

AMT630A

Video Display Controller (Product Specification)

Version 1.1

2014.10

Revision Record:

Date	Revision	Modification Description
2014-09	V1.0	Initial Version
2014-10	V1.1	Add colormatrix registers

CONFIDENTIAL

ARKMICRO TECHNOLOGIES INC. reserves the right to change or modify the information contained herein without notice. It is the customer's responsibility to obtain the most recent revision of the document. **ARKMICRO** makes no warranty for the use of its products and bears no responsibility for any errors or omissions that may appear in this document.

Content

1. GENERAL DESCRIPTION	3
2. FEATURES	4
3. APPLICATION FIELD	5
4. BLOCK DIAGRAM	5
5. PIN DIAGRAM	6
6. Register Descriptions.....	9
6.1 Global Register(I2C Address: 0xB0 MCU Address: 0xFDXX).....	9
6.2 Tcon Register(I2C Address: 0xB8 MCU Address: 0xFCXX).....	16
6.3 Decoder Register(I2C Address: 0xB2 MCU Address: 0xFEXX)	27
6.4 Video Process Register(I2C Address: 0xB4 MCU Address: 0xFFXX).....	33
6.5 GAMMA Register(I2C Address: 0xB4 MCU Address: 0xFFXX).....	38
6.6 RCRT Register(MCU Address: 0xFFXX MCU access only).....	41
6.7 Scaler Register(I2C Address: 0xB8 MCU Address: 0xFCXX).....	49
6.7 OSD 设备地址 B6, mcu 地址 0XFBXX.....	60
6.9 SPI Register(I2C Address: 0xB0 MCU Address: 0xFDXX).....	71
6.10 12bit ADC Register(I2C Address: 0xB0 MCU Address: 0xFDXX).....	73
7. AMT630A MCU Peripheral	79
7.1 SPI FLASH CONTROLER	79
7.2 MCU OPERATION:	84
7.3 TIMER	87
7.4 WATCHDOG	88
7.5 GPIO	89
8. ELECTRICAL SPECIFICATIONS.....	90

1. GENERAL DESCRIPTION

AMT630A is a video decoder and Digital TFT-LCD Panel Control SOC. The AMT630A accept analog NTSC / PAL /

SECAM CVBS from TV tuner, DVD, or VCR sources, including weak and distorted signals. Automatic gain control (AGC) and 9-bit 1-channel A/D converters provide high resolution video quantization, with automatic video source and mode detection, user can easily switch and adjust variety of signal source. Multiple internal adaptive PLL precisely extract pixel clock from video source and perform sharp-and-keen color demodulation. Build-in line-buffer supports adaptive 2-D comb-filter, 2-D sharpening, and synchronization stable in a condense manner. Build-in On Screen Display(OSD) module. The output format of AMT630A directly supports variety of TFT-LCD modules. AMT630A is one excellent efficiency for a low-cost Price and small-area PCB solution.

2. FEATURES

VIDEO DECODER

- ◆ Composite video signal(CVBS); Multiple standards supported: NTSC and NTSC-Japan; PAL (B, D, G, H, I, M, N, etc.); SECAM;
- ◆ 3 Analog Inputs: 3*CVBS Inputs
- ◆ Digital AGC,ACC
- ◆ 9-Bit 1-Channel A/D Converters with Fixed Sampling Clock
- ◆ Only One Crystal (27 MHz) required for All Standards
- ◆ Internal PLL to Generate Video Clock
- ◆ Adaptive 2-D Comb Filter for Luminance and Chrominance Separation
- ◆ Precise Chrominance Demodulation
- ◆ Internal Buffers for Video Stability Control
- ◆ Video Noise Reduction

VIDEO ENHANCE

- ◆ Frequency Directive Sharpening
- ◆ Brightness, Contrast, Color, and Tint Adjustments
- ◆ Black-Level Extension and White-Level Extension
- ◆ Digital Chrominance Transient Improvement(DCTI) and Digital Luminance Transient Improvement(DLTI)
- ◆ 3 – channel Gamma curve adjustment
- ◆ Green level enhance
- ◆ 3x3 color martrix
- ◆ Peaking
- ◆ Noise Reduction
- ◆ 9 Tap FIR filter

SCALING ENGINE

- ◆ Supports digaital panel with the resolution of 480x240, 600x480, 520x288, 800x600, 1024x768 and more.
- ◆ Supports horizontal panorama scaling.
- ◆ Supports vertical panorama scaling.

OSD BLOCK

- ◆ Built-in 512-Character Font ROM (Including Special Font Characters)
- ◆ Dynamic OSD font RAM-----4096x16 bytes
- ◆ Support Font Size upto 24x32

- ◆ 16 colors palette ,support 5 osd window
- ◆ Support 16 color bitmap
- ◆ Blending with OSD Content and Video
- ◆ Blinking and Highlight Function

INTERFACE

- ◆ Digital TTL/TCON panel
- ◆ 8-Bit/10-Bit CCIR 656 Digital Video Output Format Support
- ◆ I²C-BUS interface (slave mode)

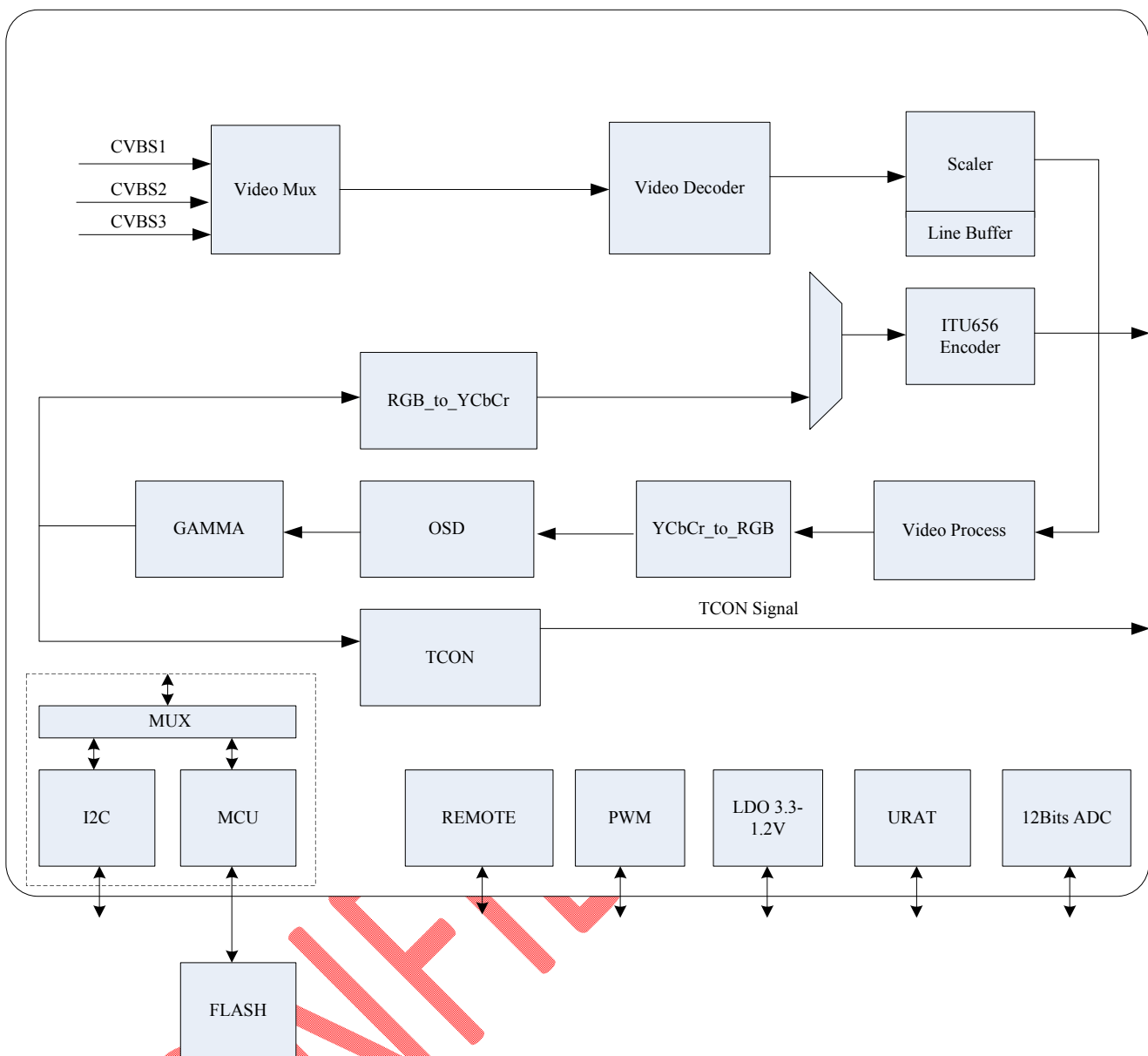
PERIPHERAL

- ◆ Build-in MCU & SPI Flash interface.
- ◆ Support SPI FLASH on line program
- ◆ Build-in 12Bit ADC
- ◆ Build-in Display PLL
- ◆ Build-in LDO for 1.2v core power
- ◆ 4 sets of Built-In PWM circuit: 4*16 bit
- ◆ 3.3V power supply only
- ◆ LQFP 64pin Package

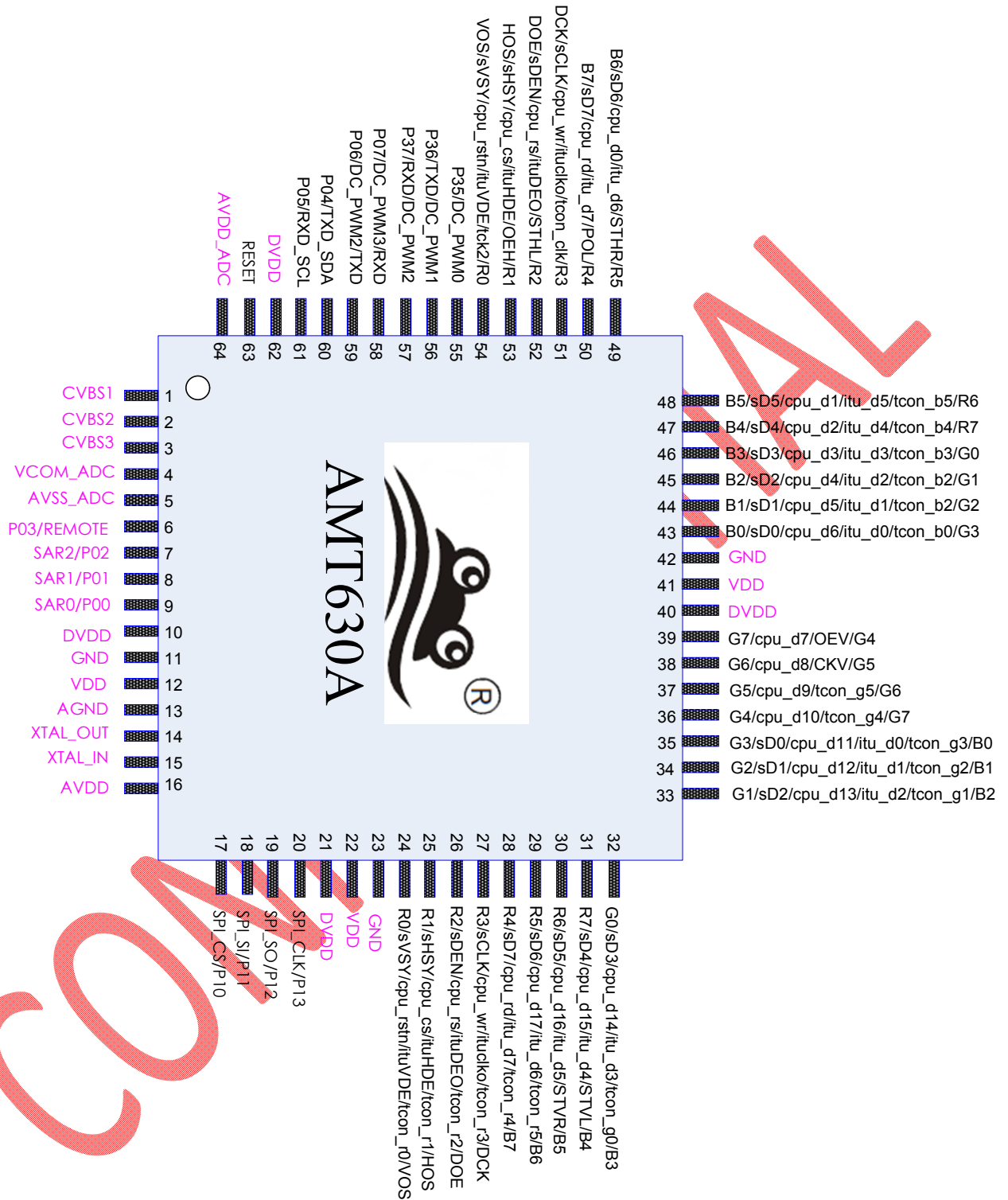
3. APPLICATION FIELD

- ◆ Portable DVD
- ◆ Small to medium sized LCD TV
- ◆ Car entertainment
- ◆ Digital photo frame
- ◆ Other application using analog panel as the display unit

4. BLOCK DIAGRAM



5. PIN DIAGRAM



AMT630A PAD Definition

PAD NAME	TYPE	64D	复用功能
CVBS1	A	1	CVBS1
CVBS2	A	2	CVBS2
CVBS3	A	3	CVBS3
VCOM_ADC	A	4	VCOM_ADC
AVSS_ADC	P	5	AVSS_ADC
REMOTE	D	6	P03/REMOTE
SAR2	A	7	SAR2/P02
SAR1	A	8	SAR1/P01
SAR0	A	9	SAR0/P00
DVDD_0	P	10	DVDD
VDD_0	P	11	VDD
VSS_0	P	12	GND
AGND	P	13	AGND(AVSS33_ANA)
XTAL_OUT	A	14	XTAL_OUT
XTAL_IN	A	15	XTAL_IN
AVDD	P	16	AVDD
pad17	D	17	SPI_CS/P10
pad18	D	18	SPI_SI/P11
pad19	D	19	SPI_SO/P12
pad20	D	20	SPI_CLK/P13
DVDD_1	P	21	DVDD
VDD_1	P	22	VDD
VSS_1	P	23	GND
pad24	D	24	R0/VOS/tcon_r0/ituVDE/sVSY/cpu_rstn/P14
pad25	D	25	R1/HOS/tcon_r1/ituHDE/sHSY/cpu_cs/P15
pad26	D	26	R2/DOE/tcon_r2/ituDEO/sDEN/cpu_rs/P16
pad27	D	27	R3/DCK/tcon_r3/ituclko/sCLK/cpu_wr/P17
pad28	D	28	R4/B7/tcon_r4/itu_d7/sD7/cpu_rd/P20
pad29	D	29	R5/B6/tcon_r5/itu_d6/sD6/cpu_d17/P21
pad30	D	30	R6/B5/STVR/itu_d5/sD5/cpu_d16/P22
pad31	D	31	R7/B4/STVL/itu_d4/sD4/cpu_d15/GPIO0
pad32	D	32	G0/B3/tcon_g0/itu_d3/sD3/cpu_d14/GPIO1
pad33	D	33	G1/B2/tcon_g1/itu_d2/sD2/cpu_d13/GPIO2
pad34	D	34	G2/B1/tcon_g2/itu_d1/sD1/cpu_d12/GPIO3
pad35	D	35	G3/B0/tcon_g3/itu_d0/sD0/cpu_d11/GPIO4
pad36	D	36	G4/G7/tcon_g4/cpu_d10/P23
pad37	D	37	G5/G6/tcon_g5/cpu_d9/P24
pad38	D	38	G6/G5/CKV/cpu_d8/P25
pad39	D	39	G7/G4/OEV/cpu_d7/P26
DVDD_3	P	40	DVDD

VDD_2	P	41	VDD
VSS_2	P	42	GND
pad43	D	43	B0/G3/tcon_b0/itu_d0/sD0/cpu_d6/GPIO5
pad44	D	44	B1/G2/tcon_b1/itu_d1/sD1/cpu_d5/GPIO6
pad45	D	45	B2/G1/tcon_b2/itu_d2/sD2/cpu_d4/GPIO7
pad46	D	46	B3/G0/tcon_b3/itu_d3/sD3/cpu_d3/GPIO8
pad47	D	47	B4/R7/tcon_b4/itu_d4/sD4/cpu_d2/GPIO9
pad48	D	48	B5/R6/tcon_b5/itu_d5/sD5/cpu_d1/GPIO10
pad49	D	49	B6/R5/STHR/itu_d6/sD6/cpu_d0/P27
pad50	D	50	B7/R4/POL/itu_d7/sD7/cpu_rd/P30
pad51	D	51	DCK/R3/tcon_clk/ituclko/sCLK/cpu_wr/P31
pad52	D	52	DOE/R2/STHL/ituDEO/sDEN/cpu_rs/P32
pad53	D	53	HOS/R1/OEH/ituHDE/sHSY/cpu_cs/P33
pad54	D	54	VOS/R0/tck2/ituVDE/sVSY/cpu_rstn/P34
pad55	D	55	P35/DC_PWM0
pad56	D	56	P36/TXD/DC_PWM1
pad57	D	57	P37/RXD/DC_PWM2
pad16	D	58	P07/DC_PWM3/RXD
pad15	D	59	P06/DC_PWM2/TXD
pad13	D	60	P04/TXD/SDA
pad14	D	61	P05/RXD/SCL
DVDD_6	P	62	DVDD
pad12	D	63	RESET
	P	64	AVDD_ADC

6. Register Descriptions

6.1 Global Register(I2C Address: 0xB0 MCU Address: 0xFDXX)

Global Register Description:

Addr	Val	Bits	Name	Description	App note
0x00	00h	[7:0]	RSTN_REG	5Ah : Soft reset Else : No action if other values	
0x01	01h	[7:1]	reserved	可以用来做变量寄存器	
		[0]	chip_en	0. 关闭 BK,ADC 等模拟电路, 屏蔽部份 CLK. 1. 正常工作	
0x02	00h	[7:1]	reserved	可以用来做变量寄存器	

		[0]	SCALE_BYPASS	SCALER bypass enable	
0x03	00h	[7:0]	reserved	可以用来做变量寄存器	
0x04	00h	[7:0]	gpio_ie[7: 0]	gpio[7:0] 输入控制 1: 输入 0: 输入高阻	
0x05	00h	[7:4]	reserved	可以用来做变量寄存器	
		[3:0]	gpio_ie[11: 8]	gpio[11:8] 输入控制 1: 输入 0: 输入高阻	
0x06	00h	[7:0]	gpio_oe[7: 0]	gpio[7:0] 输出控制 1: 输出 0: 输出高阻	
0x07	00h	[7:4]	reserved	可以用来做变量寄存器	
		[3:0]	gpio_oe[11:8]	gpio[11:8] 输出控制 1: 输出 0: 输出高阻	
0x08	00h	[7:0]	gpio_out [7: 0]	gpio[7:0] 输出 输出控制为 1 时 寄存器读的值为 GPIO_OUT, 否则为 GPIO 的输入值	
0x09	00h	[7:4]	reserved	可以用来做变量寄存器	
		[3:0]	gpio_out[11:8]	gpio[11:8] 输出 输出控制为 1 时 寄存器读的值为 GPIO_OUT, 否则为 GPIO 的输入值	
0x0A	1Dh	[7:0]	ANA_DIVA	PLLA DIV VALUE config: PLLA CLK_OUT=FIN * ANA_DIVA / OUT_DIVA	
0x0B	1Dh	[7:0]	ANA_DIVB	PLLB DIV VALUE config: PLLB CLK_OUT=FIN * ANA_DIVB / OUT_DIVB	
0x0C	33h	[7:4]	SPB	PLLA CP current trim	
		[3:0]	SPA	PLLB CP current trim	
0x0D	F0h	[7:6]	OUT_DIVB	PLLA OUT DIV 00: BYPASS; 01: 1/2; 10: 1/4; 11: 1/8	
		[5:4]	OUT_DIVA	PLLB OUT DIV 00: BYPASS; 01: 1/2; 10: 1/4; 11: 1/8	
		[3]	LPR_EN	LOW POWER DET enable; active : high;	

		[2:0]	LDO_REG	CORE LDO TRIM reg	
0x0E	20h	[7]	OSC4_EN_V12	Built-in 4M OSC enable; active high	
		[6]	SMT_EN	OSC 27MHZ SMT enable; active high;	
		[5:4]	E2_E1	OSC 27MHZ driver trim;	
		[3]	ENH_PLLB	PLLB ENABLE; active high;	
		[2]	ENH_PLLA	PLLA ENABLE; active high;	
		[1]	BAND_ADJB	PLLB BAND TRIM	
		[0]	BAND_ADJA	PLLA BAND TRIM	
0x0F	01h	[7:0]	displl_refclk_div_reg	displl_refclk 时钟源为 osc_clk_27m [7] : clk bypass [6:0] : clk 分频数	
0x10	01h	[7:0]	oscpll_refclk_div_reg	oscpll_refclk 时钟源为 osc_clk_27m [7] : clk bypass [6:0] : clk 分频数	
0x11	FFh	[7:0]	CLK_GATE_EN_REG[7:0]	CLK_GATE_EN_REG = {REG13H[7:0], REG12H[7:0], REG11H[7:0]} [0]: clk_spi gate_en [1]: clk_cache gate_en [2]: clk_flash gate_en [3]: clk_dram gate_en [4]: clk_fontram gate_en [5]: sys_clk gate_en [6]: tclk1 gate_en [7]: tclk2 gate_en	可固定 0xFF, 如果有低功耗时, 由软件动态控制。
0x12	FFh	[7:0]	CLK_GATE_EN_REG[15:8]	[0]: tclk3 gate_en [1]: srgb_clk gate_en [2]: hscale_clk gate_en [3]: display_clk gate_en [4]: mcu_adc_clk gate_en [5]: clk_pwm gate_en [6]: pll_en gate_en (no used) [7]: oscpll_en gate_en (no used)	可固定 0xFF, 如果有低功耗时, 由软件动态控制。

0x13	FFh	[7:0]	CLK_GATE_EN_REG[23:16]	[0]: reserved [1]: oversample_clk gate_en [7:2]: reserved	可固定 0xFF，如果有低功耗时，由软件动态控制。
0x14	03h	[7:0]	sca_clk_div_reg	scaler_clk_pre 时钟源为 disppll_clk [7] : clk bypass [6:0] : clk 分频数	
0x15	80h	[7:0]	dis_clk_div_reg	display_clk_pre 时钟源为 scaler_clk_pre [7] : clk bypass [6:0] : clk 分频数	
0x16	0Ah	[7:0]	tcon_src_div_reg	tcon_source_clk 时钟源为 disppll_clk [7] : clk bypass [6:0] : clk 分频数	
0x17	00h	[7:0]	mcu_adc_clk_div_reg	mcu_adc_clk_pre0 时钟源为 clk_27m [7] : clk bypass [6:0] : clk 分频数	
0x18	FFh	[7:0]	pwm_clk_div_reg	clk_pwm_pre 时钟源为 clk_pwm_src [7] : clk bypass [6:0] : clk 分频数	
0x19	42h	[7]	vadc_clk_sel	video_adc_clk 源选择 0: osc_clk_27m 1: oscpll_clk	
		[6]	pwm_clk_src_reg	clk_pwm_src 源选择 0: osc_clk_27m 1: pwm_src_clk	
		[5]	mcu_adc_clk_inv_reg	mcu_adc_clk_pre 时钟源为 mcu_adc_clk_pre0 0: bypass 1: 取反	
		[4]	srgb_clk_o_inv_reg	srgb_clk_mux 时钟源为 srgb_clk_pre 0: bypass 1: 取反	
		[3]	pwm_src_pll_sel	pwm_src_clk 源选择 0: disppll_clk 1: oscpll_clk	
		[2]	dis_clk_inv_reg	display_clk_mux 时钟源为 display_clk_pre 0: bypass 1: 取反	
		[1]	sys_clk_inv_reg	source_clk_col 时钟源为 videoadc_clk_in 0: bypass 1: 取反	
		[0]	sys_clk_div_reg	sys_clk_pre 时钟源为 source_clk_col 0: bypass 1: 2 分频	

0x1A	40h	[7:6]	adc_en_reg	[7] chip_en control (同省电模式控制) [6] ADC clock enable	
		[5:3]	reserved		
		[2:0]	tpll_div_reg	3'b000: DIV3 3'b001: DIV6 3'b010: DIV9 3'b011: DIV12 3'b100: DIV1.5 3'b101: DIV2 3'b110: DIV1 3'b111: DIV1	
0x1B	FFh	[7:0]	det_threshold[7:0]	用 rc_clk 检测晶振时钟是否存在的时限	
0x1C	FFh	[7:0]	det_threshold[15:8]	用 rc_clk 检测晶振时钟是否存在的时限	
0x1D	00h	[7]	i2c_pfi_int	Mcu soft int trigger. Active high.	
		[6:1]	reserverd		
		[0]	det_wdt_rst_en	检测到无晶振时钟时是否复位, Active high.	
0x1F	00h	[7]	pwm3_inv	PWM3 输出取反	
		[6]	pwm2_inv	PWM2 输出取反	
		[5]	pwm1_inv	PWM1 输出取反	
		[4]	pwm0_inv	PWM0 输出取反	
		[3]	pwm3_ena	PWM3 enable	
		[2]	pwm2_ena	PWM2 enable	
		[1]	pwm1_ena	PWM1 enable	
		[0]	pwm0_ena	PWM0 enable	
0x20	FFh	[7:0]	PWM0_CNTR[7:0]	PWM0_CNTR[15:0]={REG21H[7:0],REG20H[7:0]} //16'h00ff	
0x21	00h	[7:0]	PWM0_CNTR[15:8]	PWM0 周期	
0x22	FFh	[7:0]	PWM1_CNTR[7:0]	PWM1_CNTR[15:0]={REG23H[7:0],REG22H[7:0]} //16'h00ff	
0x23	00h	[7:0]	PWM1_CNTR[15:8]	PWM1 周期	
0x24	FFh	[7:0]	PWM2_CNTR[7:0]	PWM2_CNTR[15:0]={REG25H[7:0],REG24H[7:0]} //16'h00ff	
0x25	00h	[7:0]	PWM2_CNTR[15:8]	PWM2 周期	
0x26	FFh	[7:0]	PWM3_CNTR[7:0]	PWM3_CNTR[15:0]={REG27H[7:0],REG26H[7:0]} //16'h00ff	
0x27	00h	[7:0]	PWM3_CNTR[15:8]	PWM3 周期	
0x28	80h	[7:0]	PWM0_DUTY[7:0]	PWM0_DUTY[15:0]={REG29H[7:0],REG28H[7:0]} //16'h0080	
0x29	00h	[7:0]	PWM0_DUTY[15:8]	PWM0 占空比	
0x2A	80h	[7:0]	PWM1_DUTY[7:0]	PWM1_DUTY[15:0]={REG2BH[7:0],REG2AH[7:0]} //16'h0080	

0x2B	00h	[7:0]	PWM1_DUTY[15:8]	PWM1 占空比	
0x2C	80h	[7:0]	PWM2_DUTY[7:0]	PWM2_DUTY[15:0]={REG2DH[7:0],REG2CH[7:0]} //16'h0080	
0x2D	00h	[7:0]	PWM2_DUTY[15:8]	PWM2 占空比	
0x2E	80h	[7:0]	PWM3_DUTY[7:0]	PWM3_DUTY[15:0]={REG2FH[7:0],REG2EH[7:0]} //16'h0080	
0x2F	00h	[7:0]	PWM3_DUTY[15:8]	PWM3 占空比	
0x30	80h	[7:0]	padmux_reg0	padmux_reg0 ~ padmux_reg19 参考 AMT630A_PAD_MUX.xls	Pad 相关, 开发调试。
0x31	80h	[7:0]	padmux_reg1		Pad 相关, 开发调试。
0x32	98h	[7:0]	padmux_reg2		Pad 相关, 开发调试。
0x33	75h	[7:0]	padmux_reg3		Pad 相关, 开发调试。
0x34	50h	[7:0]	padmux_reg4		Pad 相关, 开发调试。
0x35	00h	[7:0]	padmux_reg5		Pad 相关, 开发调试。
0x36	00h	[7:0]	padmux_reg6		Pad 相关, 开发调试。
0x37	00h	[7:0]	padmux_reg7		Pad 相关, 开发调试。
0x38	00h	[7:0]	padmux_reg8		Pad 相关, 开发调试。
0x39	00h	[7:0]	padmux_reg9		Pad 相关, 开发调试。
0x3A	00h	[7:0]	padmux_reg10		Pad 相关, 开发调试。
0x3B	00h	[7:0]	padmux_reg11		Pad 相关, 开发调试。
0x3C	00h	[7:0]	padmux_reg12		Pad 相关, 开发调试。
0x3D	00h	[7:0]	padmux_reg13		Pad 相关, 开发调试。
0x3E	00h	[7:0]	padmux_reg14		Pad 相关, 开发调试。
0x3F	00h	[7:0]	padmux_reg15		Pad 相关, 开发调试。
0x40	00h	[7:0]	padmux_reg16		Pad 相关, 开发调试。
0x41	00h	[7:0]	padmux_reg17		Pad 相关, 开发调试。
0x42	00h	[7:0]	padmux_reg18		Pad 相关, 开发调试。
0x43	06h	[7:0]	padmux_reg19		Pad 相关, 开发调试。
0x44	01h	[7:0]	SSCG_DLY_SEL[7:0]	DCLK OUTPUT DELAY SEL LSB	
0x45	00h	[7:0]	SSCG_DLY_SEL[15:8]		
0x46	00h	[7:0]	SSCG_DLY_SEL[23:16]		
0x47	00h	[7:0]	SSCG_DLY_SEL[31:24]		
0x48	00h	[7:0]	SSCG_DLY_SEL[39:32]		
0x49	00h	[7:0]	SSCG_DLY_SEL[47:40]		
0x4A	00h	[7:0]	SSCG_DLY_SEL[55:48]		

0x4B	00h	[7:0]	SSCG_DLY_SEL[63:56]	DCLK OUTPUT DELAY SEL MSB	
0x4C	12h	[7:0]	SSCG_CP	SSCG PLL CP TRIM	
0x4D	4Fh	[7:0]	SSCG_DIV	SSCG DIV TRIM	
0x4E	02h	[7:0]	SSCG_DIV_IN	SSCG INPUT FREQ TRIM	
0x4F	27h	[7:0]	SSCG_DIV_SG	SSCG MODULE TRIM	
0x50	0Bh	[7]	SSCG_EN	SSCG PLL enable; ACTIVE HIGH;	
		[6]	SSCG_DLY_EN	SSCG MODULE enable; ACTIVE HIGH;	
		[5]	BY_SSCG	BYPASS SSCG, ACTIVE LOW;	
		[4]	SG_BAND_ADJ	SSCG PLL BAND TRIM	
		[3]	DS_INV_SEL	SSCG DATA SAMPLE EDIGE SEL	
		[2]	DCK_INV_SEL	DCK OUTPUT INV SEL	
		[1:0]	SSCG_DIV	Sscg pll DIV 00: x1; 01: 1/2; 10 1/4; 11:1/8	
0x51	55h	[7:0]	cell_type_reg0	E1 E0 00: 2mA, 01: 4mA, 10: 8mA, 11: 12mA [7:6]: pad15 E1 E0 [5:4]: pad14 E1 E0 [3:2]: pad13 E1 E0 [1:0]: pad11 E1 E0 (del)	
0x52	55h	[7:0]	cell_type_reg1	[7:6]: pad19 E1 E0 [5:4]: pad18 E1 E0 [3:2]: pad17 E1 E0 [1:0]: pad16 E1 E0	
0x53	57h	[7:0]	cell_type_reg2	[7:6]: pad26 E1 E0 [5:4]: pad25 E1 E0 [3:2]: pad24 E1 E0 [1:0]: pad20 E1 E0	
0x54	57h	[7:0]	cell_type_reg3	[7:6]: pad30 E1 E0 [5:4]: pad29 E1 E0 [3:2]: pad28 E1 E0 [1:0]: pad27 E1 E0	
0x55	55h	[7:0]	cell_type_reg4	[7:6]: pad34 E1 E0 [5:4]: pad33 E1 E0 [3:2]: pad32 E1 E0 [1:0]: pad31 E1 E0	
0x56	55h	[7:0]	cell_type_reg5	[7:6]: pad38 E1 E0 [5:4]: pad37 E1 E0 [3:2]: pad36 E1 E0 [1:0]: pad35 E1 E0	

0x57	55h	[7:0]	cell_type_reg6	[7:6]: pad45 E1 E0 [5:4]: pad44 E1 E0 [3:2]: pad43 E1 E0 [1:0]: pad39 E1 E0	
0x58	55h	[7:0]	cell_type_reg7	[7:6]: pad49 E1 E0 [5:4]: pad48 E1 E0 [3:2]: pad47 E1 E0 [1:0]: pad46 E1 E0	
0x59	5Dh	[7:0]	cell_type_reg8	[7:6]: pad53 E1 E0 [5:4]: pad52 E1 E0 [3:2]: pad51 E1 E0 [1:0]: pad50 E1 E0	
0x5A	55h	[7:0]	cell_type_reg9	[7:6]: pad57 E1 E0 [5:4]: pad56 E1 E0 [3:2]: pad55 E1 E0 [1:0]: pad54 E1 E0	
0x5B		[7]	vsyn_int_inv	场中断反相	
		[6:3]	reserved		
		[2:0]	vsyn_int_sel	场中断源选择 3'b000: vsyn_dec 3'b001: vsync_scaled 3'b010: Vsyn_vp_out 3'b011: blend_vsync_out 3'b100: osd_vsyn_int 3'b101: ga_v_sync_out 3'b110: dither_vsyn_out 3'b111: osd_vsyn_int	

6.2 Tcon Register(I2C Address: 0xB8 MCU Address: 0xFCXX)

0x00	00h	[7]	ITU_HV_SEL	输出为 itu601 时，输出行同步、HDE 信号选择。 0: 行有效点位 720; 1: 行有效点位 722。	
		[6]	INTERLACE	Itu656 输出时，该位表示逐行 656 输出还是隔行 656 输出。 0: 隔行 656，奇偶场标志跟数据源匹配; 1: 逐行 656，奇偶场标志逐行翻转，与数据源无关。	

		[5]	OUT_FIELD_INV	Itu656 输出时，奇偶场翻转标志 0: 不翻转; 1: 翻转。	
		[4]	WINDOW_EN	开窗使能，使能打开时，在水平和垂直方向可以开一大小可配的窗口，将窗口外的数据设为黑色。该窗口大小由 winh_start, winh_stop, winv_start, winv_stop 决定。 0: disable; 1: enable。	
		[3]	DAT_ORDER_INV	R G B 数据位反转选择，bit7—bit0 选择 0: 数字屏， d7-d0 (RGB) 对应 bit7-bit0; sRGB 屏， d7-d0 (sdata) 对应 bit7-bit0; Itu656 屏， d7-d0 (itu656 dat) 对应 bit7-bit0; Cpu 屏， d17-d0 (cpu_d) 对应 bit17-bit0; Tcon 屏， d5-d0 (R G B) 对应 bit7-bit0; 1: 数字屏， d7-d0 (RGB) 对应 bit0-bit7; sRGB 屏， d7-d0 (sdata) 对应 bit0-bit7; Itu656 屏， d7-d0 (itu656 dat) 对应 bit0-bit7; Cpu 屏， d17-d0 (cpu_d) 对应 bit0-bit7; Tcon 屏， d5-d0 (R G B) 对应 bit0-bit7;	
		[2:0]	RGB_ORDER_SEL	各种模式输出（包括屏、itu656 等）的 RGB 顺序选择 0: rout=rin; gout=gin; bout=bin; 1: rout=rin; gout=bin; bout=gin; 2: rout=gin; gout=rin; bout=bin; 3: rout=gin; gout=bin; bout=rin; 4: rout=bin; gout=rin; bout=gin; 5: rout=bin; gout=gin; bout=rin; Others: rout=rin; gout=gin; bout=bin;	

0x01	00h	[7]			
		[6]			
		[5]	VSCAL_UP_DOWN	Tcon 屏	
		[4]	SCA_BYPASS_SEL	Tcon 屏	
		[3]	FID_SYNC_SEL	Tcon 屏时，选择奇偶场信号 0：选择 decoder 中原始的奇偶场信号； 1：选择由 fid_sel 选择完之后的奇偶场信号。	
		[2]	FID_SEL	Tcon 屏时，选择奇偶场信号 0：每场反转； 1：采用 decoder 中的奇偶场信号,每场采样。	
		[1]	RGB_INV	Tcon 屏时，用来选择 RGB 是否反相 0：不反相； 1：反相。	
0x02	00	[0]	DTCON_EN	Tcon 屏模式选择，选择数据是否逐行倒相。 0：逐行倒相； 1：数据不倒相。	
		[7:3]			

CONFIDENTIAL

				<p>输出行同步、场同步、DE 信号 互换选择</p> <p>0：数字屏，HOS=hsyn， VOS=vsyn，DOE=hde； sRGB 屏，sHSY=hsyn， sVSY=vsyn，sDEN=hde； Itu601，ituHDE=行消隐， ituVDE=场消隐，ituDEO=数据 有效；</p> <p>Cpu 屏，cpu_rs=cpu_rs， cpu_rd=cpu_rd， cpu_cs=cpu_cs；</p> <p>1：数字屏，HOS=hsyn， VOS=hde，DOE=vsyn； sRGB 屏，sHSY=hsyn， sVSY=hde，sDEN=vsyn； Itu601，ituHDE=行消隐， ituVDE=数据有效，ituDEO=场 消隐；</p> <p>Cpu 屏，cpu_rs=cpu_rs， cpu_rd=cpu_cs， cpu_cs=cpu_rd；</p> <p>2：数字屏，HOS=vsyn， VOS=hsyn，DOE=hde； sRGB 屏，sHSY=vsyn， sVSY=hsyn，sDEN=hde； Itu601，ituHDE=场消隐， ituVDE=行消隐，ituDEO=数据 有效；</p> <p>Cpu 屏，cpu_rs=cpu_rd， cpu_rd=cpu_rs， cpu_cs=cpu_cs；</p> <p>3：数字屏，HOS=vsyn， VOS=hde，DOE=vsyn； sRGB 屏，sHSY=vsyn， sVSY=hde，sDEN=vsyn； Itu601，ituHDE=场消隐， ituVDE=数据有效，ituDEO=行 有效；</p> <p>Cpu 屏，cpu_rs=cpu_rd， cpu_rd=cpu_cs， cpu_cs=cpu_rs；</p> <p>4：数字屏，HOS=hde， VOS=hsyn，DOE=vsyn； sRGB 屏，sHSY=hde， sVSY=hsyn，sDEN=vsyn； Itu601，ituHDE=数据有效， ituVDE=行消隐，ituDEO=场消</p>	
		[2:0]	H_V_DE_SEL		

0x03	00	[7:0]	WINH_START[7:0]	开窗水平方向起始位置,低8位, 当开窗功能使能时有效	
0x04	00	[7:0]	WINH_STOP[7:0]	开窗水平方向结束位置,低8位, 当开窗功能使能时有效	
0x05	00	[7]			
		[6:4]	WINH_STOP[10:8]	开窗水平方向结束位置,高3位, 当开窗功能使能时有效	
		[3]			
		[2:0]	WINH_START[10:8]	开窗水平方向起始位置,高3位, 当开窗功能使能时有效	
0x06	00	[7:0]	WINV_START[7:0]	开窗垂直方向起始位置,低8位, 当开窗功能使能时有效	
0x07	00	[7:0]	WINV_STOP[7:0]	开窗垂直方向结束位置,低8位, 当开窗功能使能时有效	
0x08	00	[7]			
		[6:4]	WINV_STOP[10:8]	开窗垂直方向结束位置,高3位, 当开窗功能使能时有效	
		[3]			
		[2:0]	WINV_START[10:8]	开窗垂直方向起始位置,高3位, 当开窗功能使能时有效	
0x09	00h	[7]			
		[6]			
		[5]			
		[4]	YCBCR_FILTER_MODE	sRGB 模块中, CbCr444 to 422 转换时的模式选择 0: 模式 0; 1: 模式 1。	
		[3]	YCBCR_FILTER_EN	sRGB 模块中, 是否对 CbCr444 to 422 转换时的滤波选择 0: 不滤波; 1: 滤波。	
		[2]	SRGB_YUV_SEL	输入 sRGB 模块的数据源选择 0: 选择 RGB; 1: 选择 YCbCr 一般情况下, 屏模式为 sYUV422 时设 1, 其它模式设 0。	

		[1:0]	SRGB_MODE	sRGB 屏模式选择 0: through mode, R G B R G B..., sclk 为 dclk 的 3 倍; 1: sRGB dummy, R G B dummy R G B dummy ..., sclk 为 dclk 的 4 倍; 2: sYUV422, Cb Y Cr Y Cb Y Cr Y..., sclk 为 dclk 的 2 倍; 3: 输出为 0。	
0x0A	00	[7]			
		[6:4]	CF2	sRGB 屏时, 奇行或偶行对 RGB 的排序 0: BGR 1: GRB 2: RGB 3: RBG 4: GBR 5: BRG 6: BGR	
		[3]			
		[2:0]	CF1	sRGB 屏时, 偶行或奇行对 RGB 的排序, 同 CF2	
0x0B	00	[7:0]	CPU_SOFT_REG[7:0]	Cpu 屏时, 软件初始化屏寄存器, 低 8 位表示[17:0] 18 位数据线的低 8 位;	
0x0C	00	[15:8]	CPU_SOFT_REG[15:8]	Cpu 屏时, 软件初始化屏寄存器, 中 8 位表示[17:0] 18 位数据线的中 8 位;	
0x0D	00	[7:6]	CPU_SCR_STATUS	只读寄存器 Bit[7], data_refresh_state, 硬件数据刷新状态; Bit[6], adr_clr_state, 每场读取 ram 并初始化屏状态;	
		[5]	CPU_RS	Cpu 屏时, 软件初始化屏寄存器, 表示 cpu_rs 信号	
		[4]	CPU_CS	Cpu 屏时, 软件初始化屏寄存器, 表示 cpu_cs 信号	
		[3]	CPU_RD	Cpu 屏时, 软件初始化屏寄存器, 表示 cpu_rd 信号	
		[2]	CPU_WR	Cpu 屏时, 软件初始化屏寄存器, 表示 cpu_wr 信号	

		[1:0]	CPU_SOFT_REG[17:16]	Cpu 屏时，软件初始化屏寄存器，高 2 位表示[17:0] 18 位数据线的高 2 位；	
0x0E	00	[7]			
		[6]	CPU_SCR_COMB	Cpu 屏输出 RGB 信号顺序选择 0: BGR 1: RGB	
		[5]	DATA_REFRESH_EN	硬件模式下，数据刷新使能。使能有效时，每场不停望 cpu 屏的 GRAM 中刷新数据。无效时，则不更新 GRAM 中的数据。 0: disable; 1: enable。	
		[4]	ADR_CLR_SEL	每场初始化某些参数位置沿的选择 0: 场同步下降沿； 1: 场同步上升沿。	
		[3]	ADR_CLR_EN	存储于 16x20 大小的 ram 中的指令是否每场读取并生成相应的控制信号的使能选择。 0: disable，每场同步沿时不读取 16x20 ram 中的数据； 1: enable，每场同步沿时读取 16x20 ram 中的数据，并生成相应的控制信号去初始化屏的某些参数。	
		[2]	CMD_MODE	命令模式，该命令上升沿时，将用来存储命令的 16x20 大小的 ram 中的指令读取并生成相应的控制信号。	
		[1]	SOFT_MODE	Cpu 屏时，表示硬件模式与软件模式的选择 0: 硬件模式，由硬件生成各个控制信号及写入 RAM 的数据； 1: 软件模式，由软件控制各个控制信号，主要用于屏的初始化。	
		[0]	CPU_RSTN	Cpu 屏时，软件初始化屏寄存器，表示复位信号 cpu_rstn	
0x0F	00	[7:6]			

		[5:4]	DATAEN_MASK	<p>Cpu 屏硬件模式下写入 GRAM 的数据选择</p> <p>0: 与 data enable 对应的数据;</p> <p>1: 比 data enable 延迟多 2 拍, 少一个点;</p> <p>2: 比 data enable 延迟多 1 拍, 少一个点;</p> <p>3: 比 data enable 延迟多 2 拍, 不少点。</p>	
		[3]			
		[2:0]	CPU_SCR_MODE	<p>Cpu 屏模式选择</p> <p>0: 18 位数据线, 18 位屏, sclk 跟 dclk 同频;</p> <p>1: 18 位数据线, 16 位屏, bit0、bit9 固定为 0, sclk 跟 dclk 同频;</p> <p>2: 9 位数据线, 18 位屏, 低 9 位固定为 0, 使用高 9 位, sclk 为 dclk 的 2 倍;</p> <p>3: 8 位数据线, 16 位屏, 低 10 位固定为 0, 使用高 8 位, sclk 为 dclk 的 2 倍;</p> <p>4: 8 位数据线, 18 位屏, 低 12 位固定为 0, 使用高 6 位, sclk 为 dclk 的 3 倍;</p> <p>5: 8 位数据线, 24 位屏, 低 10 位固定为 0, 使用高 8 位, sclk 为 dclk 的 3 倍;</p> <p>6: 16 位数据线, 16 位屏, 低 2 位固定为 0, 使用高 16 位, sclk 与 dclk 同频;</p> <p>7: 16 位数据线, 16 位屏, 高 2 位固定为 0, 使用低 16 位, sclk 与 dclk 同频;</p>	
0x10	00	[7:4]	RAM_RD_NUM	<p>每场读取 16x20 大小 ram 并初始化屏参的个数, 0 表示 1 个, 0xF 表示 16 个。</p>	
		[3:0]	INTERVAL_COUNTER	<p>读取 16x20 大小 ram 进行初始化时, 生成控制信号的长度。</p>	
0x11	00	[7:0]	RESERVED	Reserved	
0x12	00	[7]	VS_ADJ_FID_SEL	<p>垂直方向移动位置的奇偶场标志选择。</p>	

		[6]	CKH_INV	Ckh1、ckh2 输出是否反向选择 0: 不反向; 1: 反向。	
		[5]	TCPH3_INV	Reservec	
		[4]	VCOM_ACT	VCOM 有效标志, 1 有效。	
		[3]	VCOM_INV	VCOM 取反标志, 1 取反, 0 不取反。	
		[2]	CKH2_3_MOD	Ckh2、ckh3 输出选择 0: 选择 ckh2、ckh3 1: 选择 ckh1	
		[1]	LNR	Reserved	
		[0]	UND	Reserved	
0x13	00	[7:4]			
		[3:2]	STH_SEL	OSD 的定位信号 sth1_osd, sthr_osd 信号选择 0: sth1; 1: sth2; 2: sth3; 3: ~sth1。	
		[1:0]	CKH1_MOD	Ckh1 输出选择 0: tclk1; 1: tclk2; 2: tclk3; 3: ~tclk1。	
0x14	00	[7:0]	TCON_HSHIFT[7:0]	水平方向位置移动寄存器, 低 8 位。	
0x15	00	[7]	RESERVED		
		[6:4]	TCON_VSHIFT[10:8]	垂直方向位置移动寄存器, 高 3 位	
		[3]	RESERVED		
		[2:0]	TCON_HSHIFT[10:8]	水平方向位置移动寄存器, 高 3 位。	
0x16	00	[7:0]	TCON_VSHIFT[7:0]	垂直方向位置移动寄存器, 低 8 位	
0x17	00	[7:4]	TCON_CKVFM	CKV 垂直方向寄存器	
		[3:0]	TCON_CKVFZ	CKV 垂直方向寄存器	
0x18	00	[7:0]	TCPH1_RISE[7:0]	OEH 上升沿位置, 低 8 位。	
0x19	00	[7:3]	RESERVED		
		[2:0]	TCPH1_RISE[10:8]	OEH 上升沿位置, 高 3 位。	
0x1A	00	[7:0]	TCPH1_FALL[7:0]	OEH 下降沿位置, 低 8 位。	
0x1B	00	[7:3]	RESERVED		
		[2:0]	TCPH1_FALL[10:8]	OEH 下降沿位置, 高 3 位。	

0x1C	00	[7:0]	TCPH4_RISE[7:0]	OEV 上升沿位置, 低 8 位。	
0x1D	00	[7:3]	RESERVED		
		[2:0]	TCPH4_RISE[10:8]	OEV 上升沿位置, 高 3 位。	
0x1E	00	[7:0]	TCPH4_FALL[7:0]	OEV 下降沿位置, 低 8 位。	
0x1F	00	[7:3]	RESERVED		
		[2:0]	TCPH4_FALL[10:8]	OEV 下降沿位置, 高 3 位。	
0x20	00	[7:0]	STHL_RISE[7:0]	STHL 上升沿位置, 低 8 位。	
0x21	00	[7:3]	RESERVED		
		[2:0]	STHL_RISE[10:8]	STHL 上升沿位置, 高 3 位。	
0x22	00	[7:0]	STHL_FALL[7:0]	STHL 下降沿位置, 低 8 位。	
0x23	00	[7:3]	RESERVED		
		[2:0]	STHL_FALL[10:8]	STHL 下降沿位置, 高 3 位。	
0x24	00	[7:0]	STHR_RISE[7:0]	STHR 上升沿位置, 低 8 位。	
0x25	00	[7:3]	RESERVED		
		[2:0]	STHR_RISE[10:8]	STHR 上升沿位置, 高 3 位。	
0x26	00	[7:0]	STHR_FALL[7:0]	STHR 下降沿位置, 低 8 位。	
0x27	00	[7:3]	RESERVED		
		[2:0]	STHR_FALL[10:8]	STHR 下降沿位置, 高 3 位。	
0x28	00	[7:0]	CKV_RISE[7:0]	CKV 上升沿位置, 低 8 位。	
0x29	00	[7:3]	RESERVED		
		[2:0]	CKV_RISE[10:8]	CKV 上升沿位置, 高 3 位。	
0x2A	00	[7:0]	CKV_FALL[7:0]	CKV 下降沿位置, 低 8 位。	
0x2B	00	[7:3]	RESERVED		
		[2:0]	CKV_FALL[10:8]	CKV 下降沿位置, 高 3 位。	
0x2C	00	[7:0]	VCOM_TRIG[7:0]	VCOM 翻转位置, 低 8 位。	
0x2D	00	[7:3]	RESERVED		
		[2:0]	VCOM_TRIG[10:8]	VCOM 翻转位置, 高 3 位。	
0x2E	00	[7:0]	VCOM2_TRIG[7:0]	POL 翻转位置, 低 8 位。	
0x2F	00	[7:3]	RESERVED		
		[2:0]	VCOM2_TRIG[10:8]	POL 翻转位置, 高 3 位。	
0x30	00	[7:0]	CKVSP_START[7:0]	CKV 垂直方向开始位置, 低 8 位。	
0x31	00	[7:2]	RESERVED		
		[1:0]	CKVSP_START[9:8]	CKV 垂直方向开始位置, 高 3 位。	
0x32	00	[7:0]	CKVSP_END[7:0]	CKV 垂直方向结束位置, 低 8 位。	
0x33	00	[7:2]	RESERVED		

		[1:0]	CKVSP_END[9:8]	CKV 垂直方向结束位置，高 3 位。 CKV_START, CKV_END 值相同时，CKV 在场的每行都会出，一般配置成相同的值，具体以屏为准。	
0x34	00	[7:0]	STVL_RISE[7:0]	STVL 上升沿位置。	
0x35	00	[7:0]	STVL_FALL[7:0]	STVL 下降沿位置。	
0x36	00	[7:0]	STVR_RISE[7:0]	STVR 上升沿位置。	
0x37	00	[7:0]	STVR_FALL[7:0]	STVR 下降沿位置。	
0x38	00	[7:0]	R_BIAS[7:0]	R 信号偏置电压。	
0x39	00	[7:0]	G_BIAS[7:0]	G 信号偏置电压。	
0x3A	00	[7:0]	B_BIAS[7:0]	B 信号偏置电压。	
0x3B	00	[7:0]	R_BIAS_N[7:0]	R_N 信号偏置电压。	
0x3C	00	[7:0]	G_BIAS_N[7:0]	G_N 信号偏置电压。	
0x3D	00	[7:0]	B_BIAS_N[7:0]	B_N 信号偏置电压。	
0x3E	00	[7:0]	R_RATIO[7:0]	R 信号乘积系数。	
0x3F	00	[7:0]	G_RATIO[7:0]	G 信号乘积系数。	
0x40	00	[7:0]	B_RATIO[7:0]	B 信号乘积系数。	
0x41	00	[7:0]	RESERVED		
0x42	00	[7:0]	RESERVED		
0x43	00	[7:0]	VSHIFT_ODD[7:0]	奇数场垂直方向移动行数寄存器，低 8 位。	
0x44	00	[7:4]	RESERVED		
		[3:2]	VSHIFT_EVEN[9:8]	偶数场垂直方向移动行数寄存器，高 2 位。	
		[1:0]	VSHIFT_ODD[9:8]	奇数场垂直方向移动行数寄存器，高 2 位。	
0x45	00	[7:0]	VSHIFT_EVEN[7:0]	偶数场垂直方向移动行数寄存器，低 8 位。	
0x46	39	[7:6]	RESERVED		

		[5:0]	HDEN_DEL_VAL	<p>水平点有效信号 HDEN 延迟寄存器，该寄存器值与 VP 各模块 enable 或 bypass 有关，与 color matrix 是否打开有关。具体配置的值如下所示：</p> <p>1、VP 寄存器 B1[5]=1，VP bypass，配置如下：</p> <p>VP 寄存器 FB[7]=1，HDEN_DEL_VAL = 0X10；</p> <p>VP 寄存器 FB[7]=0，HDEN_DEL_VAL = 0X0F；</p> <p>2、VP 寄存器 B1[5]=0，配置如下：</p> <p>VP 寄存器 FB[7]=1，HDEN_DEL_VAL = 0X39；</p> <p>VP 寄存器 FB[7]=0，HDEN_DEL_VAL = 0X38；</p>	
--	--	-------	--------------	--	--

6.3 Decoder Register(I2C Address: 0xB2 MCU Address: 0xFEXX)

Addr	Val	Bits	Name	Description	App note
0x00	80h	[7]	Auto_config	Auto config amtdec registers	
		[6]	Ntsc_j_sel	<p>Mcu write during automode（用于日本 NTSC-J 制式，影响黑电平）</p> <p>1. NTSC-J PAL N</p> <p>other video format</p>	
		[5]	Sample_format	<p>1. square pixel output</p> <p>0. 13.5m pixel output</p>	
		[4]	Ped_inv	1. ped invert 0. not invert	
		[3:0]	Cstd	<p>0. PAL BGDHI</p> <p>1. NTSC M</p> <p>2. NTSC 4.43</p> <p>3. PAL M</p> <p>4. PAL N</p> <p>5. PAL 60</p> <p>6. NTSC-J</p> <p>7. NTSC N</p> <p>8. PAL CN</p> <p>9. reserver</p> <p>10. SECAM</p>	
0x01	04h	[4]	Comphv_detect_en	1. comp hv detect enable 0. disable	
		[3]	reg_inq_en	1. std detected by inquire 0. normal	

0x02	00h	[2]	Pal60_mode	1. pal60 force mode 0. normal	
		[1]	cstd_pn_mode	1. simple cstd pn mode 0. all cstd mode	
		[0]	cstd_sel	1. manual set cstd 0. auto detect cstd	
		[7]	vline_625_sel	1. manual set vline 0. auto detect vline	
		[6]	vline_625_set	1. set 625 vlines 0. set 525 vlines	
		[5]	pal_sel	1. manual set pal 0. auto detect pal	
		[4]	pal_set	1. set pal 0. set ntsc	
		[3]	freq443_sel	1. manual set 4.43 0. auto detect freq	
		[2]	freq443_set	1. set 4.43M 0. set 3.58M	
0x03	80h	[1]	secam_sel	1. manual set secam 0. auto detect	
		[0]	secam_set	1. set secam 0. set no secam	
		[7:0]	comphv_slvl	Comphv detect slice level	
		[7:0]	comphv_exist_thr	Comphv exist lines threshold	
		[7:0]	comphv_hdiff	Comphv hsyncip difference	
		[2:0]	NP_DET_THLD	Burst sign detect threshold 相邻 1 行同步头极性判断阈值，一般大于 1 即可	
		[7:0]	P_CNT_THLD	NTSC/PAL detect line number threshold 是否存在 PAL 行判断阈值，越大越不容易检测到 pal 行	
		[4:0]	SECAM_DET_THLD	SECAM detect threshold for alternately line symbol 相邻两行同步头极性判断阈值	
		[5:4]	bpf_carrier_sel	色副载波带通滤波器选择，影响色度解码通道 00. cvbs_bpf_lv1 01. cvbs_bpf_lv2 10. cvbs_bpf_1	
0x0A	2fh	[3:0]	acc_num	信号检测加强倍数，用于弱信号检测	
		[7:0]	burst_gate_paldet_start	Burst gate start position for pal detect	
		[7:0]	burst_gate_paldet_end	Burst gate end position for pal detect	
		[7:0]	SECAM_BURST_THR	Secam burst_amplitude threshold 越大越易检测成 secam 制式	
		[7:0]	dgain_set[7:0]	Dgain_val low 8 bits, when dagc_sel is 1,user manual set this value	
		[3:0]	dgain_set[11:8]	Dgain_val high 4 bits, when dagc_sel is 1,user manual set this value	
		[4]	dgain_clear_en	Dgain clear enable	
		[5]	bgate_by_scope	Backporch controlled by signal scope	
		[6]	halter_clear	Hsync pll clear when hpixel change	
0x0F	07h	[3:0]	pn_detect_tc	PN detect time constant	
0x10	14h	[7:0]	hjit_max	Hsync jitter maximum value	
0x11	09h	[7:0]	hlock_num_thr	Hsync lock line number threshold	
0x12	00h	[7]	hv_sel	1. hv sliced by fixed counter 0. hv sliced by accumulate counter	
		[6:0]	psign_range	Cvbs hsyn width tolerate	

				影响弱信号行同步稳定，值越小，检测越准确。 弱信号应适当放大	
0x13	16h	[4]	pga_en	Pga enable	
		[3]	Pga_manual	1. Pga manual set 0. pga auto set	
		[2:0]	Pga_tc	Pga adjust time constant	
0x14	22h	[7:0]	pga_swch_th	Pga switch threshold	
0x15	05h	[3:0]	pga_state_fix	Pga state manual set fixed	
0x26		[3]	Chroma_lock	Chroma lock. Read only	
		[2]	Vs_lock	Vsync lock. Read only	
		[1]	Hs_lock	Hsync lock. Read only	
		[0]	Tv_inactive	No signal detect. Read only	
0x27		[3]	freq443_detected	Subc 4.43 system detected. Read only	
		[2]	vnon_standard	Vsync frequency non-standard. Read only	
		[1]	hnon_standard	hsync frequency non-standard. Read only	
0x28		[4]	Vcr_detect	Vcr detected	
		[3]	noisy_flag	Indicate input signal is noisy or not.	
		[2]	Vline_625_detected	Vertical 625 lines detected	
		[1]	Secam_verify	Secam verified in normal	
		[0]	Pal_verify	Pal verified in normal	
0x2a		[6]	comphv_detected	Comp hv input detected. 倒车检测标志	
		[5]	secamsig_detected	Secam signal detected.	
		[4]	palsig_detected	Pal signal detected.	
		[3:0]	auto_tv_standard	Auto input standard flag. Every format refer to register Cstd	
0x35	aah	[7:0]	testpat_reg[7:0]	Test pattern control register	
0x36	aah	[7:0]	testpat_reg[15:8]	Test pattern control register	
0x37	60h	[7:6]	Dgain_white_ctl	00. synctip bottom 01. synctip bottom and white adjust fast 10. synctip bottom and white adjust slowly	
		[5:0]	Dgain_bottdiff	Dgain bottom fast adjust difference	
0x38	0fh	[3:0]	dgain_fast_adj_cnt	Dgain fast adjust field counter	
0x39	08h	[3:0]	weak_color_thr	Weak color detect threshold	
0x48	07h	[7:6]	Color_kill_sel	0. 自动消色 1. 强制消色 2. 强制不消色（当 检测到 secam 制式时可设为 2）	
		[5]	reserve		
		[4]	reserve		
		[3:0]	color_kill_level	color kill level.	

				该值越小，越容易消色。反之，越不易消色。	
0x54	00h	[6]	vsync_out_sel	0. 输出滤波电路之后的场同步信号 1. 输出未滤波的场同步信号	
		[5:0]	vsync_slice_offset	场同步切割电平偏移。	
0x55	0ah	[1:0]	Vflt_coef	场锁滤波系数，越大场同步锁定越慢，反之越快。	
0x5f	C0h	[7:0]	adc_blank_level[7:0]	Specify the blank level of adc data	
0x60	03h	[7:0]	adc_blank_level[11:8]	Specify the blank level of adc data	
0x61	2dh	[7:0]	burst_gate_z_start	Burst crossZ start position	
0x62	41h	[7:0]	burst_gate_z_end	Burst crossZ end position	
0x63	C0h	[7:0]	ZCORSS_THR	crossZ detect threshold, see 0x4E[7] 色副载波频率检测门限 低于门限为 pal 为 3.58M，高于为 4.43M	
0x83	43h	[7]	Ench_en	色度细节边缘增强使能	
		[6]	Ench_coring_en	色度 coring 使能。	
		[1:0]	Ench_gain	色度边缘增强增益。	
0xa0	02h	[1]	NP_auto_switch	1. NP register auto switch 0. manual set NP register	
		[0]	Amtdecoder_rst	1. AMT decoder internal reset. 0. normal.	
0xaa	00h	[3:0]	Dgain_val[11:8]	Dgain value high 4 bits. Read only	
0xab	00h	[7:0]	Dgain_val[7:0]	Dgain value low 8 bits. Read only	
0xac	00h	[7:0]	C_mag	chroma magnitude. Read only	
0xb1	00h	[7:0]	Noise_level	Current singal's noise level. Read only	
0xb2	04h	[3:1]	peak_coeff	Peaking coefficient.	
		[0]	Peaking_en	Luma peaking enable. 结合 VP 中的 peaking 适当调整	
0xb5	6fh	[3:1]	Clock_speed	色度解码锁定速度设置，越小速度越快。	
		[0]	Clost_kill	色度解码不稳定时强制消色使能	
0xca	01h	[0]	Dagc_en	Dgain adjust enable. Active high. 当使能时，可通过 dgain_set (FE0D、FE0E) 设置 dgain 的大小。	
0xcd	32h	[7:0]	noise_thr	noise detect threshold. 越小越易检测为噪声信号。	
0xd0	00h	[7]	weak_color_frame	Read only	弱色标志
		[6]	same_color_frame	Read only	纯色标志
		[5]	Colbar_exist_flag	Read only	彩阶标志
		[4]	Graybar_exist_flag	Read only	灰阶标志
		[3]	Half_colbar_exist_flag	Read only	半彩阶标志
		[2]	Half_grabar_exist_flag	Read only	半灰阶标志
		[1]	adc1_doh	Read only	

		[0]	adc1_dol	Read only	
0xd1	19h	[6:0]	trim_vcoml	ADC command voltage trim	
0xd2	00h	[7:0]	clamp_dac[7:0]	CLAMP current trim	
0xd3	00h	[1:0]	clamp_dac[9:8]	CLAMP current trim	
0xd4	00h	[7]	ADC_IBS1_SEL1	ADC BIAS current trim	
		[6]	ADC_IBS1_SELO	ADC BIAS current trim	
		[5]	ADC_IBS0_SEL1	ADC BIAS current trim	
		[4]	ADC_IBS0_SELO	ADC BIAS current trim	
		[3]	ADC_IBLS_SEL1	ADC BIAS current trim	
		[2]	ADC_IBLS_SELO	ADC BIAS current trim	
		[1]	ADC_COM_SEL1	ADC command voltage sel	
		[0]	ADC_COM_SELO	ADC command voltage sel	
0xd5	A1h	[7]	CLAMP_MODE	CLAMP MODE	
		[6]	PF1	BAND TRIM	
		[5]	PF0	BAND TRIM	
		[4]	LPF_EN	LPF enable; active high	
		[3]	reserve		
		[2]	BAND_SEL2	BAND trim	
		[1]	BAND_SEL1	BAND trim	
		[0]	BAND_SELO	BAND trim	
0xd6	08h	[7]	COM_RANGE_SEL	Command voltage range sel	
		[6]	reserve		
		[5]	CH1_DLY_SEL1_V12	ADC CLK DELAY TRIM	
		[4]	CH1_DLY_SELO_V12	ADC CLK DELAY TRIM	
		[3]	CK_DLY_SEL3_V12	ADC CLK DELAY TRIM	
		[2]	CK_DLY_SEL2_V12	ADC CLK DELAY TRIM	
		[1]	CK_DLY_SEL1_V12	ADC CLK DELAY TRIM	
		[0]	CK_DLY_SELO_V12	ADC CLK DELAY TRIM	
0xd7	00h	[7]	LDO12_EN	ADC LDO enable, active high	
		[6]	EN_VREF	VREF enable, active high	
		[5]	EN_VHL	VH & VL enable, active high	
		[4]	EN_SOG	SOG enable, active high	
		[3]	EN_MUX_SOG	SOG MUX enable, active high	
		[2]	EN_FLOW_SOG	SOG FLOW enable, active high	
		[1]	CH1_EN_V12	ADC CLK ENABLE, active high	
		[0]	ADC1_EN	ADC ENABLE, active high	
0xd8	A3h	[7]	SOGIN_SEL1	Sog input sel	
		[6]	SOGIN_SELO	Sog input sel	
		[5]	TRIM5_VT_SOG	Sog input voltage trim	
		[4]	TRIM4_VT_SOG	Sog input voltage trim	
		[3]	TRIM3_VT_SOG	Sog input voltage trim	

		[2]	TRIM2_VT_SOG	Sog input voltage trim	
		[1]	TRIM1_VT_SOG	Sog input voltage trim	
		[0]	TRIM0_VT_SOG	Sog input voltage trim	
0xd9	40h	[7]	reserve		
		[6]	TRIM2_VIN2_SOG	Sog input voltage trim	
		[5]	TRIM1_VIN2_SOG	Sog input voltage trim	
		[4]	TRIM0_VIN2_SOG	Sog input voltage trim	
		[3]	TRIM_DRV	ADC driver trim	
		[2]	TRIM3_IBADC	ADC bias trim	
		[1]	TRIM2_IBADC	ADC bias trim	
		[0]	TRIM1_IBADC	ADC bias trim	
0xda	29	[7]	IBUF_SEL1	ADC bias trim	
		[6]	IBUF_SEL0	ADC bias trim	
		[5]	TRIM_FB_D5	ADC input range trim	
		[4]	TRIM_FB_D4	ADC input range trim	
		[3]	TRIM_FB_D3	ADC input range trim	
		[2]	TRIM_FB_D2	ADC input range trim	
		[1]	TRIM_FB_D1	ADC input range trim	
		[0]	TRIM_FB_D0	ADC input range trim	
0xdb	00h	[7]	TRIM_VREF1_SOG	Sog vref trim	
		[6]	TRIM_VREFH_D2	Vrefh trim	
		[5]	TRIM_VREFH_D1	Vrefh trim	
		[4]	TRIM_VREFH_D0	Vrefh trim	
		[3]	TRIM_VREFL_D3	Vrefl trim	
		[2]	TRIM_VREFL_D2	Vrefl trim	
		[1]	TRIM_VREFL_D1	Vrefl trim	
		[0]	TRIM_VREFL_D0	Vrefl trim	
0xdc	00h	[7]	ADC1_S1CAL1	reserve	
		[6]	ADC1_S1CAL0	reserve	
		[5]	CVBS_SEL1	CVBS input sel	
		[4]	CVBS_SEL0	CVBS input sel	
		[3]	VC_DRV_SEL	Adc bias trim	
		[2]	VC_SEL2	Adc bias trim	
		[1]	VC_SEL1	Adc bias trim	
		[0]	VC_SEL0	Adc bias trim	
0xdd	00h	[7:0]	sync_level	Synctip level. read only	
0xde	00h	[7:0]	blank_level	blank level. read only	
0xdf	00h	[7:0]	sync_midpt	Sync midpt level. read only	
0xe0	00h	[7:0]	Cagc_val	Chroma agc value. read only	
0xe1	00h	[7:0]	Max_dat[7:0]	Maximal data in a field. Read only	

0xe3	00h	[1:0]	Max_dat[9:8]	Maximal data in a field. Read only	
------	-----	-------	--------------	------------------------------------	--

6.4 Video Process Register(I2C Address: 0xB4 MCU Address: 0xFFXX)

Addr	Val	Bits	Name	Description	App note
0xB0	47h	[7]	snowflower_en	无信号时输出雪花使能, 高有效	效果相关, 开放调试。
		[6]	DENOISE_EN	Denoise enable active high 自适应降噪模块使能, see 0xc9	
		[5]	BLUE_SCREEN_EN	Blue screen enable active high 蓝屏使能开关	
		[4]	BLE_EN	Black level extension enable active high 黑电平延伸使能	
		[3]	wle_en	White level extension enable active high 白电平延伸使能	
		[2]	DLTI_EN	Digital Luminance Transient Improvement enable active high See 0xB9, 0xBA	
		[1]	DCTI_EN	Digital Chrominance Transient Improvement enable active high See 0xb9, 0xbc, 0xbd	
		[0]	PEAKING_EN	PEAKING enable active high See 0xb2~0xb8	
0xB1	0Ch	[7]	dat_valid_bypass	0. vde yuv data sampled by den 1. bypass valid signal	效果相关, 开放调试。
		[6]	Col_weak_bypass	Color weaking bypass 弱亮度消色处理, see 0xe7	
		[5]	VP_bypass	Video Process bypass VP bypass 总开关	
		[4]	vde_bypass	VDE bypass	
		[3]	Firy_bypass	Fir_9tap_y bypass 亮度多用途 9 阶 FIR 滤波器开关, 默认关闭 系数控制 0xde~0xe6	
		[2]	Firc_bypass	Fir_9tap_c bypass 色度多用途 9 阶 FIR 滤波器开关, 默认关闭	
		[1]	Col_mask_bypass	Color mask bypass 弱色度消色处理, see 0xc7[1:0], 0xc8	
		[0]	gle_bypass	Green level enpansion bypass, see 0xc2~0xc7	
0xB2	10h	[7:0]	PEAKING_COEFH	Peaking high freq gain. 0~ff 代表 0~4 倍	效果相关, 开放调试。
0xB3	10h	[7:0]	PEAKING_COEFM	Peaking medium freq gain. 0~ff 代表 0~4 倍	效果相关, 开放调试。
0xB4	10h	[7:0]	PEAKING_COEFL	Peaking low freq gain. 0~ff 代表 0~4 倍	效果相关, 开放调试。
0xB5	60h	[6:0]	PEAKING_CORING_MA	Peak coring maximum value	效果相关, 开放调试。

			X		
0xB6	10h	[6:0]	PEAKING_CORING_H_MIN	Peak coring minimum value for high freq band	效果相关，开放调试。
0xB7	10h	[7]	PEAKING_LMT_BYPASS	Peaking 输出模式选择 1-sharp 模式 0-normal 模式	效果相关，开放调试。
		[6:0]	PEAKING_CORING_M_MIN	Peak coring minimum value for median freq band	
0xB8	10h	[7]	PEAKING_YCBCR_MODE	Peaking ycbcr mode Peaking 数据格式选择 1-原始数据 0-数据限幅到 0~255	效果相关，开放调试。
		[6:0]	PEAKING_CORING_L_MIN	Peak coring minimum value for low freq band	
0xB9	22h	[6]	PRE_DCTI_EN	PRE DCTI enable	效果相关，开放调试。
		[5]	DCTI_SEL	DCTI 窗口大小选择。 0: 5 个点的窗口, 1: 9 个点的窗口	
		[4]	DLTI_SEL	DLTI 窗口大小选择。 0: 5 个点的窗口, 1: 9 个点的窗口	
		[3:0]	DLTI_GAIN	DLTI gain. 0~F 代表 0~1 倍	
0xBA	20h	[6:0]	DLTI_THRSHD	DLTI coring threshold	效果相关，开放调试。
0xBB	22h	[7:4]	DCTI_CBGAIN	DCTI cb gain 0~F 代表 0~1 倍	效果相关，开放调试。
		[3:0]	DCTI_CRGAIN	DCTI Cr gain 0~F 代表 0~1 倍	
0xBC	20h	[7:0]	DCTI_CBTHRSHD	DCTI Cb coring threshold	效果相关，开放调试。
0xBD	20h	[5:0]	DCTI_CRTHRSHD	DCTI Cr coring threshold	效果相关，开放调试。
0xBE	20h	[7:0]	BLE_THR	BLE threshold	固定为 0X00。
0xBF	20h	[7:0]	BLE_GAIN	BLE gain	固定为 0X20。
0xC0	E0h	[7:0]	WLE_THR	WLE threshold	固定为 0XE0。
0xC1	20h	[7:0]	WLE_GAIN	WLE gain	固定为 0X20。
0xC2	B5h	[7:0]	GLE_COFF_SIN	Green color axis coefficient (Sin)	固定为 0XB5。
0xC3	B5h	[7:0]	GLE_COFF_COS	Green color axis coefficient (Cos)	固定为 0XB5。
0xC4	FFh	[7:0]	GLE_Y_min	Minimum Y enhance for GLE ymin 用于防止亮度溢出,ymin 越大对溢出抑制约不明显,	固定为 0XFF。
0xC5	FFh	[7:0]	GLE_coff_tan	GLE tangent angle coefficient 用于控制绿色增强的范围,数值越大增强越多	固定为 0XFF。
0xC6	99h	[7:0]	GLE_gain	GLE gain 绿电平增强 (GLE),gain 用于控制绿色增强程度	固定为 0X6C。
0XC7	31h	[5:4]	GLE_tan_cita	GLE tangent angle select tan 用于防止亮度溢出,越大对抑制越不明显	图像效果相关，开放调试。
		[1:0]	col_cmp_ratio	消色程度控制	
0XC8	10h	[7:0]	col_lowlvl_thr	弱色判断门限	图像效果相关，开放调试。 1.减轻暗场噪点可以调试此

					门限值。
0XC9	40h	[7:0]	Dsmth_thr	VP 降噪门限	图像效果相关，开放调试。
0xCA	00h	[7:0]	offset_position	对比度中间电平	固定为 0x00。
0xCB	00h	[7]	spatial_bypass	Spatial dithring bypass	图像效果相关，开放调试。
		[6]	truncate_sel	Dithring output mode 1-truncate (w/o round) 0-dither	
		[5:4]	spatial_mux_b	when spatial_mux==2'b00, 9 bits to 8 bits dithring when spatial_mux==2'b01, 9 bits to 7 bits dithring when spatial_mux==2'b10, 9 bits to 6 bits dithring when spatial_mux==2'b11, 9 bits to 5 bits dithring	
		[3:2]	spatial_mux_g	The function same to spatial_mux_b	
		[1:0]	spatial_mux_r	The function same to spatial_mux_b	
0xCC	00h	[7]	tmporal_bypass	Temporal dithering bypass	测试图控制位，一般固定设置为 0。
		[6]	rgb_test_en	rgb test vector output enable when set to 1, the output dat is controlled only by 0xce,0xcf,0xd0	
		[5:4]	tmporal_mux_b	when tmporal_mux==2'b00: spatial_mux==2'b00, 9 bits to 8 bits dithring spatial_mux==2'b01, 9 bits to 7 bits dithring spatial_mux==2'b10, 9 bits to 6 bits dithring spatial_mux==2'b11, 9 bits to 5 bits dithring when tmporal_mux==2'b01: spatial_mux==2'b00, 9 bits to 7 bits dithring spatial_mux==2'b01, 9 bits to 6 bits dithring spatial_mux==2'b10, 9 bits to 5 bits dithring spatial_mux==2'b11, 9 bits to 4 bits dithring when tmporal_mux==others: spatial_mux==2'b00, 9 bits to 6 bits dithring spatial_mux==2'b01, 9 bits to 5 bits dithring spatial_mux==2'b10, 9 bits to 4 bits dithring spatial_mux==2'b11, 9 bits to 3 bits dithring	

		[3:2]	tmporal_mux_g	The function same to tmporal_mux_b	
		[1:0]	tmporal_mux_r	The function same to tmporal_mux_b	
0xCD	2Dh	[7:0]	diff_reg	noise value used for temporal dithering.	图像效果相关，开放调试。
0xCE	00h	[7:0]	r_test_reg	when rgb_test_en=1, output R value, or else when BLUE_SCREEN_EN=1, output Y	固定为 0X13。
0xCF	00h	[7:0]	g_test_reg	when rgb_test_en=1, output G value, or else when BLUE_SCREEN_EN=1, output Cb	固定为 0X72。
0xD0	00h	[7:0]	b_test_reg	when rgb_test_en=1, output B value, or else when BLUE_SCREEN_EN=1, output Cr	此设定值为标准的无信号蓝屏。 可作调试使用
0xD1	40h	[6:4]	rgb_protect_sel	YCbCr dat format control when convert to RGB, default 10 Bit[6] 0 - Y is controlled by bit[5:4], 1-Y is down limited to 16 Bit[5:4] 00 - raw dat 01 - Y is raw dat, C is down limited 10/11- Y in range[16,235], C in range[16,240]	固定为 0X40。
		[3:0]	reserved		
0xD2	4Fh	[7]	vde_test_vector_sel	VDE test vector select 0-output normal signal 1-output test vecotr(controlled by [3:0])	固定为 0X4F，软件动态控制。
		[6:4]	vde_protect_sel	YCbCr dat format control in VDE module. Description is the same as rgb_protect_sel	
		[3:0]	vde_out_sel	VDE output select	
0xD3	80h	[7:0]	Contrast	Contrast	图像效果相关，开放调试。
0xD4	80h	[7:0]	Brightness	Brightness	图像效果相关，开放调试。
0xD5	00h	[7:0]	Hue	Hue	图像效果相关，开放调试。
0xD6	3Fh	[7:0]	Saturation	Saturation	图像效果相关，开放调试。
0xD7	80h	[7:0]	Y_Cb_Cr_TEST_REG[7:0] COL_LUM_CFG_REG[7:0]	Y value of a test vector when vde_out_sel =7 正常显示模式时，若 d8h[7]为 0，该寄存器可设置出蓝屏的时间常数，该值越大从无信号到有信号时，出蓝屏的时间越长，反之则越短。	
0xD8	80h	[7:0]	Y_Cb_Cr_TEST_REG[15:8] LUM_STEP_REG[7:0]	Cb value of a test vector when vde_out_sel =7 正常显示模式时，若 d8h[7]为 1 则 bypass 蓝屏信号，若为 0 则根据 d7h[7:0]设置出蓝屏的时间延迟。	固定为 0X80
0xD9	80h	[7:0]	Y_Cb_Cr_TEST_REG[23:16]	Cr value of a test vector when vde_out_sel =7	固定为 0X80
0xDA	5Ch	[6]	Blue_dva_en	出蓝屏（雪花）信号时，有效窗口使能。为 1 时，DEN 信号为 1 时，出蓝屏（雪花），DEN 为 0 部分出黑信号。一般情况下配 1。	
		[5:4]	snowflower_big_s	雪花信号大小选择，可以选择在水平方向不同大	

			el	小的雪花点 0: 最小; 1: 次小; 2: 次大; 3: 最大。	
		[3:0]	snowflower_sel	雪花信号源选择, 可以改变雪花的亮点与暗点的比例分配。	
0xDB	1Bh	[7:0]	reserved	可以用来做变量寄存器	
0xDC	14h	[7:0]	reserved	可以用来做变量寄存器	
0xDD	00h	[7:0]	data_en delay	data_en signal delay control	图像位置相关, 开放调试。
0xDE	0Eh	[7:0]	Coef0	9 tap FIR coefficient	固定为 0X00
0xDF	0Eh	[7:0]	Coef1		固定为 0X00
0xE0	0Eh	[7:0]	Coef2		固定为 0X00
0xE1	0Eh	[7:0]	Coef3		固定为 0X40
0xE2	0Eh	[7:0]	Coef4		固定为 0X20
0xE3	0Eh	[7:0]	Coef5		固定为 0X00
0xE4	0Eh	[7:0]	Coef6		固定为 0X20
0xE5	0Eh	[7:0]	Coef7		固定为 0X00
0xE6	0Eh	[7:0]	Coef8		固定为 0X00
0xE7	50h	[7:6]	c_slope_ctrl_by_Y		固定为 0X50
		[5:0]	c_max_Ythr		
0xE8	10h	[7:4]	PRE_DCTI_CBGAIN	PRE_DCTI cb gain	固定为 0XFF,
		[3:0]	PRE_DCTI_CRGAIN	PRE_DCTI cr gain	此寄存器为调试色阶分界线的宽度, 可固定为 0XFF.
0xE9	22h	[7:0]	PRE_DCTI_CBTBTHRS HD	PRE_DCTI Cb coring	固定为 0X22 结合 0XE8 调整
0xEA	20h	[7:0]	PRE_DCTI_CRTBTHRS HD	PRE_DCTI Cr coring	固定为 0X20 结合 0XE8 调整
0xF0	65h	[7:0]	k00_reg[7:0]	Color matrix 系数, 计算公式如下	和图像色彩相关, 开发调试。
0XF1	C0h	[7:0]	k01_reg[7:0]	所有系数均为 2 的补码形式, 10bits 代表[-2,2]	和图像色彩相关, 开发调试。
0XF2	DAh	[7:0]	k02_reg[7:0]	$R_{out} = K_{00} * R_{in} + K_{01} * G_{in} + K_{02} * B_{in}$	和图像色彩相关, 开发调试。
0XF3	0Dh	[7:0]	k10_reg[7:0]	$G_{out} = K_{10} * R_{in} + K_{11} * G_{in} + K_{12} * B_{in}$	和图像色彩相关, 开发调试。
0XF4	3Dh	[1:0]	k00_reg[9:8]	$B_{out} = K_{20} * R_{in} + K_{21} * G_{in} + K_{22} * B_{in}$	和图像色彩相关, 开发调试。
		[3:2]	k01_reg[9:8]		
		[5:4]	k02_reg[9:8]	k00_reg[9:0]={REGF4H[1:0],REGF0H[7:0]};	
		[7:6]	k10_reg[9:8]	k01_reg[9:0]={REGF4H[3:2],REGF1H[7:0]};	

0XF4	19h	[7:0]	k11_reg[7:0]	k02_reg[9:0]={REGF4H[5:4],REGF2H[7:0]};	和图像色彩相关，开发调试。
0XF5	DAh	[7:0]	k12_reg[7:0]		和图像色彩相关，开发调试。
0XF6	CDh	[7:0]	k20_reg[7:0]	k10_reg[9:0]={REGF4H[7:6],REGF3H[7:0]};	和图像色彩相关，开发调试。
0XF7	1Ah	[7:0]	k21_reg[7:0]	k11_reg[9:0]={REGF9H[1:0],REGF5H[7:0]};	和图像色彩相关，开发调试。
0XF8	3Dh	[1:0]	k11_reg[9:8]	k12_reg[9:0]={REGF9H[3:2],REGF6H[7:0]};	和图像色彩相关，开发调试。
		[3:2]	k12_reg[9:8]		
		[5:4]	k20_reg[9:8]	k20_reg[9:0]={REGF9H[5:4],REGF7H[7:0]};	
		[7:6]	k21_reg[9:8]	k21_reg[9:0]={REGF9H[7:6],REGF8H[7:0]};	
0XF9	19h	[7:0]	k22_reg[7:0]	k22_reg[9:0]={REGFBH[1:0],REGFAH[7:0]};	和图像色彩相关，开发调试。
0XFA	81h	[7]	col_matrix_en	1 允许 COLOR MATRIX 工作 0 BYPASS COLOR MATRIX	和图像色彩相关，开发调试。
		[6:2]	reserved		
		[1:0]	k22_reg[9:8]		

6.5 GAMMA Register(I2C Address: 0xB4 MCU Address: 0xFFXX)

Addr	Val	Bits	Name	Description	App note
0x00	03h	[1]	CUSTOM_EN	1'b0: custom curve enable 1'b1: using setting curve	
		[0]	LCD_EN	GAMMA enable	
0x01	07h	[7:0]	custom_r_val01	custom_red_value_01	
0x02	10h	[7:0]	custom_r_val02	custom_red_value_02	
0x03	1Ah	[7:0]	custom_r_val03	custom_red_value_03	
0x04	23h	[7:0]	custom_r_val04	custom_red_value_04	
0x05	2Dh	[7:0]	custom_r_val05	custom_red_value_05	
0x06	35h	[7:0]	custom_r_val06	custom_red_value_06	
0x07	3Ch	[7:0]	custom_r_val07	custom_red_value_07	
0x08	43h	[7:0]	custom_r_val08	custom_red_value_08	
0x09	4Ah	[7:0]	custom_r_val09	custom_red_value_09	
0x0A	51h	[7:0]	custom_r_val10	custom_red_value_10	
0x0B	56h	[7:0]	custom_r_val11	custom_red_value_11	
0x0C	5Ch	[7:0]	custom_r_val12	custom_red_value_12	
0x0D	62h	[7:0]	custom_r_val13	custom_red_value_13	
0x0E	68h	[7:0]	custom_r_val14	custom_red_value_14	
0x0F	6Eh	[7:0]	custom_r_val15	custom_red_value_15	

0x10	74h	[7:0]	custom_r_val16	custom_red_value_16	
0x11	7Ah	[7:0]	custom_r_val17	custom_red_value_17	
0x12	81h	[7:0]	custom_r_val18	custom_red_value_18	
0x13	88h	[7:0]	custom_r_val19	custom_red_value_19	
0x14	8Fh	[7:0]	custom_r_val20	custom_red_value_20	
0x15	96h	[7:0]	custom_r_val21	custom_red_value_21	
0x16	9Dh	[7:0]	custom_r_val22	custom_red_value_22	
0x17	A4h	[7:0]	custom_r_val23	custom_red_value_23	
0x18	ABh	[7:0]	custom_r_val24	custom_red_value_24	
0x19	B2h	[7:0]	custom_r_val25	custom_red_value_25	
0x1A	BAh	[7:0]	custom_r_val26	custom_red_value_26	
0x1B	C1h	[7:0]	custom_r_val27	custom_red_value_27	
0x1C	C9h	[7:0]	custom_r_val28	custom_red_value_28	
0x1D	D2h	[7:0]	custom_r_val29	custom_red_value_29	
0x1E	DDh	[7:0]	custom_r_val30	custom_red_value_30	
0x1F	ECh	[7:0]	custom_r_val31	custom_red_value_31	
0x20	07h	[7:0]	custom_g_val01	custom_green_value_01	
0x21	10h	[7:0]	custom_g_val02	custom_green_value_02	
0x22	1Ah	[7:0]	custom_g_val03	custom_green_value_03	
0x23	23h	[7:0]	custom_g_val04	custom_green_value_04	
0x24	2Dh	[7:0]	custom_g_val05	custom_green_value_05	
0x25	35h	[7:0]	custom_g_val06	custom_green_value_06	
0x26	3Ch	[7:0]	custom_g_val07	custom_green_value_07	
0x27	43h	[7:0]	custom_g_val08	custom_green_value_08	
0x28	4Ah	[7:0]	custom_g_val09	custom_green_value_09	
0x29	51h	[7:0]	custom_g_val10	custom_green_value_10	
0x2A	56h	[7:0]	custom_g_val11	custom_green_value_11	
0x2B	5Ch	[7:0]	custom_g_val12	custom_green_value_12	
0x2C	62h	[7:0]	custom_g_val13	custom_green_value_13	
0x2D	68h	[7:0]	custom_g_val14	custom_green_value_14	
0x2E	6Eh	[7:0]	custom_g_val15	custom_green_value_15	
0x2F	74h	[7:0]	custom_g_val16	custom_green_value_16	
0x30	7Ah	[7:0]	custom_g_val17	custom_green_value_17	
0x31	81h	[7:0]	custom_g_val18	custom_green_value_18	
0x32	88h	[7:0]	custom_g_val19	custom_green_value_19	
0x33	8Fh	[7:0]	custom_g_val20	custom_green_value_20	
0x34	96h	[7:0]	custom_g_val21	custom_green_value_21	
0x35	9Dh	[7:0]	custom_g_val22	custom_green_value_22	
0x36	A4h	[7:0]	custom_g_val23	custom_green_value_23	
0x37	ABh	[7:0]	custom_g_val24	custom_green_value_24	
0x38	B2h	[7:0]	custom_g_val25	custom_green_value_25	

0x39	BAh	[7:0]	custom_g_val26	custom_green_value_26	
0x3A	C1h	[7:0]	custom_g_val27	custom_green_value_27	
0x3B	C9h	[7:0]	custom_g_val28	custom_green_value_28	
0x3C	D2h	[7:0]	custom_g_val29	custom_green_value_29	
0x3D	DDh	[7:0]	custom_g_val30	custom_green_value_30	
0x3E	ECh	[7:0]	custom_g_val31	custom_green_value_31	
0x3F	07h	[7:0]	custom_b_val01	custom_blue_value_01	
0x40	10h	[7:0]	custom_b_val02	custom_blue_value_02	
0x41	1Ah	[7:0]	custom_b_val03	custom_blue_value_03	
0x42	23h	[7:0]	custom_b_val04	custom_blue_value_04	
0x43	2Dh	[7:0]	custom_b_val05	custom_blue_value_05	
0x44	35h	[7:0]	custom_b_val06	custom_blue_value_06	
0x45	3Ch	[7:0]	custom_b_val07	custom_blue_value_07	
0x46	43h	[7:0]	custom_b_val08	custom_blue_value_08	
0x47	4Ah	[7:0]	custom_b_val09	custom_blue_value_09	
0x48	51h	[7:0]	custom_b_val10	custom_blue_value_10	
0x49	56h	[7:0]	custom_b_val11	custom_blue_value_11	
0x4A	5Ch	[7:0]	custom_b_val12	custom_blue_value_12	
0x4B	62h	[7:0]	custom_b_val13	custom_blue_value_13	
0x4C	68h	[7:0]	custom_b_val14	custom_blue_value_14	
0x4D	6Eh	[7:0]	custom_b_val15	custom_blue_value_15	
0x4E	74h	[7:0]	custom_b_val16	custom_blue_value_16	
0x4F	7Ah	[7:0]	custom_b_val17	custom_blue_value_17	
0x50	81h	[7:0]	custom_b_val18	custom_blue_value_18	
0x51	88h	[7:0]	custom_b_val19	custom_blue_value_19	
0x52	8Fh	[7:0]	custom_b_val20	custom_blue_value_20	
0x53	96h	[7:0]	custom_b_val21	custom_blue_value_21	
0x54	9Dh	[7:0]	custom_b_val22	custom_blue_value_22	
0x55	A4h	[7:0]	custom_b_val23	custom_blue_value_23	
0x56	ABh	[7:0]	custom_b_val24	custom_blue_value_24	
0x57	B2h	[7:0]	custom_b_val25	custom_blue_value_25	
0x58	BAh	[7:0]	custom_b_val26	custom_blue_value_26	
0x59	C1h	[7:0]	custom_b_val27	custom_blue_value_27	
0x5A	C9h	[7:0]	custom_b_val28	custom_blue_value_28	
0x5B	D2h	[7:0]	custom_b_val29	custom_blue_value_29	
0x5C	DDh	[7:0]	custom_b_val30	custom_blue_value_30	
0x5D	ECh	[7:0]	custom_b_val31	custom_blue_value_31	
0x5E	FFh	[7:0]	custom_r_val32	custom_red_value_32	
0x5F	FFh	[7:0]	custom_g_val32	custom_green_value_32	
0x60	FFh	[7:0]	custom_b_val32	custom_blue_value_32	

6.6 RCRT Register(MCU Address: 0xFFXX MCU access only)

Sub Address	Default Value	Bits	Name	Description	App note
FF61	01h	[7:0]	DISPERSION	The dispersion of the hardware examination pulse width, 红外遥控、倒车协议接收公用	
FF62	00h	[7:0]	PRE_DIV	Divide the clock (24M) used for counter: 0: clock 1: clock/4 2: clock/6 ... n: clock/2(n+1) 红外遥控、倒车协议公用	
FF63	01h	[7:0]	FLTR_WIDTH	The filter width of input pulse, 红外遥控、倒车协议公用	
FF64	00h	[7:3]	Reserved	Reserved (read only)	
		[2:0]	VALID_BIT	Select the valid range of the data used by hardware. 3'b000: [7:0] 3'b001: [8:1] 3'b010: [9:2] 3'b011: [10:3] 3'b100: [11:4] 3'b101: [12:5] 3'b110: [13:6] 3'b111: [14:7] 红外遥控、倒车协议公用	
FF65	14h	[7:0]	START_VALUEH	When RC5_CS=1, the value represents 1/2 of the positive pulse width of a RC5 code bit. Otherwise it represents the high pulse width of start code. 红外遥控、倒车协议公用	

FF66	0Ah	[7:0]	START_VALUEL	When RC5_CS=1, the value represents the negative pulse width of a RC5 code bit. Otherwise it represents the low pulse width of start code. 红外遥控、倒车协议公用	
FF67	04h	[7:0]	ONE_VALUEH	The width of positive pulse of “1” code. 倒车协议无效	
FF68	08h	[7:0]	ONE_VALUEL	The width of negative pulse of “1” code. 倒车协议无效	
FF69	04h	[7:0]	ZERO_VALUEH	The width of positive pulse of “0” code. 红外遥控、倒车协议公用	
FF6A	02h	[7:0]	ZERO_VALUEL	The width of negative pulse of “0” code. 红外遥控、倒车协议公用	
FF6B	04h	[7:0]	RP0_VALUEH	The width of positive pulse of “rp0” status, which is the start status of the repeat code.	
FF6C	B0h	[7:0]	RP0_VALUEL	The width of negative pulse of “rp0” status, which is the start status of the repeat code.	
FF6D	04h	[7:0]	RP1_VALUEH	The width of positive pulse of “rp1” status, which follows the “rp0”.	
FF6E	B6h	[7:0]	RP1_VALUEL	The width of negative pulse of “rp1” status, which follows the “rp0”.	
FF6F	14h	[7:0]	RP2_3_VALUEH	The width of positive pulse of “rp23” status, which follows the “rp1”.	

FF70	08h	[7:0]	RP2_3_VALUEL	The width of negative pulse of “rp23” status, which follows the “rp1”.	
FF71	04h	[7:0]	RP4_5_VALUEH	The width of positive pulse of “rp45” status, which follows the “rp23”.	
FF72	FFh	[7:0]	RP4_5_VALUEL	The width of negative pulse of “rp45” status, which follows the “rp23”.	
FF73	00h	[7:0]	USER_CODE1_L	User code1 low 8 bits	
FF74	00h	[7:0]	USER_CODE1_H	User code1 high 8 bits	
FF75	00h	[7:0]	USER_CODE2_L	User code2 low 8 bits	
FF76	00h	[7:0]	USER_CODE2_H	User code2 high 8 bits	
FF77	FFh	[7:0]	KEY_RELEASE_H	Set the key release time high 8 bits.	
FF78	FFh	[7:0]	KEY_RELEASE_L	Set the key release time low 8 bits.	
FF79	01h	[7:0]	INT_NUM	Set the interrupt position. (It's for software decoder)	
FF7A	00h	[7]	retr_sig_norm	普通红外遥控解码、倒车协议解码选择 0: 普通红外 1: 倒车协议	
		[6]	CUSM_JUD_SEL_N	是否判断客户码选择 0: 判断客户码; 1: 不判断客户码。	
		[5]	CUSM_NOT_JUD_SEL_N	是否判断客户码反码选择 0: 判断客户码反码; 1: 不判断客户码反码。	
		[4]	DATA_NOT_JUD_SEL_N	是否判断数据反码选择 0: 判断数据反码; 1: 不判断数据反码。	
		[3]	USER_CODE_SEL	0: user code select 8 bit 1: user code select 13 bit	

		[2]	NEC_RELEASE_INT_EN	NEC release interrupt enable	
		[1:0]	MODE_SEL	00: Not used. 01: Software decode mode. 10: Hardware & software decode mode. 11: Hardware decode mode.	
FF7B	00h	[7:0]	NEC_2BIT_PULSEL	NEC code 2bit pulse time counter low 8 bits	
FF7C	FFh	[7:0]	NEC_2BIT_PULSEH	NEC code 2bit pulse time counter high 8 bits	
FF7D	01h	[7:0]	DISPER_RP_5L_DELTA	Repeat 5 低电平附加 delta 值，由于一般 repeat 5 低电平时间较长，在原来的 dispersion 上加上改制为 repeat 5 低电平。	
FF7E	00h	[7:0]	USER_CODE3_L	User code3 low 8 bits，同 user_code1_l	
FF7F	00h	[7:0]	USER_CODE3_H	User code3 high 8 bits，同 user_code1_h	
FF80	00h	[7:0]	USER_CODE4_L	User code4 low 8 bits，同 user_code2_l	
FF81	00h	[7:0]	USER_CODE4_H	User code4 high 8 bits，同 user_code2_h	
FF82	00h	[7:4]	Reserved	Reserved	
		[3]	RC5_CS	Code mode 0: NEC 1: RC5	
		[2]	DATA_INV	Decoded data polarity 0: normal 1: inversion	
		[1]	CUSTOM_INV	User code polarity 0: normal 1: inversion	
		[0]	PULSE_INV	Input pulse polarity 0: normal 1: inversion	
FF83	00h	[7:0]	CUSTOM_READ	只读，客户码读取，低 8 位。	

FF84	00	[7:5]	RESERVED	RESERVED	
		[4:0]	CUSTOM_READ	只读，客户码读取，高 5 位。	
FF85	00h	[7:0]	CUSTOM_NOT_READ	只读，客户码反码读取，低 8 位。	
FF86	00	[7:5]	RESERVED	RESERVED	
		[4:0]	CUSTOM_NOT_READ	只读，客户码反码读取，高 5 位。	
FF87	00	[7:4]	RESERVED	RESERVED	
		[3:0]	RCRT_CS_OUT	只读，解码状态	
FF88	00	[7:0]	STARTH_DIS	头码高电平附加 delta 值	
FF89	00	[7:0]	STARTL_DIS	头码低电平附加 delta 值	
FF8A	00	[7:0]	RP23H_DIS	Repeat 23 高电平附加 delta 值	
FF8B	00	[7:0]	RP23L_DIS	Repeat 23 低电平附加 delta 值	
FF8C	00	[7:6]	reserved		
		[5:0]	bit_num	接收数据个数，最大为 32 个	
FF8D	00	[7:0]	Receive_data[31:24]	接收数据，高 8 位	
FF8E	00	[7:0]	Receive_data[23:16]	接收数据，中 8 位	
FF8F	00	[7:0]	Receive_data[15:8]	接收数据，中 8 位	
FF90	00	[7:0]	Receive_data[7:0]	接收数据，低 8 位	
FF91	00			发送数据使能、开始标志	
		[7:0]	Trans_wr_flag	对该寄存器进行一次写操作，数据向外发送一次。	
FF92	00	[7:2]	reserved		
		[1]	Trans_out_inv	发送数据输出高低取反选择 0: 不取反 1: 取反	
		[0]	Trans_mod	发送数据模式选择 0: 类 NEC 1: 类 RC5	
FF93	00	[7:0]	trans_start0_l	发送数据开始码第一个脉冲宽度，低 8 位	
FF94	00	[7:0]	trans_start0_h	发送数据开始码第一个脉冲宽度，高 8 位	

FF95	00	[7:0]	trans_start1_l	发送数据开始码第二个脉冲宽度，低 8 位	
FF96	00	[7:0]	trans_start1_h	发送数据开始码第二个脉冲宽度，高 8 位	
FF97	00	[7:0]	trans_bit0_0_l	发送数据 bit0 第一个脉冲的宽度，低 8 位，类 RC5 码为 0 或 1 的第一个脉冲宽度	
FF98	00	[7:0]	trans_bit0_0_h	发送数据 bit0 第一个脉冲的宽度，高 8 位，类 RC5 码为 0 或 1 的第一个脉冲宽度	
FF99	00	[7:0]	trans_bit0_l	发送数据 bit0 整个脉冲的宽度，低 8 位，类 RC5 码为 0 或 1 的整个脉冲宽度。	
FF9a	00	[7:0]	trans_bit0_h	发送数据 bit0 整个脉冲的宽度，高 8 位，类 RC5 码为 0 或 1 的整个脉冲宽度。	
FF9b	00	[7:0]	trans_bit1_0_l	发送数据 bit1 第一个脉冲的宽度，低 8 位，类 RC5 码无效	
FF9c	00	[7:0]	trans_bit1_0_h	发送数据 bit1 第一个脉冲的宽度，高 8 位，类 RC5 码无效	
FF9d	00	[7:0]	trans_bit1_l	发送数据 bit1 整个脉冲的宽度，低 8 位，类 RC5 码无效	
FF9e	00	[7:0]	trans_bit1_h	发送数据 bit1 整个脉冲的宽度，高 8 位，类 RC5 码无效	
FF9f	00	[7:6]	reserved		
		[5:0]	trans_num[5:0]	发送数据个数，最大为 32 个	
FFa0	00	[7:0]	trans_data[7:0]	发送数据值，低 8 位	
FFa1	00	[7:0]	trans_data[15:8]	发送数据值，中 8 位	
FFa2	00	[7:0]	trans_data[23:16]	发送数据值，中 8 位	
FFa3	00	[7:0]	trans_data[31:24]	发送数据值，高 8 位	

RCRT (红外)

外部中断: int0_n, 中断号 0

RCRT SFRs

1. CODE_VALUE_BUF1

Address = E1H

Default value = FFH

Bit	Default value	Name	Function
[7:0]	FFH	CODE_VALUE_BUF1	Reserve the code value used by the hardware. (read only)

2. CODE_VALUE_BUF0

Address = E2H

Default value = FFH

Bit	Default value	Name	Function
[7:0]	FFH	CODE_VALUE_BUF0	Reserve the code value used by the hardware. (read only)

3. CODE_VALUE

Address = E3H

Default value = FFH

Bit	Default value	Name	Function
[7:0]	FFH	CODE_VALUE	<p>This register reserves code status, can be used for hardware decoder or software decoder. (read only)</p> <p>The detail as follows:</p> <p>0x00: "0" receive;</p> <p>0x01: "1" receive;</p> <p>0x02: start receive;</p> <p>0x03: data end as 1;</p> <p>0x04: data end as 0;</p> <p>0x05: simple repeat start;</p> <p>0x06: simple repeat continue;</p> <p>0xff: Default</p>

4. KEY_VALUE

Address = E4H

Default value = FFH

Bit	Default value	Name	Function
[7:0]	FFH	KEY_VALUE	Set by hardware and clear by software read.

5. RCRT_STATUS

Address = E5H

Default value = FFH

Bit	Default value	Name	Function
[7:3]	0	Reserved	Reserved

[2]	0	INT_POL_FLAG	Read only
[1]	0	RP_FLAG	Read only
[0]	0	RELEASE_FLAG	Set by hardware and clear by software read.

NEC 码:

- (1)MCU 的时钟是 11.0592M
- (2)MCU 的时钟是 22.1184M
- (3)MCU 的时钟是 24M
- (4)MCU 的时钟是 25M

(二)RC5 码:

- (1)MCU 的时钟是 11.0592M
- (2)MCU 的时钟是 22.1184M
- (3)MCU 的时钟是 24M
- (4)MCU 的时钟是 25M

备注:

- 1) 红外中断给 MCU 的加了个非门。
- 2) 红外的时钟分频扩大了一倍, 可以达到 512 分频。

键值码表 NEC (uPD6121G)

按键	码值 0x	按键	码值
1	00	DEFAULT	17
2	01	CH+	12
3	02	CH-	13
4	03	VOL-	11
5	04	VOL+	10
6	05	REC	15
7	06	PP	0F
8	07	COLOR	1F
9	08	GAME	1A
0	09	OSD-SIZE	16
-/--	0A	MUTE	0C
RECALL	1C	POWER	14
ADJUST	18	ZOOM	0E
▲	0B	▼	19

键值码表 RC-5(SAA3010))

按键	码值	按键	码值
1	0x01	MUTE	0x0D
2	0x02	CH+	0x20
3	0x03	CH-	0x21
4	0x04	VOL-	0x11

5	0x05	VOL+	0x10
6	0x06	录象	0x1F
7	0x07	回放	0x17
8	0x08	暂停	0x16
9	0x09	停止	0x0B
0	0x00	快进	0x27
-/--	0x0A	快退	0x26
返回	0x29	搜台	0x1E
亮度 +	0x12	截图	0x0E
亮度 -	0x13	控制板	0x2D
微调+	0x2B	全屏	0x0F
微调-	0x2C	关机	0x25
TV/AV	0x38	自动/退出	0x0C

6.7 Scaler Register(I2C Address: 0xB8 MCU Address: 0xFCXX)

Addr	Val	Bits	Name	Description	App note
0x90	00h	[7]	CBCR_ALIGN	Cb and Cr mode select if the filter enable of Cb and Cr in 444 to 422 transfer set 1(CRCBFILTER_EN=1), if the CRCBFILTER_EN is set to 0, this register is useless. 1: Y0Cb0 Y1Cr0 Y2Cb2 Y3Cb2 0: Y0Cb0 Y1Cr1 Y2Cb2 Y3Cr3	
		[6]	CRCBFILTER_EN	Filter enable of Cb and Cr in 444 to 422 transfer 1: filter is enable 0: filter is disable	
		[5]	FIRSTISCR	First C is Cb or Cr select for YCbCr422 data in 444 to 422 transfer 1: data is YCr YCb YCr YCb 0: data is YCb YCr YCb YCr	
		[4]	CR_CUT_LSB	Cut last bit of Cr select, bit[7:0] -----→ {bit[7:1],1'b0} 1: bit[7:0] -----→ {bit[7:1],1'b0} 0: not cut	

		[3]	CB_CUT_LSB	Cut last bit of Cb select, bit[7:0] -----→ {bit[7:1],1'b0} 1: bit[7:0] -----→ {bit[7:1],1'b0} 0: not cut	
		[2]	Y_CUT_LSB	Cut last bit of Y select, bit[7:0] -----→ {bit[7:1],1'b0} 1: bit[7:0] -----→ {bit[7:1],1'b0} 0: not cut	
		[1]	VSYN_FULL_SEL	每场完整性选择, 保证每场的行数达到最小值 (vgate_stop) 0: 不做完整性处理; 1: 每场完整性处理。	
		[0]	PN_SEL_DISABLE	PAL & NTSC use same register 0: PAL use PAL registers while NTSC use NTSC registers; 1: PAL and NTSC use the same registers (NTSC registers).	
0x91	00h	[7]	FIELD_BYPASS(NTSC)	Field output of scaler select 1: field of video input select 0: field of scale output select	
		[6]	VSYN_BYPASS(NTSC)	Vertical synchronization output of scaler select 1: Vertical synchronization of video input select 0: Vertical synchronization of scale output select	
		[5]	HSYN_BYPASS(NTSC)	Horizontal synchronization output of scaler select 1: Horizontal synchronization of video input select 0: Horizontal synchronization of scale output select	
		[4]	FIELD_NO_CHANGE(NTSC)	Whether the odd field and even field is treated as the same, it is the same function to the deinterlacer. 1: the odd field and the even field is treated as the same, not do deinterlace 0: the odd field and the even field is treated as the different, do deinterlace	

		[3]	DATENA_INV(NTSC)	Data enable output of scaler invert select 1: invert 0: not invert	
		[2]	VSUNC_INV(NTSC)	Vertical synchronization output of scaler invert select 1: invert 0: not invert	
		[1]	HSUNC_INV(NTSC)	Horizontal synchronization output of scaler invert select 1: invert 0: not invert	
		[0]	FIELDINV(NTSC)	Field invert select 1: invert 0: not invert	
0x92	00h	[7]	COS_EN(NTSC)	非线性 scaler 使能	
		[6]		reserved	
		[5]	HU_SCA_BYPA(NTSC)	Horizontal up scale bypass(down scale or bypass), it determines if the up scale is used 1: up scale is bypassed 0: up scale is used	
		[4]	HD_SCA_BYPA(NTSC)	Horizontal down scale bypass(up scale or bypass), it determines if the down scale is used 1: down scale is bypassed 0: down scale is used	
		[3]	H_SCALE_BYPASS_SEL (NTSC)	Horizontal scale bypass select, including horizontal up scale and horizontal down scale. 1: horizontal down scale bypass select HD_SCA_BYPA (0xA2 bit0), horizontal up scale bypass select HU_SCA_BYPA (0xA2 bit1). 0: horizontal down scale bypass and horizontal up scale bypass select auto set which is determined by HFZ[15:0]	

		[2]	VSYN_DEL_FIELD(NTSC)	场（奇场或偶场）起始位置的延迟选择使能打开时，该信号表示选择的是奇场还是偶场。 1: 奇场 0: 偶场	
		[1]	VSYN_DEL_EN(NTSC)	场（奇场或偶场）起始位置的延迟选择使能，当使能打开时，可改变奇场或偶场的起始位置，以改变重建后输出场的位置。 1: enable; 0: disable.	
		[0]	DEN_BYPASS(NTSC)	Data enable bypass select 1: bypass 0: not bypass	
0x93	00h	[7:6]		reserved	
		[5]	CB_DELAY_SEL(NTSC)	YCbCr 在做 444 到 422 转换时，是否将 CB 延迟一拍的选择信号 0: 不延迟； 1: 延迟一拍。	
		[4]	CB_AHEAD_SEL (NTSC)	YCbCr 在做 444 到 422 转换时，是否将 CB 提前一拍的选择信号 0: 不提前； 1: 提前一拍。	
		[3]	VSYN_OUT_SEL(NTSC)	输出场同步选择 0: 输出为场同步信号； 1: 输出为场有效信号。	
		[2]	HSYN_OUT_SEL(NTSC)	输出行同步选择 0: 输出为行同步； 1: 输出为 DE 信号。	
		[1:0]	H_FLT_SEL(NTSC)	水平方向滤波选择 0: 直通，不做滤波； 1: 2 阶滤波； 2: 3 阶滤波； 3: 4 阶滤波。	
0x94	00h	[7:0]	VSYN_DEL_NUM[7:0](NTSC)	场开始（奇场或偶场）标志的延迟点数，低 8 位	
0x95	00h	[7:0]	VSYN_DEL_NUM[15:8](NTSC)	场开始（奇场或偶场）标志的延迟点数，高 8 位	
0x96	00h	[7:0]	HFZ[7:0](NTSC)	Horizontal scale parameter LSB 7~0	
0x97	04h	[7:0]	HFZ[15:8](NTSC)	Horizontal scale parameter MSB 15~8	

0x98	00h	[7:0]	VFZ[7:0](NTSC)	Vertical scale parameter LSB 7~0	
0x99	04h	[7:0]	VFZ[15:8](NTSC)	Vertical scale parameter MSB 15~8	
0x9A	00h	[7:0]	THLEN[7:0](NTSC)	Horizontal total pixel number MSB 7~0	
0x9B	04h	[7:0]	THLEN[15:8](NTSC)	Horizontal total pixel number MSB 15~8	
0x9C	03h	[7:0]	HSYN_START[7:0](NTSC)	行同步开始位置, 低 8 位	
0x9D	00h	[7:0]	HSYN_START[15:8](NTSC)	行同步开始位置, 高 8 位	
0x9E	06h	[7:0]	HSYN_STOP[7:0](NTSC)	行同步结束位置, 低 8 位	
0x9F	00h	[7:0]	HSYN_STOP[15:8](NTSC)	行同步结束位置, 高 8 位	
0xA0	20h	[7:0]	HGATE_START[7:0](NTSC)	行有效信号开始位置, 低 8 位	
0xA1	00h	[7:0]	HGATE_START[15:8](NTSC)	行有效信号开始位置, 高 8 位	
0xA2	40h	[7:0]	HGATE_STOP[7:0](NTSC)	行有效信号结束位置, 低 8 位	
0xA3	03h	[7:0]	HGATE_STOP[15:8](NTSC)	行有效信号结束位置, 高 8 位	
0xA4	03h	[7:0]	VSYN_START[7:0](NTSC)	场同步开始位置, 低 8 位	
0xA5	00h	[7:0]	VSYN_START[15:8](NTSC)	场同步开始位置, 高 8 位	
0xA6	06h	[7:0]	VSYN_STOP[7:0](NTSC)	场同步结束位置, 低 8 位	
0xA7	00h	[7:0]	VSYN_STOP[15:8](NTSC)	场同步结束位置, 高 8 位	
0xA8	20h	[7:0]	VGATE_START[7:0](NTSC)	场有效信号开始位置, 低 8 位	
0xA9	00h	[7:0]	VGATE_START[15:8](NTSC)	场有效信号开始位置, 高 8 位	
0xAA	40h	[7:0]	VGATE_STOP[7:0](NTSC)	场有效信号结束位置, 低 8 位	
0xAB	03h	[7:0]	VGATE_STOP[15:8](NTSC)	场有效信号结束位置, 高 8 位	
0xAC	00h	[7:0]	LEFT_CUT_NUM[7:0] (NTSC)	水平方向左边数据的切边, 相当于将数据左移。	
0xAD	00h	[7:0]	LEFT_BLACK_NUM[7:0] (NTSC)	水平方向左边黑像素点数, 相当于将图像向右移动, 另, 在 4:3 模式时需要使用该功能。	
0xAE	00h	[7:0]	RIGHT_BLACK_NUM[7:0] (NTSC)	水平方向右边黑像素点数, 另, 在 4:3 模式时需要使用该功能。	
0xAF	00h	[7:0]	UP_BLACK_NUM[7:0] (NTSC)	垂直方向上方黑像素行数	
0xB0	00h	[7:0]	DOWN_BLACK_NUM[7:0] (NTSC)	垂直方向下方黑像素行数	
0xB1	00h	[7:0]	HV_DELAY(NTSC)	行同步信号 (hsyn), 有效信号 (DE), 场同步信号延迟点数	
0xB2	00h	[7:0]	COS_STEP(NTSC)	水平方向非线性 scaler 的增加梯度值。	
0xB3	00h	[7:0]	COS_LEVEL[7:0](NTSC)	水平方向非线性 scaler 的增加幅度, 低 8 位。	
0xB4	00h	[7:2]		Reserved	

		[1:0]	COS_LEVEL[9:8](NTSC)	水平方向非线性 scaler 的增加幅度，高 2 位。	
0xB5	00h	[7:0]	VXMOD_ODDF_INI[7:0](NTSC)	奇场或偶场做垂直 scaler 的起始步长，用来补偿奇偶场之间的隔行关系。	
0xB6	00h	[7:2]		Reserved	
		[1:0]	VXMOD_ODDF_INI[9:8](NTSC)	奇场或偶场做垂直 scaler 的起始步长，用来补偿奇偶场之间的隔行关系。	
0xB7	00h	[7:0]	TVLEN[7:0](NTSC)	每帧总行数，低 8 位	
0xB8	00h	[7:0]	TVLEN[15:8](NTSC)	每帧总行数，高 8 位	
0xB9	00h	[7:0]		Reserved	
0xBA	00h	[7:0]		Reserved	
0xBB	00h	[7:0]	TVLEN[7:0](PAL)	每帧总行数，低 8 位	
0xBC	00h	[7:0]	TVLEN[15:8](PAL)	每帧总行数，高 8 位	
0xBD	00h	[7]	FIELD_BYPASS(PAL)	Field output of scaler select 1: field of video input select 0: field of scale output select	
		[6]	VSYN_BYPASS(PAL)	Vertical synchronization output of scaler select 1: Vertical synchronization of video input select 0: Vertical synchronization of scale output select	
		[5]	HSYN_BYPASS(PAL)	Horizontal synchronization output of scaler select 1: Horizontal synchronization of video input select 0: Horizontal synchronization of scale output select	
		[4]	FIELD_NO_CHANGE(PAL)	Whether the odd field and even field is treated as the same, it is the same function to the deinterlacer. 1: the odd field and the even field is treated as the same, not do deinterlace 0: the odd field and the even field is treated as the different, do deinterlace	奇偶场处理开关: 1: bypass

		[3]	DATENA_INV(PAL)	Data enable output of scaler invert select 1: invert 0: not invert	
		[2]	VSYNC_INV(PAL)	Vertical synchronization output of scaler invert select 1: invert 0: not invert	
		[1]	HSYNC_INV(PAL)	Horizontal synchronization output of scaler invert select 1: invert 0: not invert	
		[0]	FIELDINV(PAL)	Field invert select 1: invert 0: not invert	
0xBE	00h	[7]	COS_EN(PAL)	非线性 scaler 使能	
		[6]		reserved	
		[5]	HU_SCA_BYPA(PAL)	Horizontal up scale bypass(down scale or bypass), it determines if the up scale is used 1: up scale is bypassed 0: up scale is used	
		[4]	HD_SCA_BYPA(PAL)	Horizontal down scale bypass(up scale or bypass), it determines if the down scale is used 1: down scale is bypassed 0: down scale is used	
		[3]	H_SCALE_BYPASS_SEL (PAL)	Horizontal scale bypass select, including horizontal up scale and horizontal down scale. 1: horizontal down scale bypass select HD_SCA_BYPA (0xA2 bit0), horizontal up scale bypass select HU_SCA_BYPA (0xA2 bit1). 0: horizontal down scale bypass and horizontal up scale bypass select auto set which is determined by HFZ[15:0]	

		[2]	VSYN_DEL_FIELD(PAL)	场（奇场或偶场）起始位置的延迟选择使能打开时，该信号表示选择的是奇场还是偶场。 1: 奇场 0: 偶场	
		[1]	VSYN_DEL_EN(PAL)	场（奇场或偶场）起始位置的延迟选择使能，当使能打开时，可改变奇场或偶场的起始位置，以改变重建后输出场的位置。 1: enable; 0: disable.	
		[0]	DEN_BYPASS(PAL)	Data enable bypass select 1: bypass 0: not bypass	
0xBF	00h	[7:6]		reserved	
		[5]	CB_DELAY_SEL(PAL)	YCbCr 在做 444 到 422 转换时，是否将 CB 延迟一拍的选择信号 0: 不延迟； 1: 延迟一拍。	
		[4]	CB_AHEAD_SEL (PAL)	YCbCr 在做 444 到 422 转换时，是否将 CB 提前一拍的选择信号 0: 不提前； 1: 提前一拍。	
		[3]	VSYN_OUT_SEL(PAL)	输出场同步选择 0: 输出为场同步信号； 1: 输出为场有效信号。	
		[2]	HSYN_OUT_SEL(PAL)	输出行同步选择 0: 输出为行同步； 1: 输出为 DE 信号。	
		[1:0]	H_FLT_SEL(PAL)	水平方向滤波选择 0: 直通，不做滤波； 1: 2 阶滤波； 2: 3 阶滤波； 3: 4 阶滤波。	
0xC0	00h	[7:0]	VSYN_DEL_NUM[7:0](PAL)	场开始（奇场或偶场）标志的延迟点数，低 8 位	
0xC1	00h	[7:0]	VSYN_DEL_NUM[15:8](PAL)	场开始（奇场或偶场）标志的延迟点数，高 8 位	
0XC2	00h	[7:0]	HFZ[7:0](PAL)	Horizontal scale parameter LSB 7~0	
0XC3	04h	[7:0]	HFZ[15:8](PAL)	Horizontal scale parameter MSB 15~8	

0xC4	00h	[7:0]	VFZ[7:0](PAL)	Vertical scale parameter LSB 7~0	
0xC5	04h	[7:0]	VFZ[15:8](PAL)	Vertical scale parameter MSB 15~8	
0xC6	00h	[7:0]	THLEN[7:0](PAL)	Horizontal total pixel number MSB 7~0	
0xC7	04h	[7:0]	THLEN[15:8](PAL)	Horizontal total pixel number MSB 15~8	
0xC8	03h	[7:0]	HSYN_START[7:0](PAL)	行同步开始位置, 低 8 位	
0xC9	00h	[7:0]	HSYN_START[15:8](PAL)	行同步开始位置, 高 8 位	
0xCA	06h	[7:0]	HSYN_STOP[7:0](PAL)	行同步结束位置, 低 8 位	
0xCB	00h	[7:0]	HSYN_STOP[15:8](PAL)	行同步结束位置, 高 8 位	
0xCC	20h	[7:0]	HGATE_START[7:0](PAL)	行有效信号开始位置, 低 8 位	
0xCD	00h	[7:0]	HGATE_START[15:8](PAL)	行有效信号开始位置, 高 8 位	
0xCE	40h	[7:0]	HGATE_STOP[7:0](PAL)	行有效信号结束位置, 低 8 位	
0xCF	03h	[7:0]	HGATE_STOP[15:8](PAL)	行有效信号结束位置, 高 8 位	
0xD0	03h	[7:0]	VSYN_START[7:0](PAL)	场同步开始位置, 低 8 位	
0xD1	00h	[7:0]	VSYN_START[15:8](PAL)	场同步开始位置, 高 8 位	
0xD2	06h	[7:0]	VSYN_STOP[7:0](PAL)	场同步结束位置, 低 8 位	
0xD3	00h	[7:0]	VSYN_STOP[15:8](PAL)	场同步结束位置, 高 8 位	
0xD4	20h	[7:0]	VGATE_START[7:0](PAL)	场有效信号开始位置, 低 8 位	
0xD5	00h	[7:0]	VGATE_START[15:8](PAL)	场有效信号开始位置, 高 8 位	
0xD6	40h	[7:0]	VGATE_STOP[7:0](PAL)	场有效信号结束位置, 低 8 位	
0xD7	03h	[7:0]	VGATE_STOP[15:8](PAL)	场有效信号结束位置, 高 8 位	
0xD8	00h	[7:0]	LEFT_CUT_NUM[7:0] (PAL)	水平方向左边数据的切边, 相当于将数据左移。	
0xD9	00h	[7:0]	LEFT_BLACK_NUM[7:0] (PAL)	水平方向左边黑像素点数, 相当于将图像向右移动, 另, 在 4:3 模式时需要使用该功能。	
0xDA	00h	[7:0]	RIGHT_BLACK_NUM[7:0] (PAL)	水平方向右边黑像素点数, 另, 在 4:3 模式时需要使用该功能。	
0xDB	00h	[7:0]	UP_BLACK_NUM[7:0] (PAL)	垂直方向上方黑像素行数	
0xDC	00h	[7:0]	DOWN_BLACK_NUM[7:0] (PAL)	垂直方向下方黑像素行数	
0xDD	00h	[7:0]	HV_DELAY(PAL)	行同步信号 (hsyn), 有效信号 (DE), 场同步信号延迟点数	
0xDE	00h	[7:0]	COS_STEP(PAL)	水平方向非线性 scaler 的增加梯度值。	
0xDF	00h	[7:0]	COS_LEVEL[7:0](PAL)	水平方向非线性 scaler 的增加幅度, 低 8 位。	
0xE0	00h	[7:2]		Reserved	

		[1:0]	COS_LEVEL[9:8](PAL)	水平方向非线性 scaler 的增加幅度，高 2 位。	
0xE1	00h	[7:0]	VXMOD_ODDF_INI[7:0](PAL)	奇场或偶场做垂直 scaler 的起始步长，用来补偿奇偶场之间的隔行关系。	
0xE2	00h	[7:2]		Reserved	
		[1:0]	VXMOD_ODDF_INI[9:8](PAL)	奇场或偶场做垂直 scaler 的起始步长，用来补偿奇偶场之间的隔行关系。	
0xE3	00h	[7]	VFZ_CHG_EN	垂直缩放比例 VFZ 变化时，Thlen 实际使用值是否从配置的值开始调整，高有效。 0: VFZ 变化时，Thlen 以当前调整后的值为初始值开始调整； 1: VFZ 变化时，Thlen 以配置寄存器的值为初始值开始调整。	
		[6]	THLEN_CHG_EN	寄存器配置的行总点数 Thlen 变化时，Thlen 实际使用值是否从配置的值开始调整，高有效。 0: Thlen 变化时，Thlen 实际使用值以当前调整后的值为初始值开始调整； 1: Thlen 变化时，Thlen 实际使用值以配置寄存器的值为初始值开始调整。	
		[5]	PN_POS_MOD	PN 变化时，PN 的值变化同步位置选择。 0: 每场中间同步 PN 值，选 Tvlen/2 的位置； 1: 每场开始同步 PN 值。	
		[4]	VSYN_MOD_SEL	场同步输出模式选择。 0: 将每场信号补齐，补齐的那场行数较多； 1: 每场行数最多为 Tvlen 行，不够的不补。	
		[3]	TV_ACTIVE_DISEN	无信号标志时（即 tv_active=0），scaler 自建行场信号还是根据 decoder 输出行场信号进行同步。可配 1。 0: 当 tv_active=0 时，scaler 自建行场信号，此时需要 Tvlen； 1: 当 tv_active=0 时，采用 decoder 自建行场信号进行同步。	

		[2]	VSYN_OUT_POS_SEL	场同步输出信号位置选择, 表示场同步信号与行同步信号的相对位置 0: 场同步与行同步下降沿一致; 1: 场同步与行同步上升沿一致;	
		[1]	VSYN_EDGE_SEL	每帧起始位置选择, 根据 decoder 输出场消隐信号进行定位 0: 选择场消隐信号上升沿进行起始位置的定位; 1: 选择场消隐信号下降沿进行起始位置的定位。	
		[0]	THLEN_AUTO_ADJ_DISEN	每行总点数 (Thlen) 自动调整 disable 0: 自动调整开启; 1: 自动调整关闭。	
0xE4	00h	[7]	TV_ACTIVE_POSITION	配置寄存器时, TV_active 的同步位置。 1: 场同步位置更新; 0: 场中间位置更新。	
		[6]	HFZ_LR_BLK_EN	HFZ、左右显黑寄存器同步使能, 此功能用来做 16:9 4:3 转换没有中间状态 0: 不使能; 1: 使能。	
		[5]	VBLK_LBUF_DSN	场消隐标志来时是否继续写入行缓存空间。 0: 场消隐时继续写入; 1: 场消隐时不写。	
		[4]	HV_DELAY_MAN	Hv_delay 自动配置、手动配置选择, 建议配置 0。 0: 自动配置; 1: 手动配置, ntsc、pal 制可单独配置。	
		[3:0]	THLEN_ADJ_STEP	当自动调整开启时, 每场调节 Thlen 的步长	
0xE5	00h	[7:0]	THLEN_READ[7:0]	只读寄存器, 自动调整完之后的 Thlen 可以在该寄存器读取, 低 8 位。	
0xE6	00h	[7:0]	THLEN_READ[15:8]	只读寄存器, 自动调整完之后的 Thlen 可以在该寄存器读取, 高 8 位。	
0xE7	00h	[7:4]	RESERVED		

		[3]	THLEN_ADJ_FINISH	只读寄存器，Thlen 自动调整时，该寄存器表示是否已经调整正常 0: 调整完，图像已正常； 1: 正在调整中。	
		[2]	PN_TVACT_POS_SEL	PN 制式切换或 TV_ACTIVE 同步的位置选择 0: 场有效中间； 1: 整场中间。	
		[1]	PN_SEL	只读寄存器，用于选择寄存器配置的制式，PAL 或 NTSC 0: PAL 1: NTSC	
		[0]	PN_DETECT	只读寄存器，PAL 或 NTSC 0: PAL 1: NTSC	
0xE8	00h	[7:0]	TVLEN_READ[15:8]	只读寄存器，点屏正常后，该寄存器表示 scaler 后的总行数，高 8 位。在不同制式下，读取该寄存器，将该值配置到相应的 Tvlen 寄存器中。	
0xE9	00h	[7:0]	TVLEN_READ[7:0]	只读寄存器，点屏正常后，该寄存器表示 scaler 后的总行数，低 8 位。在不同制式下，读取该寄存器，将该值配置到相应的 Tvlen 寄存器中。	
0xEA	00h	[7:0]	HFZ_LR_BLANK_WR	HFZ, left black, right black, 寄存器同步操作寄存器。当 HFZ_LR_BLK_EN=0 时无效，当 HFZ_LR_BLK_EN=1 时，只有对该寄存器进行写操作才将 HFZ, LEFT BLACK, RIGHT BLACK 这三个寄存器的值更新进去。通常用在 16:9, 4:3 模式切换时。	

6.7 OSD 设备地址 B6，mcu 地址 0XFBXX

FB00H	index ram addr	写 index ram 地址，低 8bit
FB01H	index ram data	写 index ram 数据，低 8bit
FB02H	font ram addr	写 font ram 地址，低 8bit

FB03H	font ram data	写 font ram 高 4 位数据
FB04H	font ram data	写 font ram 低 8 位数据
FB05H	osd_con1	<p>[0] osdc_en : osd block 0 使能。0 无效, 1 有效</p> <p>[1] osdc1_en : osd block 1 使能。0 无效, 1 有效</p> <p>[2] osdc2_en : osd block 2 使能。0 无效, 1 有效</p> <p>[3] osdc3_en : osd block 3 使能。0 无效, 1 有效</p> <p>[4] osdc4_en : osd block 4 使能。0 无效, 1 有效</p> <p>[5] osd_bl_en : 闪烁区域使能 0 无效 1 有效</p> <p>[6] osdc_color_sel : color index select for 5 osd block 0: color from color index ram; 1: select front and back color.</p> <p>[7] osdc_bitmap_en : bitmap enable for 5 osd block, when set 1 it means there are bitmap font in font ram, when set 0 it means there are not bitmap font in font ram.</p>
FB06H	osd_con2	<p>[5:0] osd_blink : osd 闪烁区域闪烁频率调节</p> <p>[6] osd_mix_en: osd 与 video 混合使能</p> <p>[7] osd_mix_mode : OSD 的与 VIDEO blending 模式选择, 1: 整块 OSD 与 VIDEO 混合; 0: OSD 背景色与 VIDEO 混合, 该种模式只支持普通 OSD。</p>
FB07H	osdc_sizex	<p>[6:0] osd block 0 水平方向最大字符数</p> <p>[7] Reserved</p>
FB08H	osdc_sizey	<p>[5:0] osd block 0 垂直方向最大字符数</p> <p>[7:6] Reserved</p>
FB09H	osdc_posx osdc_posy	<p>[2:0] 确定 osd block 0 起始点像素的水平位置的高 3 位</p> <p>[6:4] 确定 osd block 0 起始点像素的垂直位置的高 3 位</p>
FB0AH	osdc_posx	确定 osd block 0 起始点像素的水平位置的低 8 位
FB0BH	osdc_posy	确定 osd block 0 起始点像素的垂直位置的低 8 位
FB0CH	osd_alpha_bright	<p>[2:0]osd_alpha : 5 块 osd 与 video 混合参数</p> <p>[7:3]osd_bright : 5 块 osd 亮度调节参数</p>
FB0DH	index_ram_addr_hb	<p>[0] 写 index ram 地址, 高 1bit</p> <p>[7:1] Reserved</p>
FB0EH	Index_ram_data_hb	<p>[1:0] 写 index ram 数据, 高 2bit</p> <p>[7:2] Reserved</p>
FB0FH	Font_ram_addr_hb	<p>[3:0] 写 font ram 地址, 高 4bit</p> <p>[7:4] Reserved</p>
FB10H	Index_color_data	<p>写 index ram color data, 8bit, 用来表示每个 ICON 的前景色与背景色选择。</p> <p>[2:0]: 选择前景色调色盘</p> <p>[6:4]: 选择背景色调色盘</p>
FB11H	bitmap_start_icon[7:0]	Start ICON number of bitmap font in font ram。Font ram 中, 该 ICON 后面存储的 ICON 都认为是 bitmap 的信息。低 7bit, 高 2bit 见 FB70H。
FB12H	Osdc1_sizex	[6:0] osd block 1 水平方向最大字符数
FB13H	Osdc1_sizey	[5:0] osd block 1 垂直方向最大字符数

FB14H	Osdcl_posy Osdcl_posx	[2:0] 确定 osd block 1 起始点像素的水平位置的高 2 位 [6:4] 确定 osd block 1 起始点像素的垂直位置的高 2 位 [7] 确定 osd block 1 对应的 index ram 起始地址高 1 位
FB15H	Osdcl_posx	确定 osd block 1 起始点像素的水平位置的低 8 位
FB16H	Osdcl_posy	确定 osd block 1 起始点像素的垂直位置的低 8 位
FB17H	Osdcl_index_start_a	确定 osd block 1 对应的 index ram 起始地址低 8 位
FB18H	Osd2_sizex	[6:0] osd block 2 水平方向最大字符数
FB19H	Osd2_sizey	[5:0] osd block 2 垂直方向最大字符数
FB1AH	Osd2_posx Osd2_posy	[2:0] 确定 osd block 2 起始点像素的水平位置的高 2 位 [6:4] 确定 osd block 2 起始点像素的垂直位置的高 2 位 [7] 确定 osd block 2 对应的 index ram 起始地址高 1 位
FB1BH	Osd2_posx	确定 osd block 2 起始点像素的水平位置的低 8 位
FB1CH	Osd2_posy	确定 osd block 2 起始点像素的垂直位置的低 8 位
FB1DH	Osd2_index_start_a	确定 osd block 2 对应的 index ram 起始地址低 8 位
FB1EH	Osd3_sizex	[6:0] osd block 3 水平方向最大字符数
FB1FH	Osd3_sizey	[5:0] osd block 3 垂直方向最大字符数
FB20H	Osd3_posx Osd3_posy	[2:0] 确定 osd block 3 起始点像素的水平位置的高 2 位 [6:4] 确定 osd block 3 起始点像素的垂直位置的高 2 位 [7] 确定 osd block 3 对应的 index ram 起始地址高 1 位
FB21H	Osd3_posx	确定 osd block 3 起始点像素的水平位置的低 8 位
FB22H	Osd3_posy	确定 osd block 3 起始点像素的垂直位置的低 8 位
FB23H	Osd3_index_start_a	确定 osd block 3 对应的 index ram 起始地址低 8 位
FB24H	Osd4_sizex	[6:0] osd block 4 水平方向最大字符数
FB25H	Osd4_sizey	[5:0] osd block 4 垂直方向最大字符数
FB26H	Osd4_posx Osd4_posy	[2:0] 确定 osd block 4 起始点像素的水平位置的高 2 位 [6:4] 确定 osd block 4 起始点像素的垂直位置的高 2 位 [7] 确定 osd block 4 对应的 index ram 起始地址高 1 位
FB27H	Osd4_posx	确定 osd block 4 起始点像素的水平位置的低 8 位
FB28H	Osd4_posy	确定 osd block 4 起始点像素的垂直位置的低 8 位
FB29H	Osd4_index_start_a	确定 osd block 4 对应的 index ram 起始地址低 8 位
FB2AH	osdc_clr	5 块 osd 调色盘选择，在 osdc_color_sel 为 1 时有效 [7:4] 选择背景色 [3:0] 选择前景色
FB2BH	v_scaler[7:0]	OSD block 0 垂直方向缩放，低 8bit， 位数 0-7 对应单个字符垂直方向的行数，最低位对应的是第 16 行。 其中位数为 1 的，则对应行放大，放大倍数由 v_scale_coef 决定，否则不变
FB2CH	v_scaler[15:8]	OSD block 0 垂直方向缩放，中 8bit 最高位对应的是第 1 行。 其中位数为 1 的，则对应行放大，放大倍数由 v_scale_coef 决定，否则不变
FB2DH	v_scaler[23:16]	OSD block 0 垂直方向缩放，中 8bit 最高位对应的是第 1 行。

		其中位数为 1 的, 则对应行放大, 放大倍数由 v_scale_coef 决定, 否则不变
FB2EH	v_scaler[31:24]	OSD block 0 垂直方向缩放, 高 8bit 最高位对应的是第 1 行。 其中位数为 1 的, 则对应行放大, 放大倍数由 v_scale_coef 决定, 否则不变
FB2FH	h_scaler[7:0]	OSD block 0 水平方向缩放, 低 8bit 位数 0~7 对应单个字符水平方向的点数, 最低位对应的是第 16 点。 其中位数为 1 的, 则对应点放大, 放大倍数由 h_scale_coef 决定, 否则不变
FB30H	h_scaler[15:8]	OSD block 0 水平方向缩放, 中 8bit 最高位对应的是第 1 点。 其中位数为 1 的, 则对应点放大, 放大倍数由 h_scale_coef 决定, 否则不变
FB31H	h_scaler[23:16]	OSD block 0 水平方向缩放, 高 8bit 最高位对应的是第 1 点。 其中位数为 1 的, 则对应点放大, 放大倍数由 h_scale_coef 决定, 否则不变
FB32H	hv_scale_coef	[1:0] h_scale_coef 水平方向放大倍数 0: 2 倍; 1: 3 倍; 2: 4 倍; 3: 5 倍 [3:2] v_scale_coef 垂直方向放大倍数 0: 2 倍; 1: 3 倍; 2: 4 倍; 3: 5 倍 [7:4] reserved
FB33H	hv_scale_coef for osdc1 osdc2	h_scale_coef and v_scale_coef for osdc1 and osdc2. [1:0] h_scale_coef_osdc1: osd block 1 的水平方向放大倍数 0: 不放大; 1: 2 倍; 2: 3 倍; 3: 4 倍 [3:2] v_scale_coef_osdc1: osd block 1 的垂直方向放大倍数 0: 不放大; 1: 2 倍; 2: 3 倍; 3: 4 倍 [5:4] h_scale_coef_osdc2: osd block 2 的水平方向放大倍数 0: 不放大; 1: 2 倍; 2: 3 倍; 3: 4 倍 [7:6] v_scale_coef_osdc2: osd block 2 的垂直方向放大倍数 0: 不放大; 1: 2 倍; 2: 3 倍; 3: 4 倍
FB34H	hv_scale_coef for osdc3 osdc4	h_scale_coef and v_scale_coef for osdc3 and osdc4. [1:0] h_scale_coef_osdc3: osd block 3 的水平方向放大倍数 0: 不放大; 1: 2 倍; 2: 3 倍; 3: 4 倍 [3:2] v_scale_coef_osdc3: osd block 3 的垂直方向放大倍数 0: 不放大; 1: 2 倍; 2: 3 倍; 3: 4 倍 [5:4] h_scale_coef_osdc4: osd block 4 的水平方向放大倍数 0: 不放大; 1: 2 倍; 2: 3 倍; 3: 4 倍 [7:6] v_scale_coef_osdc4: osd block 4 的垂直方向放大倍数 0: 不放大; 1: 2 倍; 2: 3 倍; 3: 4 倍
FB35H	osd_h_mirror	[2:0] blink_block_sel: osd 闪烁块选择, 当 osd_bl_en 为 1 时有

	& blink_block_sel	<p>效 0: osd block 0 闪烁; 1: osd block 1 闪烁; 2: osd block 2 闪烁; 3: osd block 3 闪烁; 4: osd block 4 闪烁; others: osd block 0 闪烁。</p> <p>[3] hsyn_edge_sel: 输入行同步上升沿、下降沿选择, 用来定位 osd 位置;</p> <p>1: 选择下降沿;</p> <p>0: 选择上升沿;</p> <p>[4] osd_bitmap_cr0_touming_sel : OSD 的 bitmap 调色盘 cr0 是否为透明选择, 1: 不透明, 为调色盘 cr0 的颜色; 0: 透明</p> <p>[5] vsyn_edge_sel: 输入场同步上升沿、下降沿选择, 用来定位 osd 位置;</p> <p>1: 选择下降沿;</p> <p>0: 选择上升沿;</p>
		FB36-FB55 为 bitmap 使用的调色盘
FB36H	Cr0	调色盘 0
FB37H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB38H	cr1	调色盘 1
FB39H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB3AH	cr2	调色盘 2
FB3BH		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB3CH	cr3	调色盘 3
FB3DH		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB3EH	cr4	调色盘 4
FB3FH		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB40H	cr5	调色盘 5
FB41H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB42H	cr6	调色盘 6
FB43H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB44H	cr7	调色盘 7
FB45H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB46H	cr8	调色盘 8
FB47H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB48H	cr9	调色盘 9
FB49H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB4AH	cra	调色盘 10
FB4BH		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB4CH	crb	调色盘 11
FB4DH		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB4EH	crc	调色盘 12
FB4FH		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB50H	crd	调色盘 13
FB51H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R

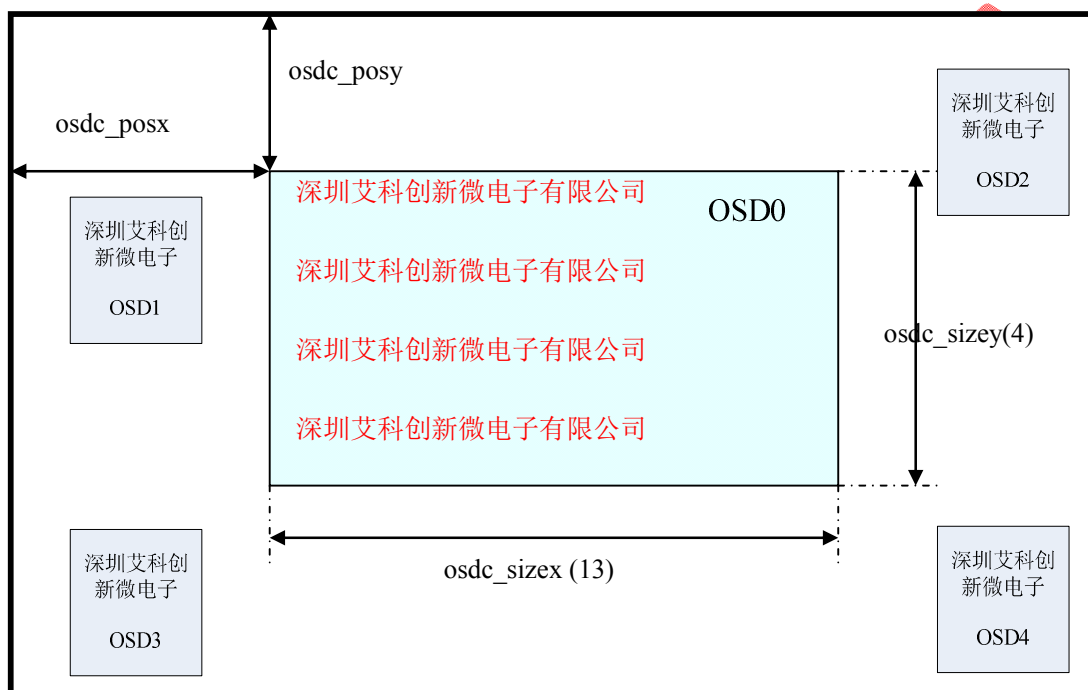
FB52H	cre	调色盘 14
FB53H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB54H	crf	调色盘 15
FB55H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
		FB56—FB62 为普通 osd 的调色盘
FB56H	cr1_normal	调色盘 1
FB57H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB58H	cr2_normal	调色盘 2
FB59H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB5AH	cr3_normal	调色盘 3
FB5BH		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB5CH	cr4_normal	调色盘 4
FB5DH		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB5EH	c5_normal	调色盘 5
FB5FH		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
FB60H	c6_normal	调色盘 6
FB61H		[11:8] 颜色 B , [7:4] 颜色 G , [3:0] 颜色 R
		FB62—FB6F 为渐变色 (color transition) 寄存器
FB70H	bitmap_start_icon[9:8]	[2:0] Start ICON number of bitmap font in font ram. Font ram 中, 该 ICON 后面存储的 ICON 都认为是 bitmap 的信息。高 2bit。 [7:3] reserved
FB71H	bitmap_cr_chag_en	[0] bitmap 调色盘互换使能, 1 有效。当该使能有效时, 根据 index color ram 中的信息, 可以将原来选中的调色盘 cr0-crf, 换成由下面寄存器决定的目标调色盘。 Index color ram data=0, 所对应 icon 中的调色盘不变; Index color ram data=1, 所对应 icon 中的调色盘, 可以有两个可以换, bitmap_cr_chag_00[3: 0], bitmap_cr_chag_11[3:0] 调色盘可以换成 bitmap_cr_chag_00[7:4], bitmap_cr_chag_11[7:4]。 Index color ram data=2, 所对应 icon 中的调色盘, 可以有三个可以换, bitmap_cr_chag_00[3 : 0], bitmap_cr_chag_11[3:0], bitmap_cr_chag_22[3:0] 调色盘可以换成 bitmap_cr_chag_00[7:4], bitmap_cr_chag_11[7:4], bitmap_cr_chag_22[7:4]。 Index color ram data=3, 所对应 icon 中的调色盘, 可以有四个可以换, bitmap_cr_chag_00[3 : 0], bitmap_cr_chag_11[3:0], bitmap_cr_chag_22[3:0], bitmap_cr_chag_33[3:0] 调色盘可以换成 bitmap_cr_chag_00[7:4], bitmap_cr_chag_11[7:4], bitmap_cr_chag_22[7:4], bitmap_cr_chag_33[7:4]。 [7:1] reserved
FB72H	bitmap_cr_chag_00	Bitmap 调色盘互换时的源调色盘与目标调色盘。 [3: 0] 源调色盘; [7: 4] 目标调色盘。

FB73H	bitmap_cr_chag_11	Bitmap 调色盘互换时的源调色盘与目标调色盘。 [3: 0] 源调色盘; [7: 4] 目标调色盘。
FB74H	bitmap_cr_chag_22	Bitmap 调色盘互换时的源调色盘与目标调色盘。 [3: 0] 源调色盘; [7: 4] 目标调色盘。
FB75H	bitmap_cr_chag_33	Bitmap 调色盘互换时的源调色盘与目标调色盘。 [3: 0] 源调色盘; [7: 4] 目标调色盘。
FB76H	Char_size_x	[4:0] 字符大小, 每个 icon 的水平像素点数 [7:5] Reserved
FB77H	Char_size_y	[4:0] 字符大小, 每个 icon 的垂直行数 [7:5] Reserved
FB78H	Osdram_sync_delay	[2:0] osdram_sync_delay, 通过 i2c 或 mcu 向 index ram, index color ram, osd font ram 中写数据时, 当 display 时钟比 27M osc 时钟小时, 需要配置次寄存器, 将 27M 写时序拉长, 不至于将写入时许丢失。 0: 不变, 1 cycle; 1: 2 cycle; 2: 3 cycle 3: 4 cycle 4: 5 cycle 5: 6 cycle 6: 7 cycle 7: 8 cycle [3] VSYN_INT_INV : OSD VSYN interrupt invert select [4]: osd_h_mirror0 osd 水平方向每个 icon 的 12bit 对应显示的镜像使能, 1: 使能; 0: 正常, 非镜像。 [5]: osd_h_mirror1 osd 水平方向 icon 的镜像使能, 1: 使能; 0: 正常, 非镜像。 [6]: osd_v_mirror0 osd 垂直方向每个 icon 的镜像使能, 1: 使能; 0: 正常, 非镜像。 [7]: osd_v_mirror1 osd 垂直方向 icon 的镜像使能, 使每个块中的第一行 icon 显示到最后一行 icon, 以 icon 为单位, 1: 使能; 0: 正常, 非镜像。
FB79H	osdc_bly_start	[5:0] 闪烁区域垂直方向起始字符数 [7:6] Reserved
FB7AH	osdc_bly_stop	[5:0] 闪烁区域垂直方向结束字符数 [7:6] Reserved
FB7BH	osdc_blx_start	[6:0] 闪烁区域水平方向起始字符数 [7] Reserved
FB7CH	osdc_blx_stop	[6:0] 闪烁区域水平方向结束字符数 [7] Reserved

FB7DH	Block1_blink_sel	[2:0] block1_blk_block_sel, osd 第二闪烁块的 block 块选择, 当 blok1_bl_en 为 1 时有效 0: osd block 0 闪烁; 1: osd block 1 闪烁; 2: osd block 2 闪烁; 3: osd block 3 闪烁; 4: osd block 4 闪烁; others: osd block 0 闪烁。 [3] blok1_bl_en, osd 第二闪烁块使能, 1 有效。 [7:4] reserved
FB7EH	Block1_blink	[5:0] block1_blink : osd 第二闪烁区域闪烁频率调节 [7:6] Reserved
FB7FH	Blok1_bly_start	[5:0] 第二闪烁区域垂直方向起始字符数 [7:6] Reserved
FB80H	Blok1_bly_stop	[5:0] 第二闪烁区域垂直方向结束字符数 [7:6] Reserved
FB81H	Blok1_blx_start	[6:0] 第二闪烁区域水平方向起始字符数 [7] Reserved
FB82H	Blok1_blx_stop	[6:0] 第二闪烁区域水平方向结束字符数 [7] Reserved
FB83H	Block2_blink_sel	[2:0] block2_blk_block_sel, osd 第三闪烁块的 block 块选择, 当 blok2_bl_en 为 1 时有效 0: osd block 0 闪烁; 1: osd block 1 闪烁; 2: osd block 2 闪烁; 3: osd block 3 闪烁; 4: osd block 4 闪烁; others: osd block 0 闪烁。 [3] blok2_bl_en, osd 第二闪烁块使能, 1 有效。 [7:4] reserved
FB84H	Block2_blink	[5:0] block2_blink : osd 第三闪烁区域闪烁频率调节 [7:6] Reserved
FB85H	Blok2_bly_start	[5:0] 第三闪烁区域垂直方向起始字符数 [7:6] Reserved
FB86H	Blok2_bly_stop	[5:0] 第三闪烁区域垂直方向结束字符数 [7:6] Reserved
FB87H	Blok2_blx_start	[6:0] 第三闪烁区域水平方向起始字符数 [7] Reserved
FB88H	Blok2_blx_stop	[6:0] 第三闪烁区域水平方向结束字符数 [7] Reserved
FB89H	Osd_hsyn_vsyn_sel	[1:0] osd_hsyn_sel, OSD 用来定位水平方向的信号选择 0: hsyn, 行同步信号; 1: hden, 行有效信号; 2: sthl, tcon 屏信号; 3: sthr, tcon 屏信号; [3:2] osd_vsyn_sel, OSD 用来定位垂直方向的信号选择 0: vsyn, 场同步信号; 1: vsyn, 场同步信号; 2: stvl, tcon 屏信号; 3: stvr, tcon 屏信号。

1) 可任意定位的 5 个 OSD BLOCK 区域

可以在同一屏幕上同时显示 5 块 OSD，每块 OSD 可以通过相应的位置寄存器及大小寄存器配置。由 `osdc_posx[10:0]` 和 `osdc_posy[10:0]` 来确定 OSD0 在屏幕上的具体位置，其区域大小由 `osdc_size[6:0]` 和 `osdc_sizey[5:0]` 来确定。OSD1, OSD2, OSD3, OSD4 块相应的寄存器见上述寄存器列表。



2) 可容纳 512 个 ICON 的 FONT ROM，192 个 ICON 的 FONT RAM 以及字符索引存储器 (512x10)，字符色彩存储器 (512x6)

OSD 的 ICON 存放于 FONT ROM 和 FONT RAM 中。其中 FONT ROM 大小为 8K X 12，共可放置 512 个普通 ICON，FONT RAM 大小为 3072 X 12，可以存放 192 个普通 ICON。3072x12 大小的 FONT RAM 也可以存放 bitmap 的信息，其数据量是普通 ICON 的四倍。应用时，配置 bitmap 的相关寄存器即可以 bitmap 的形式显示。MCU 可将字符写入 OSD FONT RAM，或从 FLASH 直接导入 OSD FONT RAM，OSD 控制器读取其中的 ICON。

INDEX RAM 大小为 512 X 10，检索值为 **000H~3FFH**，其中索引值 **2E0H~39FH** 为读取 FONT RAM 中 ICON，其他索引值为读取 FONT ROM ICON 提供首地址。

INDEX COLOR RAM 大小为 512x6，其中存放每个 ICON 对应的前景色与背景色信息，3bit 对应前景色的调色盘，另外 3bit 对应背景色的调色盘。

3) ICON 大小为 12X16

OSD ICON 规定格式为 12 X 16, 即水平像素点为 12 个, 垂直像素点为 16。ICON 在 FONT RAM 或 FONT ROM 中存储的方式为前景写 1, 背景写 0。

4) bitmap 有 16 个调色盘(color pallete), 普通 ICON 提供 8 个调色盘

OSD 提供 16 个调色盘给 bitmap, 其中 0 号调色盘用户可以选择透明或不透明。每个调色盘为 12 位宽, 颜色 RGB 分别对应其中的[3:0]、[7:4]、[11:8]。

提供 8 个调色盘给普通 ICON, 其中 0 号调色盘为透明, 7 号调色盘为黑色, 用户不可配置, 其他 6 个调色盘用户可以根据需要灵活配置。

6) 一个闪烁区域(blink)

OSD 可以设置 1 个闪烁区域, 其闪烁的频率由参数 `osd_blink` 决定, 同时只能选中 5 个 OSD 中的一个。

寄存器 `osd_bl_en` 可以设置闪烁区域使能, `blink_block_sel` 可以选择闪烁的 OSD。

7) 字符缩放功能 (font scaler)

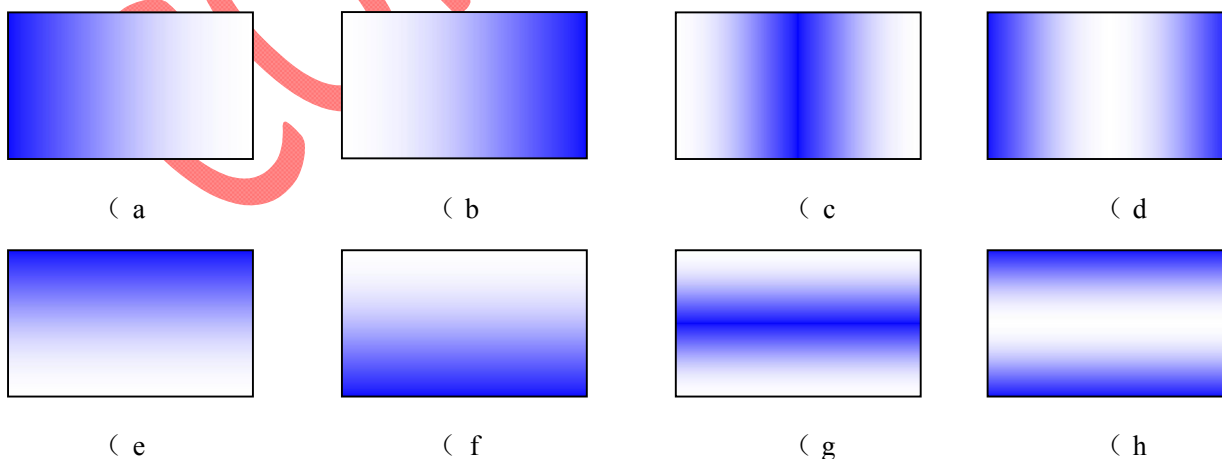
提供最大 5 倍的字符放大, 其中 OSD0 最大倍数为 5 倍, 其余 4 个 OSD 的最大倍数为 4 倍。OSD0 中, 也可对单个 ICON 的水平方向某点或垂直方向某行单独进行相应的放大, 相应的参数为 `v_scaler` 和 `h_scaler`。其余 4 块 OSD 由相应的 `h_scale_coef`、`v_scale_coef` 决定其放大倍数。

8) 场中断 (vsyn_int)

在一屏 osd 显示中, 有一些字符需要经常改变, 如菜单里可调的选项, 显示根据用户的遥控器输入需要相应的改变。如果在程序中直接更改字符对应的 `index num`, 则容易引起 osd 闪烁。为了避免这种闪烁, 则必须保证在有 osd 显示时其对应的字符 `index num` 不变。在场中断中对 `index num` 进行操作, 则从时间上看, 是在场消隐的时间内更改 `index num`, 由此避免出现上述问题。

9、颜色渐变 (color transition)

其效果包括如下:



FB62	[0]	block_en	颜色渐变使能
	[3:1]	block_mode	颜色渐变模式选择 00: 水平方向颜色渐变方式(颜色 1→颜色 2) 如 (a) (b) 01: 水平方向颜色渐变方式(颜色 1→颜色 2→颜色 1) 如 (c) (d) 10: 垂直方向颜色渐变方式(颜色 1→颜色 2) 如 (e) (f) 11: 垂直方向颜色渐变方式(颜色 1→颜色 2→颜色 1) 如 (g) (h)
	[7:4]	block_blend	Video 与 block 混合的透明度大小 (16 种混合效果)
FB63	[0]	block_r_comp	渐变的两种颜色值的比较 (左: 颜色 1, 右: 颜色 2) 1: 颜色 1>颜色 2 (表示从左至右颜色递减) 0: 颜色 2>颜色 1 (表示从左至右颜色递增)
	[1]	block_g_comp	渐变的两种颜色值的比较 (左: 颜色 1, 右: 颜色 2) 1: 颜色 1>颜色 2 (表示从左至右颜色递减) 0: 颜色 2>颜色 1 (表示从左至右颜色递增)
	[2]	block_b_comp	渐变的两种颜色值的比较 (左: 颜色 1, 右: 颜色 2) 1: 颜色 1>颜色 2 (表示从左至右颜色递减) 0: 颜色 2>颜色 1 (表示从左至右颜色递增)
	[4:3]	block_video_sel	Video 与 block 混合方式 00: video 01: block 10,11: block 混合在 video 中
	[5]	hsyn_inv_reg	0: 行保持 1: 行取反
	[6]	vsyn_inv_reg	0: 行保持 1: 行取反
	[7]	hsyn_pos_reg	0: 判断行信号的上升沿 1: 判断行信号的下降沿
FB64	[7:0]	color_rout	Block 中的 R 值
FB65	[7:0]	color_gout	Block 中的 G 值
FB66	[7:0]	color_bout	Block 中的 B 值
FB67	[2:0]	block_x_start[10:8]	Block 窗口的水平起始位置, 高 3 位
	[3]	Reserved	Reserved
	[6:4]	block_y_start[10:8]	Block 窗口的垂直起始位置, 高 3 位
	[7]	Reserved	Reserved
FB68	[7:0]	block_x_start[7:0]	Block 窗口的水平起始位置, 低 8 位
FB69	[7:0]	block_y_start[7:0]	Block 窗口的垂直起始位置, 低 8 位
FB6A	[2:0]	block_length [10:8]	Block 窗口的长度, 高 3 位
	[3]	Reserved	Reserved

	[6:4]	block_width[10:8]	Block 窗口的宽度，高 3 位
	[7]	Reserved	Reserved
FB6B	[7:0]	block_length[7:0]	Block 窗口的长度，低 8 位
FB6C	[7:0]	block_width [7:0]	Block 窗口的宽度，低 8 位
FB6D	[7:4]	block_r_step	颜色 R 值渐变的步长
	[3:0]	block_r_base	颜色 R 值渐变的基色
FB6E	[7:4]	block_g_step	颜色 G 值渐变的步长
	[3:0]	block_g_base	颜色 G 值渐变的基色
FB6F	[7:4]	block_b_step	颜色 B 值渐变的步长
	[3:0]	block_b_base	颜色 B 值渐变的基色

6.9 SPI Register(I2C Address: 0xB0 MCU Address: 0xFDXX)

Addr	Val	Bits	Name	Description	App note
0XD0	0x00	[7:0]	INSTR_REG	Bit0: flash_id_read 1:enable Bit1: flash_sr_read 1:enable Bit2: flash_sr_write 1:enable Bit3: flash_erase 1:enable Bit4: flash_data_read 1:enable Bit5: flash_data_write 1:enable Bit6: flash_load_ram 1:enable Bit7: flash_program 1:enable (Note: only one bit can be 1 at the same time)	
0xD1	0x00	[7:0]	WRDATA_REG	the data byte writted to flash	
0xD2	0x00	[7:0]	RDDATA_REG	the data byte read from flash	
0xD3	0x00	[7:0]	FETCH_ID0_REG	the fist byte of ID read from flash	
0xD4	0x00	[7:0]	FETCH_ID1_REG	the second byte of ID read from flash	
0xD5	0x00	[7:0]	FETCH_ID2_REG	the third byte of ID read from flash	
0xD6	0x00	[7:0]	FADDR_REG0	the 7:0 bit of the target address of flash	
0xD7	0x00	[7:0]	FADDR_REG1	the 15:8 bit of the target address of flash	
0xD8	0x00	[7:0]	FADDR_REG2	the 23:16 bit of the target address of flash	

0xD9	0x00	[7:0]	DRADDR_REG0	the 7:0 bit of the target address of data_ram or font_ram	
0xDA	0x00	[7:0]	DRADDR_REG1	[5:0] the 13:8 bit of the target address of data_ram or font_ram [6] N/A [7] select font_ram or data_ram 0:font_ram 1:data_ram	
0xDB	0x00	[7:0]	LOAD_LENGTH0	the lowest 8 bits of load length	
0xDC	0x00	[7:0]	LOAD_LENGTH1	the highest 6 bit of load length	
0xDD	0x80	[7:0]	STATUS_REG	Bit7: dtfl_pro_dw Bit6: dtfl_load_dw Bit5: dtfl_wr_dw Bit4: dtfl_rd_dw Bit3: dtfl_erase_dw Bit2: dtfl_wsr_dw Bit1: dtfl_rsr_dw Bit0: dtfl_id_dw	
0xDE	0x00	[7:0]	Ctrl_reg	7: interrupt enable 6: fast read enable (if the serial flash support the fast read command, you must set this bit to 1 to enable the fast read function) 5: sst_aai_pro enable (if you use the SST serial flash, you must set this bit to 1'b1 to enable the flash_program fuction) 4: sst_pro_word (1: support the SST aai word program 0: support the SST aai program) 3: chip erase (if you need chip erase, you must set this bit to 1)	
0xDF	0x80	[7:0]	swhw_handshaking	7 dflash_idle 0: indicate that serial flash is busy 1: indicate that serail flash is idle now, you cand send the next command	
0xE0	0x03	[7:0]	read_command	set the read command according to different serial flash	

0xE1	0x02	[7:0]	program_command	set the program command according to different serial flash	
0xE2	0x05	[7:0]	rdsr_command	set the read status register command according to different serial flash	
0xE3	0x01	[7:0]	wrsr_command	set the write status register command according to different serial flash	
0xE4	0xB8	[7:0]	erase_command	set the erase command according to different serial flash	
0xE5	0x9F	[7:0]	rdid_command	set the read id command according to different serial flash	
0xE6	0x06	[7:0]	wren_command	set the write enable command according to different serial flash	
0xE7	0x07	[7:0]	p_dummy_num		
0xF0	0x06	[2:0]	cache_size	001: 2BYTE 010: 4BYTE 011: 8BYTE; 100: 16BYTE 101: 32BYTE 110: 64BYTE; 111: 128BYTE 000: 64BYTE	
0xF1	0x01	[7:0]	bank_sel	00000001 bank1 00000010 bank2 11111111 bank255	

6.10 12bit ADC Register(I2C Address: 0xB0 MCU Address: 0xFDXX)

12-bit ADC 不支持触摸屏，相关寄存器不开放。

The 12-bit ADC module is an APB slave, and it can transform analog input into digital data. And the measured data will be stored in the related register through SPI interface.

12-bit ADC main features:

- Key-pressure measurement.
- Battery measurement.
- Analog inputs (Temperature , Humidity ,etc.)measurement.

12-bit ADC Architecture

The APB_12bitADC has two interfaces that connect to APB bus and ICU.

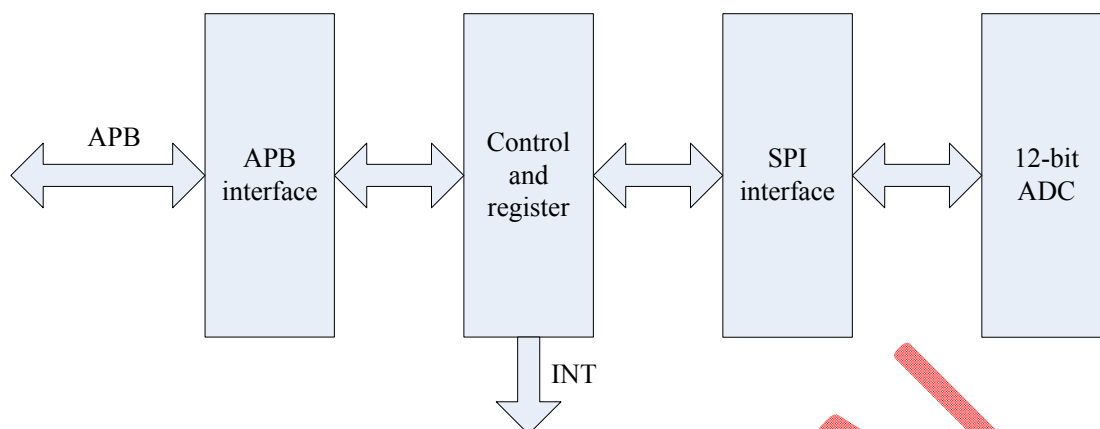


Figure 26-1 12-bit ADC Architecture

12-bit ADC Function Block Description:

APB Interface

The APB interface is used to connect the APB_12bitADC to AMBA APB bus.

Control and Register Logic

This block contains registers and control logic. The control registers can config the work mode of the module. These registers can be written or read through the APB-Interface.

SPI interface

The measured data will be stored in the related register through SPI (serial parallel interface).

Interrupt

This block controls the interrupt generating. It generates interrupt to the APB ICU (interrupt control unit).

Work mode

The module can work in hard mode or soft mode. If SOF_MOD=1 (software mode), the ADC will work by configing ADCSCTR. When SOF_MOD=0 (hardware mode) and CMD bit is not set, the hardware will auto detect and measure key or touch panel pressure if ENA bit is set. When SOF_MOD=0 (hardware mode) and CMD bit is set, and the hardware will measure the related analog input if ENA bit is set.

12-bit ADC Register Description:

Sub Address	Bits	Default Value	Name	Description
0xB0	[7]	0	ADC_Control_0	Must be 0.
	[6]	0		AUX3_ENA(no used) 0: aux3 disable. 1: aux3 enable.
	[5]	0		AUX2_ENA 0: aux2 disable. 1: aux2 enable.
	[4]	0		AUX1_ENA 0: aux1 disable. 1: aux1 enable.
	[3]	0		AUX0_ENA 0: aux0 disable. 1: aux0 enable.
	[2]	0		Must be 0.
	[1]	0		BAT_ENA 0: battery disable. 1: battery enable.

	[0]	0		ADC_RST ADC soft rst : 0 : normal. 1 : reset.
0xB1	[7]	0	ADC_Control_1	AUX2_CMD 0: aux2 force transform disable.. 1: aux2 force transform enable.
	[6]	0		AUX1_CMD 0: aux1 force transform disable.. 1: aux1 force transform enable.
	[5]	0		AUX0_CMD 0: aux0 force transform disable.. 1: aux0 force transform enable.
	[4]	0		Must be 0.
	[3]	0		AUX3_DET_INV(no used) 0: aux3_det low valid. 1: aux3_det high valid.
	[2]	0		AUX2_DET_INV 0: aux2_det low valid. 1: aux2_det high valid.
	[1]	0		AUX1_DET_INV 0: aux1_det low valid. 1: aux1_det high valid.
	[0]	0		AUX0_DET_INV 0: aux0_det low valid. 1: aux0_det high valid.
0xB2	[7:6]	00	ADC_Control_2	reserved
	[5]	1		PD_VREF_TH(no used) to analog block : touch pannel vref enable
	[4]	0		VREF_SWITCH_TH to analog block : 12 bit adc vref select ctl
	[3]	0		TRIM_KREF0 to analog block : KEY INT VOLTAGE ADJUST
	[2]	1		TRIM_KREF1 to analog block : KEY INT VOLTAGE ADJUST
	[1]	0		SOF_MOD 0: hardware mode.. 1: software mode.
	[0]	0		AUX3_CMD(no used) 0: aux3 force transform disable.. 1: aux3 force transform enable.
0xB3	[7:0]	0	ADC_Control_3	reserved
0xB4	[7]	0	ADC_Config_0	AUX_ADR[0] 110:AUX
	[6:4]	010		BAT_ADR

				010: BAT
	[3]	0		ADC_MOD For analog block.
	[2]	0		ADC_SER For analog block. 0: differential reference. 1: single-ended reference
	[1:0]	10		ADC_PD For analog block.
0xB5	[7:2]	001101	ADC_Config_1	Reserved.
	[1:0]	11		AUX_ADR[2:1] 110:AUX
0xB6	[7]	1	ADC_Int_Mask_0	AUX2_STOP_IM Aux2 stop interrupt mask: 0: mask disabled. 1: mask enabled.
	[6]	1		AUX2_START_IM Aux2 start interrupt mask: 0: mask disabled. 1: mask enabled.
	[5]	1		AUX1_VALUE_IM Aux1 value interrupt mask: 0: mask disabled. 1: mask enabled.
	[4]	1		AUX1_STOP_IM Aux1 stop interrupt mask: 0: mask disabled. 1: mask enabled.
	[3]	1		AUX1_START_IM Aux1 start interrupt mask: 0: mask disabled. 1: mask enabled.
	[2]	1		AUX0_VALUE_IM Aux0 value interrupt mask: 0: mask disabled. 1: mask enabled.
	[1]	1		AUX0_STOP_IM Aux0 stop interrupt mask: 0: mask disabled. 1: mask enabled.
	[0]	1		AUX0_START_IM Aux0 start interrupt mask: 0: mask disabled. 1: mask enabled.
0xB7	[7]	1	ADC_Int_Mask_1	BAT_VALUE_IM Battery value interrupt mask: 0: mask disabled. 1: mask enabled.
	[6:1]	6'h3f		Must be 6'h3f.
	[0]	1		AUX2_VALUE_IM Aux2 value interrupt mask: 0: mask disabled. 1: mask enabled.
0xB8	[7]	0	ADC_Status_0	AUX2_STOP_INT Aux2 stop interrupt status.
	[6]	0		AUX2_START_INT Aux2 start interrupt status.
	[5]	0		AUX1_VALUE_INT Aux1 value interrupt status.
	[4]	0		AUX1_STOP_INT Aux1 stop interrupt status.
	[3]	0		AUX1_START_INT Aux1 start interrupt status.
	[2]	0		AUX0_VALUE_INT

				Aux0 value interrupt status.
	[1]	0		AUX0_STOP_INT Aux0 stop interrupt status.
	[0]	0		AUX0_START_INT Aux0 start interrupt status.
0xB9	[7]	0	ADC_Status_1	BAT_VALUE_INT Battery value interrupt status
	[6:1]	0		Reserved.
	[0]	0		AUX2_VALUE_INT Aux2 value interrupt status.
0xBA	[7:0]	0	ADC_BAT_0	ADCBAT[7:0] ADC battery value. Read only.
0xBB	[3:0]	0	ADC_BAT_1	ADCBAT[11:8] ADC battery value. Read only.
0xBC	[7:0]	0	ADC_AUX0_0	ADCAUX0[7:0] ADC AUX0 value. Read only.
0xBD	[3:0]	0	ADC_AUX0_1	ADCAUX0[11:8] ADC AUX0 value Read only.
0xBE	[7:0]	0	ADC_AUX1_0	ADCAUX1[7:0] ADC AUX1 value. Read only.
0xBF	[3:0]	0	ADC_AUX1_1	ADCAUX1[11:8] ADC AUX1 value Read only.
0xC0	[7:0]	0	ADC_AUX2_0	ADCAUX2[7:0] ADC AUX2 value. Read only.
0xC1	[3:0]	0	ADC_AUX2_1	ADCAUX2[11:8] ADC AUX2 value Read only.
0xC2	[7:0]	0	ADC_AUX3_0	ADCAUX3[7:0](no used) ADC AUX3 value. Read only.
0xC3	[3:0]	0	ADC_AUX3_1	ADCAUX3[11:8] (no used) ADC AUX3 value Read only.
0xC8	[7:0]	8'hff	Debounce_Counter_0	DBNCNT[7:0] Debounce counter for eliminating bounce
0xC9	[7:0]	8'h0f	Debounce_Counter_1	DBNCNT[15:8] Debounce counter for eliminating bounce
0xCA	[7:0]	00	Debounce_Counter_2	DBNCNT[23:16] Debounce counter for eliminating bounce
0xCC	[7:0]	8'hff	ADC_Trans_Interval_0	INTER[7:0] ADC transform interval.
0xCD	[7:0]	8'hff	ADC_Trans_Interval_1	INTER[15:8] ADC transform interval.
0xCE	[7]	0	ADC_soft_control_0 See fig1,table1,table2	ADC_CLK adc_clk when SOF_MOD = 1.

	[6]	0	For details	ADC_SO adc_so . Read only.
	[5]	0		ADC_BUSY adc_busy . Read only.
	[4:3]	00		Read only.
	[2]	0		AUX2_REQ AUX2 request. Read only.
	[1]	0		AUX1_REQ AUX1 request. Read only.
	[0]	0		AUX0_REQ AUX0 request. Read only.
0xCF	[3:2]	00	ADC_soft_control_1	AUX_SEL_S AUX_SEL when SOF_MOD = 1.
	[1]	1		ADC_CSn adc_csn when SOF_MOD = 1.
	[0]	0		ADC_SI adc_si when SOF_MOD = 1.

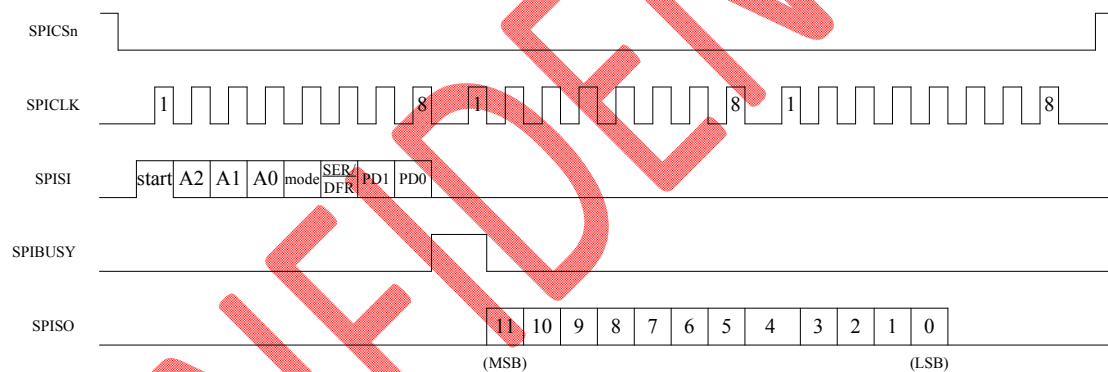


Fig 1. Conversion Timing, 24 Clocks-per-Conversion.

(MSB)							(LSB)
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	A2	A1	A0	MODE	SER/DFR	PD1	PD0

Table 1. Order of the Control Bits in the Control Byte

BIT	NAME	DESCRIPTION
7	S	Start bit. Control byte starts with first high bit on ADC_SI.
6-4	A2-A0	Channel Select bits.

		Along with the SER/DFR bit, these bits control the setting of the multiplexer input, touch driver switches, and reference inputs(see Table 3).
3	MODE	12-Bit/8-Bit Conversion Select bit. 0: 12-Bit 1: 8-Bit
2	SER/DFR	Single-Ended/Differential Reference Select bit.
1-0	PD1-PD0	Power-Down Mode Select bits.

Table2. Description of the Control Bits in the Control Byte

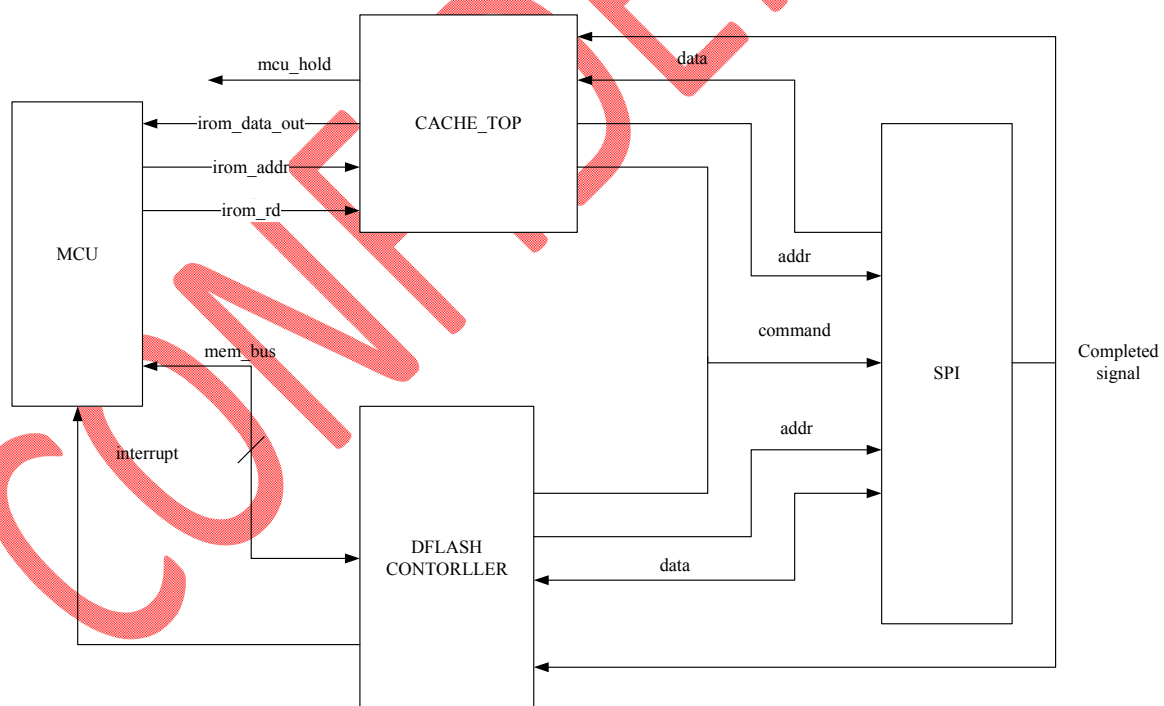
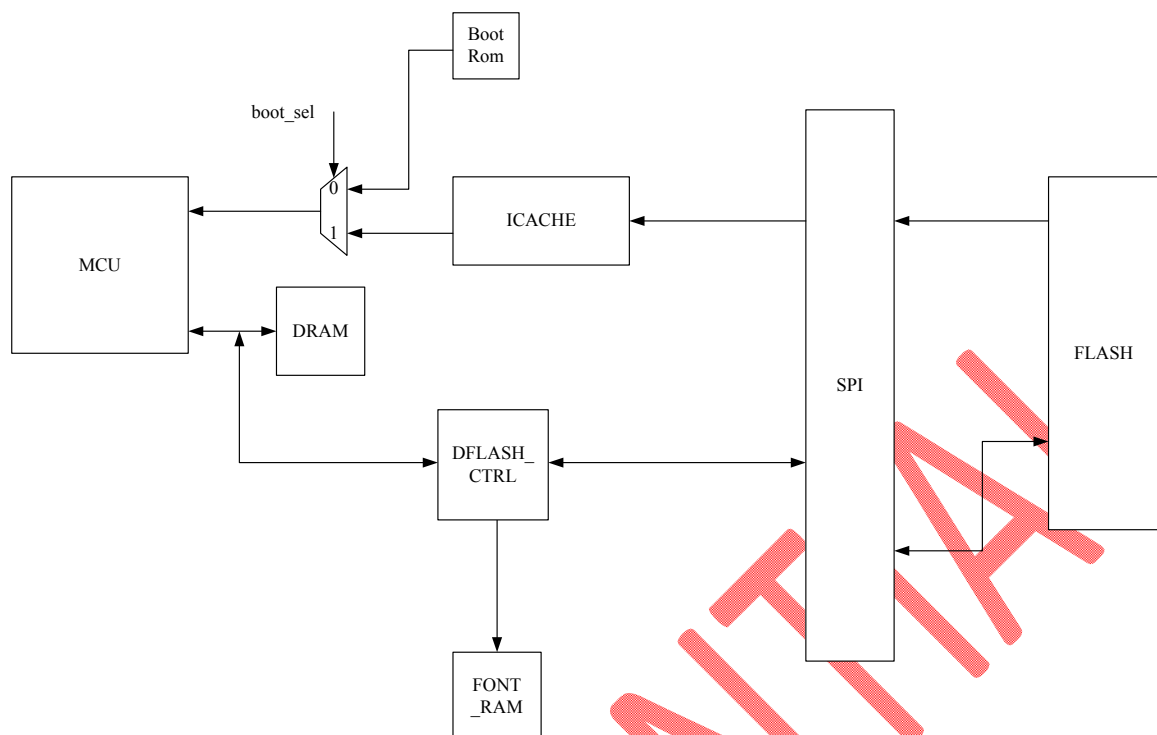
A2 A1 A0	AUX_SEL	INPUT
000	-	-
001	-	-
010	-	BAT
011	-	-
100	-	-
101	-	-
110	00	AUX0
	01	AUX1
	10	AUX2
	11	-
111	-	-

Table3. Inputs Switch

7. AMT630A MCU Peripheral

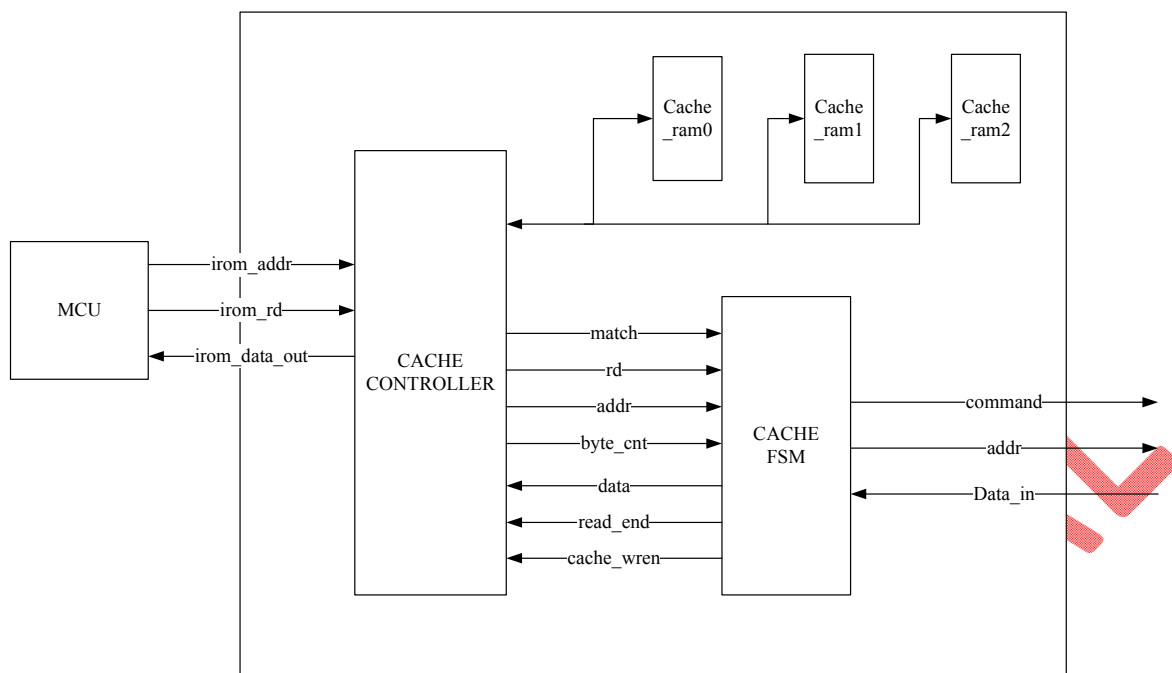
7.1 SPI FLASH CONTROLLER

1. Architecture

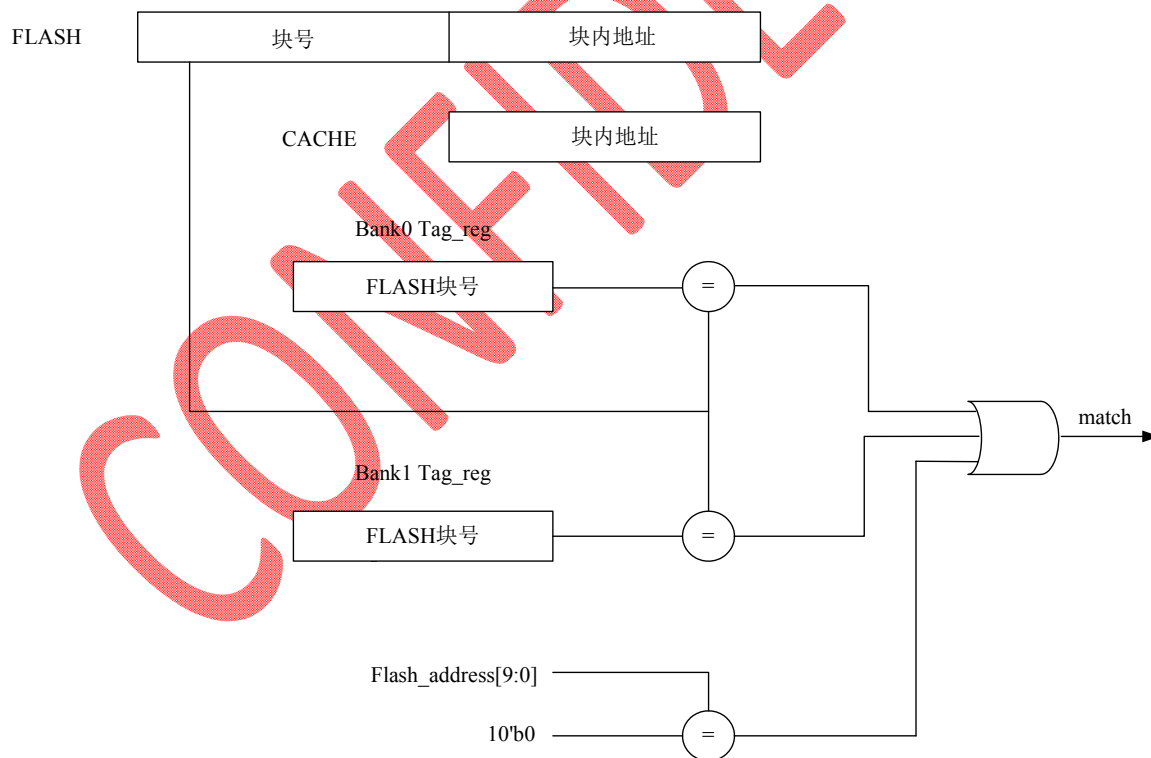


2. Cache

Cache Top 如下图所示:



2.2 cache 的地址映射



Note 1 : cache 大小可配 (FFF0[2:0]: 001: 2BYTE; 010: 4BYTE; 011: 8BYTE; 100: 16BYTE;
101: 32BYTE; 110: 64BYTE; 111: 128BYTE; 000: 64BYTE)

2.3 BANK 跳转:

MCU地址空间与Bank 跳转之间的关系



Note 1: Bank 的跳转需配寄存器 FFF1(因为 Bank0 与 command area 地址空间重叠, 所以 REG FFF1 的可配值为 00000001 – 11111111)

3. Data Flash

NAME	ADDRESS	DEFAULT	FUNCTION
INSTR_REG	16'hFFD0	0000_0000	Bit0: flash_id_read 1:enable Bit1: flash_sr_read 1:enable Bit2: flash_sr_write 1:enable Bit3: flash_erase 1:enable Bit4: flash_data_read 1:enable Bit5: flash_data_write 1:enable Bit6: flash_load_ram 1:enable Bit7: flash_program 1:enable (Note: only one bit can be 1 at the same time)
WRDATA_REG	16'hFFD1	0000_0000	7:0 the byte writted to flash
RDDATA_REG	16'hFFD2	0000_0000	7:0 the byte read from flash
FETCH_ID0_REG	16'hFFD3	0000_0000	7:0 the fist byte of ID read from flash
FETCH_ID1_REG	16'hFFD4	0000_0000	7:0 the second byte of ID read from flash
FETCH_ID2_REG	16'hFFD5	0000_0000	7:0 the third byte of ID read from flash
FADDR_REG0	16'hFFD6	0000_0000	7:0 the 7:0 bit of the target address of flash
FADDR_REG1	16'hFFD7	0000_0000	7:0 the 15:8 bit of the target address of flash
FADDR_REG2	16'hFFD8	0000_0000	7:0 the 23:16 bit of the target address of flash

DRADDR_REG0	16'hFFD9	0000_0000	7:0 the 7:0 bit of the target address of data_ram or font_ram
DRADDR_REG1	16'hFFDA	0000_0000	5:0 the 13:8 bit of the target address of data_ram or font_ram 6 N/A 7 select font_ram or data_ram 0:font_ram 1:data_ram
LOAD_LENGTH0	16'hFFDB	0000_0000	7:0 the lowest 8 bits of load length
LOAD_LENGTH1	16'hFFDC	0000_0000	5:0 the highest 6 bit of load length
STATUS_REG	16'hFFDD	1000_0000	Bit7: dtfl_pro_dw Bit6: dtfl_load_dw Bit5: dtfl_wr_dw Bit4: dtfl_rd_dw Bit3: dtfl_erase_dw Bit2: dtfl_wsr_dw Bit1: dtfl_rsr_dw Bit0: dtfl_id_dw
Ctrl_reg	16'hFFDE	0000_0000	7: interrupt enable 6: fast read enable (if the serial flash support the fast read command, you must set this bit to 1 to enable the fast read function) 5: sst_aai_pro enable (if you use the SST serial flash, you must set this bit to 1 to enable the flash_program fuction) 4: sst_pro_word (1: support the SST aai word program 0: support the SST aai program) 3: chip erase (if you need chip erase, you must set this bit to 1)
swhw_handshaking	16'hFFDF	1000_0000 (read only)	7 dflash_idle 0: indicate that serial flash is busy 1: indicate that serail flash is idle now, you cand send the next command
read_command	16'hFFE0	0000_0011	set the read command according to different serial flash
program_command	16'hFFE1	0000_0010	set the program command according to different serial flash
rdsr_command	16'hFFE2	0000_0101	set the read status register command according to different serial flash
wrsr_command	16'hFFE3	0000_0001	set the write status register command command according to different serial flash
erase_command	16'hFFE4	1101_1000	set the erase command according to different serial flash
rdid_command	16'hFFE5	1001_1111	set the read id command according to different serial

			flash
wren_command	16'hFFE6	0000_0110	set the write enable command according to different serial flash

7.2 MCU OPERATION:

1. ID READ:

- (1) 检查 dflash_idle (FFDF bit7), 确定为 1
- (2) 设置 interrupt_en (FFDE bit7), 使能中断
- (3) 设置 flash_id_read (FFD0 bit0) 为 1, 进入 ID READ 模式
- (4) 等待 dtfl_id_dw (FFDD bit0) 为 1, 标志 ID READ 结束, 从 FETCH_ID_REG0~3 (FFD3~FFD5) 读出 ID 的值
- (5) 等待 dflash_idle 为 1 (FFDF bit7), 标志 DATA FLASH 处于闲置状态, 可以进行下一步操作

2. SR READ

- (1) 检查 dflash_idle (FFDF bit7), 确定为 1
- (2) 设置 interrupt_en (FFDE bit7), 使能中断
- (3) 设置 flash_sr_read (FFD0 bit1) 为 1, 进入 ID READ 模式
- (4) 等待 dtfl_rsr_dw (FFDD bit1) 为 1, 标志 SR READ 结束, 从 RDDATA_REG (FFD2) 读出 sr 的值
- (5) 等待 dflash_idle (FFDF bit7), 标志 DATA FLASH 处于闲置状态, 可以进行下一步操作

3. SR WRITE

- (1) 检查 dflash_idle (FFDD bit7), 确定为 1
- (2) 设置 interrupt_en (FFDE bit7), 使能中断
- (3) 将欲写的值写入 WRDATA_REG (FFD1)
- (4) 设置 flash_sr_write (FFD0 bit2) 为 1, 进入 ID WRITE 模式
- (5) 等待 dtfl_wsr_dw (FFDD bit2) 为 1, 标志 SR WRITE 结束 (Note: 此时串行 flash 仍处于工作状态, 必须执行完下一步才可以进行下一个指令)
- (6) 等待 dflash_idle (FFDF bit7), 标志 DATA FLASH 处于闲置状态, 可以进行下一步操作

4. SECTOR ERASE

- (1) 检查 dflash_idle (FFDF bit7), 确定为 1
- (2) 设置 interrupt_en (FFDE bit7), 使能中断
- (3) 设置欲擦除的地址 flash_addr[23:0] (FFD6~FFD8)
- (4) 设置 flash_erase (FFD0 bit3) 为 1, 进入 FLASH ERASE 模式
- (5) 等待 dtfl_erase_dw (FFDD bit3) 为 1, 标志 SR READ 结束 (Note: 此时串行 flash 仍处于工作状态, 必须执行完下一步才可以进行下一个指令)
- (6) 等待 dflash_idle (FFDF bit7), 标志 DATA FLASH 处于闲置状态, 可以进行下一步操作

5. DATA READ

- (1) 检查 dflash_idle (FFDF bit7), 确定为 1
- (2) 设置 interrupt_en (FFDE bit7), 使能中断
- (3) 设置欲写入的地址 flash_addr[23:0] (FFD6~FFD8)
- (4) 设置 flash_data_read (FFD0 bit4) 为 1, 进入 DATA READ 模式
- (5) 等待 dtfl_rd_dw (FFDF bit4) 为 1, 标志 DATA READ 结束, 从 RDDATA_REG (FFD2) 读出 DATA 的值
- (6) 等待 dflash_idle (FFDF bit7), 标志 DATA FLASH 处于闲置状态, 可以进行下一步操作

6. DATA WRITE

- (1) 检查 dflash_idle (FFDF bit7), 确定为 1
- (2) 设置 interrupt_en (FFDE bit7), 使能中断

- (3) 设置欲写入的地址 flash_addr[23:0] (FFD6~FFD8)
- (4) 将欲写的值写入 WRDATA_REG (FFD1)
- (5) 设置 flash_data_write (FFD0 bit5) 为 1, 进入 DATA WRITE 模式
- (6) 等待 dtfl_wr_dw (FFDD bit5) 为 1, 标志 DATA WRITE 结束 (Note: 此时串行 flash 仍处于工作状态, 必须执行完下一步才可以进行下一个指令)
- (7) 等待 dflash_idle (FFDF bit7), 标志 DATA FLASH 处于闲置状态, 可以进行下一步操作

7. DATA RAM or FONT RAM LOAD

- (1) 检查 dflash_idle (FFDF bit7), 确定为 1
 - (2) 设置 interrupt_en (FFDE bit7), 使能中断
 - (3) 设置源地址 flash_addr[23:0] (FFD6~FFD8)
 - (4) 设置目的地址 ram_addr[13:0] (FFD9~FFDA)
- Note: 目的 ram 为 font_ram 时, 须将 FF0A bit7 设置为 0
目的 ram 为 data_ram 时, 须将 FF0A bit7 设置为 1

- (5) 设置欲装载的长度 load_length[13:0] (FFDB~FFDC)
- (6) 设置 flash_load_ram (FFD0 bit6) 为 1, 进入 RAM LOAD 模式
- (7) 等待 dtfl_load_dw (FFDD bit6) 为 1, 标志 RAM LOAD 结束
- (8) 等待 dflash_idle (FFDF bit7), 标志 DATA FLASH 处于闲置状态, 可以进行下一步操作

8. Flash Program

- (1) 检查 dflash_idle (FFDF bit7), 确定为 1
 - (2) 设置 interrupt_en (FFDE bit7), 使能中断
 - (3) 设置目的地址 flash_addr[23:0] (FFD6~FFD8)
 - (4) 设置源地址 ram_addr[13:0] (FFD9~FFDA)
- Note: 目的 ram 须为 data_ram 时, 将 FF0A bit7 设置为 1
- (5) 设置欲装载的长度 load_length[13:0] (FFDB~FFDC)
 - (6) 设置 flash_load_ram (FFD0 bit7) 为 1, 进入 RAM LOAD 模式
 - (7) 等待 dtfl_load_dw (FFDD bit7) 为 1, 标志 RAM LOAD 结束 (Note: 此时串行 flash 仍处于工作状态, 必须执行完下一步才可以进行下一个指令)
 - (8) 等待 dflash_idle (FFDF bit7), 标志 DATA FLASH 处于闲置状态, 可以进行下一步操作

Note:

1. 如果使用 SST 的串行 flash, 当使用 FLASH PROGRAM 的时候, 必须将 ctrl_reg[5](FFDE)设置为 1, 再根据 SST 的类型配置 ctrl_reg[4](FFDE)
2. 根据 FLASH 类型的不同, 在执行读操作时, 如果频率超过某一规定的值, 应该使用 Fast read 指令, 这时除了要配置 read_command 寄存器外, 还要将 ctrl_reg[6](FFDE)置为 1
3. 但进行片擦除时, 要将 ctrl_reg[3] (FFDE)置为 1

MCU INTERRUPTS

Interrupt	Description	Natural Priority	Interrupt Vector	AMT630A 配置
pfi	Power fail interrupt	0	33h	不开放
int0_n	External interrupt 0	1	03h	红外中断（反相）
TF0	Timer0 interrupt	2	0Bh	开放予外部使用
int1_n	External interrupt 1	3	13h	FLASH 中断
TF1	Timer 1 interrupt	4	1Bh	开放予外部使用
TI_0 or RI_0	Serial Port 0 transmit or receive	5	23h	
TF2 or EXF2	Timer 2 interrupt	6	2Bh	开放予外部使用
TI_1 or RI_1	Serial Port 1 transmit or receive	7	3Bh	不开放
int2	External interrupt 2	8	43h	采样 ADC 中断
int3_n	External interrupt 3	9	4Bh	开放予外部使用
int4	External interrupt 4	10	53h	开放予外部使用
int5_n	External interrupt 5	11	5Bh	OSD 场同步中断
wdti	Watchdog timer interrupt	12	63h	TIMER34 中断

Internal SFR Area

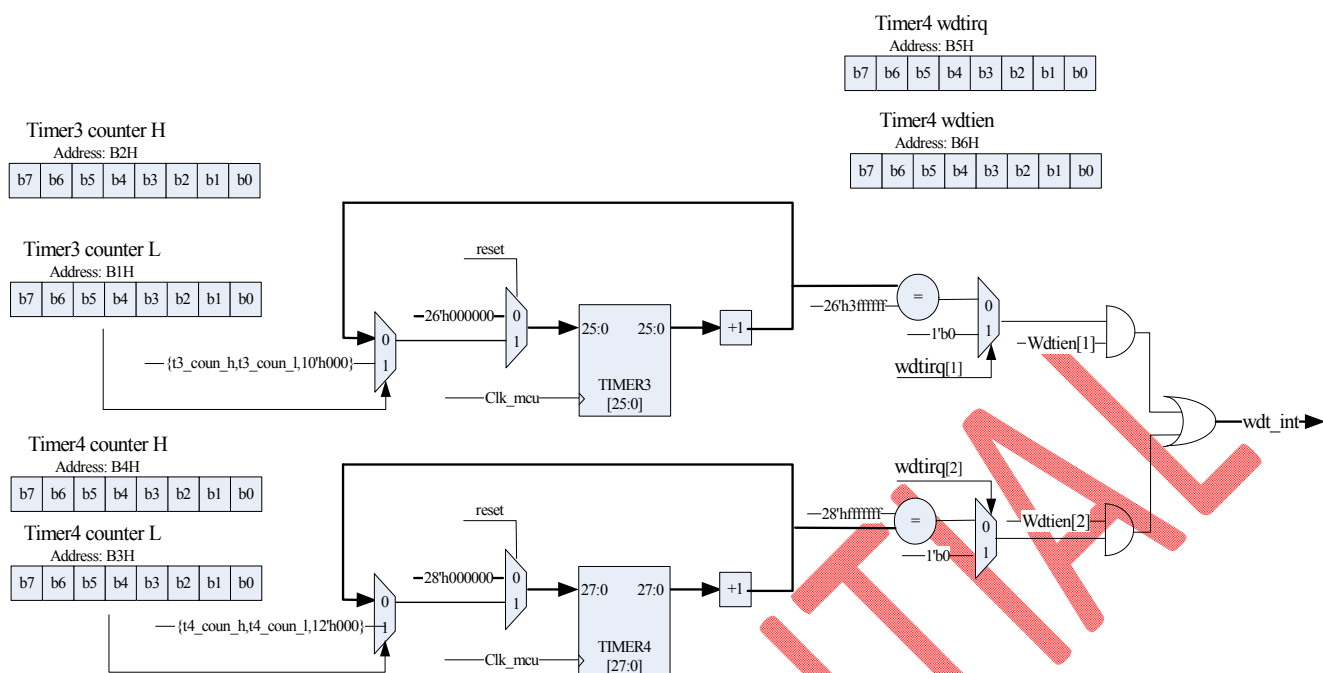
SFR Address	Register	SFR	Bit specification								Reset value
81h	SP	8051									07H
82h	DPL0	8051									00H
83h	DPH0	8051									00H
84h	DPL1 (1)	8051									00H
85h	DPH1 (1)	8051									00H
86h	DPS (1)	8051	0	0	0	0	0	0	0	SEL	00H
88h	TCON	8051	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
89h	TMOD	8051	GATE	C/Tn	M1	M0	GATE	C/Tn	M1	M0	00H
8Ah	TL0	8051									00H
8Bh	TL1	8051									00H

8Ch	TH0	8051									00H
8Dh	TH1	8051									00H
8Eh	CKCON (1,7)	8051			T2M	T1M	T0M	MD2	MD1	MD0	01H
8Fh	SPC_FNC (1)	8051	0	0	0	0	0	0	0	WRS	00H
91h	EXIF (1,4)	8051	IE5	IE4	IE3	IE2	1	0	0	0	08H
92h	MPAGE (1)	8051									00H
A8h	IE (6) (1,4)	8051	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	00H
B8h	IP (6)	8051	1	PS1	PT2	PS0	PT1	PX1	PT0	PX0	00H
C8h	T2CON (2)	8051	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2n	00H
CAh	RCAP2L (2)	8051									00H
CBh	RCAP2H (2)	8051									00H
CCh	TL2 (2)	8051									00H
CDh	TH2 (2)	8051									00H
D0h	PSW	8051	CY	AC	F0	RS1	RS0	OV	F1	P	00H
D8h	EICON (1,6)	8051	SMOD1	1	EPFI	PFI	WDTI	0	0	0	40H
E0h	ACC	8051									00H
E8h	EIE (1,4)	8051	1	1	1	EWDI	EX5	EX4	EX3	EX2	E0H
F0h	B	8051									00H
F8h	EIP	8051	1	1	1	PWDI	PX5	PX4	PX3	PX2	E0H

7.3 TIMER

外部中断: wdti, 中断号 12

B1H	cpu_cs_t3cntl	TIMER3 计数器高 16 位的低 8 位
B2H	cpu_cs_t3cnth	TIMER3 计数器高 16 位的高 8 位
B3H	cpu_cs_t4cntl	TIMER4 计数器高 16 位的低 8 位
B4H	cpu_cs_t4cnth	TIMER4 计数器高 16 位的高 8 位
B5H	cpu_cs_wdtirq	TIMER34 中断清零 [1] TIMER3 中断清零 [2] TIMER4 中断清零
B6H	cpu_cs_wdtien	TIMER34 中断使能 [1] TIMER3 中断使能 [2] TIMER4 中断使能

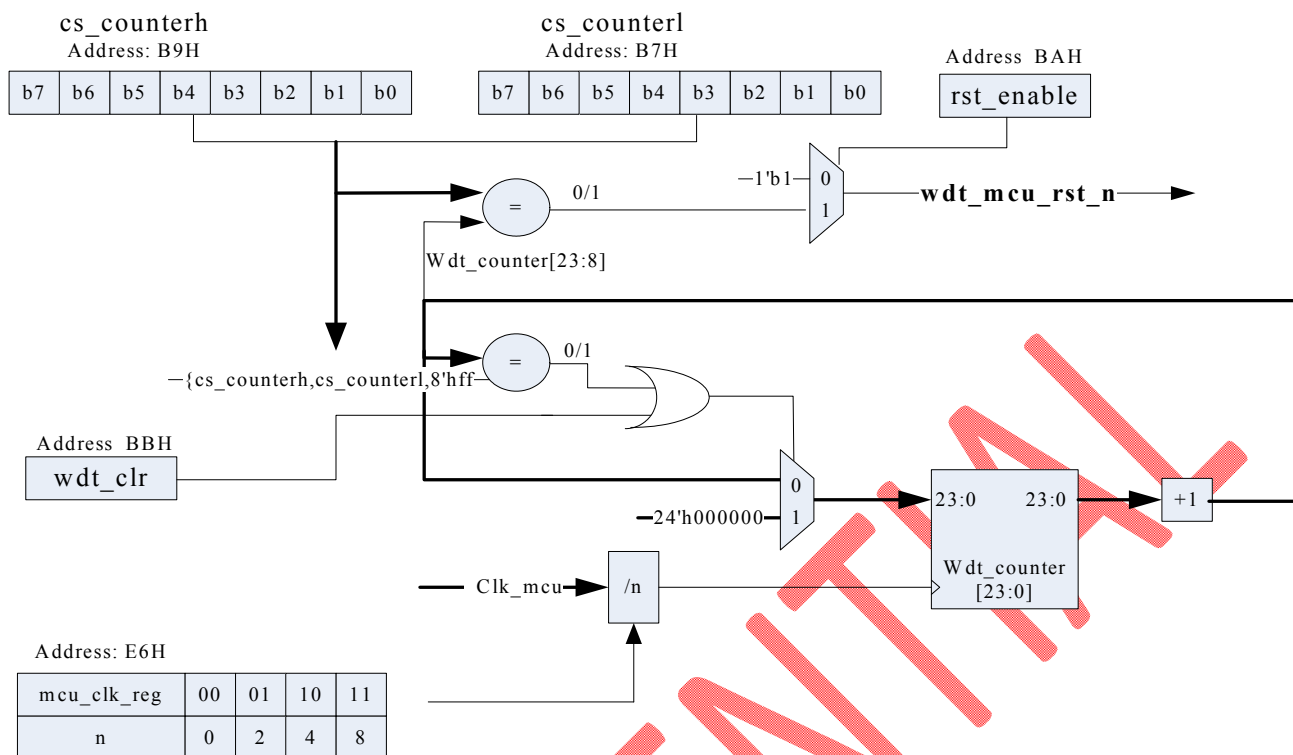


测试时钟：25MHz

T3 = 1024 * (1/25M) = 40.96us 1024 分频 40.96us 加 1
T4 = 4096 * (1/25M) = 163.84us 4096 分频 163.84us 加 1

7.4 WATCHDOG

BCH	cs_counterl	看门狗计数器中 8 位
BDH	cs_counterh	看门狗计数器高 8 位
BAH	cs_rst_enable	[0] : 0 将看门狗复位信号置为 1 1 看门狗复位信号不变
BBH	cs_wdt_clr	将看门狗计数器清零
B9H	mcu_clk_reg	[1:0] 对输入时钟进行分频 00: 不分频 01: 2 分频 10: 4 分频 11: 8 分频



测试时钟频率：25MHz $T = 256 * (1/25M) = 10.24\mu s$ 256 分频

2 分频：20.48us

4 分频：40.96us

8 分频：81.92us

7.5 GPIO

80H	p0_data[7:0]	<p>P0[7:6] gpio</p> <p>p0[5] scl_in 复用 I2C MASTER 输出</p> <p>p0[4] sda_in 复用 I2C MASTER 输出</p> <p>p0[2] rxd0 串口输入</p> <p>p0[1] remote_in 红外输入</p> <p>p0[0] gpio</p>
90H	p1_data[7:0]	<p>p1[0] t0</p> <p>p1[1] t1</p> <p>p1[2] t2ex</p> <p>p1[3] t2</p> <p>p1[4] int3_n</p> <p>p1[5] int4 p1[7:6] gpio</p>
A0H	p2_data[7:0]	gpio

B0H	p3_data[7:0]	GPIO
E9H	p0_oen	0 有效
EAH	p1_oen	0 有效
EBH	p2_oen	0 有效
ECH	p3_oen	0 有效
EDH	p0_ien	0 有效 PO_IEN[5]与 PO_IEN[4]只能为 0
EEH	p1_ien	0 有效
EFH	p2_ien	0 有效
F4H	p3_ien	0 有效
C6H	Mcu_cfg	<p>Bit 7 mcu_i2c_sel default 0 , logic 0 means i2c can read or write the internal register Logic 1 means mcu can read or write the internal register</p> <p>Bit[6:4] i2c_debounce sel default 100 001– debounce pulse width 0.08-0.16us 010 – debounce pulse width 0.16-0.33us 011 – debounce pulse width 0.33-0.66us 100 – debounce pulse width 0.66-1.3us 101 – debounce pulse width 1.3 -2.6us 110 – debounce pulse width 2.6-5.3us 111 – debounce pulse width 5.3-10.6us</p> <p>Bit3 font_ram_access_sel default 0 Logic 1 means data from spi write to fontram Logic 0 means mcu write data to font ram</p> <p>Bit2 Reserverd default 0</p> <p>Bit1 default 0 (mcu control) logic 0 means mcu run code from irom Logic 1 meas mcu run code form cache</p> <p>Bit0 mcu_soft_reset default 0(mcu control) Logic 1 only reset mcu</p>

当 P0、P1、P2 口设为输入状态（oen = 1'b1 且 ien=1'b0）时，内部 P0、P1、P2 对应寄存器接收外部输入值，其他状态时 P0、P1、P2 对应寄存器可由 MCU 配置，可作为 GPIO 使用。

8. ELECTRICAL SPECIFICATIONS

Parameter	Symbol	Min.	Typ.	Max.	Unit
3.3V Supply Voltages	DVDD	3.0	3.3	3.6	V
Input voltage	Vin	3.0	3.3	3.6	V
Abmient Operating Temperature	T _A	-20		80	°C
Storage Temperature	T _{STG}	-40		125	°C
Junction Temperature	T _J			125	°C