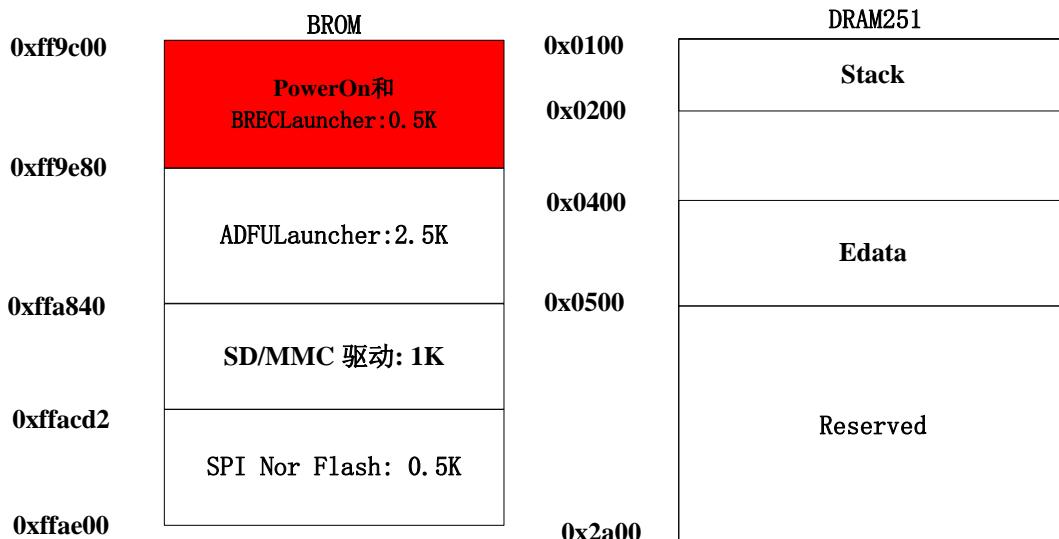


Figure 15-1 BROM 和 DRAM 的空间分配

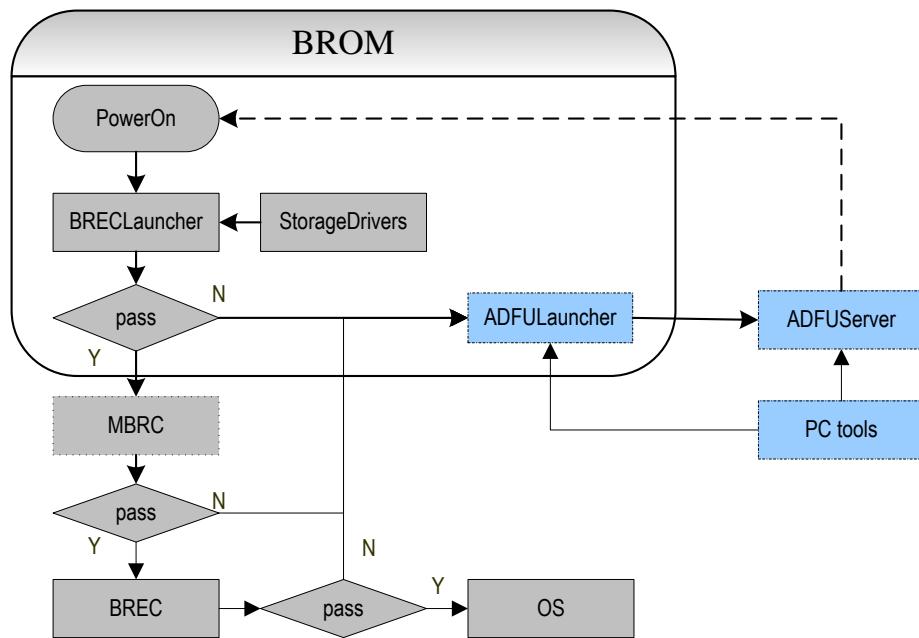


Figure 15-2 总体启动流程

## 15.3 Module Description

### 15.3.1 POWER ON

Power On 模块主要完成以下几个功能：

- 1、设置看门狗时间并使能看门狗。测试阶段的看门狗时间为 1.4s，测试过后改为 5.6s。
- 2、使能 24M 晶振，默认就已经是打开。配置 RMU 寄存器，关闭所有模块使能位。
- 3、配置堆栈指针 SP，当前 brom 的堆栈是从 100h 到 200h。
- 4、使能所有 memory clk，并选择 memory clk 为 mcu clk，MURAM 切为 DATA RAM
- 5、Vcc、vdd 配置默认值，分别为 3.1v、1.7v。
- 6、调用 Brec Launcher 进行存储介质的引导。
- 7、如果引导 Brec Launcher 不成功，则回到 poweron 将中断向量表切到 0xff0000，并填入 0xff0043 和 0xff007b 的入口地址，然后跳到 ADFU Launcher 程序运行。

### 15.3.2 BREC Launcher

GL5115 中 BREC Launcher 主要是实现两种存储的引导：NorFLASH、SD/MMC，并且按照此顺序进行引导。具体的流程和步骤如 Figure2.4 所示：

- 1、如果 Nor Flash driver, SDMMC Card driver 成功检测到一种引导介质，就会从该介质中读取 512 字节的 MBRC 到 PRAM 的 0xff0200 地址，并将控制权交给它。
- 2、若以上驱动都没有检测到相应的 MBRC，系统将重新回到 PowerOn 模块，执行 ADFULauncher 后进入 ADFU，此时可以通过与 PC 交互实现量产和升级。

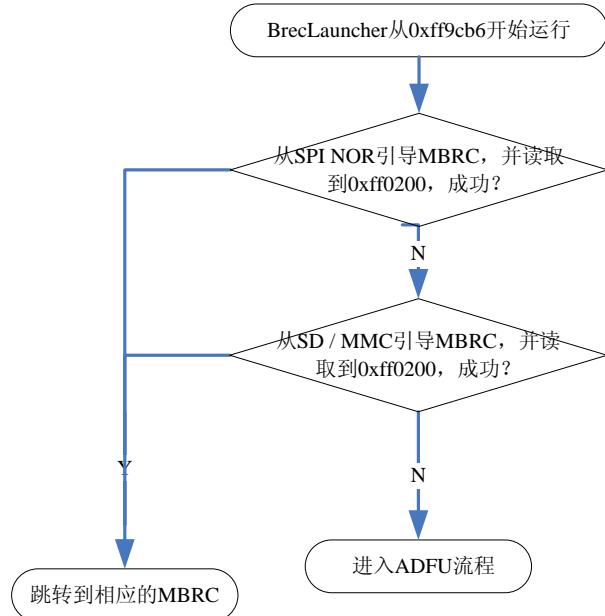


Figure 15-3 BrecLauncher 引导路程

### 15.3.3 NorFlash 引导

- 采用 3 线模式，则只有 CLK、SS 和 DIO，不区分 MISO 和 MOSI
- Randomizer 在引导阶段不会开启
- 使用 12M clk 传输
- 支持所有支持 03 读命令的 NorFlash

NorFLASH 引导流程图如下图所示：

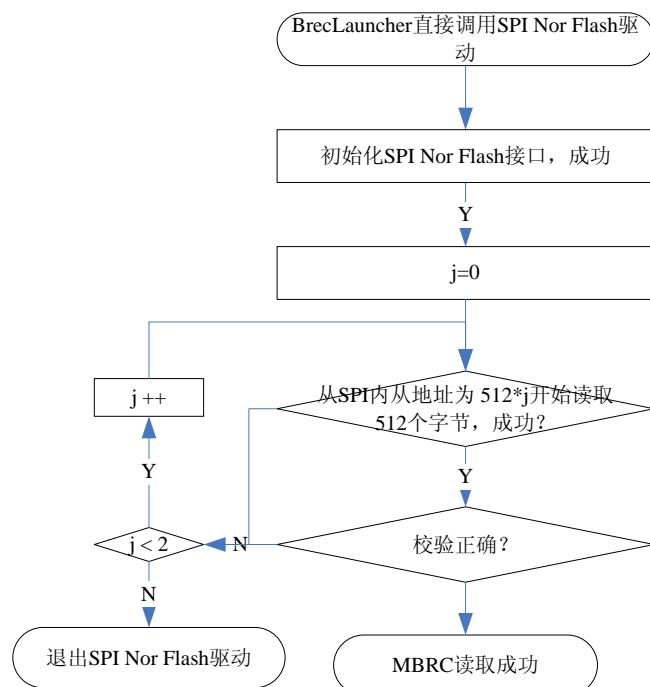


Figure 15-4 SPI Nor Flash 引导流程

MBRC 在 SPI Nor Flash 上可放置的位置包括偏移 0, 512, 1024 byte, 每个 MBRC 长度为 512 bytes, 引导流程如图 3.6。数据结构如表 3.3。

表 1 SDMMC 的 MBRC 结构

| Name     | Offset (byte) | Length (byte) | Descriptor                 |
|----------|---------------|---------------|----------------------------|
| Code     | 0             | 0x1fc         | BrecLauncher code segment  |
| Act flag | 0x1f8         | 4             | “Acts”                     |
| Flag     | 0x1fc         | 2             | Flag:0x55aa                |
| CheckSum | 0x1fe         | 2             | Check sum of MBRC + 0x1234 |

### 15.3.4 SDMMC 引导

- 支持 SD、MMC 和 MMC PLUS
- 支持从 EMMC boot 区引导
- 使用 12M clk 传输
- 只支持从 clk0 引导, 不 try clk1

上电后进入 BREC Launcher, 引导 NorFLASH 失败, 则进入 SD/MMC 的引导。SD/MMC 引导流程图如下图所示:

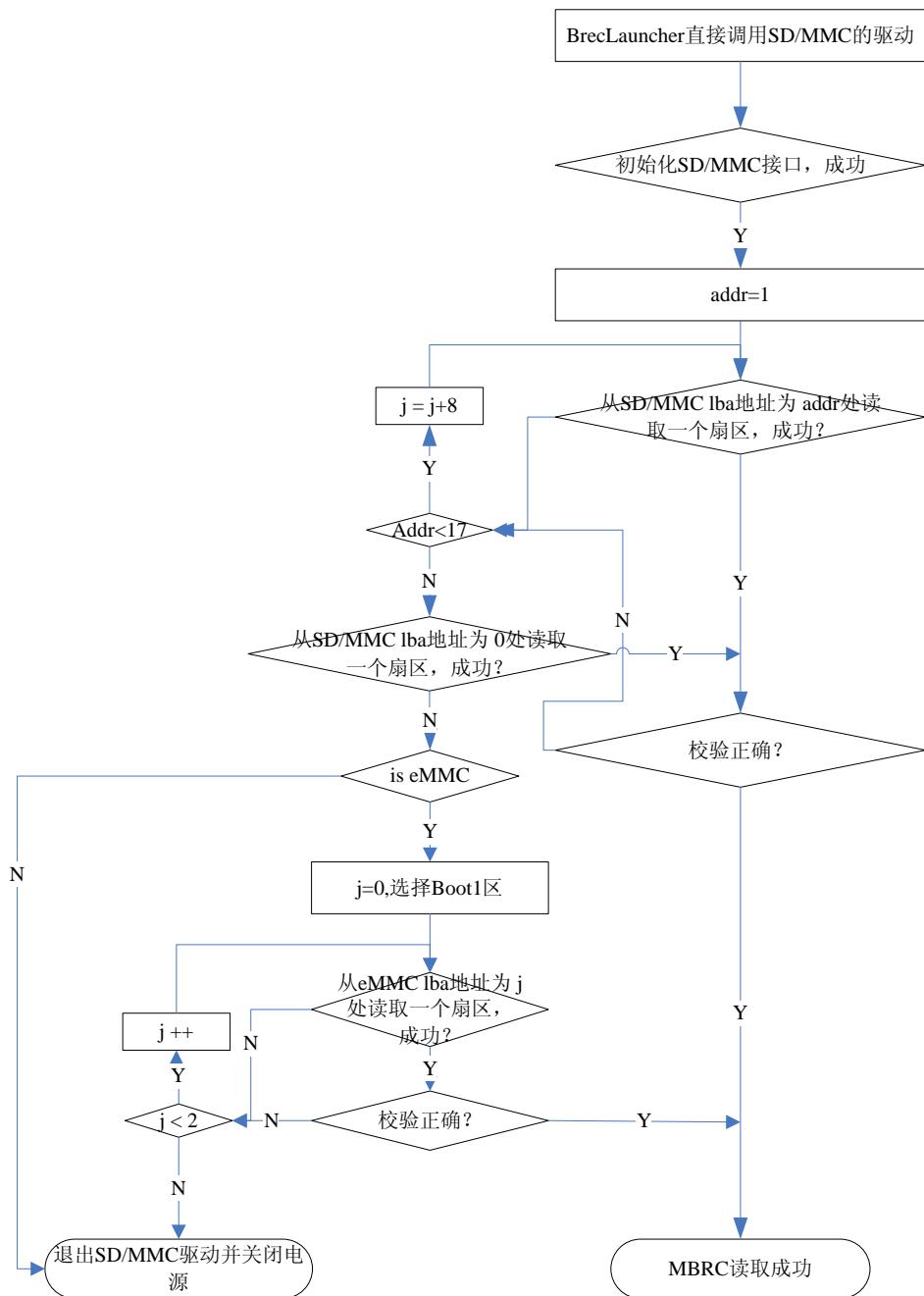


Figure 15-5 SD/MMC 引导流程

SD/MMC 的总体引导过程如上图，需要注意的是 MBRC 在 SD/MMC 上的放置区域，由于普通卡（eMMC 卡也可以当普通卡使用）可以直接被读卡器使用，其第 0 个 sector 通常会放置 MBR(供文件系统使用)，故第一份 MBRC 不能不在 0 扇区，我们将它放在第 1 扇区，第二份放在第 9 个扇区，第三份放置在第 17 个扇区，但假如有客户的卡只用于存放代码，所以可以将 mbrec 直接放到第 0 个扇区。如果是 eMMC 卡，MBRC 除了可以放在如上述的数据区外（user 区），还可以放在 Boot1 区，可以发命令设置选择 boot1 区，将数据读出，原像放在第 0 扇区，镜像放在第 1 扇区。其数据结构与 NAND Flash 一样，见下表。

表 2 SDMMC 的 MBRC 数据结构

| Name | Offset (byte) | Length (byte) | Descriptor |
|------|---------------|---------------|------------|
|------|---------------|---------------|------------|

|          |       |       |                            |
|----------|-------|-------|----------------------------|
| Code     | 0     | 0x1fc | BrecLauncher code segment  |
| Act flag | 0x1f8 | 4     | “Acts”                     |
| Flag     | 0x1fc | 2     | Flag:0x55aa                |
| CheckSum | 0x1fe | 2     | Check sum of MBRC + 0x1234 |

### 15.3.5 ADFULauncher 流程

- 将中断向量表切换到 0xff0000，不和系统重叠
- 将 MURAM 切换到 CODE 空间，后续 ADFUS 会搬到这里该地方运行
- 设置 URAM、FIR\_RDS\_RAM 和 PCMRAM 为 USBc1k

当存储介质引导均失败后，进入 ADFU Launcher 部分。

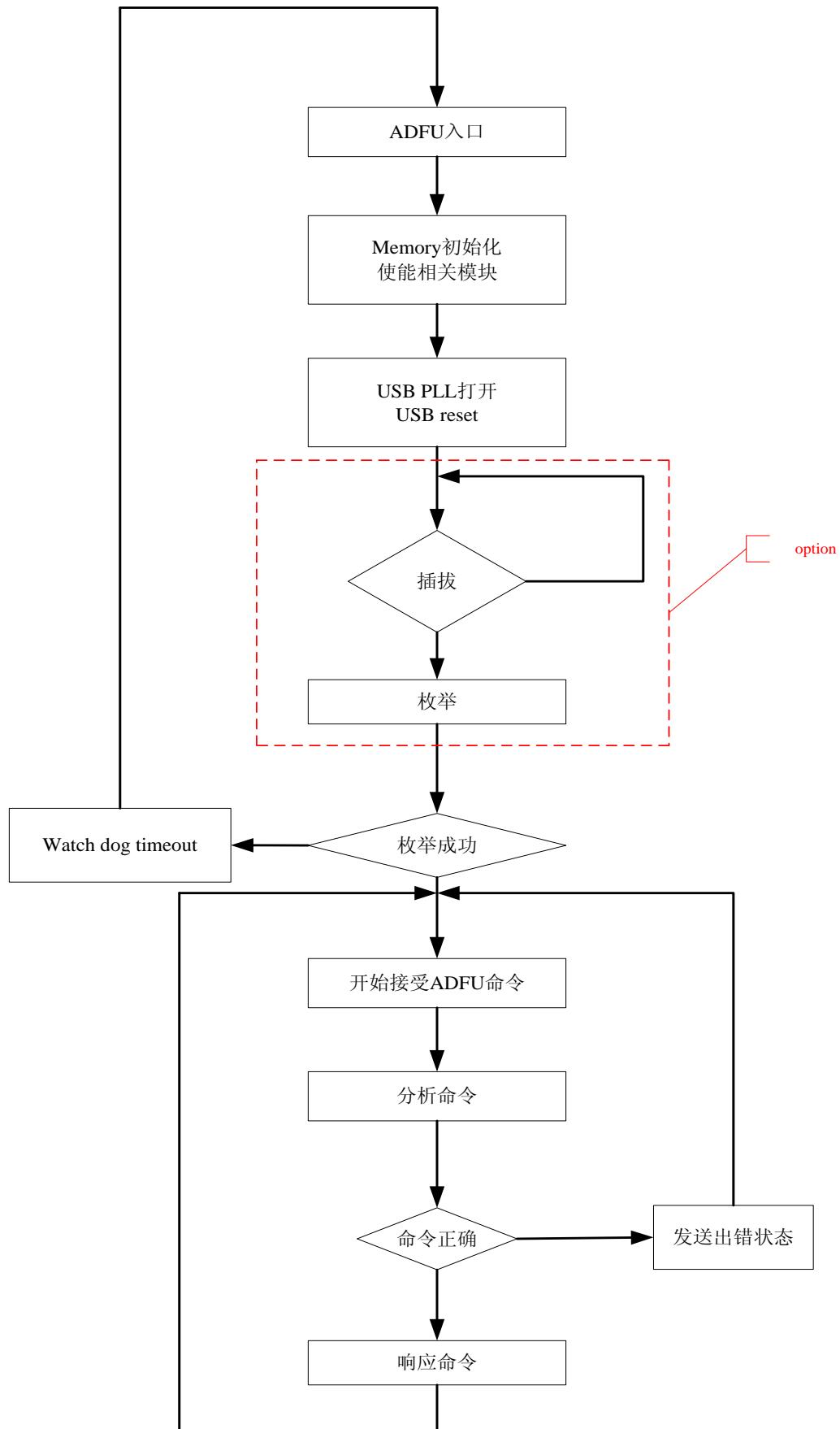


Figure 15-6 ADFU Launcher 引导流程图

## 15.4 BROM 中个物理模块的接口说明

### 15.4.1 SDMMC 物理驱动接口

#### 15.4.1.1 概述

SDMMC 卡驱动主要提供初始化和读函数，以便 BROM 进行 SDMMC 卡的检测和引导。具体构架图如下：

#### 15.4.1.2 文件名说明

sdmmcdrv\_boot.A51

#### 15.4.1.3 函数接口

表 3 SDMMC 卡驱动函数结构列表

| 物理层函数名称         | 功能           |
|-----------------|--------------|
| phyCard_init    | 卡初始化         |
| phyCard_exit    | 卡退出          |
| phyCard_read    | 卡物理读         |
| is_MMC_card     | 检查是否为 eMMC 卡 |
| eMMC_switch_cmd | 使能 eMMC 启动区  |

#### 15.4.1.4 函数接口说明

##### 1) phyCard\_init

Input: 无

Output: R11 0:卡初始化正确

1:没有卡或卡初始化失败

Description: 卡初始化

##### 2) phyCard\_exit

Input: 无

Output: 无

Description: 卡退出

##### 3) phyCard\_read

Input: WR6 卡读写数据结构指针: 4byte LBA 读写地址, 以 512 BYTE 为单位  
1byte 一次需要读写的扇区数目  
3byte 读写到内存中的地址

Output: R11 0: 数据读取成功; 1: 数据读取出错; 0xff: CRC 出错;

Description: 卡物理读

#### 4) is\_MMC\_card

Input: 无

Output: R11 0 is mmc card, 1 not mmc card

Description: 检查是否为 eMMC 卡

#### 5) eMMC\_switch\_cmd

Input:

Output: R11 0 is OK, 1 no rsp

Description: 使能 eMMC 启动区

## 15.4.2 SPI 物理驱动接口

### 15.4.2.1 概述

SPI 驱动主要提供初始化和按扇区读写函数, 以便 BROM 进行 SPI 的检测和引导及进行系统的初始化等工作。具体构架图如下:

### 15.4.2.2 文件名说明

Spinordrv\_boot.A51

### 15.4.2.3 函数接口

表 4 SPI Nor Flash 驱动函数结构列表

| 物理层函数名称      | 功能          |
|--------------|-------------|
| sSpinor_init | Nor 初始化     |
| sSpinor_exit | Nor 退出      |
| sSpinor_read | 对 SPI 进行读访问 |

### 15.4.2.4 函数接口说明

#### 1) sSpinor\_init

Input: 无

Output: v0, 0 初始化成功 1 初始化失败

Description: 初始化接口

#### 2) sSpinor\_exit

Input: 无

Output: 无

Description: 关闭接口

#### 3) sSpinor\_read

Input: WR6 卡读写数据结构指针: 4byte LBA 读写地址, 以 512 BYTE 为单位

1byte 一次需要读写的扇区数目

3byte 读写到内存中的地址

Output: R11 0: 数据读取成功; 1: 数据读取出错; 0xff: CRC 出错;

Description: 物理读

## 16 TEST MODE (李克伟、蔡瑞仁)

| 日期         | 版本    | 描述           | 修订人 |
|------------|-------|--------------|-----|
| 2012-12-12 | V2.02 | Initial (新增) | 李克伟 |

### 16.1 Features

There is a P\_TEST pin at GL5115, when this pin is high (VCC), GL5115 come to TEST MODE。然后通过 P\_RESETB PIN 决定进入哪个TEST MODE n, 只要P\_RESETB PIN 从高电平变换到低电平一次, 都会转换到下一个test mode, 并且循环转换。

在test模式下, 需要屏蔽POWEROK。因为在test模式下, 电源均为外灌, 防止上电等待时间过长。只需要等待CLK。

VCC/VDD在test mode下需要将电压设置寄存器设置为000, 及对应最低可调电压, 并且寄存器在test mode下可通过CPU进行配置以调节电压。

There are 3 test modes in GL5115:

- TEST MODE1: 外置 80251, 机台可以通过 pad 向所有的 memory 读写数据, 并进行 efuse 的烧写。
- TEST MODE2: reserved。
- TEST MODE3: function 模式, 在外灌 HOSC clock 的情况下基本等同于 normal mode。
- TEST MODE4: scan 模式, 所有的 memory 使能需要给关掉。增加 standby 模块的 scan。

其中 BIST 通过 test1 转 test3 来实现, 每个 memory 单独做 BIST controller。除 USB 外所有其他模式的 function 测试都是通过 test1 转 test3 来实现。

## 16.2 Function Description

### 16.2.1 TEST MODE 1

#### 16.2.1.1 TESE MODE 1 简介

1. 在 test mode 1 模式下, 80251 外置, 关闭 80251 的 clock 和 reset 80251;
2. SFR bus 拉到 PAD 上;
3. EFUSE 在此 mode 下进行烧写。因为 IC 内存在 4 组 EFUSE, 为节约资源, 4 组 EFUSE 使用同一条 pad bus, 通过 2 个 GPIO pad 来选择其中一组进行烧写和读操作。

#### 16.2.1.2 TEST MODE 1 所需 PAD

TEST MODE1 下需要的 pad 有:

| PAD instance name | TMODE1  | direction |
|-------------------|---------|-----------|
| P_TEST            | test    | I         |
| P_RESETB          | reset_n | I         |
| P_GPIOA0          | T1_A[0] | I         |
| P_GPIOA1          | T1_A[1] | I         |
| P_GPIOA2          | T1_A[2] | I         |
| P_GPIOA3          | T1_A[3] | I         |
| P_GPIOA4          | T1_A[4] | I         |
| P_GPIOA5          | T1_A[5] | I         |
| P_GPIOA6          | T1_A[6] | I         |
| P_GPIOA7          | T1_A[7] | I         |
| P_GPIOB0          | T1_A[8] | I         |
| P_GPIOB1          | T1_A[9] | I         |

|          |               |   |
|----------|---------------|---|
| P_GPIOB2 | T1_A[10]      | I |
| P_GPIOB3 | T1_A[11]      | I |
| P_GPIOB4 | T1_A[12]      | I |
| P_GPIOB5 | T1_A[13]      | I |
| P_GPIOB6 | T1_A[14]      | I |
| P_GPIOB7 | T1_A[15]      | I |
| P_GPIOC0 | T1_A[16]      | I |
| P_GPIOC1 | T1_A[17]      | I |
| P_GPIOC2 | T1_A[18]      | I |
| P_GPIOC3 | T1_A[19]      | I |
| P_GPIOC4 | T1_A[20]      | I |
| P_GPIOC5 | T1_A[21]      | I |
| P_GPIOC6 | T1_A[22]      | I |
| P_GPIOC7 | T1_A[23]      | I |
| P_GPIOD0 | T1_D[0]       | B |
| P_GPIOD1 | T1_D[1]       | B |
| P_GPIOD2 | T1_D[2]       | B |
| P_GPIOD3 | T1_D[3]       | B |
| P_GPIOD4 | T1_D[4]       | B |
| P_GPIOD5 | T1_D[5]       | B |
| P_GPIOD6 | T1_D[6]       | B |
| P_GPIOD7 | T1_D[7]       | B |
| P_GPIOE0 | T1_REB        | I |
| P_GPIOE1 | T1_WEB        | I |
| P_GPIOE2 | T1_DOE        | I |
| P_GPIOE3 | efuse_sel[0]  | I |
| P_GPIOE4 | efuse_sel[1]  | I |
| P_GPIOE5 | efuse_pa[0]   | I |
| P_GPIOE6 | efuse_pa[1]   | I |
| P_GPIOE7 | efuse_pa[2]   | I |
| P_GPIOF0 | efuse_enb     | I |
| P_GPIOF1 | efuse_read    | I |
| P_GPIOF2 | efuse_sdout   | O |
| P_GPIOF3 | efuse_program | I |
| P_GPIOF5 | efuse_test    | I |
| P_GPIOF6 |               |   |
| P_GPIOF7 |               |   |

注释：

efuse\_enb/efuse\_test 等输入口需要在上电后立刻配置为无效，在进行 EFUSE 测试时再配置为有效，否则可能会导致 efuse 误烧写。

## 16.2.2 TEST MODE 3

1. 在外灌 HOSC clock 情况下等同于 normal mode，所有功能模块可实现 spec 中描述的功能。该模式下直接外灌精准 24MHZ 来做为系统时钟。
2. 在 TEST MODE 3 模式下，需要所有的 PLL 可以正常工作，以方便提高测试效率；
3. RTC test clock 都需要切换到 HOSC；
4. BIST mode 只用 SFR bus 配置，不做 pad 直接控制；
5. 只做 function BIST，不做 speed BIST。

## 16.2.3 TEST MODE 4

TEST MODE4 设计为 scan mode。

### 16.2.3.1 SCAN 基本原理

Scan 测试的基本原理：在 scan 模式下，通过控制逻辑输入端口，依据从已知的被测试电路（DUT）的数字逻辑推断出的逻辑值来检测输出端口上的逻辑电平，如果输出端口的逻辑电平与推断完全相符，则测试指标达到工具算得的测试覆盖率，如果输出端口的逻辑电平与推断不相符，则该被测试电路存在生产缺陷。

### 16.2.3.2 TEST MODE 4 所需 PAD

TEST MODE4 下需要的 pad 有：

| PAD instance name | TMODE4     | direction |
|-------------------|------------|-----------|
| P_TEST            | test       | I         |
| P_RESETB          | reset_n    | I         |
| P_GPIOA0          | scan_in[0] | I         |
| P_GPIOA1          | scan_in[1] | I         |
| P_GPIOA2          | scan_in[2] | I         |
| P_GPIOA3          | scan_in[3] | I         |
| P_GPIOA4          | scan_in[4] | I         |
| P_GPIOA5          | scan_in[5] | I         |
| P_GPIOA6          | scan_in[6] | I         |
| P_GPIOA7          | scan_in[7] | I         |

|          |              |   |
|----------|--------------|---|
| P_GPIOB0 | scan_in[8]   | I |
| P_GPIOB1 | scan_in[9]   | I |
| P_GPIOB2 | scan_in[10]  | I |
| P_GPIOB3 | scan_in[11]  | I |
| P_GPIOB4 | scan_in[12]  | I |
| P_GPIOB5 | scan_in[13]  | I |
| P_GPIOB6 | scan_in[14]  | I |
| P_GPIOB7 | scan_in[15]  | I |
| P_GPIOC0 | scan_out[0]  | 0 |
| P_GPIOC1 | scan_out[1]  | 0 |
| P_GPIOC2 | scan_out[2]  | 0 |
| P_GPIOC3 | scan_out[3]  | 0 |
| P_GPIOC4 | scan_out[4]  | 0 |
| P_GPIOC5 | scan_out[5]  | 0 |
| P_GPIOC6 | scan_out[6]  | 0 |
| P_GPIOC7 | scan_out[7]  | 0 |
| P_GPIOD0 | scan_out[8]  | 0 |
| P_GPIOD1 | scan_out[9]  | 0 |
| P_GPIOD2 | scan_out[10] | 0 |
| P_GPIOD3 | scan_out[11] | 0 |
| P_GPIOD4 | scan_out[12] | 0 |
| P_GPIOD5 | scan_out[13] | 0 |
| P_GPIOD6 | scan_out[14] | 0 |
| P_GPIOD7 | scan_out[15] | 0 |
| P_GPIOE0 | test_se      | I |
| P_GPIOE1 | ate_clk      | I |
| P_GPIOE2 | comp_en      | I |

### 16.2.3.3 SCAN 测试过程

Scan 的测试过程：先控制 Scan\_en 置位为 1，控制寄存器的数据来源为扫描链的 Scan\_in 或前一寄存器的输出值，然后 Scan 时钟依次把 Pattern 的值置位到寄存器的输出端，扫描链上所有寄存器都置值完成后，Scan\_en 置位为 0，寄存器的数据来源为正常功能下的数据输入，Scan 时钟把各寄存器的前一级的寄存器输出端形成的逻辑值置位到输出端后，Scan\_en 置位为 1，控制寄存器的数据来源为扫描链中前一寄存器的输出值，Scan 时钟把各寄存器的值依次送达被测试电路的扫描链输出端 Scan\_out 与特定逻辑值比对。

SCAN 过程中数字送出的高频跳变信号对模拟模块会产生不良影响：可能会造成模拟模块产生大电流，或者会造成电压波动较大，或者会造成 clock 不受控制，因此模拟模块一些不需要的功能最好 disable。

### 16.2.3.4 SCAN 注意事项

1. Testmode4 只做 scan pad 切换，scan enable 需要 pad 灌入；
2. 在 scan 模式，模拟部分需要 hold 一些寄存器，由模拟整合负责人直接提供给数字整合负责人，无需在此具体描述。
3. Standby 电源域信号需要进行 scan 扫描

### 16.2.4 TEST MODE 优先级

TESTMODE > EJTAG > DEBUG > GPIO > MFP。

## 16.3 Module Description

| 模块                 | 测试方法                          | 备注 |
|--------------------|-------------------------------|----|
| 80251              | scan+function (test1 转 test3) |    |
| DMA 0/1/2/3/4      | scan+function (test1 转 test3) |    |
| USB                | scan                          |    |
| USB PHY            | BIST                          |    |
| LCD/LED            | scan+BIST (test1 转 test3)     |    |
| External Interface | scan+function (test1 转 test3) |    |
| ACC                | scan+function (test1 转 test3) |    |
| SRC                | scan+function (test1 转 test3) |    |
| UART               | scan+function (test1 转 test3) |    |
| IRC                | scan+function (test1 转 test3) |    |
| SPI                | scan+function (test1 转 test3) |    |
| CARD               | scan+function (test1 转 test3) |    |
| AUDIOIP            | scan+function (test1 转 test3) |    |
| DAC                | scan+function (test1 转 test3) |    |
| ADC                | scan+function (test1 转 test3) |    |
| GPIO               | scan+function (test1 转 test3) |    |
| MFP                | scan+function (间接测试)          |    |

|         |  |  |
|---------|--|--|
| MEMORY  | BIST+function (test1 转 test3)                            |  |
| PMU     | function (test1 转 test3)                                 |  |
| bandgap | function   |  |
| EFUSE   | function (test1 烧写并测试)                                   |  |
| RTC     | scan+function (test1 转 test3)                            |  |
| EJTAG   | function (test1 转 test3)                                 |  |
| SPDIF   | function (test1 转 test3)                                 |  |
| I2S     | function (test1 转 test3)                                 |  |
| CMU     | function (test1 转 test3)<br><br>HOSC、LOSC 部分在 wafer 上面不测 |  |
| PWM     | scan+function (test1 转 test3)                            |  |

### 16.3.1 CMU Module TEST

一、高校低的测试：

在 GPIO debug 模式中加入高校低的 debug 信号输出，以测试高校低电路是否工作正常。

二、对于 HOSC 部分在 wafer 上面不测，只在 Pakage 上测试。

三、MCUPLL 和 AudioPLL 通过间接方法进行测试。

### 16.3.2 AUDIO Module TEST

一、DAC 和 ADC analog 部分：不在 CP 测试模拟电路，需要时间很长，只测试数字部分的电路 function。

## 16.4 时序注意事项

在 test 拉高的情况下，特别需要注意 POWEROK 和 PMU\_ISOLATION 2 个信号。

TEST 不能控制 RTCVDDOK，否则会导致 normal 时提前发 powerok 等；

SYSON 实现 reset 的电路有 40ms debounce；

PreokA 为 VCC 域的信号；powerok 为 VDD 域的信号，在上电过程中需要小心处理。

## 16.5 Operation Manual

注意事项：

灌电顺序：RTCVDD——>VCC/VDD/BAT/DC5V 等

# 17 Electrical Characteristics

## 17.1 Absolute Maximum Ratings

Table10-1 Absolute Maximun Ratings

| Parameter           | Symbol                 | Min  | Max  | Unit |
|---------------------|------------------------|------|------|------|
| Ambient Temperature | Tamb                   | -10  | +70  | °C   |
| Storage temperature | Tstg                   | -55  | +150 | °C   |
| Supply voltage      | DC5V/ BAT              | -0.3 | 5.5  | V    |
|                     | VCC /UVCC              | -0.3 | 3.6  | V    |
|                     | VDD                    | -0.3 | 2.0  | V    |
| Input Voltage       | +3.3V IO               | -0.3 | 3.6  | V    |
|                     | +1.8V IO               | -0.3 | 2.0  | V    |
| ESD stress voltage  | Vesd(Human body model) | 2K   |      | V    |

**Note:**

- 1) Even if one of the above parameters exceeds the absolute maximum ratings momentarily, the quality of the product may be degraded. The absolute maximum ratings, therefore, specify the value exceeding, which the product may be physically damaged. Use the product well within these ratings.
- 2) All voltage values are with respect to GND
- 3) +3.3V IO/+1.8V IO are defined in the Pin-list.
- 4) **IC stability temperature must be less than 70 °C.**

## 17.2 DC Characteristics

Table10-2 DC Parameters for +3.3V IO PIN

| Parameter                 | Symbol | MIN. | MAX. | Unit | Condition                         |
|---------------------------|--------|------|------|------|-----------------------------------|
| Low-level input voltage   | VIL    |      | 0.8  | V    | VCC = 3.1V<br>Tamb = -10 to 70 °C |
| High-level input voltage  | VIH    | 2.0  |      | V    |                                   |
| Low-level output voltage  | VOL    |      | 0.4  | V    |                                   |
| High-level output voltage | VOH    | 2.4  |      | V    |                                   |

DC Parameter for EJTAG/SIRQ/I2C/I2S/SPI\_SS Pin

| Parameter                                | Symbol | MIN. | MAX. | Unit | Condition                       |
|--|--------|------|------|------|---------------------------------|
| Schmitt trigger positive-going threshold | VT+    |      | 1.9  | V    | VCC=3.1V<br>Tamb = -10 to 70 °C |
| Schmitt trigger                          | VT-    | 1.2  |      | V    |                                 |

|                             |  |  |  |  |  |
|-----------------------------|--|--|--|--|--|
| negative-going<br>threshold |  |  |  |  |  |
|-----------------------------|--|--|--|--|--|

Note: A schmitt trigger is used for reset pin

### 17.3 Recommended Power Supply

Table10-3 Recommended Power Supply

| Supply Voltage | Min | Typ | Max | Unit |
|----------------|-----|-----|-----|------|
| BAT (Li)       | 3.4 | 3.8 | 4.2 | V    |
| DC5V (No Bat)  | 4.5 | 5.0 | 5.2 | V    |
| VCC            | 2.8 | 3.1 | 3.4 | V    |
| UVCC           | 3.0 | 3.1 | 3.4 | V    |
| VDD            | 1.4 | 1.8 | 2.0 | V    |

**Note:**

- 1) According to different application, the VDD can be setting different voltage. For optimum CPU performance, the VDD should be higher than 1.8V; for reduced the power consumption, the VDD can supply with 1.4V.

### 17.4 Initialization Parameter

Table10-4 Initialization Paramter

| Parameter                           | Symbol   | Condition | MIN.  | MAX.  | Unit |
|-------------------------------------|----------|-----------|-------|-------|------|
| Data Sampling Time<br>(from RESET#) | $t_{ss}$ |           | —     | 61.04 | us   |
| Output delay time (from RESET#)     | $t_{OD}$ |           | 61.04 | —     | us   |

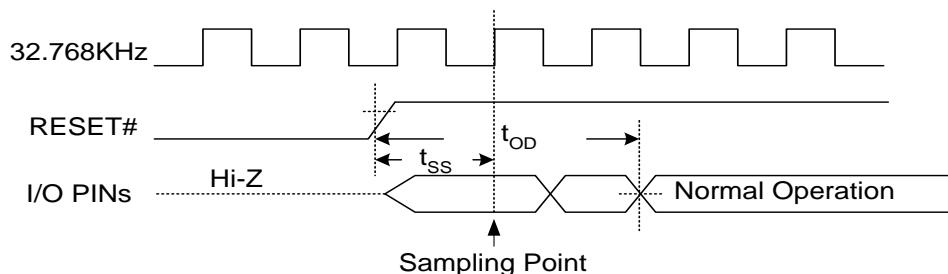


Figure10-1 Initialization Timing

### 17.5 USB DC Electrical Characteristics

Table10-5 Input Level for Low/Full speed

| Parameter | Symbol | Min | Max | Units |
|-----------|--------|-----|-----|-------|
|-----------|--------|-----|-----|-------|

|                                |     |     |     |   |
|--------------------------------|-----|-----|-----|---|
| HIGH                           | VIH | 1.5 |     | V |
| LOW                            | VIL |     | 0.8 | V |
| Differential Input Sensitivity | VDI | 0.2 |     | V |
| Differential Common Mode Range | VCM | 0.8 | 2.5 | V |

**Table10-6 Input Level for High speed**

| Parameter  | Symbol | Min | Max | Units |
|--|--------|-----|-----|-------|
| High-speed disconnect detection threshold (differential signal amplitude)    | VHSDSC | 525 | 625 | mV    |
| High-speed data signaling common mode voltage range (guideline for receiver) | VHSCM  | -50 | 500 | mV    |

**Table10-7 Output Level for Low/Full speed**

| Parameter                       | Symbol | Min | Max | Units |
|---------------------------------|--------|-----|-----|-------|
| HIGH                            | VOH    | 2.8 | 3.6 | V     |
| LOW                             | VOL    | 0.0 | 0.3 | V     |
| SE1                             | VOSE1  | 0.8 |     | V     |
| Output Signal Crossover Voltage | VCRS   | 1.3 | 2.0 | V     |

**Table10-8 Output Level for High Speed**

| Parameter                            | Symbol  | Min   | Max  | Units |
|--------------------------------------|---------|-------|------|-------|
| High-speed idle level                | VHSOI   | -10.0 | 10.0 | mV    |
| High-speed data signaling high       | VHSOH   | 360   | 440  | mV    |
| High-speed data signaling low        | VHSOL   | -10.0 | 10.0 | mV    |
| Chirp J level (differential voltage) | VCHIRPJ | 700   | 1100 | mV    |
| Chirp K level (differential voltage) | VCHIRPK | -900  | -500 | mV    |

**Table10-9 Terminations**

| Parameter  | Symbol | Min   | Max   | Units |
|--|--------|-------|-------|-------|
| Bus Pull-up Resistor on Upstream Facing Port                       | RPU    | 1.425 | 1.575 | kΩ    |
| Bus Pull-down Resistor on Downstream Facing Port                   | RPD    | 14.25 | 15.75 | kΩ    |
| Input impedance exclusive of pullup/pulldown (for low-/full-speed) | ZINP   | 300   |       | kΩ    |
| Termination voltage for upstream facing port pullup (RPU)          | VTERM  | 3.0   | 3.6   | V     |

## 17.6 SPI Interface Electrical Parameter

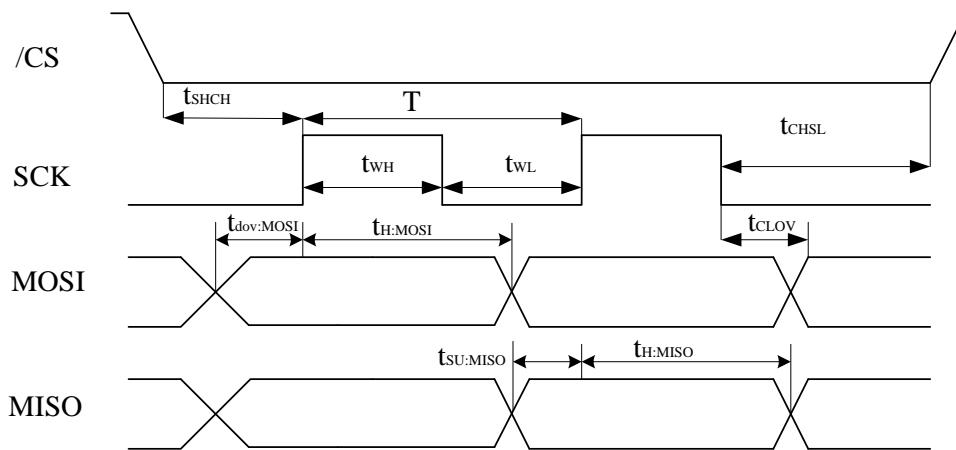


Figure10-2 SPI Timing

Table10-10 SPI Interface Electrical Parameter

| Parameter                     | Symbol     | MIN | MAX | Unit |
|-------------------------------|------------|-----|-----|------|
| SCK Clock                     | fclk       | -   | 60  | MHz  |
| SCK High time                 | tWH        | 5   | -   | ns   |
| SCK Low time                  | tWL        | 6   | -   | ns   |
| SCK rise time                 | tr         |     | 15  | ns   |
| SCK fall time                 | tf         |     | 15  | ns   |
| Data output valid(setup time) | tDOV: MOSI | 4   | -   | ns   |
| Data output hold              | tH: MOSI   | 8   | -   | ns   |
| Data in setup time            | tSU: MISO  | 2   | -   | ns   |
| Data in hold time             | tH: MISO   | 5   | -   | ns   |
| Clock low to output valid     | tCLOV      |     | 7   | ns   |

## 17.7 UART Timing Fault-tolerant Parameter

For UART's baud rate of 3 standards, the error rate measurements are as follows:

**Table10-11 UART Timing Fault-tolerant Parameter**

| <b>115200 Standard Baut Rate</b> |                                   |                                 |                   |
|----------------------------------|-----------------------------------|---------------------------------|-------------------|
| <b>Baut Rate</b>                 | <b>Theoretical Bit Width (us)</b> | <b>Practical Bit Width (us)</b> | <b>Error Rate</b> |
| 1200                             | 833                               | 832                             | 0.12%             |
| 2400                             | 416.67                            | 416                             | 0.16%             |
| 3600                             | 277.8                             | 277.5                           | 0.11%             |
| 4800                             | 208.33                            | 208                             | 0.15%             |
| 7200                             | 138.8                             | 138.8                           | 0                 |
| 9600                             | 104.17                            | 104                             | 0.16%             |
| 14400                            | 69.4                              | 69.3                            | 0.14%             |
| 19200                            | 52.08                             | 52                              | 0.15%             |
| 28800                            | 34.7                              | 34.65                           | 0.14%             |
| 38400                            | 26.04                             | 26                              | 0.15%             |
| 57600                            | 17.36                             | 17.35                           | 0.05%             |
| 115200                           | 8.68                              | 8.66                            | 0.02%             |

| <b>921600 Standard Baut Rate</b> |                                   |                                 |                   |
|----------------------------------|-----------------------------------|---------------------------------|-------------------|
| <b>Baut Rate</b>                 | <b>Theoretical Bit Width (us)</b> | <b>Practical Bit Width (us)</b> | <b>Error Rate</b> |
| 3600                             | 277.8                             | 277.5                           | 0.11%             |
| 4800                             | 208.33                            | 208                             | 0.16%             |
| 7200                             | 138.8                             | 138.8                           | 0                 |
| 9600                             | 104.17                            | 104                             | 0.16%             |
| 14400                            | 69.4                              | 69.3                            | 0.14%             |
| 19200                            | 52.08                             | 52                              | 0.15%             |
| 28800                            | 34.7                              | 34.7                            | 0                 |
| 38400                            | 26.04                             | 26                              | 0.15%             |
| 57600                            | 17.36                             | 17.32                           | 0.23%             |
| 115200                           | 8.68                              | 8.66                            | 0.23%             |
| 230400                           | 4.34                              | 4.33                            | 0.23%             |
| 460800                           | 2.17                              | 2.165                           | 0.23%             |
| 921600                           | 1.085                             | 1.084                           | 0.1%              |

| <b>1.5M Baut Rate</b> |                                   |                                 |                   |
|-----------------------|-----------------------------------|---------------------------------|-------------------|
| <b>Baut Rate</b>      | <b>Theoretical Bit Width (us)</b> | <b>Practical Bit Width (us)</b> | <b>Error Rate</b> |
| 7200                  | 138.8                             | 138.8                           | 0                 |

|         |        |       |       |
|---------|--------|-------|-------|
| 9600    | 104.17 | 104   | 0.16% |
| 14400   | 69.4   | 69.3  | 0.14% |
| 19200   | 52.08  | 52    | 0.15% |
| 28800   | 34.7   | 34.65 | 0.14% |
| 38400   | 26.04  | 26    | 0.15% |
| 57600   | 17.36  | 17.34 | 0.12% |
| 115200  | 8.68   | 8.68  | 0     |
| 750000  | 1.33   | 1.334 | 0.3%  |
| 1500000 | 0.667  | 0.666 | 0.15% |

| <b>6M Baut Rate</b> |                                   |                                 |                   |
|---------------------|-----------------------------------|---------------------------------|-------------------|
| <b>Baut Rate</b>    | <b>Theoretical Bit Width (us)</b> | <b>Practical Bit Width (us)</b> | <b>Error Rate</b> |
| 1000000             | 1                                 | 1                               | 0.00%             |
| 1200000             | 0.833                             | 0.835                           | 0.166%            |
| 1500000             | 0.667                             | 0.668                           | 0.133%            |
| 2000000             | 0.5                               | 0.5                             | 0.00%             |
| 3000000             | 0.333                             | 0.336                           | 0.66%             |
| 6000000             | 0.167                             | 0.167                           | 0.00%             |

## 17.8 IR Timing fault-tolerant parameter

Table10-12 IR Timing Fault-tolerant Parameter

|                                 | Min | TYP | Max | Unit |
|---------------------------------|-----|-----|-----|------|
| Infrared wave carrier frequency | 36  | 38  | 40  | Khz  |

## 17.9 SD Card Interface AC Parameter

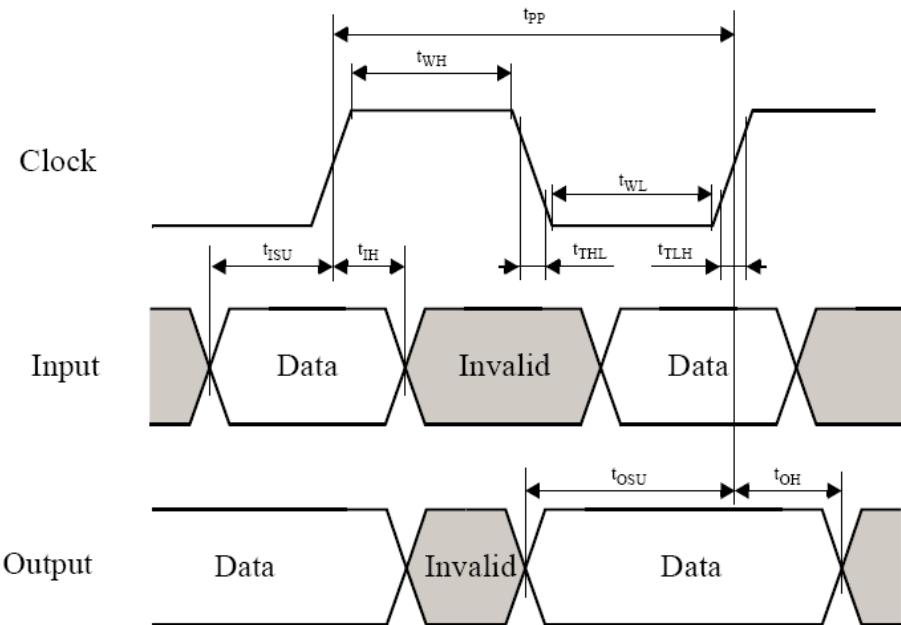


Figure 10-3 3.3V signaling default speed timing diagram

Table10-13 Threshold level for 3.3V voltage range

| Parameter           | Symbol | Min       | Max      | Unit | Remark            |
|---------------------|--------|-----------|----------|------|-------------------|
| Supply voltage      | VDD    | 2.7       | 3.6      | V    |                   |
| Output high voltage | VOH    | 0.625*VDD | VDD+0.3  | V    |                   |
| Output low voltage  | VOL    | VSS-0.3   | 0.25VDD  | V    |                   |
| Input high voltage  | VIH    | 0.75VDD   |          | V    | IOH=-2mA          |
| Input low voltage   | VIL    |           | 0.125VDD | V    | IOL=2mA           |
| Power up time       |        |           | 250      | ms   | From 0 to VDD min |

CL=CHost+Cbus+Ccard

Table10-14 3.3V signals timing Default speed mode

| Parameter   | symbol | Min | Max   | unit | Remark                      |
|---|--------|-----|-------|------|-----------------------------|
| Clock CLK   |        |     |       |      |                             |
| Clock frequency data Transfer Mode<br>(Push Pull) | fpp    | 0   | 25/26 | MHz  | CL<=30pF(tolerance +100KHz) |
| Clock frequency identification Mode(Open Drain)   | fOD    | 0   | 400   | KHz  | Tolerance:+20KHz            |
| Clock low time                                    | tWL    | 10  |       | ns   | Chost+Cbus<=30pf            |
| Clock low time                                    | tWH    | 10  |       | ns   | Chost+Cbus<=30pf            |
| Clock rise time                                   | tTLH   |     | 10    | ns   | Chost+Cbus<=30pf            |
| Clock fall time                                   | tTHL   |     | 10    | ns   | Chost+Cbus<=30pf            |

|                                  |      |   |  |             |
|----------------------------------|------|---|--|-------------|
|                                  |      |   |  |             |
| Inputs CMD DAT(reference to CLK) |      |   |  |             |
| Input setup time                 | tISU | 3 |  | ns CL<=30pF |
| Input hold time                  | tIH  | 3 |  | ns CL<=30pF |
| Output CMD DAT(reference to CLK) |      |   |  |             |
| Output setup time                | tOSU | 5 |  | ns CL<=30pF |
| Output hold time                 | tOH  | 5 |  | ns CL<=30pF |

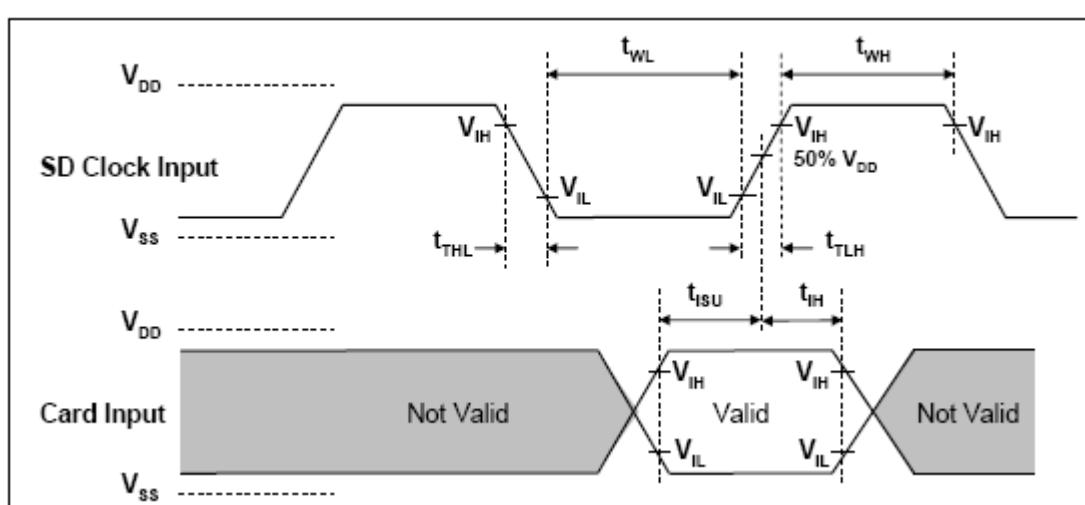


Figure 10-4 3.3V signaling high speed card input timing.(SDC output)

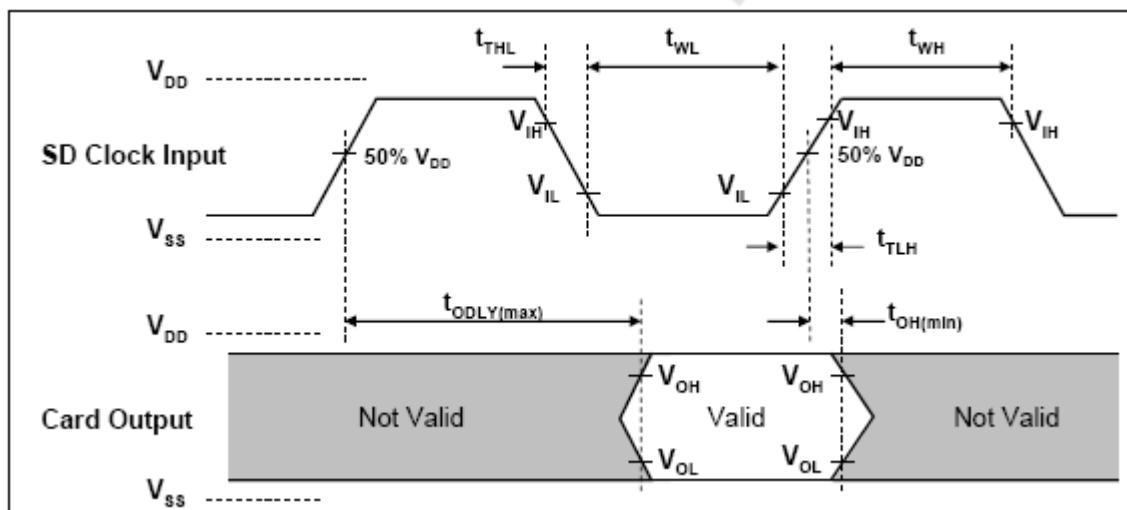


Figure10-5 3.3V signaling high speed card output timing.(SDC input)

Table10-15 3.3V signals timing high speed mode

| Parameter   | symbol | Min | Max   | unit | Remark                      |
|---|--------|-----|-------|------|-----------------------------|
| Clock CLK   |        |     |       |      |                             |
| Clock frequency data Transfer Mode<br>(Push Pull) | fpp    | 0   | 50/52 | MHz  | CL<=30pF(tolerance +100KHz) |

|   |      |         |     |     |                  |
|---|------|---------|-----|-----|------------------|
| Clock frequency identification Mode(Open Drain) | fOD  | 0       | 400 | KHz | Tolerance:+20KHz |
| Clock low time                                  | tWL  | 7       |     | ns  | Chost+Cbus<=30pf |
| Clock low time                                  | tWH  | 7       |     | ns  | Chost+Cbus<=30pf |
| Clock rise time                                 | tTLH |         | 3   | ns  | Chost+Cbus<=30pf |
| Clock fall time                                 | tTHL |         | 3   | ns  | Chost+Cbus<=30pf |
| Peak voltage on all lines                       |      | -0.3    | 0.3 | V   |                  |
| Inputs CMD DAT(reference to CLK)                |      |         |     |     |                  |
| Input setup time                                | tISU | 6-delay |     | ns  | CL<=30pF         |
| Input hold time                                 | tIH  | 2.5     |     | ns  | CL<=30pF         |
| Output CMD DAT(reference to CLK)                |      |         |     |     |                  |
| Output setup time                               | tOSU | 6       |     | ns  | CL<=30pF         |
| Output hold time                                | tOH  | 2       |     | ns  | CL<=30pF         |

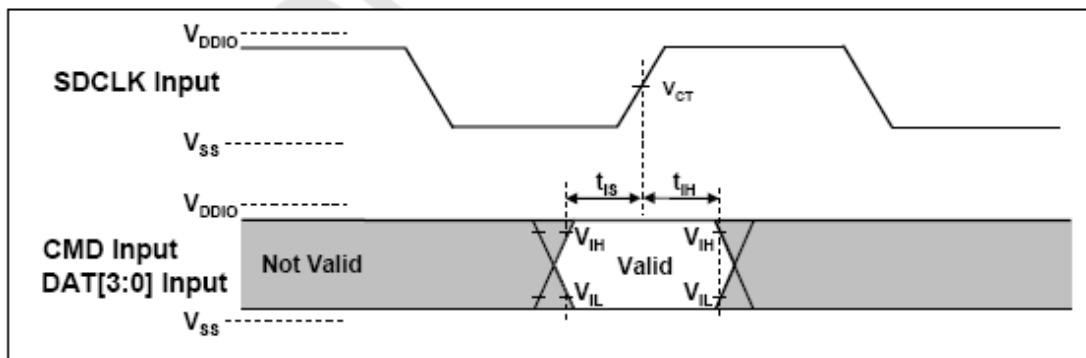


Figure 10-6 SRD50 card input timing. (SDC output)

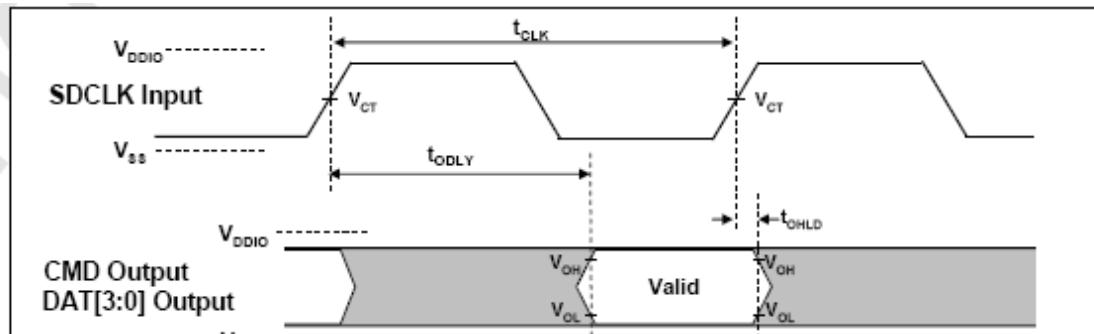


Figure 10-7 SRD50 card output timing. (SDC input)

Table 10-16 Threshold level for 1.8V voltage range

| Parameter           | Symbol | Min     | Max  | Unit | Remark   |
|---------------------|--------|---------|------|------|----------|
| Supply voltage      | VDD    | 2.7     | 3.6  | V    |          |
| Output high voltage | VOH    | 1.27    | 2.0  | V    |          |
| Output low voltage  | VOL    | Vss-0.3 | 0.58 | V    |          |
| Input high voltage  | VIH    | 1.4     |      | V    | IOH=-2mA |

|                   |     |  |      |   |         |
|-------------------|-----|--|------|---|---------|
| Input low voltage | VIL |  | 0.45 | V | IOL=2mA |
|-------------------|-----|--|------|---|---------|

CL=CHost+Cbus+Ccard

**Table10-17 1.8V signals timing SDR50 mode**

| Parameter   | symbol | Min | Max | unit | Remark                         |
|---|--------|-----|-----|------|--------------------------------|
| Clock CLK   |        |     |     |      |                                |
| Clock frequency data Transfer Mode<br>(Push Pull) | fpp    | 0   | 100 | MHz  | CL<=30pF(tolerance<br>+100KHz) |
| Clock duty  | tWL    | 30  | 70  | %    |                                |
| Clock low time                                    | tWH    |     |     | ns   |                                |
| Clock rise time                                   | tTLH   |     | 2   | ns   | Ccard=10pf                     |
| Clock fall time                                   | tTHL   |     | 2   | ns   | Ccard=10pf                     |
| Inputs CMD DAT(reference to CLK)                  |        |     |     |      |                                |
| Input setup time                                  | tISU   | 2.5 |     | ns   | CL=30pF,                       |
| Input hold time                                   | tIH    | 0.5 |     | ns   | CL=15pf                        |
| Output CMD DAT(reference to CLK)                  |        |     |     |      |                                |
| Output setup time                                 | tOSU   | 3   |     | ns   | Ccard=10pf,Vct=0.975V          |
| Output hold time                                  | tOH    | 0.8 |     | ns   | Ccard=5pf,Vct-0.975V           |

**Table10-18 Bus signal Line Load**

| Parameter                      | Symbol | Min | Normal | Max | Unit | Remark                  |
|--------------------------------|--------|-----|--------|-----|------|-------------------------|
| Pull up resistance for CMD     | Rcmd   | 4.7 | 20     | 50  | KOhm | To prevent bus floating |
| Pull up resistance for dat0-7  | Rdat   | 4.7 | 20     | 50  | KOhm | To prevent bus floating |
| Bus signal line capacitance    | CL     |     |        | 30  | pF   | Single card             |
| Signal card capacitance        | Ccard  |     |        | 7   | pF   |                         |
| Maximum signal line inductance |        |     |        | 16  | nH   |                         |

CL=CHost+Cbus+Ccard CHost+Cbus&lt;=30pf

## 17.10 Audio channel Characteristics Parameter

### 17.10.1.1 DAC+PA channel Characteristics Table

Testing environment: Audiprecision2722, VCC=3.1V, VDD=1.7V, 16.5ohm + 220uF;

**Table10-19 headphone Driver Characteristics**

| Characteristics      | Min | Typ | Max | Unit |
|----------------------|-----|-----|-----|------|
| Max Ampl (0dB Input) |     | 540 |     | mV   |
| Noise                |     | 9   |     | uV   |
| Max Power            |     | 18  |     | mW   |

|  |  |           |  |    |
|--|--|-----------|--|----|
| SNR  |  | 95.5      |  | dB |
| SNR(A-Weighting)                             |  | 97.5      |  | dB |
| Dynamic Range (-48dB Input)                  |  | 93.5      |  | dB |
| Dynamic Range (A-Weighting, -48 dBFS Input)  |  | 97        |  | dB |
| Total Harmonic Distortion + Noise(0dB Input) |  | -81       |  | dB |
| Inter channel Gain Mismatch(1KHz)            |  | $\pm 0.1$ |  | dB |



Figure10-8 Frequency Response Diagram of Headphone Driver

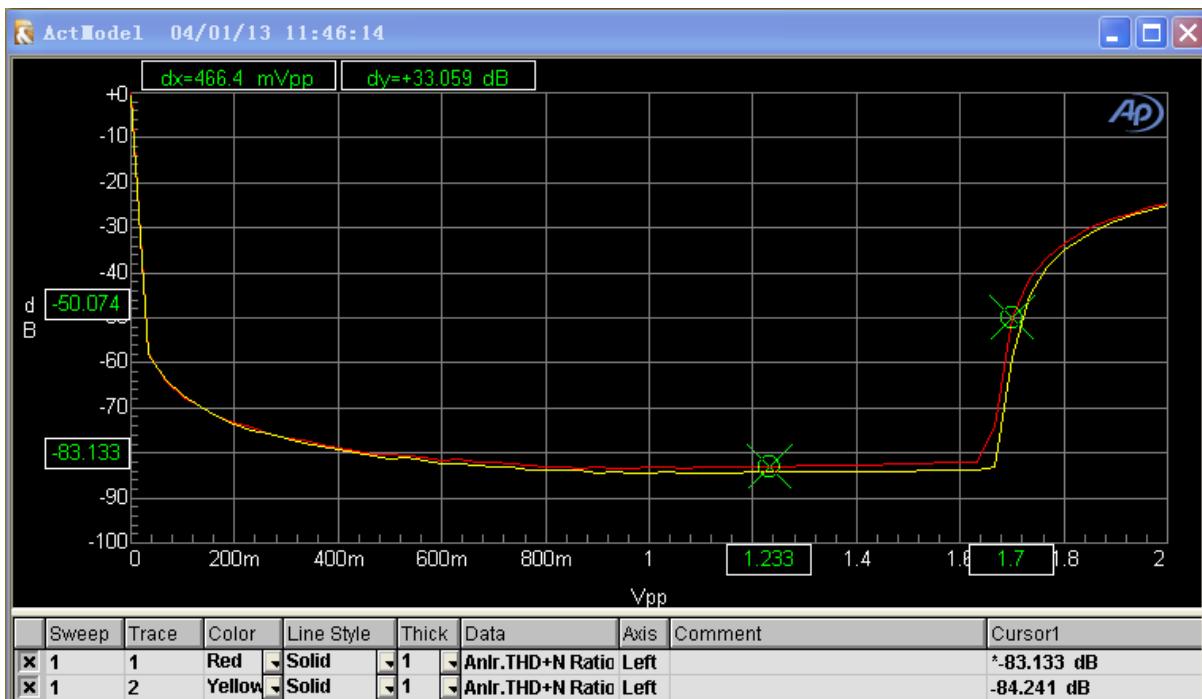


Figure10-9 THD + N Amplitude Diagram of Headphone Driver

### 17.10.1.2 ADC channel Characteristics Table

Testing environment: AudipPrecision2722, VCC=3.1V, VDD=1.7V;

Table10-20 headphone Driver Characteristics

| Characteristics                             | Min  | Typ         | Max | Unit |
|---|------|-------------|-----|------|
| Noise                                       |      | 39          |     | uV   |
| Dynamic Range (-40 dBFS Input)              |      | 82/83       |     | dB   |
| Dynamic Range (A-Weighting, -40 dBFS Input) |      | 83.5/84.5   |     | dB   |
| Total Harmonic Distortion+Noise             |      | -75         |     | dB   |
| Frequency Response 20-18KHz                 | -0.5 |             | 0.1 | dB   |
| Input Common Mode Voltage                   |      | 1.499/1.500 |     | V    |
| Full Scale Input Voltage                    |      | 2.8         |     | Vpp  |
| Crosstalk @ 1.6Vpp                          |      | -65         |     |      |

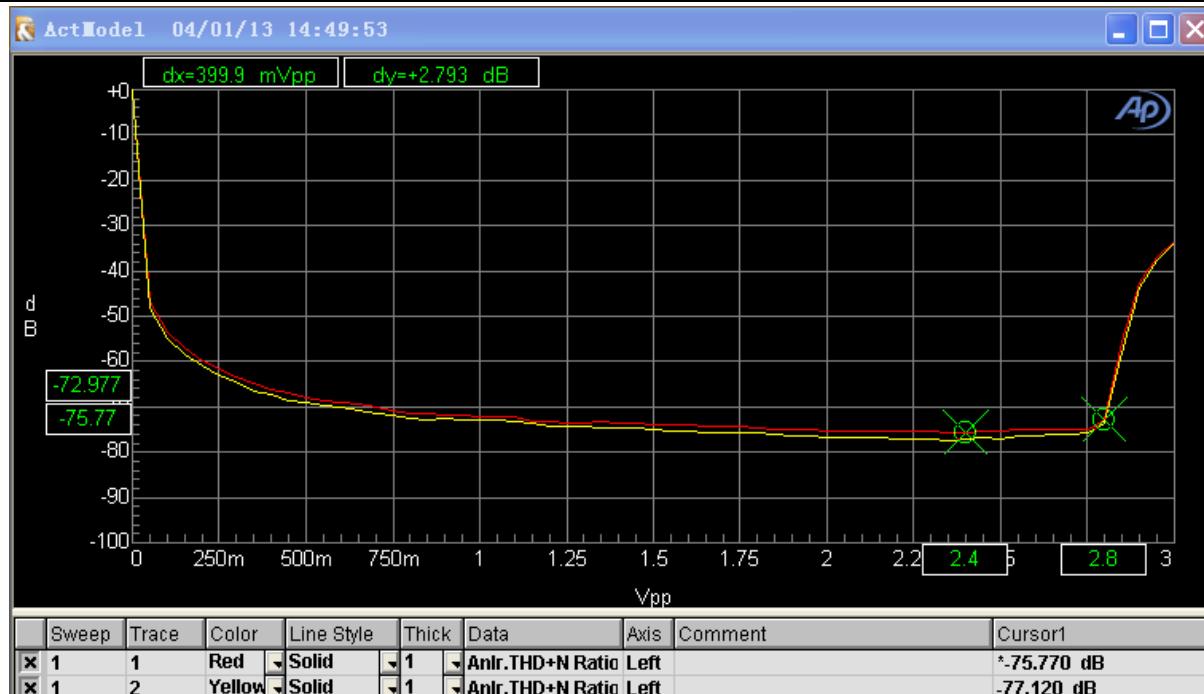


Figure10-10 THD + N Amplitude Diagram of Headphone Driver

## 17.11 GPIO Interface Parameter

Table10-21 GPIO Interface Parameter

| Parameter                    | Symbol       | Condition        | MIN.           | MAX. | Unit |
|------------------------------|--------------|------------------|----------------|------|------|
| Input level width            | $t_{GPIN}$   | Normal operation | $8/f_{mcuclk}$ |      | s    |
| GPIO output rise time(@50pf) | $t_{GPRISE}$ |                  | 5              | 20   | ns   |
| GPIO output fall time(@50pf) | $t_{GPFALL}$ |                  | 5              | 20   | ns   |
| Output level width           | $t_{GPOUT}$  |                  | $8/f_{mcuclk}$ |      | s    |

Notes:

$f_{MCUCLK}$  is the frequency that MCU is running upon.

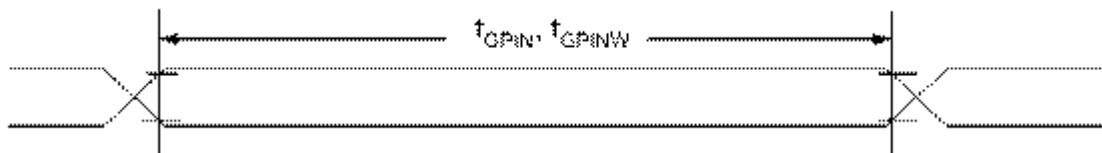


Figure10-11 Input Level Width



Figure10-12 Output Rise/Fall Time

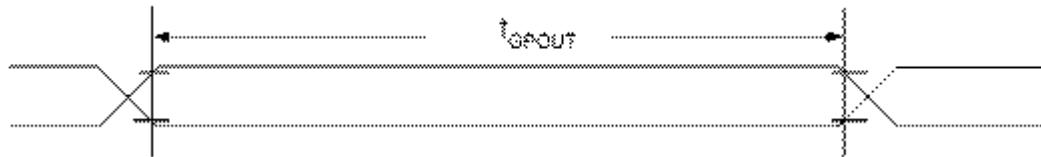


Figure10-11 Output Level Width

## 17.12 LCM Driver Parameter

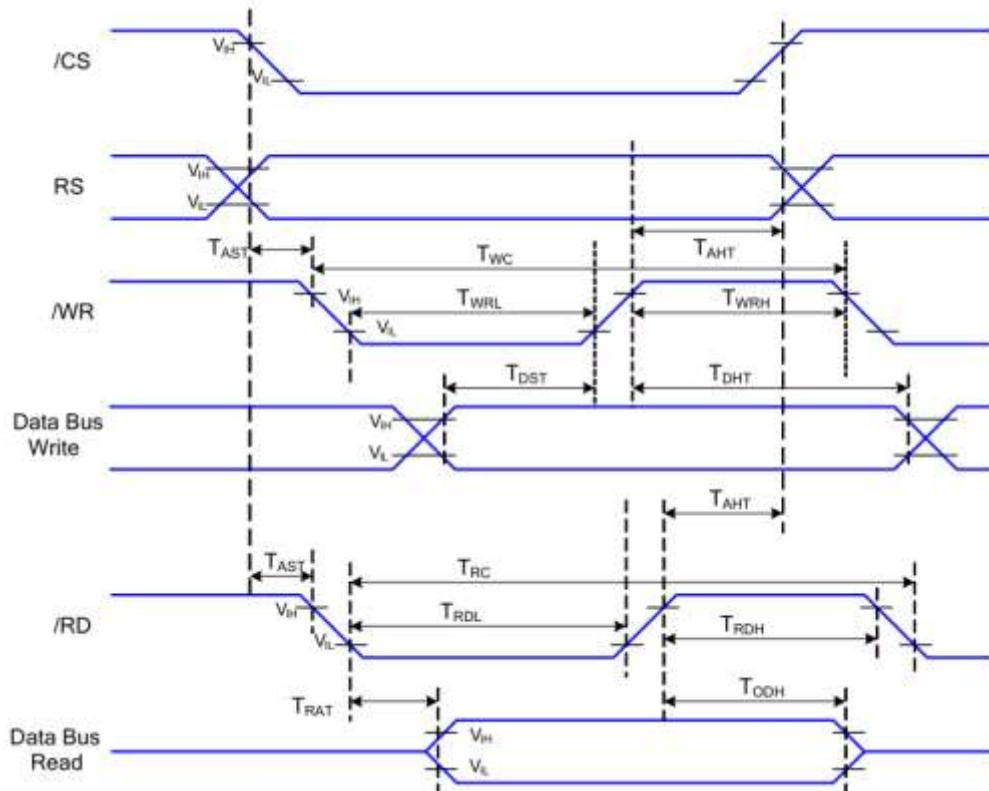


Figure10-12 LCM Interface Timing

Table10-22 LCM Driver Parameter

| Signal | Symbol           | Parameter                     | Condition  | Min | Max | Unit |
|--------|------------------|-------------------------------|------------|-----|-----|------|
| RS     | T <sub>AST</sub> | Address Setup Time            | HOSC=24MHZ | 50  |     | ns   |
|        | T <sub>AHT</sub> | Address Hold Time(Write/Read) | HOSC=24MHZ | 50  |     | ns   |

|     |                  |                            |            |     |      |    |
|-----|------------------|----------------------------|------------|-----|------|----|
| /WR | T <sub>WC</sub>  | Write Cycle                | HOSC=24MHZ | 100 | 1000 | ns |
|     | T <sub>WRH</sub> | Control Pulse "H" Duration | HOSC=24MHZ | 50  | 500  | ns |
|     | T <sub>WRL</sub> | Control Pulse "L" Duration | HOSC=24MHZ | 50  | 500  | ns |
| /RD | T <sub>RC</sub>  | Read Cycle                 | HOSC=24MHZ | 100 | 1000 | ns |
|     | T <sub>RDH</sub> | Control Pulse "H" Duration | HOSC=24MHZ | 50  | 500  | ns |
|     | T <sub>RDL</sub> | Control Pulse "L" Duration | HOSC=24MHZ | 50  | 500  | ns |
| DB  | T <sub>DST</sub> | Data Setup Time            | HOSC=24MHZ | 50  | 500  | ns |
|     | T <sub>DHT</sub> | Data Hold Time             | HOSC=24MHZ | 50  | 500  | ns |
|     | T <sub>RAT</sub> | Read Access Time           | HOSC=24MHZ |     | 200  | ns |
|     | T <sub>ODH</sub> | Output Disable Time        | HOSC=24MHZ | 50  |      | ns |

## 17.13 External System Bus Parameter

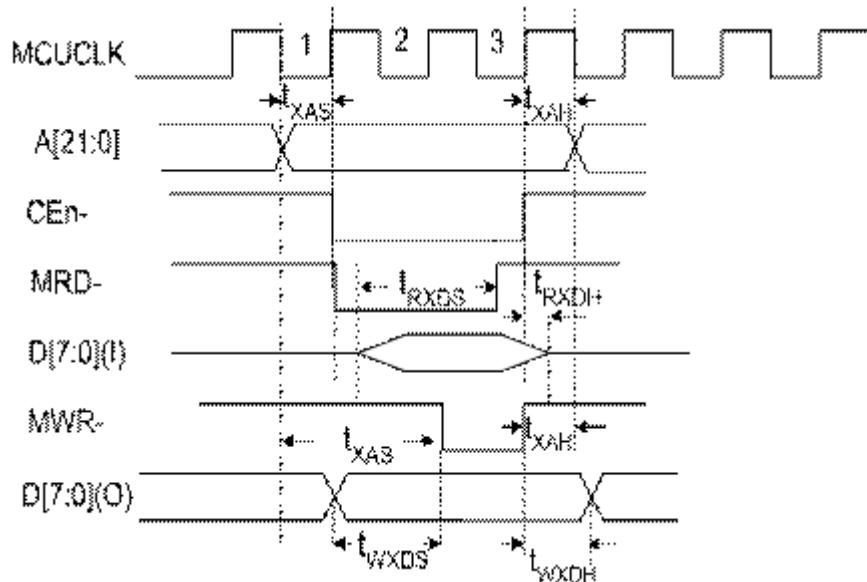


Figure10-13 External System Bus Parameter

Table9-23 External System Bus Parameter

| Parameter  | Symbol            | Condition    | MIN. | MAX. | Unit |
|--|-------------------|--------------|------|------|------|
| Address setup time (to command signal) <sup>Note 1, 2</sup>  | t <sub>XAS</sub>  | Memory Read  | 10   |      | ns   |
|  | t <sub>XAS</sub>  | Memory Write | 10   |      | ns   |
| Address hold time (from command signal) <sup>Note 1, 2</sup> | t <sub>XAH</sub>  |              | 5    |      | ns   |
| Data output setup time (to command signal) <sup>Note 1</sup> | t <sub>WXDS</sub> |              | 20   |      | ns   |

|  |                   |  |    |  |    |
|--|-------------------|--|----|--|----|
| Data output hold time(from command signal) <sup>Note 1</sup> | t <sub>WXDH</sub> |  | 10 |  | ns |
| Data input setup time (to command signal) <sup>Note 1</sup>  | t <sub>RXDS</sub> |  | 20 |  | ns |
| Data input hold time (from command signal) <sup>Note 1</sup> | t <sub>RXDH</sub> |  | 10 |  | ns |

**Notes:**

1. MRD#, MWR# is called the command signals for the External System Bus Interface.

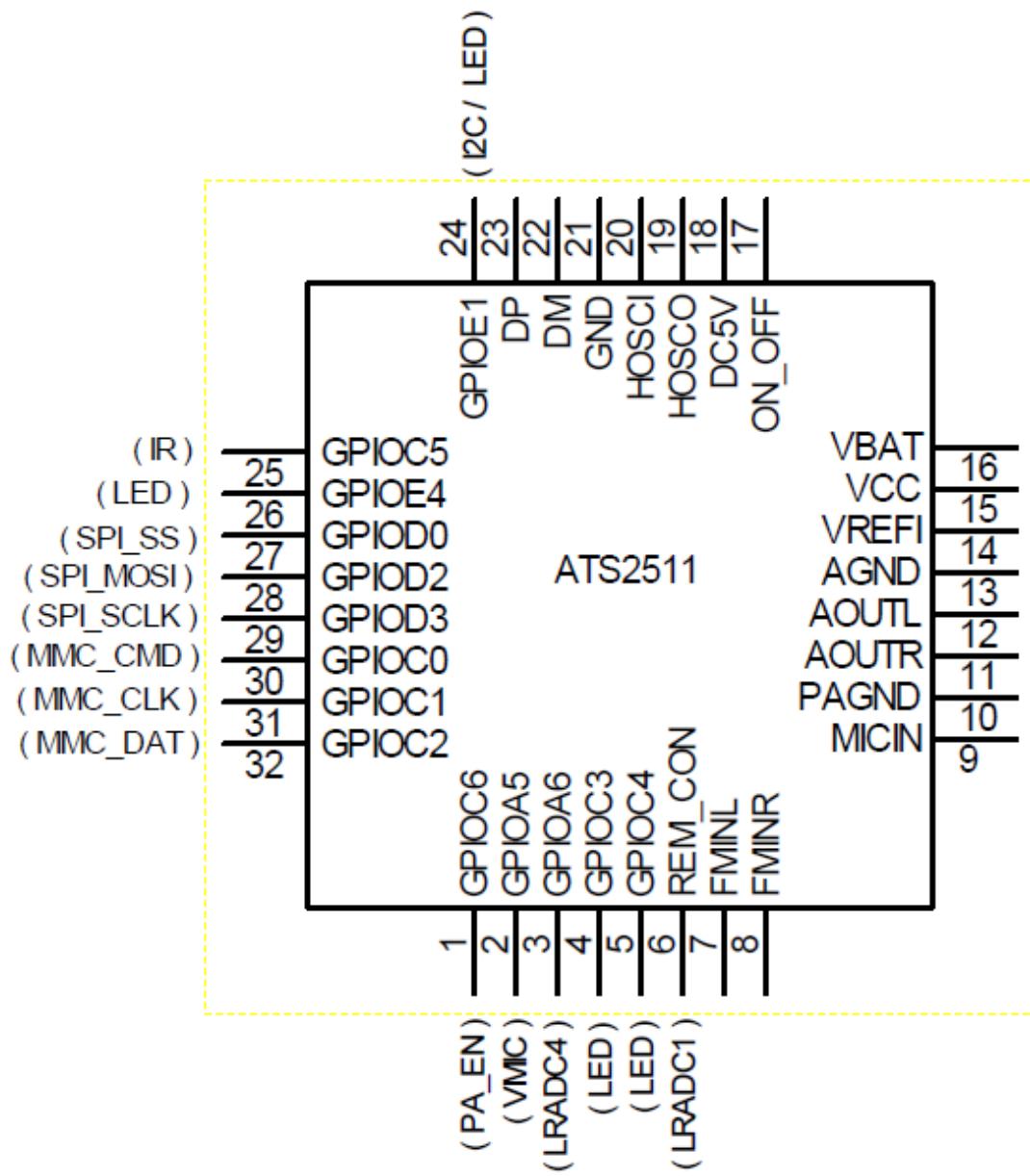
$$2. T \text{ (ns)} = 1/f_{MCUCLK}$$

## 18 Pin Assignment

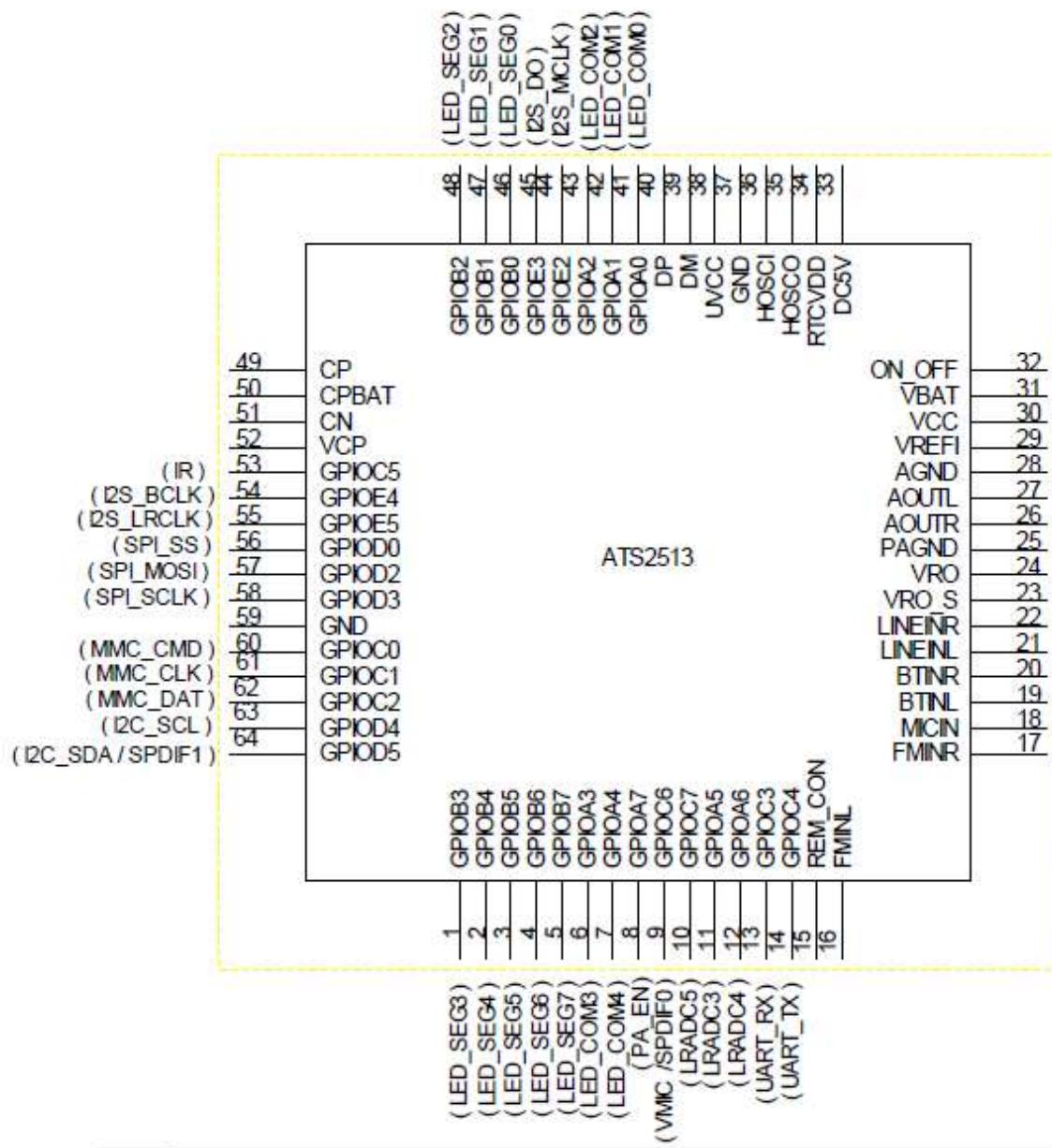
详见《GL5115\_Pin\_AssignmentV2.2(强制).xls》

## 19 Package and Drawing

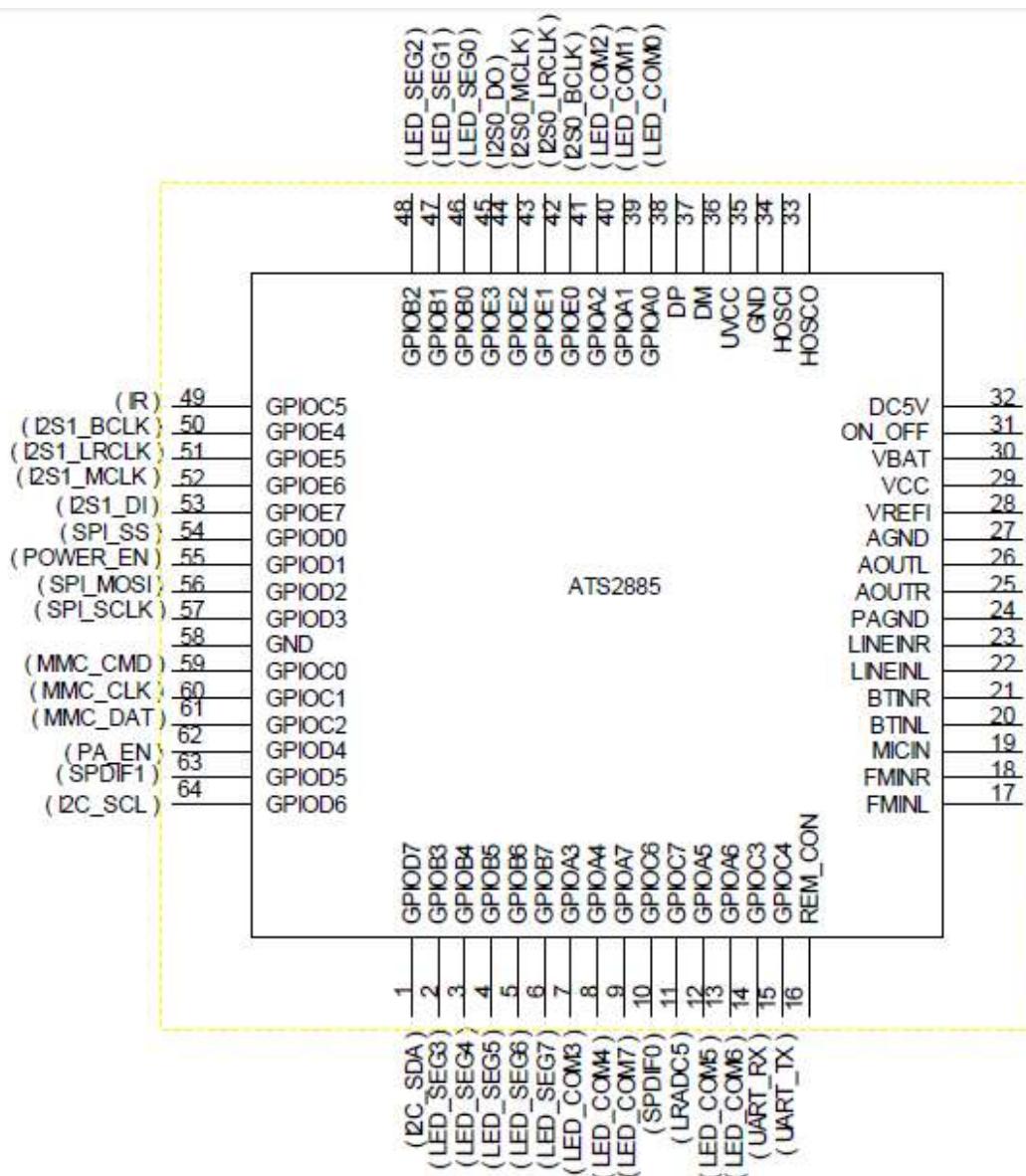
### 19.1 ATS2511(无屏 boombox, LQFP-32)



## 19.2 ATS2513(点阵屏/数码管 boombox, LQFP-64)



### 19.3 ATS2885(soundbar, LQFP-64)



## 19.4 ESC5115(for IC verify, LQFP-128)

