SPL161001A Programming Guide

V1.0 – Feb 13, 2004

SUNPLUS TECHNOLOGY CO. reserves the right to change this documentation without prior notice. Information provided by SUNPLUS TECHNOLOGY CO. is believed to be accurate and reliable. However, SUNPLUS TECHNOLOGY CO. makes no warranty for any errors which may appear in this document. Contact SUNPLUS TECHNOLOGY CO. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by SUNPLUS TECHNOLOGY CO. for any infringement of patent or other rights of third parties which may result from its use. In addition, SUNPLUS products are not authorized for use as critical components in life support devices/ systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Sunplus.

19, Innovation First Road • Science-Based Industrial Park • Hsin-Chu • Taiwan • R.O.C.

886-3-578-6005 886-3-578-4418 www.sunplus.com.tw

0 Table of Content

PAGE

SPL161001A PROGRAMMING GUIDE **1**
0 TABLE OF CONTENT **2**
1 REVISION HISTORY **6**
2 IMPORTANT NOTICE BEFORE WRITING YOUR PROGRAM **7**
3 INTRODUCTION **11**
3.1 General Description 11
3.2 Significant Features 11
3.3 Applications 12
4 SYSTEM CONTROL **13**
4.1 Introduction 13
4.2 Device Identification 14
4.3 Body Information 14
4.4 Reset Control 14
4.5 Clock Generation 16
4.6 System Reliability Control 18
4.7 Operation Mode Control 20
5 MEMORY **22**
5.1 Memory Mappings 22
5.2 Memory Access Pin Configuration 24
5.3 Control Registers 24
5.4 ROM/RAM size/Initializations 36
5.5 Main program and Test program Locations 36
5.6 Interrupt Vectors 36
5.7 Stack Location 37
5.8 Values for Unused Area 37
5.9 Chip-Select Settings in μ'nSP IDE 37
5.10 Program Examples 37
6 I/O PORTS **39**
6.1 Available Ports 39
6.2 General Purpose I/O Configuration 42
6.3 General Purpose I/Os Function Table 42
6.4 Control Register 43
6.5 I/O Structure Diagram 49

6.6 Special Notes 49
6.7 Program Examples 50
7 INTERRUPT 51
7.1 Introduction 51
7.2 Peripheral Interrupt Arrangement 52
7.3 Control Registers 54
7.4 Program Examples 60
8 TIMER/COUNTER 62
8.1 Timer Introduction 62
8.2 Timer Structure and Clock Source 62
8.3 Control Registers 68
8.4 Program Examples 76
9 TIMEBASE 77
9.1 Introduction 77
9.2 Timebase structure and clock source 77
9.3 Control Registers 77
9.4 Program Examples 80
10 REAL TIME CLOCK (RTC) 81
10.1 Introduction 81
10.2 RTC Structure and clock source 81
10.3 Control Registers 82
10.4 Program Examples 86
11 AUDIO OUTPUT 88
11.1 DAC/PWM 88
11.2 DAC Operation 89
11.3 PWM Operation 89
11.4 Structure 92
11.5 Speech Mode and Tone Mode 93
11.6 Control Registers 93
11.7 Mute level 98
11.8 Ramp Up and Ramp Down 98
11.9 Program Examples (Playing a PCM Speech) 99
12 LCD 101
12.1 Introduction 101
12.2 LCD Control Pin Configuration 101
12.3 LCD Buffer 101
12.4 LCD Buffer Layout 103

12.5 Control Registers 103
12.6 Operation during Wait/Halt/Standby & Wakeup Procedure 110
12.7 LCD Image Resource File Tooling 111
12.8 Program Examples 111
12.9 LCD Driver Selection Guide 112
13 UART / IRDA INTERFACE 115
13.1 Introduction 115
13.2 Structure and Block Diagram 116
13.3 UART / IrDA SIR Frame Scheme 117
13.4 UART/IrDA Control Pin Configuration 118
13.5 Control registers 118
13.6 Program Examples 129
14 SERIAL PERIPHERAL INTERFACE (SPI) 130
14.1 Introduction 130
14.2 Structure 130
14.3 SPI Control Pin Configuration 131
14.4 Master Mode 131
14.5 Control Registers 133
14.6 Program Examples 136
15 SERIAL INTERFACE 138
15.1 Introduction 138
15.2 Control Registers 139
15.3 Program Example 141
16 NAND FLASH INTERFACE 143
16.1 Introduction 143
16.2 Control Registers 143
16.3 Program Example 150
16.3.1 8-bit Mode Without DMA 150
16.3.2 16-bit Mode With DMA 152
17 MISCELLANEOUS 155
17.1 Introduction 155
17.2 Specified Register 155
17.3 Program Example 156
18 LINK YOUR PROGRAM 158
19 APPENDIX 160
19.1 Commonly Used Abbreviations 160
19.2 Control Register Mapping List (by function) 161

19.3 Control Registers Summary....162
19.4 System Clock Usage Guide....167
19.5 32768Hz Crystal Usage Guide....167
19.6 Development (Emulation) Board Configuration....168
19.6.1 Power Adjustment & Selection....171
19.6.2 The Dual Power System....171
19.6.3 Reset....171
19.6.4 ICE....172
19.6.5 Memory....173
19.6.6 NAND Flash Interface....174
19.6.7 Audio Output....174
19.6.8 Others Input and Output....175
19.7 Development (Emulation) Board Schematic....177
19.8 Header File for C or Assembly Language....186
19.9 Differences Between Emulation Chip and Real Chip....193
19.10 Differences Between SPL16256A and SPL161001A....194
19.11 CPU Performance Downgrading Issue....195
19.12 Audio Output Components Selection Guide....197
19.13 32768 Crystal and PLL Power-on Stable Time....198
19.14 Reset Type....199
19.15 Important Note for the Setup of Memory Access Time....201
19.16 Project Setting on IDE....202

1 REVISION HISTORY

Version	Date	Author	Description
1.0	02/13/2004	Bryan Wang	First edition

2 Important Notice Before Writing Your Program

- **Confirmation Sheet**

  It is highly recommended that programmers obtain a copy of the appropriate confirmation sheet before writing a program. The confirmation sheet, as a requisite document before placing orders, contains useful information and a checklist that helps avoid mistakes during development. Due to the updates of the confirmation sheet, please download the newest confirmation sheet from http://www.sunplus.com.tw. The following checklist is an abstract of the confirmation sheet for each series.

  Also, please refer to ***appendix: difference between emulation chip and real chip*** to avoid possible errors.

Customer		Date	
Project title		Code No.	
Project description			
CPU working voltage (except VDDC/VDDP)	☐2.4V ~ 3.6V	VDDC/VDDP voltage	☐2.4V ~ 3.6V
CPU Operation Frequency	_____MHz	Crystal (32768Hz)	☐Check
LCD Configuration	#COM:_____ #SEG: _____ Bias: _____ Duty: _____ V_{LCD}: _____V Mode: ☐ Gray-level ☐ Mono (Black & White) Bit-per-pixel: ☐ 1 ☐ 4		
LCD Driver Interface	Frame Rate: _____ Hz Interface Data Width: ☐ 1-bit ☐4-bit (shared with IOD[6..4]) ☐ 8-bit (shared with IOD[6..4] & IOA[3..0]) LCD Driver Components: ☐ SPLD Series Body no: _____ ☐ SPLC Series Body no: _____ ☐ Others Body no: _____		
Release code file (fill "0xFFFF" for unused area)			
Binary filename (*.tsk):		Binary file check-sum:	
SRAM size: 6K words (0x00000~0x01FFF)		☐ Check	
Page0 ROM size: 31K words (0x08000~0x0FBFF)		☐ Check	
Non-Page0 ROM size: 96K words (0x10000~0x27FFF)		☐ Check	
SUNPLUS test program: 1K words (0xFC00~0xFFEF)		☐ Check (Reserved Area)	

Input / Output																
IOA Port																
	b15	b14	b13	b12	b11	b10	b9	B8	b7	b6	B5	b4	b3	b2	b1	b0
Input																
Output																
Used																
eup																
Special Function	TimerB CCP	TimerA CCP	EXTA	EXTB									LCD Interface D[7..4]			
Used																
IOB Port																
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Input																
Output																
Wakeup																
Special Function					SPI CLK,DO,DI											
Used																
IOC Port																
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Input																
Output																
Special Function	MRE	MWE	CLE	ALE	RX	TX	WEB	REB	ND7	ND6	ND5	ND4	ND3	ND2	ND1 SCK	ND0 SDA
Used																
IOD Port																
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Input																
Output																
Special Function			Memory Address Bus MA[22:16]							LCD Interface D[3..0]			Memory Chip Select CS0,CS1,CS2,CS3			
Used																

Hardware /Software	
LVR (Low Voltage Reset)	Always Enable: (☐2.2V ☐2.5V ☐2.7V)
LVD (Low Voltage detector)	☐Disable ☐Enable: (☐2.3V ☐2.5V ☐2.8V ☐3.1V)
Internal Memory Configuration	_____ (Value written to 0x704C) SUNPLUS recommend value is 0x0002
External Memory (CS0) Configuration	☐Disable ☐Enable: Start Address: _____ End Address: _____ Type: ☐SRAM ☐ROM ☐FLASH ☐Others Size:_____ Access Time:_____ Extra Wait Cycle:_____ (Value written to 0x7020)
External Memory (CS1) Configuration	☐Disable ☐Enable: Start Address: _____ End Address: _____ Type: ☐SRAM ☐ROM ☐FLASH ☐Others Size:_____ Access Time:_____ Extra Wait Cycle:_____ (Value written to 0x7021)
External Memory (CS2) Configuration	☐Disable ☐Enable: Start Address: _____ End Address: _____ Type: ☐SRAM ☐ROM ☐FLASH ☐Others Size:_____ Access Time_____ Extra Wait Cycle:_____ (Value written to 0x7022)
External Memory (CS3) Configuration	☐Disable ☐Enable: Start Address: _____ End Address: _____ Type: ☐SRAM ☐ROM ☐FLASH ☐Others Size:_____ Access Time:_____ Extra Wait Cycle:_____ (Value written to 0x7023)
UART	☐Disable ☐Enable: if Enable , Maximum Baud Rate = _____
IrDA	☐Disable ☐Enable: if Enable , Maximum Baud Rate = _____
LCD Buffer Never configures on internal mask ROM	☐Check

Do NOT use system clock cycle to make important and accuracy-demanded timing	☐Check
Watchdog enable (Software Write-once Option) If watchdog is enabled, please check the following: (A). Watchdog port ($7012H) must be cleared by writing 0xAXX5 within designated duration	☐Yes ☐No ☐Check

General programming checklist

The general programming checklist intends to provide some general characteristics about SUNPLUS devices. It is the customer's responsibility to check all the information in the list. No responsibility is assumed by SUNPLUS for any non-checked box even this confirmation sheet has been approved by SUNPLUS. Make sure the following conditions are met and verified.

All used SRAM must be initialized after power on (strongly recommended).	☐Check
Make sure that the used SRAM variables are not over stack reserved area.	☐Check
Make sure that the Interrupt section is located in the page0 or is declared as .TEXT section ($08000-$0FBFF).	☐Check
(A) Make sure that no current leakage in I/O or speaker amplifier during sleeping. (B) Make sure that all I/Os are not floating during sleeping.	☐Check ☐Check
Non-used I/O ports must be masked off (for input process). Example, if IOA[0:7] are input: R1 = [P_IOA_Data]; ; Read I/O port A Data R1 & = 0x00FF; ; Mask higher byte CMP R1,0x0011; ; Lower byte is available for compare operation	☐Check

Document version

To make sure that the correct version of document is used, please fill out the followings:

(A) SPL161001A Programming Guide Version_____

(B) μ'nSP™ IDE User Manual Title and Version_____

(C) Other documents (if any)_____

Development tool / board version

To make sure the correct version of hardware is used, please fill out the followings:

(A) Emulation Board Version_____

(B) Emulation Chip Version_____

(C) Piggyback Version (if used) _____

(D) Others (if any)_____

To make sure that the correct version of software is used, please fill out the following:

(A) μ'nSP™ IDE Version_____

(B) SACM Library Version (if used)_____

(C) Others (if any)_____

For Third Party Application

Cyberon voice recognition solution is applied in this code ☐Yes ☐No

If yes, please fill out the followings:

(A) Types of solutions: ☐SI only ☐SD only ☐SID(both SI & SD) ☐SV

(B) VStar Modeling Toolkit IDE Version_____

(C) VStar Library Version: BSRV_____.lib

(D) VStar-SDK Programming Guide Version _____

(E) Others (if any)_____

3 Introduction

3.1 General Description

The SPL161001A is a 16-bit architecture LCD controller product which carries the Sunplus 16-bit microprocessor, called *μ'nSP®* (pronounced as *micro-n-SP*). Its high performance assures that the *μ'nSP®* is capable of handling complex digital signal process easily and efficiently. Therefore, the SPL161001A is applicable, but not limited, to the areas of digital sound processing and voice recognition. The memory capacity includes 128K-word mask ROM plus an 8K-word SRAM. The LCD controller is able to support a maximum of 320x320 dots 16 gray-level or monochrome display. Other features include 43 programmable multi-functional I/Os, three 16-bit timers/counters, 32768Hz real time clock, low voltage reset/detection, NAND-gate flash interface, SPI, and plus many others.

3.2 Significant Features

- SUNPLUS 16-bit CPU (*μ'nSP®*) running at 50MHz @ 2.4V ~ 3.6V.
- Triple-Clock System (Phase Lock Loop, R-oscillator, and 32768 Crystal)
- Flexible Operations: Wait//Halt/Sleep for power-management
- Dual-Power System (VDDC/VDDP can be supplied with a lower voltage de decrease power consumption)
- Address extensible to 4M words
- Built-in Internal 8K-word SRAM, 128K-word high speed ROM
- Powerful LCD controller supporting up to 320x320 dots 16 gray-level or monochrome display
- 2 Channel 10-bit PWM or DAC audio outputs
- UART & IrDA with receive and transmit FIFO
- 4 chip selection pins to access external memory
- 3 16-bit re-loadable timer/counters with capture, comparison, and PWM functions
- 1 SPI (Serial Peripheral Interface)
- Real Time Clock (RTC) supports auto-update to hour and an alarm comparison register
- Programmable 4-level (2.3V, 2.5V, 2.8V and 3.1V) low voltage detector
- Built-in 3-level (2.2V, 2.5V and 2.7V) low voltage reset
- Embedded In-Circuit-Emulation
- More system reliability features: watchdog, illegal write reset flag, mode protection for write error, watchdog protection for write error.

3.3 Applications

- Electronic spoken/non-spoken dictionary
- Kid storybook, E-book
- Advanced educational toys
- Hand-Held multimedia LCD game
- Electronic learning assistant
- Personal organizer

4 System Control

4.1 Introduction

This chapter describes the body information, reset option, system clock, system reliability, and operation mode. The features are as follows:

- Body Information
- Built-in 32768Hz crystal circuit
- Built-in R-oscillator clock source from 1.06MHz to 8.50MHz
- Built-in Phase-Lock Loop (PLL), which pumps from 32768Hz up to 51.6096MHz
- Adjustable PLL clock output frequency, from 3.6864MHz to 51.6096MHz
- Support high/slow speed mode for CPU (Slow speed comes from 32768Hz and R-oscillator, and high speed from PLL)
- 4 Level Low Voltage Detect (LVD)
- 3 Level Low Voltage Reset (LVR)
- Build-in Watchdog Timer
- Support wait mode, halt mode and sleep mode for power management

System Control Register Summary Table:

Name	Address	Description
P_BodyID	0x7000	Body Identification Number Register
P_BodyInfo	0x7001	Body Information
P_Reset_Flag	0x7006	Reset Event Flag Register
P_Clock_Ctrl	0x7007	System Clock Control Register
P_LVR_Ctrl	0x7008	Low Voltage Reset Control Register
P_LVD_Ctrl	0x7009	Low Voltage Detection Control Register
P_Watchdog_Ctrl	0x700A	Watchdog Control Register
P_Watchdog_Clear	0x700B	Watchdog Clear Register
P_WAIT	0x700C	Wait Mode Entrance Register
P_HALT	0x700D	Halt Mode Entrance Register
P_SLEEP	0x700E	Sleep Mode Entrance Register

4.2 Device Identification

P_BodyID **0x7000** **SUNPLUS Body ID Number**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	0x9999															
Default	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	BODYID	R	Body Identification Register For SPL161001A, the ID number is fixed to 0x9999	0x9999 for SPL161001A

4.3 Body Information

P_BodyInfo **0x7001** **SUNPLUS Body Information**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IRAMSIZE						IROMSIZE									
Default	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Function	Type	Description	Condition
[15:10]	IRAMSIZE	R	Internal SRAM size (unit: 1KW) These 6 bits defines the internal SRAM size	001000 for SPL161001A (8KW internal SRAM)
[9:0]	IROMSIZE	R	Internal ROM size (unit: 32KW) These 10 bits defines the internal ROM size.	0000000100 for SPL161001A (128KW internal ROM)

4.4 Reset Control

P_Reset_Flag **0x7006** **Reset Event Flag Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	WDG	WDE	MPE	ILWR	LVR
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

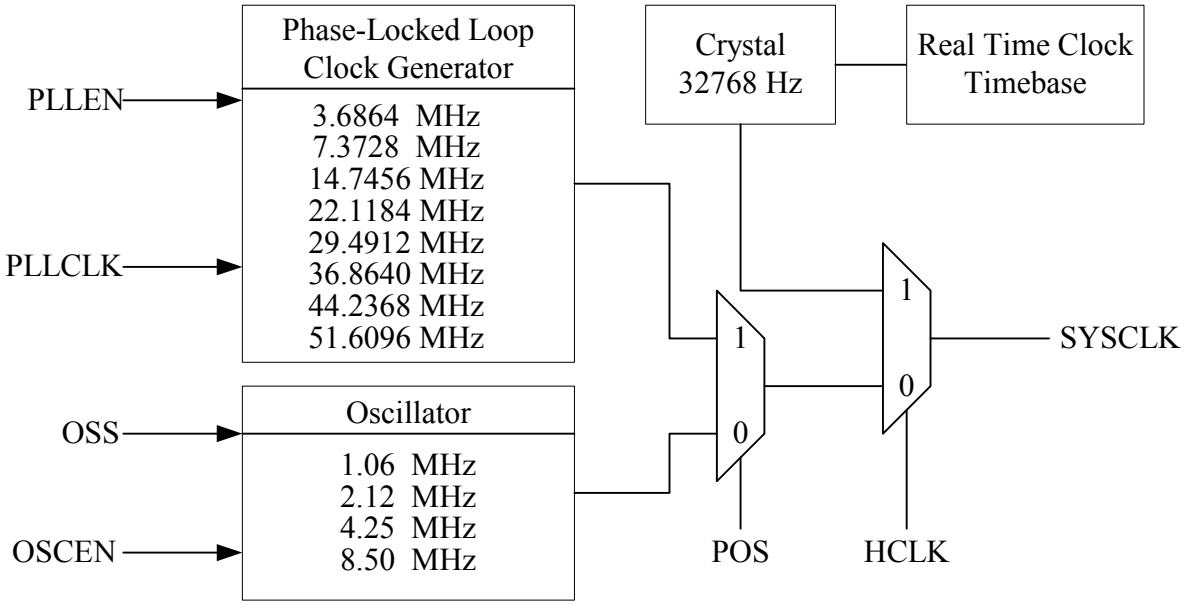
Bit	Function	Type	Description	Condition
[15:5]			Reserved	
4	WDG	R/W	Watchdog Timeout Reset Flag	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect

Bit	Function	Type	Description	Condition
				Write 1= Clear the flag
3	WDE	R/W	Watchdog Error Write Flag If programmers do NOT write 0xA005 to clear watchdog timer, this bit will be set to “1” by SPL161001A, causing CPU reset	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
2	MPE	R/W	Mode Protect Error Write Flag If programmers do NOT write 0x5005 to enter wait mode, or NOT write 0x500A to enter halt mode, or NOT write 0xA00A to enter standby mode, this bit will be set to”1” by SPL161001A, causing CPU to be reset.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
1	ILWR	R/W	Illegal Write Reset Flag If chip selects x, device is configured as read-only (CSxEN=0) and CPU attempts to perform write operation on chip select x area. Next, this flag will be set to “1” by SPL161001A, causing CPU to be reset.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
0	LVR	R/W	Low Voltage Reset Flag If SPL161001A power is below designated threshold voltage, this flag will be set to “1” by SPL161001A, causing CPU to be reset. The threshold voltage is defined in P_LVR_Ctrl.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag

Note: Program can confirm the type of reset which is activated by reading the corresponding bit.

4.5 Clock Generation

External 32768Hz crystal accompanied with SPL161001A built-in circuit provides precise clock source. On the other hand, the phase-lock-loop (PLL) circuit provides the required clocks for CPU and peripherals. The PLL whose default frequency is 3.6864MHz, pumps the frequency from 3.6864MHz to 51.6096Mhz. Eight PLL pump frequencies can be easily switched by software. Also, a low frequency built-in R-oscillator is provided to save power. Programmers can switch the system clock frequency anytime when needed. Following diagram shows the control block diagram of clock sources.



P_Clock_Ctrl **0x7007** **System Clock Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PLLEN	HCLK	OSE	POS	-	-	-	-	-	-	OSS		-	PLLCLK		
Default	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	PLLEN	R/W	System Clock Enable This control bit is used to enable internal PLL logic circuit that supports master clock source for most of modules in SPL161001A. This master clock frequency is programmable by only modifying PLLCLK control bits Note that “System clock” will be abbreviated as	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
			SYSCLK in the programming guide.	
14	HCLK	R/W	CPU Clock Selection This control bit is used to select system clock between 32768Hz, PLL, and R-oscillator. System clock and R-oscillator will be abbreviated as “SYSCLK” and “R-OSC” in the reset of the programming guide.	0= Low speed clock, 32768Hz 1= High Speed clock, PLL / R-OSC
13	OSE	R/W	R-OSC enable	0= R-OSC off 1= R-OSC on
12	POS	R/W	System clock selection	0= R-OSC 1= PLL
[11:6]			Reserved	
[5:4]	OSS	R/W	R-OSC clock selection	00=1.06MHz 01=2.12MHz 10=4.25MHz 11=8.50MHz
3			Reserved	
[2:0]	PLLCLK		PLL Clock Selection Note: CPU will first run an additional 512 SYSCLK cycles (based on 32768Hz frequency) before any changing system clock.	000= SYSCLK = 3.6864MHZ 001= SYSCLK = 7.3728MHZ 010= SYSCLK = 14.7456MHZ 011= SYSCLK = 22.1184MHZ 100= SYSCLK = 29.4912MHZ 101= SYSCLK = 36.8640MHZ 110= SYSCLK = 44.2368MHZ 111= SYSCLK = 51.6096MHZ

Note that not any combination of the control bits PLLEN, HCLK, OSCEN and POS are effective. The valid combination are as the following table. All other settings are invalid.

PLLEN	HCLK	OSCEN	POS	SYSCLK Source
0	0	1	0	32768 XTAL
0	1	1	0	R-OSC
1	0	0	1	32768 XTAL
1	0	1	0	32768 XTAL
1	0	1	1	32768 XTAL
1	1	0	1	PLL
1	1	1	0	R-OSC
1	1	1	1	PLL

Changing bits 12 to 15 with bits 0 to 2 within the same instruction will cause the clock to be unstable. It is also the case with bits 12 to 15 and bits 4 to 5. For example, if the user wants to change the clock setting of 0x7007 from 0xD000 to 0x6037, he or she should follow the sequence: 0xD000, 0xD007, 0xD037, and finally 0x6037.
If user wants to change the clock source from R-OSC to PLL, please wait after 512 clock cycles to ensure that the PLL clock is stable. Also, before turning on the R-oscillator, OSS should be set to the desired value.

4.6 System Reliability Control

P_LVR_Ctrl **0x7008** **Low Voltage Reset Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LVREN	-	-	-	-	-	-	-	-	-	-	-	-	-	LVRS	
Default	1	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*

Bit	Function	Type	Description	Condition
15	LVREN	R/W	Write “0” once to disable low voltage reset	0=Disabled 1=Enabled
[14:2]			Reserved	
[1:0]	LVRS	R/W	Low Voltage Reset Level Selection This control bit is to select low voltage reset level. For voltage drops below 2.2V, 2.5V or 2.7V, LVR activates. The LVR only resets the kernel CPU, not for other peripheral functional blocks.	00=2.2V, 01=2.5V 10=2.7V, 11=2.7V *The default value depends on the combination of EMUEN and IROMEN.

EMUEN	IROMEN	Default
0	0	10
0	1	00
1	X	10

P_LVD_Ctrl **0x7009** **Low Voltage Detect Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LVDIF/C	LVDIEN	LVDEN	-	-	-	-	-	-	-	-	-	-	-	LVDS	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Function	Type	Description	Condition
15	LVDIF/C	R/W	Low Voltage Detection Interrupt flag This bit is set when the voltage on LVDP pin drops below the programmed LVD voltage level	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect

Bit	Function	Type	Description	Condition
			and is cleared by software writing this bit '1'.	Write 1= Clear the flag
14	LVDIEN	R/W	Low Voltage Detection Interrupt Enable If this bit is set to "1", and LVD interrupt occurred, hardware will issue an IRQ6 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked off. To select IRQ6 and FIQ, please refer to ***Chapter Interrupt***	0= Disabled 1= Enabled
13	LVDEN	R/W	Low Voltage Detection Enable If this bit is set to "1", low voltage detection function is enabled. If this bit is cleared to "0", the LVD will be disabled.	0= Disabled 1= Enabled
[12:2]			Reserved	
[1:0]	LVDS	R/W	Low Voltage Detection Level selection	00=2.3V 01=2.5V 10=2.8V 11=3.1V

P_Watchdog_Ctrl **0x700A** **Watchdog Reset Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	WDGEN	WDGS	-	-	-	-	-	-	-	-	-	-	-	WDGPD		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	WDGEN	R/W	Write once to enable Watchdog reset	0= Disabled 1= Enabled
14	WDGS	W	Write once to select Reset Target For more information about system-reset and CPU-reset, refer to ***Appendix***.	0= Reset System (including CPU) 1= Reset CPU
[13:3]			Reserved	
[2:0]	WDGPD	R/W	Interval of watchdog reset taking place	000= 2 seconds 001= 1 second 010= 0.5 seconds 011= 0.25 seconds 1X0= 0.125 seconds 1X1= 62.5 milliseconds

P_Watchdog_Clear **0X700B** **Watchdog Clear Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	WDGC															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	WDGC	W	Watchdog Clear Register Write AXX5 to clear watchdog timer only when watchdog is enabled (i.e. 0X700A[15]=1). Write other values to reset the system or CPU.	

4.7 Operation Mode Control

The SPL161001A has three operation modes - Wait, Halt, and Sleep modes. In the sleep mode, it shuts down CPU to save power, almost no power is consumed. Please refer to the following table. Note that these three modes will all yield CPU to be powered down.

	CPU	System Clock (PLL / R-OSC)	32768Hz Xtal	After wakeup
Wait Mode	OFF	ON	ON	Next Instruction
Halt Mode	OFF	OFF	ON (RTC)	Reset CPU
Sleep Mode	OFF	OFF	OFF	Reset CPU

To determine if CPU is power-on reset or wakeup from Halt mode or sleep mode in reset procedure, refer to ***chapter: Interrupt***

P_WAIT **0x700C** **Wait Mode Entrance Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	WAIT															
Default																

Bit	Function	Type	Description	Condition
[15:0]	WAIT	W	Wait Mode Entrance Register Writing 5XX5 enters Wait mode (stop SYSCLK source only, but PLL / R-OSC and 32768Hz are still valid)	

P_HALT **0x700D** **Halt Mode Entrance Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	HALT															
Default																

Bit	Function	Type	Description	Condition
[15:0]	HALT	W	Halt Mode Entrance Register Write 5XXA to enter halt mode (stop SYSCLK, but 32768Hz remains working). The RTC is still capable of running in this mode.	

P_SLEEP **0x700E** **Sleep Mode Entrance Register**

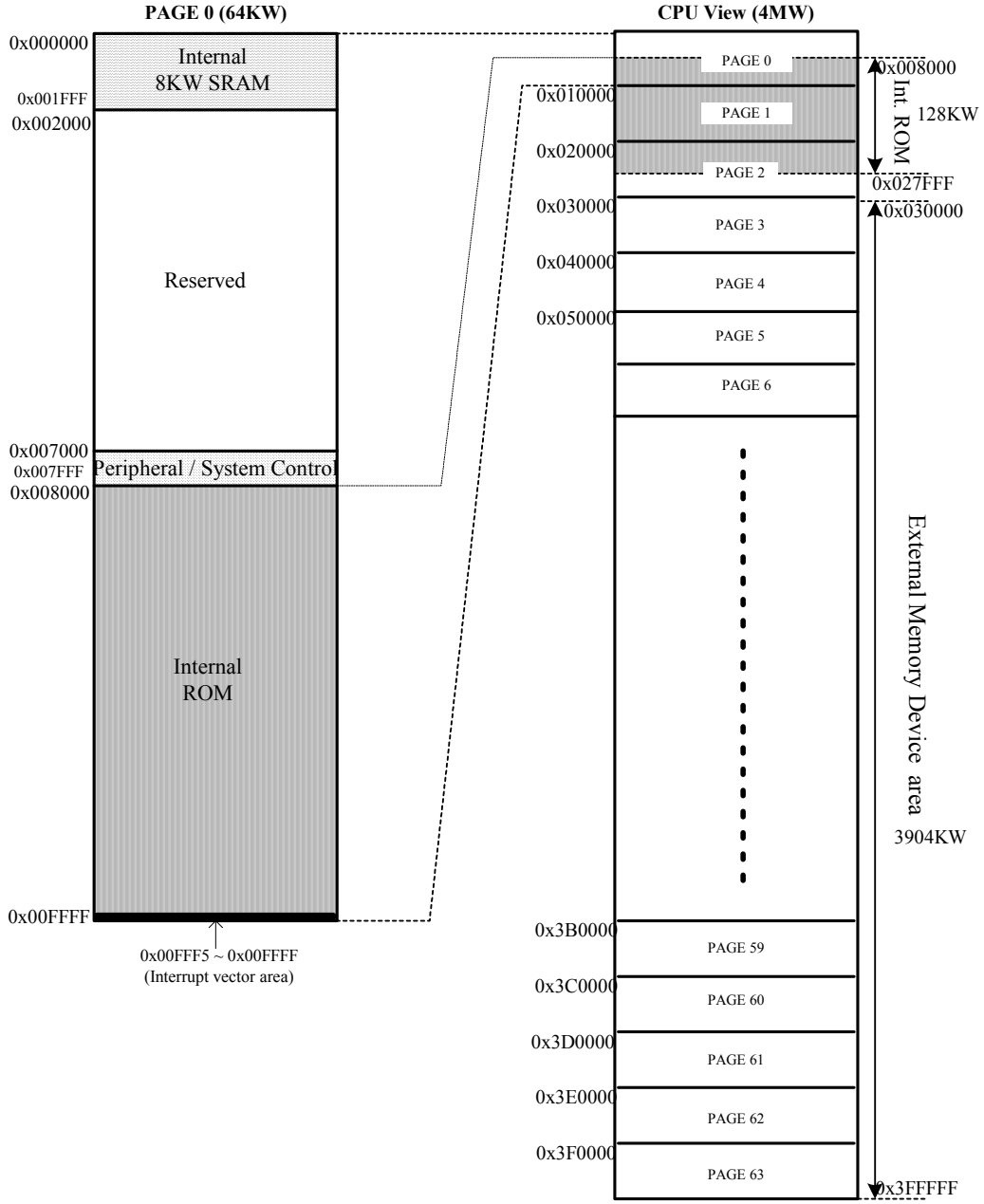
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SLP															
Default																

Bit	Function	Type	Description	Condition
[15:0]	SLP	W	Sleep Mode Entrance Register Write AXXA to enter standby mode (stop all clock sources)	

5 Memory

5.1 Memory Mappings

The SPL161001A has an 8K-word SRAM and a 128K-word ROM. Associated with external memory devices. The SPL161001A is able to address up to 4M-word locations*.



* Note: with certain configurations and controls, the SPL161001A can address up to 32M-words. Please refer to the description of P_Page60_BR through P_Page63_BR.

Each page size is 64K-word. The SPL161001A can access up to 64 pages. The page starts from Page0, Page1, Page2, …, to Page63. The internal resource resides in first 3 pages and external memory (device) locates in the rest 61 pages. In addition, theSPL161001A supports four chip selection signals to enable four external memories (devices).

The 8K-word Internal SRAM (including stack) area is located in 0x000000 ~ 0x001FFF, and system control registers and peripheral control registers reside in the 4K-word area, from 0x007000 to 0x007FFF. Note that the 16-bit CPU (*μ'nSP®*) inside the SPL161001A offers high performance operation on the first 64 words SRAM (located at 0x000000 to 0x00003F). For detailed CPU cycle performance, refer to ***SUNPLUS μ'nSP® Assembly Tools User Manual***.

Internal SRAM, system control register and peripheral control registers are positioned in the upper 32K-word of Page0. The layout of internal ROM starts from the lower 32K-word in page0, and ends up in the upper 32K-word of Page4 (from 0x008000 to 0x027FFF). Therefore, internal resource will be stored in Page0, Page1, and Page2. Note that the higher 32K-word of Page2 is reserved (0x028000~0x02FFFF).

On the other hand, the four external devices can be arbitrarily distributed in other 61 pages, totally 3904K-word. In other words, the size of each four devices can be programmable by software, but the addressing space of these four devices must be mutually exclusive.

See the following formula:

Size of (External Device0 + External Device1 + External Device2 + External Device3) ≤ 3904K words

The method used to define the address mapping of four external memory devices is to setup the size of each memory device. In the SPL161001A, the start address of CS0 is 0x030000. After the size of CS0 is given, the CS1 address is defined implicitly. That is 0x030000 + size of CS0 (the unit is 64K or 0x010000). After the CS1 is determined, the start address of CS2 is automatically defined. That is also the case for CS3.

Example:

Suppose we have two 64K-Word SRAMs, a 256K-Word Flash memory and a 512K-Word ROM

Arrangement the following:

CS0: 0x030000 ~ 0x03FFFF as first SRAM

CS1: 0x040000 ~ 0x04FFFF as second SRAM

CS2: 0x050000 ~ 0x08FFFF as 256KW Flash

CS3: 0x90000 ~ 0x10FFFF as 512KW ROM

In most cases, not all four external device are present, nor all 23 address lines used. Therefore, the SPL161001A allows system designer to alternate unused CS0, CS1, CS2, CS3 and address bus pins (A16 ~ A22) or even read/write signal pins (#WE, #RE) to general purposed I/O. These pins' default settings are memory control signals after CPU resets. After configuring the control registers, these signals can be one of the GPIOs in PortD or PortC. For more details about its position, please refer to ***Chapter***: ***I/O Ports***. The GPIO alternation is accomplished by write-once mechanism. That is, unless the system is reset or these pins cannot be return to their default.

5.2 Memory Access Pin Configuration

Name	I/O	Description
MD[15:0]	I/O	Memory 16-bit data bus (Dedicated)
MA[15:0]	O	Memory address bus [15:0] (Dedicated)
MA[21:16]	O	Memory address bus [22:16] (shared with GPIO PortD[12:7])
CS[3:0]	O	Memory chip selection signal (shared with GPIO PortD[0:3])
WE	O	Memory write enable control signal (shared with GPIO PortC14)
RE	O	Memory read enable control signal (shared with GPIO PortC15)

5.3 Control Registers

Memory Control Register Summary Table

Name	Address	Description
P_MCS0_Ctrl	0x7020	Chip Selection 0 Memory Device Control Register
P_MCS1_Ctrl	0x7021	Chip Selection 1 Memory Device Control Register
P_MCS2_Ctrl	0x7022	Chip Selection 2 Memory Device Control Register
P_MCS3_Ctrl	0x7023	Chip Selection 3 Memory Device Control Register
P_MCS3_Timing	0x7024	Chip Selection 3 Memory Device Timing Control Register
P_Page63_BR	0x7028	Page 63 Bank Register
P_Page62_BR	0x7029	Page 62 Bank Register
P_Page61_BR	0x702A	Page 61 Bank Register
P_Page60_BR	0x702B	Page 60 Bank Register
P_MA16_Disable	0x7030	Memory Address 16 Disable (GPIO PortD7 Enable) Write-once Register
P_MA17_Disable	0x7031	Memory Address 17 Disable (GPIO PortD8 Enable) Write-once Register
P_MA18_Disable	0x7032	Memory Address 18 Disable (GPIO PortD9 Enable) Write-once Register
P_MA19_Disable	0x7033	Memory Address 19 Disable (GPIO PortD10 Enable) Write-once Register
P_MA20_Disable	0x7034	Memory Address 20 Disable (GPIO PortD11 Enable) Write-once Register
P_MA21_Disable	0x7035	Memory Address 21 Disable (GPIO PortD12 Enable) Write-once Register

Name	Address	Description
P_MA22_Disable	0x7036	Memory Address 22 Disable (GPIO PortD13 Enable) Write-once Register
P_MCS0_Disable	0x7040	Chip Selection 0 Disable (GPIO PortD3 Enable) Write-once Register
P_MCS1_Disable	0x7041	Chip Selection 1 Disable (GPIO PortD2 Enable) Write-once Register
P_MCS2_Disable	0x7042	Chip Selection 2 Disable (GPIO PortD1 Enable) Write-once Register
P_MCS3_Disable	0x7043	Chip Selection 3 Disable (GPIO PortD0 Enable) Write-once Register
P_MRE_Disable	0x7048	Memory Read Enable Signal Disable (GPIO PortC15 Enable) Write-once Register
P_MWE_Disable	0x7049	Memory Write Enable Signal Disable (GPIO PortC14 Enable) Write-once Register
P_Madr_Status	0x704A	Memory Address Signal Enable Status Register
P_MCtrl_Status	0x704B	Memory Control Signal (CS[3:0], WE, RE) Enable Status Register
P_InMem_Ctrl	0x704C	Internal Memory Device Control Register

P_InMem_Ctrl, P_MCS0_Ctrl, P_MCS1_Ctrl, P_MCS2_Ctrl, and P_MCS3_Ctrl are control registers for internal SRAM, internal ROM and memory devices on chip selection 0, 1, 2, and 3. To make system operation more reliable, programmers should setup appropriate wait cycles for each internal or external device on these five control registers. The longer wait period is, the more reliable system is. The shorter the wait cycle is, the higher performance the system has, but the less reliable the system will be.

Device access time should **always** be smaller than designated wait cycle determined by number of system clocks. The access time specification of memory device is usually taken under certain operating condition. Therefore, margins should be considered while determining the period of wait cycle, especially when system voltage varies and when there are many memory devices in the system.

P_InMem_Ctrl **0x704C** **Internal Memory Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	IRAMW		IOW	-	-	-	IROMW			
Default	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:8]	IRAMW	R/W	Internal RAM Access Wait State Setup Criterion: [(IRAMW[3:0] + 2) * SYSCLK cycle] > Internal RAM access time Note that IROMW should be set at least “1”, that is IROMW can NOT be set as “0000”	Range= 1 ~3 Tw= (IRAMW[1:0] + 2) * SYSCLK

Bit	Function	Type	Description	Condition
7	IOW		I/O Port Access Wait State	Tw=(IOW + 2)*SYSCLK
[6:4]			Reserved	
[3:0]	IROMW	R/W	Internal ROM Access Wait State Setup Criterion: [(IROMW[3:0] + 2) * SYSCLK cycle] > Internal ROM access time Note that IROMW should be set at least “1”, that is IROMW can NOT be set as “0000”	Range= 1 ~15 Tw= (IROMW[3:0] + 2) * SYSCLK

P_MCS0_Ctrl **0x7020** **CS0 Device Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	CS0SIZE							CS0MD	-	-	-	CS0WAIT			
Default	0	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
15			Reserved	
[14:8]	CS0SIZE	R/W	Memory Device Size on Chip Selection 0 (CS0), unit is 64K words. “CS0SIZE[6:0] - 2 ” defines the number of page for the entire memory device on CS0. Page size is 64K-word.	Range= 0 ~58 Size=(CS0SIZE[6:0] -2)* 64K-word Size Range= 64K-word ~ 3904K-word
7	CS0MD	R/W	CS0 Memory Device Access Mode If this bit is set to “1”, the memory device on CS0 supports both read and write operations and if this bit is cleared to “0”, the device can only support read access.	0= Read Only 1= Read / Write
[6:4]			Reserved	
[3:0]	CS0WAIT	R/W	CS0 Memory Device Access Wait State Setup Criterion: [(CS0WAIT[3:0] + 2) * SYSCLK cycle] > memory device access time	Range= 0 ~15 Tw= (CS0WAIT[3:0] + 2) * SYSCLK

P_MCS1_Ctrl **0x7021** **CS1 Device Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	CS1SIZE							CS1MD	-	-	-	CS1WAIT			
Default	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
15			Reserved	
[14:8]	CS1SIZE	R/W	Memory Device Size on Chip Selection 1 (CS1), unit is 64K words. “CS1SIZE[6:0] + 1” defines the number of page for the whole memory device on CS0. Page size is 64K-word.	Range= 0 ~58 Size=(CS1SIZE [6:0] +1)* 64K-word Size Range= 64K-word ~ 3904K-word
7	CS1MD	R/W	CS1 Memory Device Access Mode If this bit is set to “1”, the memory device on CS1 supports both read and write operations. If this bit is cleared to “0”, the device can only support read access.	0= Read Only 1= Read / Write
[6:4]			Reserved	
[3:0]	CS1WAIT	R/W	CS1 Memory Device Access Wait State Setup Criterion: [(CS1WAIT[3:0] + 2) * SYSCLK cycle] > memory device access time	Range= 0 ~15 T= (CS1WAIT[3:0] + 2) * SYSCLK

P_MCS2_Ctrl **0x7022** **CS2 Device Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	CS2SIZE							CS2MD	-	-	-	CS2WAIT			
Default	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
15			Reserved	
[14:8]	CS2SIZE	R/W	Memory Device Size on Chip Selection 2 (CS2) unit is 64K words. “CS2SIZE[6:0] + 1” defines the number of page for entire memory device on CS0. Page size is 64K-word.	Range= 0 ~58 Size=(CS2SIZE[6:0] + 1)* 64K-word Size Range= 64K-word ~ 3904K-word
7	CS2MD	R/W	CS2 Memory Device Access Mode If this bit is set to “1”, the memory device on CS2 supports both read and write operations and if this bit is cleared to “0”, the device can only support read access.	0= Read Only 1= Read / Write
[6:4]			Reserved	
[3:0]	CS2WAIT	R/W	CS2 Memory Device Access Wait State Setup Criterion: [(CS2WAIT[3:0] + 2) * SYSCLK cycle] > memory device access time	Range= 0 ~15 Tw= (CS2WAIT[3:0] + 2) * SYSCLK

P_MCS3_Ctrl **0x7023** **CS3 Device Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	CS3SIZE							CS3MD	-	-	-	CS3WAIT			
Default	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
15			Reserved	
[14:8]	CS3SIZE	R/W	Memory Device Size on Chip Selection 3 (CS3), unit is 64K words. “CS3SIZE[6:0] + 1” defines the number of page for the whole memory device on CS0. Page size is 64K-word.	Range= 0 ~58 Size=(CS3SIZE[6:0] +1) *64K-word Size Range= 64K-word ~ 3904K-word
7	CS3MD	R/W	CS3 Memory Device Access Mode If this bit is set to “1”, the memory device on CS3 supports both read and write operations. If this bit is cleared to “0”, the device can only support read access.	0= Read Only 1= Read / Write
[6:4]			Reserved	
[3:0]	CS3WAIT	R/W	CS3 Memory Device Access Wait State Setup Criterion: [(CS3WAIT[3:0] + 2) x SYSCLK cycle] > memory device access time	Range= 0 ~15 Tw= (CS3WAIT[3:0] + 2) * SYSCLK

Note that the total size of four external memory devices is restricted to 3904K-word (61 pages). Therefore, value of (CS0SIZE + CS1SIZE + CS2SIZE + CS3SIZE-1) should be smaller than 61. The memory map location of these four external memories is sequentially starting from CS0 to CS3.

The CS0MD, CS1MD, CS2MD, and CS3MD are control registers to define the external memory for read only or read/write option.

The SPL161001A allows programmers to define wait cycle for each internal and external device by setting IRAMW, IROMW, CS0WAIT, CS1WAIT, CS2WAIT, and CS3WAIT. The default wait cycles are 17 system clock cycles to ensure reliability.

Following control registers are used to change the memory address signals and memory control signals to GPIO function. After power-on reset, these pins are in default as memory access signals. If the number of external memory devices are less than 4 or they are of small size, some of the memory addresses or control signals can be GPIO. To change these pins to GPIO, programmers must write certain pattern to the corresponding write-once control register. That is, they can be written once only, and further writing to the same register has no effect. After system reset, programmers have to re-disable these memory addresses or control pins to GPIO.

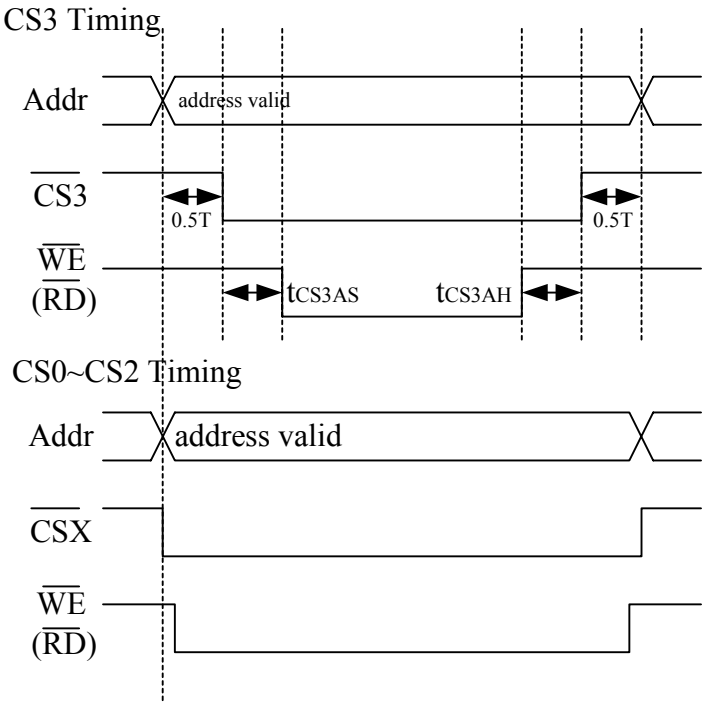
Read P_Madr_Status and P_MCtrl_Status to determine if the memory access signals are enabled or are used as GPIO.

P_MCS3_Timing 0x7024 CS3 Device Timing Control Register

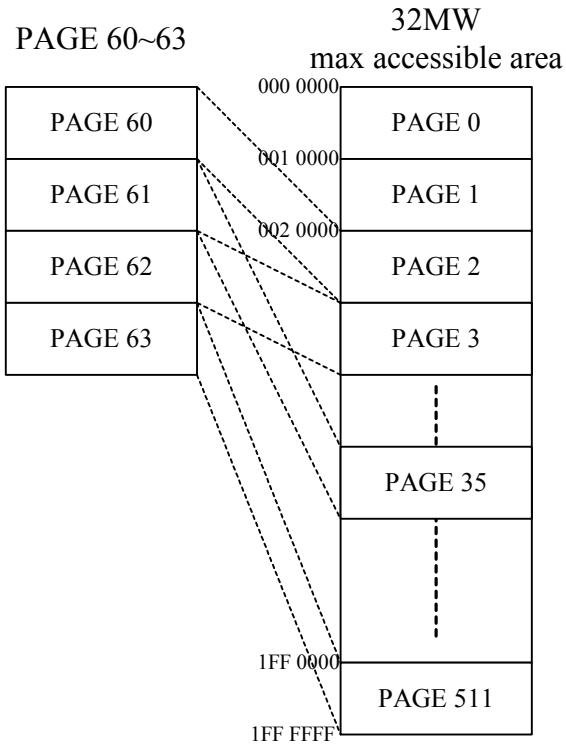
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CS3AS		CS3AH		-											
Init	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
[15:14]	CS3AS	R/W	Address setup time (address valid to WEB falling edge)	Bit[15:14]=00 : 0.5 SYSCLK cycle 01: 1.5 SYSCLK cycle 10: 2.5 SYSCLK cycle 11: 3.5 SYSCLK cycle
[13:12]	CS3AH	R/W	Address hold time (WEB rising edge to address)	Bit[13:12]=00 : 0.5 SYSCLK cycle 01: 1.5 SYSCLK cycle 10: 2.5 SYSCLK cycle 11: 3.5 SYSCLK cycle
[11:0]			Reserved	

Among all the chip select signals, only CS3 address setup time and hold time can be adjusted. This makes CS3 more flexible for special memory devices, say, Compact Flash cards. Please refer to the following diagram of the timings. In the diagram, T stands for a SYSCLK cycle.



As mentioned in the previous sections, the SPL161001A is able to totally access 32MW. However, in the CPU view, it can only see 4MW. To access 32MW, page 60~63 are able to be mapped onto any page in a total memory space of 32MW. 32MW can be divided into 512 pages, which can be represented in 9 bits. Note that since the SPL161001A only has 23 address pins, each chip select can connect up to an 8MW memory device. That is, 32MW can only consist of four 8MW memory devices. In this example, page 60 to 63 are mapped onto page 2, 35, 3, 511, respectively.



P_Page63_BR **0x7028** **Page 63 Bank Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	MAP3F								
Init	-	-	-	-	-	-	-	0	0	0	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
[8:0]	MAP3F	R/W	Page 63 Bank Register In order to expand CPU addressing space, fill in the page number in 32MW memory space. Default = 63 (0x3F)	

P_Page62_BR **0x7029** **Page 62 Bank Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	MAP3E								
Init	-	-	-	-	-	-	-	0	0	0	1	1	1	1	1	0

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
[8:0]	MAP3E	R/W	PAGE 62 Bank Register In order to expand CPU addressing space, fill in the page number in 32MW memory space. Default = 62 (0x3E)	

P_Page61_BR **0x702A** **Page 61 Bank Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	MAP3D								
Init	-	-	-	-	-	-	-	0	0	0	1	1	1	1	0	1

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
[8:0]	MAP3D	R/W	PAGE 61 Bank Register In order to expand CPU addressing space, fill in the page number in 32MW memory space. Default = 61 (0x3D)	

P_Page60_BR **0x702B** **Page 60 Bank Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	MAP3C								
Init	-	-	-	-	-	-	-	0	0	0	1	1	1	1	0	0

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
[8:0]	MAP3C	R/W	PAGE 60 Bank Register In order to expand CPU addressing space, fill in the page number in 32MW memory space. Default = 60 (0x3C)	

P_MA16_Disable **0x7030** **Memory Address 16 Disable (Write Once)**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	MA16DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	MA16DIS	W	When write “0x5A5A” to this control register, PortD7 becomes GPIO. Otherwise, PortD7 remains Memory Address 16 as the default.	MA16EN=0 if write “0x5A5A” once

P_MA17_Disable 0x7031 Memory Address 17 Disable (Write Once)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	MA17DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	MA17DIS	W	When write “0x5A5A” to this control register, PortD8 becomes GPIO. Otherwise, PortD8 remains Memory Address 17 as the default.	MA17EN=0 if write “0x5A5A” once

P_MA18_Disable 0x7032 Memory Address 18 Disable (Write Once)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	MA18DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	MA18DIS	W	When write “0x5A5A” to this control register, PortD9 becomes GPIO. Otherwise, PortD9 remains Memory Address 18 as the default.	MA18EN=0 if write “0x5A5A” once

P_MA19_Disable 0x7033 Memory Address 19 Disable (Write Once)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	MA19DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	MA19DIS	W	When write “0x5A5A” to this control register, PortD10 becomes GPIO. Otherwise, PortD10 remains Memory Address 19 as the default.	MA19EN=0 if write “0x5A5A” once

P_MA20_Disable 0x7034 Memory Address 20 Disable (Write Once)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	MA20DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	MA20DIS	W	When write “0x5A5A” to this control register, PortD11 becomes GPIO. Otherwise, PortD11 remains Memory Address 20 as the default.	MA20EN=0 if write “0x5A5A” once.

P_MA21_Disable 0x7035 Memory Address 21 Disable (Write Once)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	MA21DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	MA21DIS	W	When write “0x5A5A” to this control register, PortD12 becomes GPIO. Otherwise, PortD12 remains Memory Address 21 as the default.	MA21EN=0 if write “0x5A5A” once.

P_MA22_Disable 0x7036 Memory Address 22 Disable (Write Once)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	MA22DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	MA22DIS	W	When write “0x5A5A” to this control register, PortD13 becomes GPIO. Otherwise, PortD13 remains Memory Address 22 as the default.	MA22EN=0 if write “0x5A5A” once.

P_MCS0_Disable 0x7040 Chip Selection 0 Disable (Write Once)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CS0DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	CS0DIS	W	When write “0x5A5A” to this control register, PortD3 becomes GPIO. Otherwise, PortD3 remains Chip Selection 0 as the default.	CS0EN=0 if write “0x5A5A” once.

P_MCS1_Disable 0x7041 Chip Selection 1 Disable (Write Once)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CS1DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	CS1DIS	W	When write “0x5A5A” to this control register, PortD2 becomes GPIO. Otherwise, PortD2 remains Chip Selection 1 as the default.	CS1EN=0 if write “0x5A5A” once.

P_MCS2_Disable **0x7042** **Chip Selection 2 Disable (Write Once)**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CS2DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	CS2DIS	W	When write “0x5A5A” to this control register, PortD1 becomes GPIO. Otherwise, PortD1 remains Chip Selection 2 as the default.	CS2EN=0 if write “0x5A5A” once.

P_MCS3_Disable **0x7043** **Chip Selection 0 Disable (Write Once)**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CS3DIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	CS3DIS	W	When write “0x5A5A” to this control register, PortD0 becomes GPIO. Otherwise, PortD0 remains Chip Selection 3 as the default.	CS3EN=0 if write “0x5A5A” once.

P_MRE_Disable **0x7048** **Memory Read Enable Disable (Write Once)**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	REDIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	REDIS	W	When write “0x5A5A” to this control register, PortC15 becomes GPIO. Otherwise, PortC15 remains Read Enable Signal as the default.	REEN =0 if write “0x5A5A” once.

P_MWE_Disable **0x7049** **Memory Write Enable Disable (Write Once)**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	WEDIS															
Default																

Bit	Function	Type	Description	Condition
[15:0]	WEDIS	W	When write “0x5A5A” to this control register, PortC14 becomes GPIO. Or, PortC14 remains Write Enable Signal as the default.	WEEN=0 if write “0x5A5A” once.

P_MAdr_Status 0x704A Memory Address Signal Status Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	MA22EN	MA21EN	MA20EN	MA19EN	MA18EN	MA17EN	MA16EN
Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
6	MA22EN	R	Memory Address 22 Enable status.	0= Disabled (PortD13 is GPIO) 1= Enabled (PortD13 is MA21)
5	MA21EN	R	Memory Address 21 Enable status.	0= Disabled (PortD12 is GPIO) 1= Enabled (PortD12 is MA21)
4	MA20EN	R	Memory Address 20 Enable status.	0= Disabled (PortD11 is GPIO) 1= Enabled (PortD11 is MA20)
3	MA19EN	R	Memory Address 19 Enable status.	0= Disabled (PortD10 is GPIO) 1= Enabled (PortD10 is MA19)
2	MA18EN	R	Memory Address 18 Enable status.	0= Disabled (PortD9 is GPIO) 1= Enabled (PortD9 is MA18)
1	MA17EN	R	Memory Address 17 Enable status.	0= Disabled (PortD8 is GPIO) 1= Enabled (PortD8 is MA17)
0	MA16EN	R	Memory Address 16 Enable status.	0= Disabled (PortD7 is GPIO) 1= Enabled (PortD7 is MA16)

P_MCtrl_Status 0x704B Memory Control Signal Status Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	CS3EN	CS2EN	CS1EN	CS0EN	WEEN	REEN
Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
5	CS3EN	R	Memory Device Chip Selection 3 Enable Status.	0= Disabled (PortD0 is GPIO) 1= Enabled (PortD0 is CS3)
4	CS2EN	R	Memory Device Chip Selection 2 Enable Status.	0= Disabled (PortD1 is GPIO) 1= Enabled (PortD0 is CS2)
3	CS1EN	R	Memory Device Chip Selection 1 Enable Status	0= Disabled (PortD2 is GPIO)

Bit	Function	Type	Description	Condition
				1= Enabled (PortD0 is CS1)
2	CS0EN	R	Memory Device Chip Selection 0 Enable Status.	0= Disabled (PortD3 is GPIO) 1= Enabled (PortD0 is CS0)
1	WEEN	R	Memory Device Write Signal Enable Status.	0= Disabled (PortC14 is GPIO) 1= Enabled (PortD0 is WE)
0	REEN	R	Memory Device Read Signal Enable Status.	0= Disabled (PortC15 is GPIO) 1= Enabled (PortD0 is RE)

5.4 ROM/RAM size/Initializations

The 8K-word SRAM (including stack) area is located in 0x000000 ~ 0x001FFF. The operating cycle in the first 64 words (0x000000 ~ 0x00003F) is 5-cycle, and operating cycle in other areas (0x000040 ~ 0x001FFF) is 7-cycle. Programmers may take advantage of the first 64 words of SRAM to achieve higher performance. In other words, placing the most frequently used variables in the first 64 words will speed up the running of software.

5.5 Main program and Test program Locations

Main program can be started from 0x008000 residing with Internal ROM. However, SUNPLUS reserves 0x00FC00 ~ 0X00FFEF(about 1K-word) as test program area. Since SUNPLUS will add the test pattern after code is released, programmers should not read nor write this area for any purpose.

5.6 Interrupt Vectors

Interrupt Vector	
Address	Function
0x00FFF5	Software Break
0x00FFF6	Fast IRQ
0x00FFF7	RESET
0x00FFF8	IRQ0
0x00FFF9	IRQ1
0x00FFFA	IRQ2
0x00FFFB	IRQ3
0x00FFFC	IRQ4
0x00FFFD	IRQ5
0x00FFFE	IRQ6
0x00FFFF	IRQ7

5.7 Stack Location

SUNPLUS recommends that stack starts at the end of Internal SRAM; that is, stack pointer is set to 0x001FFF. Stack area will grow bottom-up.

5.8 Values for Unused Area

SUNPLUS recommends that users write 0xFFFF to unused ROM area.

5.9 Chip-Select Settings in µ'nSP IDE

SUNPLUS µ'nSP IDE provides a dialog box to set up external CS0,1,2,3 configuration. With which, programmers can setup the CS0,1,2,3 Size, Mode and Wait State easily and quickly. Moreover, programmers can review start and end addresses of each CS0,1,2,3 memory in other dialog boxes after setting.

Note that the setting in this dialog box is for downloading only. Therefore, programmers should add appropriate program codes of setting P_MCS0_Ctrl, P_MCS1_Ctrl, P_MCS2_Ctrl and P_MCS3_Ctrl in their projects.

SUNPLUS recommends programmers to set this dialog box before downloading programs; otherwise, IDE may fail to download. Furthermore, the dialog box can also be used to determine whether each memory device (CS0, CS1, CS2, CS3) needs to be downloaded to save time on downloading.

For more detail, refer to ***Appendix: Project Setting in IDE***.

5.10 Program Examples

```
R1 = 0x0381                 // Internal mask ROM wait state = 3 system cycle
[P_InMEM_Ctrl] = R1         // Internal RAM wait state = 5 system cycle
                            // I/O wait state = 3 system cycle

R1 = 0x0380                 // SRAM, 64KW,   35ns access time
[P_MCS0_Ctrl] = R1          // External CS0 memory wait state = 2 system cycle
                            // CPU memory mapping is 0x030000 ~ 0x03FFFF

R1 = 0x0181                 // SRAM, 64KW,   50ns access time
[P_MCS1_Ctrl] = R1          // External CS1 memory wait state = 3 system cycle
                            // CPU memory mapping is 0x040000 ~ 0x04FFFF
```

```
R1 = 0x0782                 // FLASH, 256KW,   70ns access time
[P_MCS2_Ctrl] = R1          // External CS2 memory wait state = 4 system cycle
                            // CPU memory mapping is 0x050000 ~ 0x08FFFF

R1 = 0x0703                 // ROM, 512KW,   90ns access time
[P_MCS3_Ctrl] = R1          // External CS3 memory wait state = 5 system cycle
                            // CPU memory mapping is 0x90000 ~ 0x10FFFF
```

6 I/O Ports

6.1 Available Ports

Four programmable I/O ports are available in SPL161001A: PortA, PortB, PortC, and PortD. Each I/O pin can be bit-by-bit configured by software programming. Virtually every I/O pin on these 4 ports can be programmed as special function I/Os. In other words, many special function control signals are shared with I/O ports. PortA and PortB also provide wakeup capability.

For PortA, PortB, PortC[13..0], PortD[6..4], and PortD[15:14], default configuration is GPIO. On the other hand, the default configuration of PortC[15,14], PortD[13:7] and PortD[3:0] are memory access signals. To change PortA, PortB, PortC[13..0], PortD[6..4], and PortD[15:14] from GPIO functions to special functions, programmers only need to enable the corresponding special functions. When special functions are activated, GPIO function will be disabled. Any setup on GPIO will be invalid. To change each memory control signals to GPIO function, programmers have to setup corresponding write-once register for each I/O pin. These registers are write-once for system reliability.

The following table depicts shared information about I/O port and the special functions.

PortA and Special Functions Shared Information

	15	14	13	12									3	2	1	0
Special Function	Timer Specified Output		External Input										LCD Interface D[7..4]			
Signal	CCPB	CCPA	EXTB NF_B	EXTA									LCD4	LCD5	LCD6	LCD7
In/Out	I/O	I/O	I	I									O	O	O	O

PortB and Special Functions Shared Information

					11	10	9									
Special Function					Serial Peripheral Interface											
Signal					SPICK	SPIDO	SPIDI									
In/Out					I/O	O	I									

PORT C and Special Functions Shared Information

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Special Function	Memory Signals		NAND Flash I/F		UART/IrDA Interface		Nand Flash Interface/Sunplus Serial Interface									
Signal	RE#	WE#	CLE	ALE	RX	TX	WEB	REB	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
															SCK	SDA
In/Out	O	O	I	O	I	O	O	O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O

Note that PortC[15,14] are initialized as special function (Memory control signal WE#, RE#) after power on reset. To configure PortC[15,14] as GPIO(s), programmers must use write-once registers which depicts in ***Chapter: Memory***.

PortD and Special Functions Shared Information

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Special Function			Memory Address Bus [24:16]							LCD Interface D[3..1]			Memory Chip Selections			
Signal			MA22	MA21	MA20	MA19	MA18	MA17	MA16	LCD3	LCD2	LCD1	MCS0	MCS1	MCS2	MCS3
In/Out	I/O	I/O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

Note that all I/O pins on PortD[3:0] and PortD[13:7] are initialized as special functions (Memory control signal) after power on reset. To configure memory access signal from PortD as GPIO(s), programmers must use write-once registers depicted in ***chapter: Memory***.

Unlike other GPIOs, IOA[15:12] pins have more than one special function. The following tables depict the special functions for each port respectively.

IOA12 Pin Special Function Shared Information

Special Function	Supported I/O Mode	Enable Control bit	Description
External Interrupt A	Floating With pull-low resistor With pull-high resistor	P_MINT_Ctrl.bit0=1	When one of these modes enabled, direction control bit (P_IOA_Dir.bit12) is forced to “0”. (That is, original content of direction control bit will be ignored). Programmers can configure 3 input modes by modifying corresponding data and attribute control bit directly (P_IOA_Data.bit12, P_IOA_Attrib.bit12). 00 = with pull-low resistor, 01= with pull–high resistor. 1X= floating
TimerA Clock Source	Floating With pull-low resistor With pull-high resistor	P_TimerA_Ctrl.bit[3:0]=7	
TimerB Clock Source	Floating With pull-low resistor With pull-high resistor	P_TimerB_Ctrl.bit[3:0]=7	
TimerC Clock Source	Floating With pull-low resistor With pull-high resistor	P_TimerC_Ctrl.bit[3:0]=7	
GPIO	All modes		

IOA13 Pin Special Function Shared Information

Special Function	I/O Mode Supported	Enable Control bit	Description
External Interrupt B	Floating With pull-low resistor With pull-high resistor	P_MINT_Ctrl.bit1=1	When one of these modes is enabled, direction control bit (P_IOA_Dir.bit13) is forced to "0". (That is, original content of direction control bit will be ignored). Programmers can configure 3 input modes by modifying corresponding data and attribute control bit directly (P_IOA_Data.bit13, P_IOA_Attrib.bit13). 00 = with pull-low resistor, 01= with pull–high resistor. 1X= floating
TimerA Clock Source	Floating With pull-low resistor With pull-high resistor	P_TimerA_Ctrl.bit[6:4]=7	
NAND Flash B/R	Floating With pull-low resistor With pull-high resistor	P_NF_Ctrl.bit[13]=1	
GPIO	All modes		

IOA14 Pin Special Function Shared Information

Special Function	I/O Mode Supported	Enable Control bit	Description
TimerA Capture Mode Trigger Input	Floating	P_TimerA_CCP_Ctrl. bit[15:14]=01	When one of these modes is enabled, direction/attribute/data control bits are forced to corresponding mode. The contents of direction/attribute/data control bits have no effect on this I/O pad.
TimerA Comparison Mode Event Output	Output Buffer (High or Low)	P_TimerA_CCP_Ctrl. bit[15:14]=10	
TimerA PWM Mode Signal Output	Output Buffer (High or Low)	P_TimerA_CCP_Ctrl. bit[15:14]=11	
GPIO	All modes		

IOA15 Pin Special Function Shared Information

Special Function	I/O Mode Supported	Enable Control bit	Description
TimerB Capture Mode Trigger Input	Floating	P_TimerB_CCP_Ctrl. bit[15:14]=01	When one of these modes is enabled, direction/attribute/data control bits are forced to corresponding mode. The contents of the direction/attribute/data control bits have no effect on this I/O pad.
TimerB Comparison Mode Event Output	Output Buffer (High or Low)	P_TimerB_CCP_Ctrl. bit[15:14]=10	
TimerB PWM Mode Signal Output	Output Buffer (High or Low)	P_TimerB_CCP_Ctrl. bit[15:14]=11	
GPIO	All modes		

Note that Key wakeup function of PortA and PortB are valid only when their corresponding special functions are disabled. In other words, if one of the PortA or PortB pins is configured as its special function, any voltage level change on this pin will NOT wake up CPU, even though key change enable bit (P_MINT_Ctrl.bit14) is set to “1” before CPU going to sleep.

On the other hand, if key change enable bit (P_MINT_Ctrl.bit14) is set to “1”, CPU can still be awakened even though IOA13 and IOA12 pins are configured to special function (EXTB, EXTA). For details, refer to ***Chapter: Interrupt***.

6.2 General Purpose I/O Configuration

The SPL161001A provides a bit-to-bit I/O configuration and every I/O configuration can be defined individually. To setup a bit configuration, three control registers must be setup: Data, Attribute, and Direction. The following table is a summary of I/O configuration setup. Each corresponding bit in these three control registers should be given a value. For example, suppose PortA.0 is used as input with pull low. The bit0 in PortA DIRECTION, ATTRIBUTE and DATA control registers should be given “000” in binary. If PortA.1 is used as input with float and wakeup functions, the bit1 in PortA DIRECTION, ATTRIBUTE and DATA control registers should be given “010” in binary.

Reading operation on I/O DATA control registers will perform status read from external I/O pad. On the other hand, the SPL161001A also provides a control register named I/O BUFFER register. This I/O BUFFER control register holds the setting data (value) that is previously written into I/O DATA control register. Therefore, programmers do not need extra SRAM to store (hold) the previous setting on DATA control register.

6.3 General Purpose I/Os Function Table

Direction Register	Attribution Register	Data Register	Function	Wakeup	Description
0	0	0	Pull Low*	Yes**	Input with pull low
0	0	1	Pull High	Yes**	Input with pull high
0	1	0	Float	Yes**	Float (High Impendence)
0	1	1	Float	No	Float (High Impendence)
1	0	0	Output High	No	Output with buffer (inverted -content of buffer register)
1	0	1	Output Low	No	Output with buffer (inverted -content of buffer register)

Direction Register	Attribution Register	Data Register	Function	Wakeup	Description
1	1	0	Output Low	No	Output with buffer
1	1	1	Output High	No	Output with buffer

* Default is input mode with pull-low state.

** Only PortA and PortB in the state of “000”, “001” and “010” have wake up capability.

ODP output configuration can be done with changing between float state (“011” in binary) and output high state (“111” in binary) by only modifying the direction bit from “0” to “1”. On the other hand, **ODN output configuration** can be done with change between float state (“010” in binary) and output low state (“110” in binary) by only modifying the direction bit from “0” to “1”.

6.4 Control Register

I/O Port Control Register Summary Table

Name	Address	Description
P_IOA_Data	0x7060	I/O PortA Data Register
P_IOA_Buffer	0x7061	I/O PortA Buffer Register
P_IOA_Dir	0x7062	I/O PortA Direction Register
P_IOA_Attrib	0x7063	I/O PortA Attribute Register
P_IOA_Latch	0x7064	I/O PortA Latch Register for Wakeup
P_IOB_Data	0x7068	I/O PortB Data Register
P_IOB_Buffer	0x7069	I/O PortB Buffer Register
P_IOB_Dir	0x706A	I/O PortB Direction Register
P_IOB_Attrib	0x706B	I/O PortB Attribute Register
P_IOB_Latch	0x706C	I/O PortB Latch Register for Wakeup
P_IOC_Data	0x7070	I/O PortC Data Register
P_IOC_Buffer	0x7071	I/O PortC Buffer Register
P_IOC_Dir	0x7072	I/O PortC Direction Register
P_IOC_Attrib	0x7073	I/O PortC Attribute Register
P_IOD_Data	0x7078	I/O PortD Data Register
P_IOD_Buffer	0x7079	I/O PortD Buffer Register
P_IOD_Dir	0x707A	I/O PortD Direction Register
P_IOD_Attrib	0x707B	I/O PortD Attribute Register

P_IOA_Data **0x7060** **IOA Data Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOADATA															
Default	0	0	0	0	-	-	-	-	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOADATA	R/W	Writing this register will latch setup value into I/O PortA data register. Similarly, reading this register will read the status from I/O PortA external pad.	Refer to the above table, ***I/O port configuration and function.***

P_IOA_Buffer **0x7061** **IOA Buffer Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOABUF															
Default	0	0	0	0	-	-	-	-	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOABUF	R/W	Reading this register will read the setup value from I/O PortA data register, which is previously latched by IOADATA write operation.	IOABUF(R)=IOADATA(W) =IOABUF(W)

P_IOA_Dir **0x7062** **IOA Direction Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOADIR															
Default	0	0	0	0	-	-	-	-	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOADIR	R/W	This control register sets the direction of I/O PortA. The direction setup value can be read back from the same control register.	Refer to the above table, ***I/O port configuration and function.***

P_IOA_Attrib **0x7063** **IOA Attribute Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOAATT															
Default	0	0	0	0	-	-	-	-	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOAATT	R/W	This control register defines the attribute of I/O PortA. The attribute setup value can be read back from the same control register.	Refer to the above table, ***I/O port configuration and function.***

P_IOA_Latch **0x7064** **IOA Latch for Key Change Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOALH															
Default	0	0	0	0	-	-	-	-	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOAWLH	R	This control register latches the I/O PortA status for key-changed wakeup purpose. Wakeup is triggered if any I/O state of PortA is different from that latched. This latch operation must be done before entering sleep mode (for more information, refer to ***Chapter: System Control***).	

P_IOB_Data **0x7068** **IOB Data Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-				IOBDATA			-								
Default	-	-	-	-	0	0	0	-	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	IOBDATA	R/W	Writing this register will latch setup value into I/O PortB data register. Similarly, reading this register will read the status from I/O PortB external pad.	Refer to the above table, ***I/O port configuration and function.***

P_IOB_Buffer **0x7069** **IOB Buffer Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-				IOBBUF			-								
Default	-	-	-	-	0	0	0	-	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	IOBBUF	R/W	Reading this register will read the setup value from I/O PortB data register, which is previously latched by IOBDATA write operation.	IOBBUF(R)=IOBDATA(W) =IOBBUF(W)

P_IOB_Dir **0x706A** **IOB Direction Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-				IOBDIR			-								
Default	-	-	-	-	0	0	0	-	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	IOBDIR	R/W	This control register sets the direction of I/O PortB. The direction setup value can be read back from the same control register.	Refer to the above table, ***I/O port configuration and function.***

P_IOB_Attrib **0x706B** **IOB Attribute Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-				IOBATT			-								
Default	-	-	-	-	0	0	0	-	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	IOBATT	R/W	This control register defines the attribute of I/O PortB. The attribute setup value can be read back from the same control register.	Refer to the above table, ***I/O port configuration and function.***

P_IOB_Latch **0x706C** **IOB Latch for Key Change Wakeup**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-				IOBLHW			-								
Default	-	-	-	-	0	0	0	-	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	IOBLHW	R	This control register latches the I/O PortA status for key-changed wakeup purpose. Wakeup is triggered if any I/O state of PortA is different from that latched. This latch operation must be done before entering sleep mode (for more information, refer to ***Chapter: System Control***).	

P_IOC_Data **0x7070** **IOC Data Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOCDATA															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOCDATA	R/W	Writing this register will latch setup value into I/O PortC data register. Similarly, reading this register will read the status from I/O PortC external pad.	Refer to the above table, ***I/O port configuration and function.***

P_IOC_Buffer **0x7071** **IOC Buffer Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOCBUF															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOCBUF	R/W	Reading this register will read the setup value from I/O PortC data register, which is previously latched by IOCDATA write operation.	IOCBUF(R)=IOCDATA(W) =IOCBUF(W)

P_IOC_Dir **0x7072** **IOC Direction Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOCDIR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOCDIR	R/W	This control register sets the direction of I/O PortC. The direction setup value can be read back from the same control register.	Refer to the above table, ***I/O port configuration and function.***

P_IOC_Attrib **0x7073** **IOC Attribute Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOCATT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOCATT	R/W	This control register defines the attribute of I/O PortA. The attribute setup value can be read back from the same control register.	Refer to the above table, ***I/O port configuration and function.***

P_IOD_Data **0x7078** **IOD Data Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IODDATA															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IODDATA	R/W	Writing this register will latch setup value into I/O PortD data register. Similarly, reading this register will read the status from I/O PortD external pad.	Refer to the above table, ***I/O port configuration and function.***

P_IOD_Buffer **0x7079** **IOD Buffer Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOBBUF															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IODBUF	R/W	Reading this register will read the setup value from I/O PortD data register, which is previously latched by IODDATA write operation.	IODBUF(R)=IODDATA(W) =IODBUF(W)

P_IOD_Dir **0x707A** **IOD Direction Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IODDIR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

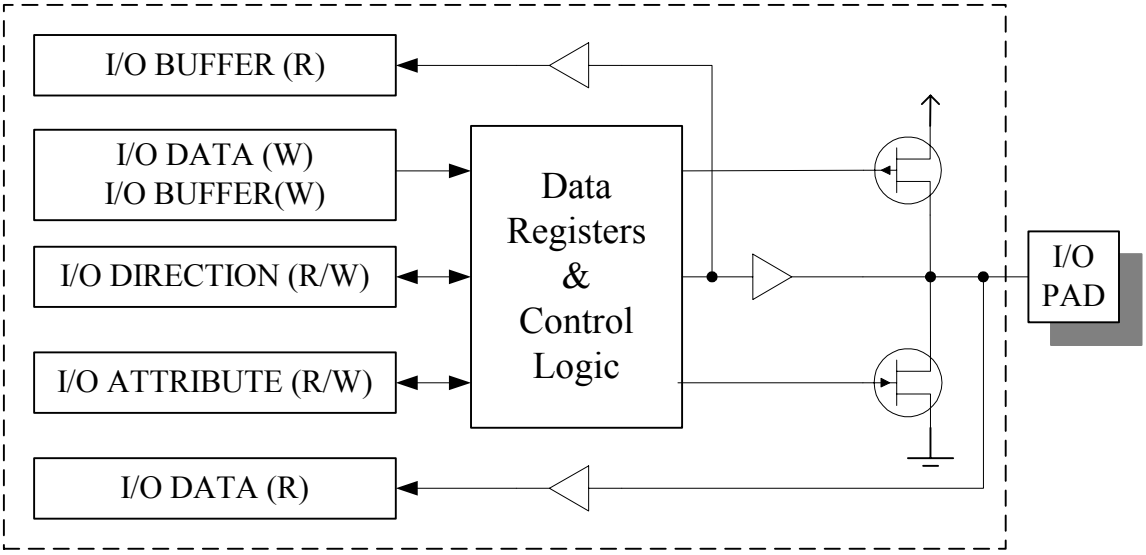
Bit	Function	Type	Description	Condition
[15:0]	IOADIR	R/W	This control register sets the direction of I/O PortD. The direction setup value can be read back from the same control register.	Refer to the above table, ***I/O port configuration and function.***

P_IOD_Attrib **0x707B** **IOD Attribute Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IODATT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IODATT	R/W	This control register defines the attribute of I/O PortD. The attribution setup value can be read back from the same control register.	Refer to the above table, ***I/O port configuration and function.***

6.5 I/O Structure Diagram



6.6 Special Notes

If a programmer intends to change the I/O configuration and finds that two control registers need to be setup, the various setup sequence may have different outcomes. For example, changing the I/O mode from input pull-low to output buffer requires both Direction and Attribution registers to be configured. Setting the Direction bit to "1" before Attribute bit to be "1" causes an unexpected short pulse on the I/O pad. In contrast, setting the Attribute bit to "1" before the Direction bit will not cause an additional pulse.

6.7 Program Examples

```
R1 = 0xF000                 // IOA[3..0] are input with pull-low resistor
[P_IOA_Data] = R1           // IOA[15..8] are all floating
R1 = 0xF000
[P_IOA_Attrib] = R1
R1 = 0x0000
[P_IOA_Dir] = R1


R1 = 0x5A5A
[P_MRE_Disable] = R1        // Disable MRE and IOC15 becomes GPIO
[P_MWE_Disable] = R1        // Disable MWE and IOC14 becomes GPIO

R1 = 0xF0F0                 // IOC[3..0] are output buffer high (PMOS driven)
[P_IOC_Data] = R1           // IOC[7..4] are output buffer low (NMOS driven)
R1 = 0xFF00                 // IOC[11..8] are output buffer low (NMOS driven)
[P_IOC_Attrib] = R1         // IOC[15..12] are output buffer high (PMOS driven)
R1 = 0xFFFF
[P_IOC_Dir] = R1
```

7 Interrupt

7.1 Introduction

The SPL161001A provides many interrupt sources, which can also be wakeup sources. In other word, after system enters into wait, halt, or sleep mode, interrupt event will wake CPU up. For interrupt wakeup capability from different modes, please refer to following table:

Interrupt Wakeup Capability Table for Wait, Halt, and Sleep Mode

Interrupt / Wakeup Source	Wait Mode	Halt Mode	Sleep Mode
PortA and PortB pads (key) change wakeup	O	O	O
External A wakeup	O	O	O
External B wakeup	O	O	O
UART/ IrDA receiving wakeup	O	X	X
Serial Peripheral Interface (SPI) receiving wakeup	O	X	X
LCD frame pulse (rising edge) wakeup	O	X	X
TimerA wakeup	O	X	X
TimerB wakeup	O	X	X
TimerC wakeup	O	X	X
TimebaseA wakeup	O	O	X
TimebaseB wakeup	O	O	X
TimebaseC wakeup	O	O	X
HMS (Hour/Minute/Second) wakeup	O	O	X
Alarm wakeup	O	O	X
Scheduler wakeup	O	O	X
Low voltage detect wakeup	O	O	X
Audio Channel A FIFO Empty wakeup	O	X	X
Audio Channel B FIFO Empty wakeup	O	X	X

O: Supported, X: Not Supported

All interrupts are level-triggered type. That is, an interrupt flag has to be cleared when interrupt service routine begins; otherwise, CPU will re-enter the Interrupt service routine again. Most interrupt flags depicted in this chapter are read-only (reference-only). To enable or clear these interrupts, programmers must write corresponding control bit individually, as depicted in the following chapter for each functional block. The exceptions are key change interrupts (PortA and PortB) and external interrupts (EXTA and EXTB).

In addition, There are also Interrupt flags which have the same function with these read-only interrupt flags, depicted in this chapter, in one of the control registers for corresponding functional block. For example, reading from P_LVD_Ctrl.bit15 is the same as reading from P_INT_Status2.bit7.

Note that once any I/O pin of PortA (except IOA12 and IOA13) or PortB has been configured to their special function, the key-change interrupt (wakeup) ability is disabled. On the other hand, key-change interrupt or wakeup ability of two special pins IOA12 (EXTA) and IOA13 (EXTB) will still function, even though their special function is enabled by choosing them as timer clock source or enabling external (rising or falling) interrupt. Please refer to following table.

	Key-change Interrupt	IO Special Function	Key-Change Interrupt or Wakeup Ability
IOA12 and IOA13	Disable	Disable	NO
	Disable	Enable	NO
	Enable	Disable	YES
	Enable	Enable	**YES**
PortA and PortB (Except IOA12 and IOA13)	Disable	Disable	NO
	Disable	Enable	NO
	Enable	Disable	YES
	Enable	Enable	**NO**

7.2 Peripheral Interrupt Arrangement

As depicted in ***Section: Memory, Vectors***, there are 11 interrupt events in the SUNPLUS 16-bit CPU (*μ'nSP*™): Software Break, Fast Interrupt (FIQ), Reset and IRQ[7..0]. SPL161001A peripheral interrupts are distributed in nine of above interrupt sources, FIQ and IRQ[7..0]. The following table depicts the peripheral interrupt arrangement. Note that some peripheral interrupts can be configured as FIQ or as one of the IRQs. Refer to P_INT_Priority1 and P_INT_Priority2 control register for details.

Interrupt Type	Possible Peripheral Interrupt	Flag Register
FIQ	Key Change (PortA and PortB only) Interrupt UART/ IrDA Interrupt* Serial Peripheral Interface (SPI) Interrupt LCD Frame Pulse Rising Edge (FP) Interrupt Audio Channel A FIFO Empty Interrupt Audio Channel B FIFO Empty Interrupt External A Interrupt (rising or falling edge of IOA12) External B Interrupt (rising or falling edge of IOA13) TimerA Up-Counter Overflow or Capture or Comparison event Interrupt** TimerB Up-Counter Overflow or Capture or Comparison event Interrupt** TimerC Up-Counter Overflow Interrupt Low Voltage Detection Interrupt	P_INT_Status1 P_INT_Status2
IRQ0	Audio Channel A FIFO Empty Interrupt Audio Channel B FIFO Empty Interrupt	P_INT_Status1
IRQ1	NAND Flash DMA Read/Write Interrupt	P_INT_Status1
IRQ2	External A Interrupt (rising or falling edge of IOA12) External B Interrupt (rising or falling edge of IOA13)	P_INT_Status1
IRQ3	UART/ IrDA Interrupt* Serial Peripheral Interface (SPI) Interrupt	P_INT_Status1
IRQ4	TimerA Up-Counter Overflow or Capture or Comparison event Interrupt** TimerB Up-Counter Overflow or Capture or Comparison event Interrupt** TimerC Up-Counter Overflow Interrupt	P_INT_Status2
IRQ5	Key Change (PortA and PortB only) Interrupt LCD Frame Pulse Rising Edge (FP) Interrupt	P_INT_Status1
IRQ6	Low Voltage Detection Interrupt TimeBaseC Interrupt Scheduler Interrupt	P_INT_Status2
IRQ7	TimeBaseA Interrupt TimeBaseB Interrupt Alarm Interrupt Hour / Minute / Second / Half-Second Interrupt***	P_INT_Status2

*There are three interrupt events in the UART/IrDA module: reception, transmission and reception timeout interrupt. Any one of these three interrupt sources can trigger FIQ or IRQ3. Therefore, in the FIQ or IRQ3 interrupt service routine, programmers should read P_UARTIrDA_Ctrl.bit[15..13] to distinguish which interrupt event has happened. These interrupts may occur at the same time. Please refer to ***chapter: UART/IrDA Interface.***

There are three interrupt events in the TimerA and TimerB module: up-counter overflow, capture event and comparison event. One of the above interrupt sources can trigger FIQ or IRQ4. Therefore, in the FIQ or IRQ4 interrupt service routine, programmers should read P_TimerA_CCP_Ctrl.bit[15:14] or P_TimerB_CCP_Ctrl.bit[15:14] to distinguish which interrupt event takes place. Unlike UART/IrDA interrupts, only one of the above three events could occur for each conifguration. Please refer to ***chapter: Timer/Counter*

There are four interrupt events in Hour / Minute / Second / Half-Second Interrupt: Hour, Minute, Second, and Half-Second Interrupt. Any one of these four interrupt sources can trigger IRQ7. Therefore, in the IRQ7 interrupt service routine, programmers should read P_RTC_INT_Status[3..0] to distinguish which interrupt event is generated. These interrupts may occur at the same time. For example, reading from P_RTC_INT_Status [3..0] will be “1111” when clock register is from 15:59:59 to 16:0:0. Please refer to ***chapter: Real Time Clock.

7.3 Control Registers

Global Interrupt Control Register Summary Table

Name	Address	Description
P_INT_Status1	0x70A0	Interrupt Status Register 1
P_INT_Status2	0x70A1	Interrupt Status Register 2
P_INT_Priority1	0x70A4	Interrupt Priority Register 1
P_INT_Priority2	0x70A5	Interrupt Priority Register 2
P_MINT_Ctrl	0x70A8	Miscellaneous Interrupt Control Register

P_INT_Status1 0x70A0 Interrupt Status 1 Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	KEYIF/C	-	-	-	UTIRIF	SPIIF	FPIF	-	-	-	AUDBIF	AUDAIF	NFWIF	NFRIF	EXTBIF/C	EXTAIF/C
Default	0	-	-	-	0	0	0	-	-	-	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	KEYIF/C	R/W	Key-change Interrupt status Write “1” to clear the flag. This bit is set to “1” by hardware if the key change interrupt is asserted. (PortA and PortB only)	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
[14:12]			Reserved	
11	UTIRIF	R	UART/IrDA Interrupt status This bit is set to “1” by hardware if the UART/IrDA interrupt is asserted. For detail, refer to ***chapter: UART/IrDA interface.***	0= Not Occurred 1= Occurred
10	SPIIF	R	Serial Peripheral Interface (SPI) Interrupt status This bit is set to “1” by hardware if the SPI interrupt is asserted. For detail, refer to ***chapter: Serial Peripheral interface.***	0= Not Occurred 1= Occurred
9	FPIF	R	FP Interrupt status This bit is set to “1” by hardware if the LCD-FP-signal rising edge interrupt is asserted. For detail, refer to ***chapter: LCD interface.***	0= Not Occurred 1= Occurred
[8:6]			Reserved	
5	AUDBIF	R	Audio Channel B FIFO Empty Interrupt status This bit is set to “1” by hardware if the audio channel B FIFO empty interrupt is asserted. For detail, refer to ***chapter: Audio Output interface.***	0= Not Occurred 1= Occurred
4	AUDAIF	R	Audio Channel A FIFO Empty Interrupt status This bit is set to “1” by hardware if the audio channel A FIFO empty interrupt is asserted. For detail, refer to ***chapter: Audio Output interface.***	0= Not Occurred 1= Occurred
3	NFWIF	R	NAND Flash DMA channel write complete interrupt status This bit is set to “1” by hardware if the writing to NAND Flash with DMA channel is completed. For	0= Not Occurred 1= Occurred

Bit	Function	Type	Description	Condition
			detail, refer to ***chapter: NAND Flash Interface.***	
2	NFRIF	R	NAND Flash DMA channel read complete interrupt status This bit is set to "1" by hardware if the reading to NAND Flash with DMA channel is completed. For detail, refer to ***chapter: NAND Flash Interface.***	0= Not Occurred 1= Occurred
1	EXTBIF/C	R/W	External Interrupt B status Write '1' to clear the flag This bit is set to "1" by hardware if the external interrupt B is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag
0	EXTAIF/C	R/W	External Interrupt A status Write '1' to clear the flag This bit is set to "1" by hardware if the external interrupt A is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag

P_INT_Status2 **0x70A1** **Interrupt Status 2 Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	TMCIF	TMBIF	TMAIF	-	TMBCIF	TMBBIF	TMBAIF	LVDIF	-	-	-	-	SCHIF	ALMIF	HMSIF
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15			Reserved	
14	TMCIF	R	TimerC Interrupt Flag This bit is set to "1" by hardware if the TimerC up-counter overflow event interrupt is asserted. For detail, refer to ***chapter: Timer/Counter interface.***	0= Not Occurred 1= Occurred
13	TMBIF	R	TimerB Interrupt Flag This bit is set to "1" by hardware if the TimerB up-counter overflow, capture, or comparison event interrupt is asserted. For detail, refer to ***chapter: Timer/Counter interface.***	0= Not Occurred 1= Occurred
12	TMAIF	R	TimerA Interrupt Flag This bit is set to "1" by hardware if the TimerA up-counter overflow, capture, or comparison event interrupt is asserted. For detail, refer to ***chapter: Timer/Counter interface.***	0= Not Occurred 1= Occurred
11			Reserved	

Bit	Function	Type	Description	Condition
10	TMBCIF	R	TimeBaseC Interrupt Flag This bit is set to “1” by hardware if the TimebaseC interrupt is asserted. For detail, refer to ***chapter: TimeBase interface.***	0= Not Occurred 1= Occurred
9	TMBBIF	R	TimeBaseB Interrupt Flag This bit is set to “1” by hardware if the TimebaseB interrupt is asserted. For detail, refer to ***chapter: TimeBase interface.***	0= Not Occurred 1= Occurred
8	TMBAIF	R	TimeBaseA Interrupt Flag This bit is set to “1” by hardware if the TimebaseA interrupt is asserted. For detail, refer to ***chapter: TimeBase interface.***	0= Not Occurred 1= Occurred
7	LVDIF	R	Low Voltage Detection Interrupt Flag This bit is set to “1” by hardware if the low voltage detection interrupt is asserted. For detail, refer to ***chapter: System Control***	0= Not Occurred 1= Occurred
[6:3]			Reserved	
2	SCHIF	R	Schedule Interrupt Flag This bit is set to “1” by hardware if the scheduler interrupt is asserted. For detail, refer to ***chapter: Real Time Clock***	0= Not Occurred 1= Occurred
1	ALMIF	R	Alarm Interrupt Flag This bit is set to “1” by hardware if the alarm interrupt is asserted. For detail, refer to ***chapter: Real Time Clock***	0= Not Occurred 1= Occurred
0	HMSIF	R	HMS Interrupt Flag This bit is set to “1” by hardware if the hour or minute or second or half-second interrupt is asserted. For detail, refer to ***chapter: Real Time Clock***	0= Not Occurred 1= Occurred

P_INT_Priority1 **0x70A4** **Interrupt Priority 1 Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	KEYIP	-	-	-	UTIRIP	SPIIP	FPIP	-	-	-	AUDBIP	AUDAIP	NFIP	-	EXTBIP	EXTAIP
Default	0	-	-	-	0	0	0	-	-	-	0	0	0	-	0	0

Bit	Function	Type	Description	Condition
15	KEYIP	R/W	Key-change interrupt priority (PortA and PortB only)	0= IRQ5 1= FIQ
[14:12]			Reserved	
11	UTIRIP	R/W	UART/ IrDA interrupt priority	0= IRQ3 1= FIQ
10	SPIIP	R/W	Serial Peripheral Interface (SPI) interrupt priority	0= IRQ3 1= FIQ
9	FPIP	R/W	LCD Frame Pulse Signal Rising Edge (FP) interrupt priority	0= IRQ5 1= FIQ
[8:6]			Reserved	
5	AUDBIP	R/W	Audio Channel B FIFO Empty Interrupt priority	0= IRQ0 1= FIQ
4	AUDAIP	R/W	Audio Channel A FIFO Empty Interrupt priority	0= IRQ0 1= FIQ
3	NFIP	R/W	NAND Flash interrupt priority	0= IRQ1 1= FIQ
2			Reserved	
1	EXTBIP	R/W	External Interrupt B priority	0= IRQ2 1= FIQ
0	EXTAIP	R/W	External Interrupt A priority	0= IRQ2 1= FIQ

P_INT_Priority2 **0x70A5** **Interrupt Priority 2 Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	TMCIP	TMBIP	TMAIP	-	TMBCIP	TMBBIP	TMBAIP	LVDIP	-	-	-	-	SCHIP	ALMIP	RTCIP
Default	-	0	0	0	-	0	0	0	0	-	-	-	-	-	0	0

Bit	Function	Type	Description	Condition
15			Reserved	
14	TMCIP	R/W	TimerC up-counter overflow interrupt priority	0= IRQ4 1= FIQ
13	TMBIP	R/W	TimerB up-counter or capture or comparison	0= IRQ4

Bit	Function	Type	Description	Condition
			event overflow interrupt priority	1= FIQ
12	TMAIP	R/W	TimerA up-counter or capture or comparison event overflow interrupt priority	0= IRQ4 1= FIQ
11			Reserved	
10	TMBCIP	R/W	TimeBaseC interrupt priority is set as IRQ6	0= IRQ6 1= FIQ
9	TMBBIP	R/W	TimeBaseB interrupt priority is set as IRQ7	0= IRQ7 1= FIQ
8	TMBAIP	R/W	TimeBaseA interrupt priority is set as IRQ7	0= IRQ7 1= FIQ
7	LVDIP	R/W	Low Voltage Detection Interrupt priority	0= IRQ6 1= FIQ
[6:3]			Reserved	
2	SCHIP	R/W	Schedule Interrupt priority	0= IRQ6 1= FIQ
1	ALMIP	R/W	Alarm interrupt priority	0= IRQ7 1= FIQ
0	RTCIP	R/W	RTC interrupt priority	0= IRQ7 1= FIQ

P_MINT_Ctrl **0x70A8** **Miscellaneous Interrupt Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	KEYIEN	-	-	-	-	-	-	-	-	-	EXTBIS	EXTAIS	-	-	EXTBIEN	EXTAIEN
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	KEYIEN	R/W	Key-change Interrupt Enable If this bit is set to “1”, key change Interrupt is enabled. This feature is only valid on PORTA and PORTB.	0= Disabled 1= Enabled
[14:6]			Reserved	
5	EXTBIS	R/W	EXTB Interrupt Edge Selection This bit is valid only when EXTBIEN control bit is set to “1”. If this bit is set to “1”, the EXTB is rising edge triggered. Otherwise, it is falling edge triggered.	0= Falling edge triggered 1= Rising edge triggered
4	EXTAIS	R/W	EXTA Interrupt Edge Selection	0= Falling edge triggered

Bit	Function	Type	Description	Condition
			This bit is valid only when EXTAIEN control bit is set to “1”. If this bit is set to “1”, the EXTA is rising edge triggered. Otherwise, it is falling edge triggered.	1= Rising edge triggered
[3:2]			Reserved	
1	EXTBIEN	R/W	EXTB interrupt enable If this bit is set as “1”, PortA13 will be configured as external interrupt B input pin. If this bit is cleared to “0”, PortA13 remains GPIO.	0= Disabled 1= Enabled
0	EXTAIEN	R/W	EXTA interrupt enable If this bit is set as “1”, PortA12 will be configured as external interrupt A input pin. If this bit is cleared to “0”, PortA12 remains GPIO.	0= Disabled 1= Enabled

7.4 Program Examples

```
.PUBLIC                 _BREAK,_FIQ,_IRQ6;

//****************************
_BREAK:                                         // Software Break ISR
                        reti
//****************************
_FIQ:
                        push r1 to [sp]         // Save destroyed CPU register r1 to stack
                        r1 = [P_INT_Status1]
                        r1 = r1 & C_INT_AUDAFIFOEmpty
                        jz     L_End_AudioCHA_ISR?
                             [P_CHA_Ctrl] = r1
                             r1 = [P_CHA_Ctrl]  // Clear & Disable Interrupt Flag
                                ............
                                ............
L_End_AudioCHA_ISR?:
                        pop r1 from [sp]        // Restore original CPU register r1 from stack
                        reti
//****************************
_IRQ6:
                        push r1, r3 to [sp]     // Save destroyed CPU register r1, r2, r3 to stack
                        r2 = [P_INT_Status2]
                        r1 = r2 & C_INT_LVD
                        jz     L_End_LVD_ISR?
```

```
                    R1 = 0x8000
                    [P_LVD_Ctrl] = r1          // Clear & Disable Interrupt Flag
                    ............
                    ............
L_End_LVD_ISR?:
                r1 = r2 & C_INT_TimeBaseC
                jz      L_End_TimeBaseC_ISR?
                    r1 = [P_TimeBaseC_Ctrl]
                    [P_TimeBaseC_Ctrl] = r1    // Clear & Disable Interrupt Flag
                    ............
                    ............
L_End_TimeBaseC_ISR?:
                pop r1, r3 from [sp]           // Restore original CPU register r1, r2, r3 from stack
                reti
```

8 Timer/Counter

8.1 Timer Introduction

The SPL161001A contains three 16-bit timer/counters: TimerA, TimerB, and TimerC. TimerA and TimerB support Capture/Comparison/PWM (CCP) functions when cooperating with their own two 16-bit registers (preload registers and CCP registers). On the other hand, TimerC has only one 16-bit preload resister and therefore, TimerC does not provide CCP function. The clock sources of these three timers can be programmed by internal clock sources (timer mode) or from external I/O pin (counter mode).

Main features

- Variant clock source selection for each timer source
- Support two external clock sources
- One external clock source provides 1/4, 1/16 pre-scalar
- Each programmable clock source can be synchronous with CPU clock as for each timer source
- Two timers support Capture function
- Two timers support Comparison function
- Two timers support Pulse Width Modulation (PWM) function

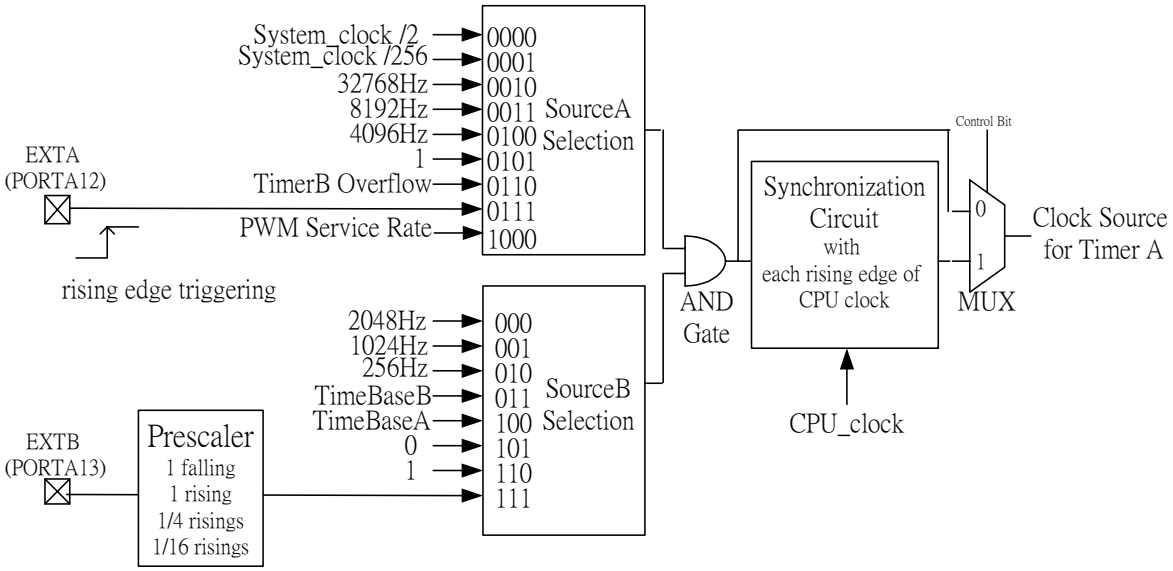
8.2 Timer Structure and Clock Source

There are two basic operation modes in the SPL161001A timer/counter module - timer mode and counter mode.

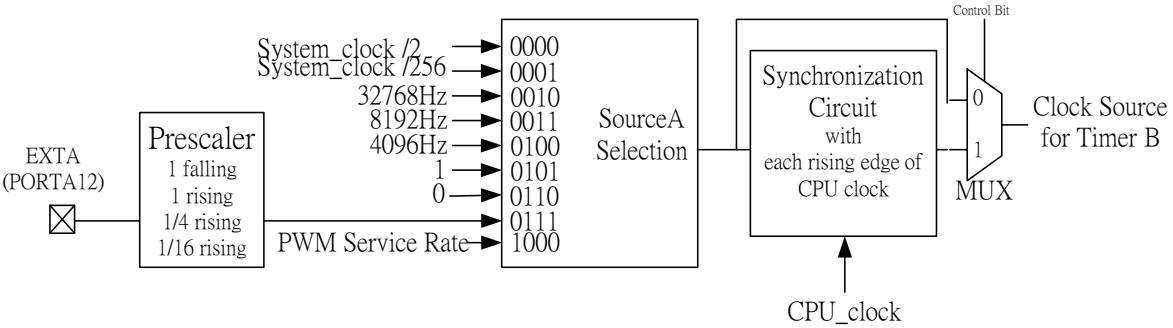
In the timer mode, the clock source is internal clock sources such as CPUCLK/256, CPUCLK/2, 32768Hz, 8192Hz, 4096Hz, 1024Hz, 4Hz, 2Hz, 1Hz, or combinations (for detail, refer to clock source block diagram). in addition, clock source can be synchronized by CPU clock in the way of setting the corresponding control bits.

In the counter mode, the clock source is supported by external clock source pin (PortA12 or PortA13). Additionally, signal from external clock source can be divided by 4 or 16 pre-scalars and will be capable of selecting trigger edge (rising or falling). To ensure the CPU clock synchronization procedure is completed, this external clock frequency has to be less than or equal to a half of CPUCLK.

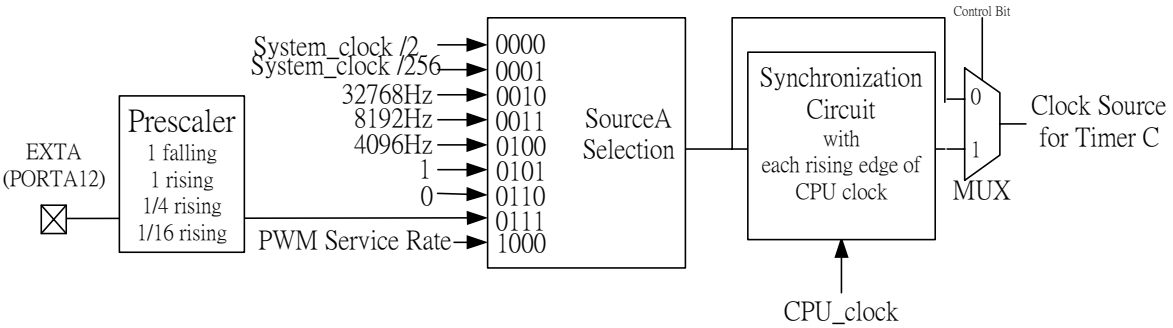
Following are functional diagrams of TimerA, TimerB, and TimerC source clock selection.



TimerA Source Clock Selection



TimerB Source Clock Selection



TimerC Source Clock Selection

Each of these 16-bit timer/counter has an up-counter and increment on the rising edge of internal clock source, or increment either on the rising or falling edge of external clock source. Initial value of the up-counter is stored in the preload register. When the timer/counter is enabled and overflow occurs, the initial value is loaded into counter on the next increment clock edge (synchronous load). At the same time, corresponding interrupt flag is set. If corresponding interrupt is enabled, it will issue interrupt to CPU. For example, if the initial value is 0xFFFC, the timer/counter will count by the sequence 0xFFFC, 0xFFFD, 0xFFFE, 0xFFFF, 0xFFFC, 0xFFFD, and so on.

Formula:

$$\text{Timer/Counter Overflow Interrupt Frequency} = \frac{\text{Source clock frequency}}{65536 - \text{Preload Register Value}}$$

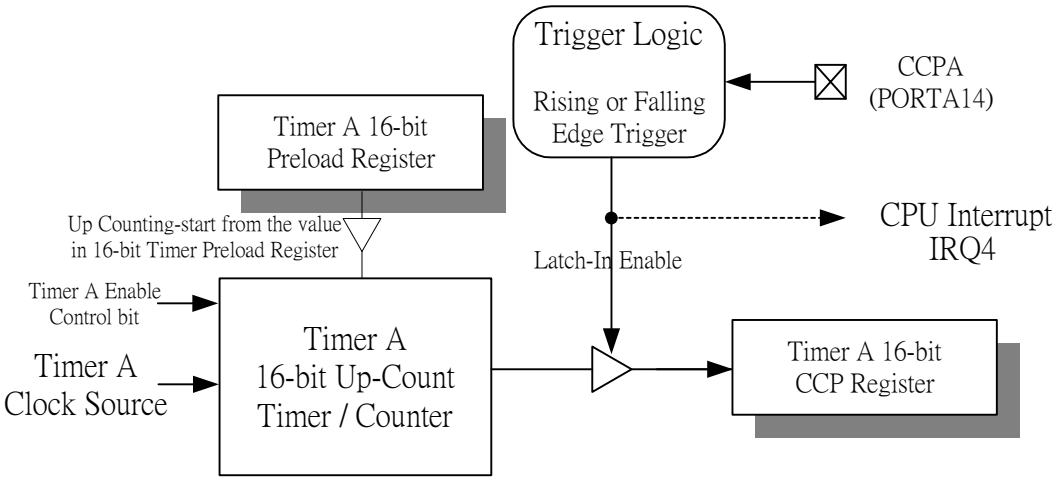
Note that if external Input A (PortA12) is selected as counter clock source, this pin cannot be GPIO function. In other words, any GPIO setting on PortA12 will have no effect. Similarly, if external Input B (PortA13) is selected as counter clock source, this pin cannot be GPIO function. That is, any GPIO setting on PortA13 will have no effect.

In addition, TimerA and TimerB offer Capture / Comparison / Pulse-Width-Modulation (CCP) special functions. If one of these two special functions is enabled, Control Register 2 and CCP Register of TimerA and TimerB should be setup appropriately. Note that TimerC does not support the CCP function and therefore, it does not have control over register 2 and CCP register. The following three sections will depict the detailed operation of the CCP function.

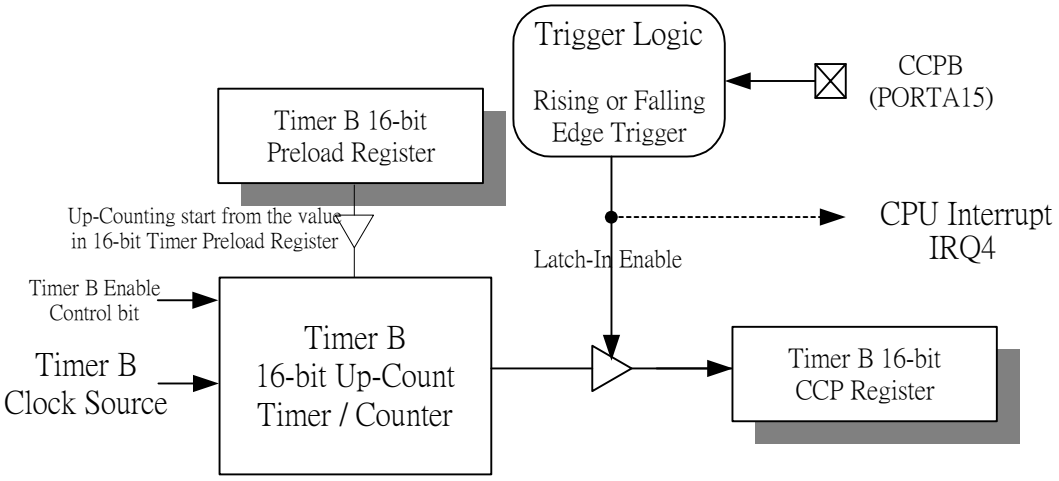
Capture Mode

In the capture mode, the value in Timer/Counter register is latched in Capture/Comparison/PWM registers (CCP Register) at the selected edge (rising or falling) of external I/O pin (PortA14 or PortA15). The value in Timer/Counter Register can be latched every rising or falling edge of external I/O pin (refer to control register 2 for details). When a capture is made, the interrupt flag is set and the CPU is interrupted via IRQ4. The interrupt flag must be cleared by software.

Note that if another capture occurs before the value in the CCP Register is read, the old captured value will be lost. That is, the clock frequency from external I/O pin has to be at most one half of the timer/counter clock source frequency. The corresponding GPIO pin, PortA14 or PortA15, for the timer/counter is configured as INPUT pin automatically when TimerA or TimerB is set in the capture mode.



TimerA Capture Mode Function Diagram

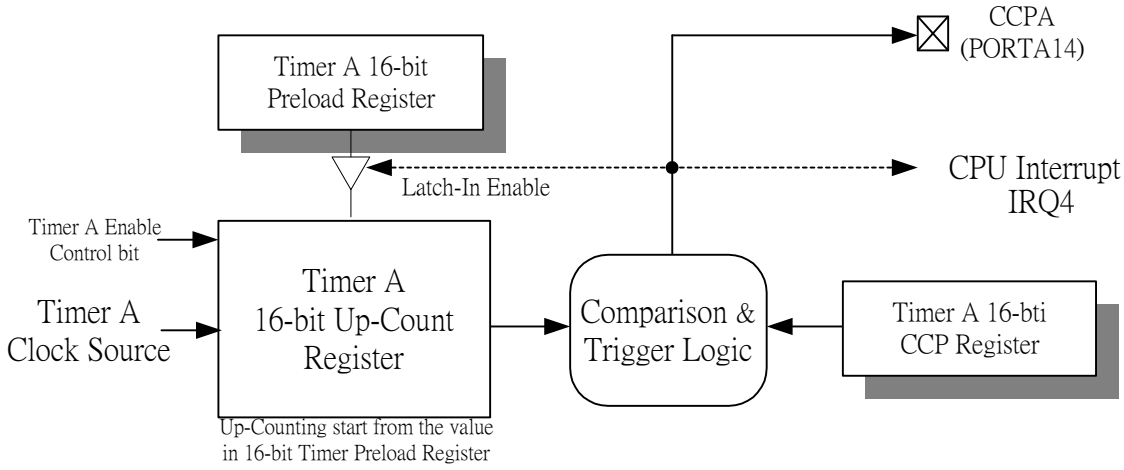


TimerB Capture Mode Function Diagram

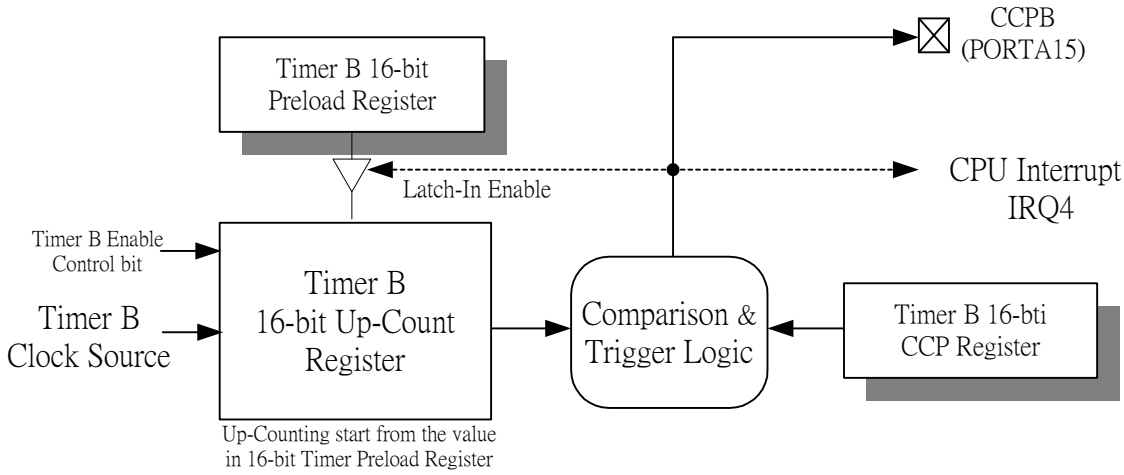
When TimerA or TimerB is configured in the capture mode (P_TimerX_CCP_Ctrl.bit[15,14]="01"), programmers can enable, disable, clear and read status of capture event interrupt by accessing P_TimerX_Ctrl.bit[15.14]. At this time, timerA or timerB up-counter overflow interrupt will have no effect on any operation of P_TimerX_Ctrl.bit[15.14]. In other words, only one interrupt event is valid on the control bits (P_TimerX_Ctrl.bit[15.14]) at one time, either up-counter overflow interrupt or capture event interrupt.

Comparison Mode

In the comparison mode, the value to be compared with the 16-bit timer/counter registers is stored in the CCP Register. The preloaded value is reloaded to 16-bit timer/counter and the interrupt flag is set if they are matched. The corresponding PortA14 or PortA15 I/O pin is configured as an OUTPUT pin automatically when comparison mode is set. Note that in this mode, the overflow of the 16-bit timer/counter will not reload the preloaded value..



TimerA Comparison Mode Function Diagram

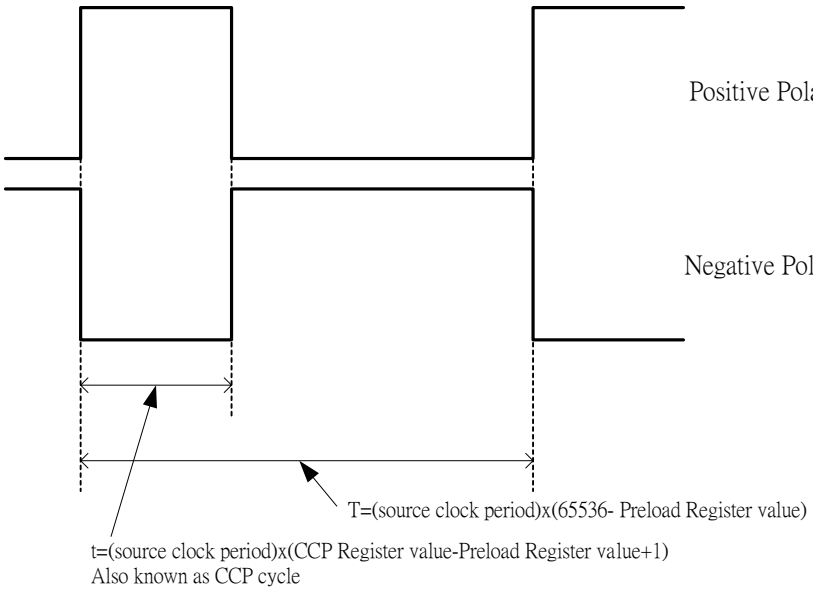


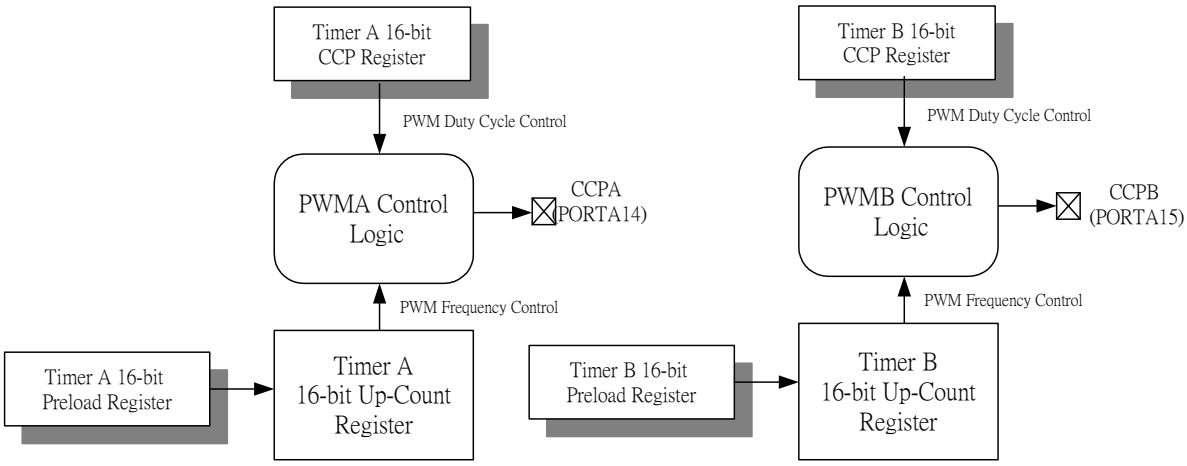
TimerB Comparison Mode Function Diagram

When TimerA or TimerB is configured as comparison mode (P_TimerX_CCP_Ctrl.bit[15,14]="10"), programmers can enable, disable, clear and read status of comparison event interrupt by accessing P_TimerX_Ctrl.bit[15.14]. At this time, timerA or timerB up-counter overflow interrupt will have no effect on any operation of P_TimerX_Ctrl.bit[15.14]. In other words, only one interrupt event is valid on the control bits (P_TimerX_Ctrl.bit[15.14]) at one time, either up-counter overflow interrupt or comparison event interrupt.

Pulse-Width-Modulation Mode

When timer/counter is set in the Pulse-Width-Modulation (PWM) mode, the PortA14 or PortA15 pin is configured as OUTPUT pins automatically. When the timer/counter is enabled, the PWM output will toggle at the overflow of timer/counter and at the end of CCP cycle. There are two output signal polarities on PWM output pin, positive and negative.





In the Pulse-Width-Modulation (PWM) normal mode the PWM frequency is set by the preloaded register value. See the formula below:

$$\textbf{PWM Output Frequency} = \frac{\textbf{Source clock frequency}}{\textbf{65536 – Preload Register Value}}$$

$$\textbf{PWM Period} = \frac{1}{\textbf{PWM Output Frequency}}$$

The PWM duty cycle is set by the CCP register value. See the formula below:

$$\textbf{PWM Output Duty Cycle} = \frac{\textbf{CCP Register Value – Preload Register Value +1}}{\textbf{65536 – Preload Register Value}}$$

8.3 Control Registers

Timer Control Register Summary Table

Name	Address	Description
P_TimerA_Ctrl	0x70C0	TimerA Control Register
P_TimerA_CCP_Ctrl	0x70C1	TimerA Capture / Comparison / PWM Control Register
P_TimerA_Preload	0x70C2	TimerA Preload Register
P_TimerA_CCP_Reg	0x70C3	TimerA Capture / Comparison / PWM Register
P_TimerA_UpCount	0x70C4	TimerA up-count value
P_TimerB_Ctrl	0x70C8	TimerB Control Register
P_TimerB_CCP_Ctrl	0x70C9	TimerB Capture / Comparison / PWM Control Register
P_TimerB_Preload	0x70CA	TimerB Preload Register
P_TimerB_CCP_Reg	0x70CB	TimerB Capture / Comparison / PWM Register
P_TimerB_UpCount	0x70CC	TimerB up-count value
P_TimerC_Ctrl	0x70D0	TimerC Control Register
P_TimerC_Preload	0x70D2	TimerC Preload Register
P_TimerC_UpCount	0x70D4	TimerC up-count value

P_TimerA_Ctrl **0x70C0** **TimerA Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMAIF/C	TMAIE	TMAEN	SYNC	-	-	EXTBSEL		-	SRCBSEL			SRCASEL			
Default	0	0	0	0	-	-	0	0	-	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	TMAIF/C	R/W	TimerA Interrupt Flag Write “1” to clear the flag In timer /counter mode, this bit is set when TimerA rolls over FFFF and is cleared by writing this bit to “1”. In capture mode, This bit is set when the external signals is driven, and timer/counter value is latched into TimerA CCP register. In comparison mode, this bit is set when the timer/counter value is the same as the value stored in TimerA CCP register In PWM mode, reading and writing this bit have no effect.	Read 0= Not Occurred Read 1= Occurred Write 0 = No effect Write 1= Clear the flag
14	TMAIE	R/W	TimerA Interrupt Enable If this bit is set to “1”, and TimeA (Timer overflow or capture or compare event) interrupt happened, hardware will issue an IRQ4 or FIQ to CPU. If this bit is clear to “0”, this interrupt will be masked. To select between IRQ4 and FIQ, please refer to ***Chapter Interrupt***	0= Disabled 1= Enabled
13	TMAEN	R/W	TimerA Enable If this bit is set to “1”, TimerA will start to up count its 16-bit timer/counter register according the frequency of selected clock source. If this bit is clear to “0”, the TimerA will stop counting.	0= Disabled 1= Enabled
12	SYNC	R/W	Clock Source Synchronization Enable If this bit is set to “1”, the clock source of TimerA is synchronized by CPU clock.	0= Disabled 1= Enabled
[11:10]			Reserved	
[9:8]	EXTBSEL	R/W	External Input B (IOA13) pre-scalar setup These two bits are used to select the pre-scalar function between external input signal and timer clock source.	00= EXTB every falling 01= EXTB every rising 10= EXTB every 4 risings 11= EXTB every 16 risings
7			Reserved	
[6:4]	SRCBSEL	R/W	Clock Source Group B selection	000= 2048Hz

Bit	Function	Type	Description	Condition
			[1]When TimebaseB or TimebaseA is selected, the corresponding timebase enable bit should be set, otherwise the timer will not count. [2]When external Input B ("111" in binary) is selected, PortA13 is configured as counter clock source. It cannot be GPIO function; that is, any GPIO setting on PortA13 will have no effect.	001= 1024Hz 010= 256Hz 011= TimeBaseB[1] 100= TimeBaseA[1] 101= 0 (logic low) 110= 1 (logic high) 111= EXTB with Pre-scalar[2]
[3:0]	SRCASEL	R/W	Clock Source Group A selection [3]TimerB and TimerA form a 32-bit counter if configured as a cascade mode when TimerB overflow ("0110" in binary) is selected. [4]When external Input A ("0111" in binary) is selected, IOA12 is configured as counter clock source. It cannot be GPIO function; that is, any GPIO setting on IOA12 will have no effect. [5]When PWM service rate is selected, PWM mode should be selected and either audio ChannelA or ChannelB should be enabled, otherwise the timer will not count.	0000= SYSCLK/2 0001= SYSCLK/256 0010= 32768Hz 0011= 8192Hz 0100= 4096Hz 0101= 1 0110= TimerB Overflow[3] 0111= EXTA[4] 1000= PWM Service Rate[5] 1001~1111 = Reserved

P_TimerA_CCP_Ctrl **0x70C1** **TimerA CCP Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CCPAEN						CAPASEL				CMPASEL				PWMASEL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:14]	CCPAEN	R/W	Operation Mode Selection (Default is timer/counter mode) When capture mode is selected, PortA14 becomes INPUT automatically and any GPIO setting will be no effect. When comparison and PWM mode is selected, PortA14 becomes OUTPUT automatically; any GPIO setting will be no effect.	00= CCP Mode Disabled 01= Capture Enabled 10= Comparison Enabled 11= PWM Enabled
[13:10]			Reserved	
[9:8]	CAPASEL	R/W	Capture Operation mode Selection These 2 bits are valid only when CCPAEN is set to "01" in binary.	00= every falling 01= every rising 10= reserved 11= reserved
[7:6]			Reserved	
[5:4]	CMPASEL	R/W	Comparison Operation Mode Selection	00= high pulse on IOA14

Bit	Function	Type	Description	Condition
			These 2 bits are valid only when CCPAEN is set to “10” in binary.	01= low pulse on IOA14 10= unaffected on IOA14 11= reserved
[3:2]			Reserved	
[1:0]	PWMASEL	R/W	PWM Operation Mode Selection These 2 bits are valid only when CCPAEN is set to “11” in binary.	00= PWM mode, positive output 01= PWM mode, negative output 10= reserved 11= reserved

P_TimerA_Preload 0x70C2 TimerA Preload Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMAPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMAPLR	R/W	TimerA preload value	

P_TimerA_CCP_Reg 0x70C3 TimerA Capture / Comparison / PWM Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMACCPR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMACCPR	R/W	TimerA Capture/Comparison/PWM register	

P_TimerA_UpCount 0x70C4 TimerA Up-Count

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMAUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMAUCR	R	TimerA up-counter value The up-counter’s value can be read from this register.	

P_TimerB_Ctrl 0x70C8 TimerB Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBIF/C	TMBIE	TMBEN	SYNC	-	-	EXTASEL		-	-	-	-	SRCASEL			
Default	0	0	0	0	-	-	0	0	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
15	TMBIF/C	R/W	TimerB Interrupt Flag Write “1” to clear the flag. In timer /counter mode, this bit is set when TimerB rolls over FFFF and is cleared by writing this bit to “1”. In capture mode, this bit is set when the external signals is driven, and timer/counter value is latched into TimerB CCP register. In comparison mode, this bit is set when the timer/counter value is the same as the value stored in TimerB CCP register. In PWM mode, reading and writing this bit has no effect.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
14	TMBIE	R/W	TimerB Interrupt Enable If this bit is set to “1” and TimeB (timer overflow or capture or compare event) interrupt occurred, hardware will issue an IRQ4 or FIQ to CPU. If this bit is cleared to “0”, this interrupt will be masked. To select IRQ4 and FIQ, please refer to ***Chapter Interrupt***	0= Disabled 1= Enabled
13	TMBEN	R/W	TimerB Module Enable If this bit is set to “1”, TimerB will start up-counting according the frequency of selected clock source. If this bit is cleared to “0”, the TimerB will stop counting.	0= Disabled 1= Enabled
12	SYNC	R/W	Clock Source Synchronization Enable If this bit is set to “1”, the clock source of TimerB is synchronized by CPU clock	0= Disabled 1= Enabled
[11:10]			Reserved	
[9:8]	EXTASEL	R/W	External Input A (IOA12) pre-scalar setup These two bits are used to select the pre-scalar function between external input signal and timer clock source.	00= EXTA every falling 01= EXTA every rising 10= EXTA every 4 risings 11= EXTA every 16 risings
[7:4]			Reserved	
[3:0]	SRCASEL		Clock Source Group A selection [1]When external Input A (“111” in binary) is selected, PortA12 is configured as counter clock	0000= SYSCLK/2 0001= SYSCLK/256 0010= 32768Hz

Bit	Function	Type	Description	Condition
			source. It cannot be GPIO function; that is, any GPIO setting on PortA12 will have no effect. [2]When PWM service rate is selected, PWM mode should be selected and either audio ChannelA or ChannelB should be enabled; otherwise, the timer will not count.	0011= 8192Hz 0100= 4096Hz 0101= 1 (logic high) 0110= 0 (logic low) 0111= EXTA with pre-scalar[1] 1000= PWM Service Rate[2] 1001~1111 = Reserved

P_TimerB_CCP_Ctrl **0x70C9** **TimerB Control CCP Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CCPBEN		-	-	-	-	CAPBSEL		-	-	CMPBSEL		-	-	PWMBSEL	
Default	0	0	-	-	-	-	0	0	-	-	0	0	-	-	0	0

Bit	Function	Type	Description	Condition
15	CCPBEN	R/W	Operation Mode Selection (Default is timer/counter mode) When capture mode is selected, PortA15 becomes INPUT automatically. Any GPIO setting will have no effect. When comparison and PWM mode are selected, PortA15 becomes OUTPUT automatically; any GPIO setting will have no effect.	00= CCP Mode Disabled 01= Capture Enabled 10= Comparison Enabled 11= PWM Enabled
[14:10]			Reserved	
[9:8]	CAPBSEL	R/W	Capture operation mode selection These 2 bits are valid only when CCPBEN is set to “01” in binary.	00= every falling 01= every rising 10= reserved 11= reserved
[7:6]			Reserved	
[5:4]	CMPBSEL	R/W	Comparison Operation Mode Selection These 2 bits are valid only when CCPBEN is set to “10” in binary.	00= high pulse on IOA15 01= low pulse on IOA15 10= unaffected on IOA15 11= reserved
[3:2]			Reserved	
[1:0]	PWMBSEL	R/W	PWM Operation Mode Selection These 2 bits are valid only when CCPBEN is set to “11” in binary.	00= PWM mode, NRO output 01= PWM mode, NRZ output 10= reserved 11= reserved

P_TimerB_Preload 0x70CA TimerB Preload Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMBPLR	R/W	TimerB preload value	

P_TimerB_CCP_Reg 0x70CB TimerB Capture / Comparison / PWM Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBCCPR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMBCCPR	R/W	TimerB Capture/Comparison/PWM register	

P_TimerB_UpCount 0x70CC TimerB Up-Count

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMBUCR	R	TimerB Up-Counter value The up-counter's value can be read from this register.	

P_TimerC_Ctrl 0x70D0 TimerC Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCIF/C	TMCIE	TMCEN	SYNC	-	-	EXTASEL		-	-	-	-	SRCASEL			
Default	0	0	0	0	-	-	0	0	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
15	TMCIF/C	R/W	TimerC Interrupt Flag This bit is set when TimerC rolls over FFFF and is cleared by writing this bit to "1".	Read 0= Not Occurred Read 1= Occurred Write 0 = No effect Write 1= Clear the flag
14	TMCIE	R/W	TimerC Interrupt Enable If this bit is set to "1", and TimerC interrupt occurs, hardware will issue an IRQ4 or FIQ to CPU. If this bit	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
			is cleared to “0”, this interrupt will be masked. To select between IRQ4 and FIQ, please refer to ***Chapter Interrupt***	
13	TMCEN	R/W	TimerC Module Enable If this bit is set to “1”, TimerC will start up-counting based on the frequency of selected clock source. If this bit is cleared to “0”, TimerC will stop counting.	0= Disabled 1= Enabled
12	SYNC	R/W	Clock Source Synchronization Enable If this bit is set to “1”, the clock source of TimerC is synchronized by CPU clock.	0= Disabled 1= Enabled
[11:10]			Reserved	
[9:8]	EXTASEL	R/W	External Input A (IOA12) Prescaler Setup These 2 bit are used to select the precaler function between external input signal and timer clock source	00= EXTA every falling 01= EXTA every rising 10= EXTA every 4 risings 11= EXTA every 16 risings
[7:4]			Reserved	
[3:0]	SRCASEL	R/W	Clock Source Group A selection. [1]When external Input A (“111” in binary) is selected, PortA12 is configured as counter clock source. It cannot be GPIO function, i.e., any GPIO setting on PortA12 will have no effect. [2]When PWM service rate is selected, PWM mode should be selected and either audio ChannelA or ChannelB should be enabled. Or, the timer will not count.	0000= SYSCLK/2 0001= SYSCLK/256 0010= 32768Hz 0011= 8192Hz 0100= 4096Hz 0101= 1 (logic high) 0110= 0 (logic low) 0111= EXTA with pre-scalar[1] 1000= PWM Service Rate[2] 1001~1111 = Reserved

P_TimerC_Preload **0x70D2** **TimerC Preload Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMCPLR	R/W	TimerC preload value	

P_TimerC_UpCount **0x70D4** **TimerC Up-Count**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMCUCR	R	TimerC Up-Counter value The up-counter's value can be read from this register.	

8.4 Program Examples

```
.DEFINE                     SYSCLK                      51609600
L_TimerCClockSourcePolling:
                            r1 = 0x0001
                            [P_IOC_Attrib] = r1
                            [P_IOC_Dir] = r1            //Setup IOC0 as output buffer low

                            Int off
                            r1 = (0x10000-SYSCLK/2/8000)  //Setup TimerC preload value
                            [P_TimerC_Preload] = r1     //so that up-counter overflow frequency = 8000Hz
                            r1 = 0x2000
                            [P_TimerC_Ctrl] = r1        //Enable TimerC
L_CheckTimerCOverflow?:
                            r1 = [P_TimerC_Ctrl]
                            jpl L_CheckTimerCOverflow?
                            [P_TimerC_Ctrl] = r1        //Clear TimerC up-counter-overflow interrupt flag
                            r1 =   [P_IOC_Buffer]       //Read Previous PortC Setup
                            r1 = r1 xor 0x0001
                            [P_IOC_Data] = r1           //Toggle PortC0 for 50% duty square wave
                            jmp   L_CheckTimerCOverflow?  // and frequency is 4000 Hz (8000/2)
L_TimerAPWMPolling:
                            r1 = 0x1000                 //IOA14 will output a square wave with 1/16 duty
                            [P_TimerA_CCP_Reg] = r1     //cycle and frequency is CPUCLK/2/65536
                            r1 = 0x0000
                            [P_TimerA_Preload] = r1
                            r1 = 0xC000
                            [TimerA_CCP_Ctrl] = r1
                            r1 = 0xA060
                            [P_TimerA_Ctrl] = r1
                            jmp $
```

9 Timebase

9.1 Introduction

Timebase, generated from a 32768Hz source, is a combination of frequency selections. The SPL161001A supports three timebases, accompanied with their interrupt mechanism and it facilitates timing control for most projects. In addition, timebase also provides a variety of frequency selections as clock source of timers.

These three timebases are as follows:

- 1Hz/ 2Hz/ 4Hz frequency programmable for TimebaseA
- 8Hz/ 16Hz/ 32Hz/ 64Hz frequency programmable for TimebaseB
- 128Hz/ 256Hz/ 512Hz/ 1024Hz frequency programmable for TimebaseC

9.2 Timebase structure and clock source

9.3 Control Registers

Timebase Control Register Summary Table

Name	Address	Description
P_TimeBaseA_Ctrl	0x70B0	TimeBaseA Control Register
P_TimeBaseB_Ctrl	0x70B1	TimeBaseB Control Register
P_TimeBaseC_Ctrl	0x70B2	TimeBaseC Control Register
P_TimeBase_Reset	0x70B8	TimeBase Counter Reset Register

P_TimeBaseA_Ctrl **0x70B0** **TimeBaseA Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBAIF/C	TMBAIE	TMBAEN	-	-	-	-	-	-	-	-	-	-	-	TMBAS	
Default	0	0	1	-	-	-	-	-	-	-	-	-	-	-	1	0

Bit	Function	Type	Description	Condition
15	TMBAIF/C	R/W	TimebaseA interrupt Flag Write “1” to clear the flag If TimebaseA interrupt occurs, this flag is set to “1” by hardware.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
14	TMBAIE	R/W	TimebaseA Interrupt Enable If this bit is set to “1” and TimebaseA interrupt occurs, hardware will issue an IRQ7 to CPU. If this bit is cleared to “0”, this interrupt will be masked.	0= Disabled 1= Enabled
13	TMBAEN	R/W	TimebaseA Module Enable If this bit is set to “1”, TimebaseA module will be enabled; otherwise, it will be disabled to save power.	0= Disabled 1= Enabled
[12:2]			Reserved	
[1:0]	TMBAS	R/W	TimebaseA frequency selection There are three frequency sources on TimebaseA, 1Hz, 2Hz, and 4Hz. These two control bits are to select one of three frequencies.	00= reserved 01= 1Hz 10= 2Hz 11= 4Hz

P_TimeBaseB_Ctrl **0x70B1** **TimeBaseB Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBBIF/C	TMBBIE	TMBBEN	-	-	-	-	-	-	-	-	-	-	-	TMBBS	
Default	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	0

Bit	Function	Type	Description	Condition
15	TMBBIF/C	R/W	TimebaseB interrupt Flag Write “1” to clear the flag. If TimebaseB interrupt occurs, this flag is set to “1” by hardware.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
14	TMBBIE	R/W	TimebaseB Interrupt enable If this bit is set to “1”, and TimebaseB interrupt occurs, hardware will issue an IRQ7 to CPU. If this bit is cleared to “0”, this interrupt will be masked.	0= Disabled 1= Enabled
13	TMBBEN	R/W	TimebaseB Module Enable If this bit is set to “1”, TimebaseB module will be enabled; otherwise, it will be disabled to save power.	0= Disabled 1= Enabled
[12:2]			Reserved	

Bit	Function	Type	Description	Condition
[1:0]	TMBBS	R/W	TimebaseB Frequency Selection There are four frequency sources on TimebaseB, 8Hz, 16Hz, 32Hz, and 64Hz. These two control bits are to select one of four frequencies.	00= 8Hz 01= 16Hz 10= 32Hz 11= 64Hz

P_TimeBaseC_Ctrl **0x70B2** **TimeBaseC Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBCIF/C	TMBCIE	TMBCEN	-	-	-	-	-	-	-	-	-	-	-	TMBCS	
Default	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	0

Bit	Function	Type	Description	Condition
15	TMBCIF/C	R/W	TimebaseC Interrupt Flag Write “1” to clear the flag If TimebaseC interrupt occurs, this flag is set to “1” by hardware.	Read 0= Not Occurred Read 1= occurred Write 0 = No Effect Write 1= Clear the flag
14	TMBCIE	R/W	TimebaseC Interrupt enable If this bit is set to “1” and TimebaseC interrupt occurs, hardware will issue an IRQ6 to CPU. If this bit is cleared to “0”, this interrupt will be masked.	0= Disabled 1= Enabled
13	TMBCEN	R/W	TimebaseC Module Enable If this bit is set to “1”, TimebaseC module will be enabled; otherwise, it will be disabled for power consumption consideration.	0= Disabled 1= Enabled
[12:2]			Reserved	
[1:0]	TMBCS	R/W	TimebaseC Frequency Selection There are four frequency sources on TimebaseC, 128Hz, 256Hz, 512Hz, and 1024Hz. These two control bits are to select one of four frequencies.	00= 128Hz 01= 256Hz 10= 512Hz 11= 1024Hz

P_TimeBase_Reset **0x70B8** **TimeBase Reset Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCCR															
Default																

Bit	Function	Type	Description	Condition
[15:0]	TMBCCR	W	Writing 0x5555 to this control register will reset the internal timebase counter for precise timing correction/control.	

9.4 Program Examples

```
L_PollingTimeBase:
                    r1 = 0x0007
                    [P_IOB_Attrib] = r1
                    [P_IOB_Dir] = r1                //Setup IOB[2..0] as output buffer low

                    r1 = 0xA001
                    [P_TimeBaseA_Ctrl] = r1         // Setup TimeBaseA Frequency as 1Hz
                    [P_TimeBaseB_Ctrl] = r1         // Setup TimeBaseB Frequency as 16Hz
                    [P_TimeBaseC_Ctrl] = r1         // Setup TimeBaseC Frequency as 128Hz
L_CheckTimeBaseOverflow?:
                    r1=[ P_TimeBaseC_Ctrl]
                    jpl     L_EndCheckTimeBaseC?
                            [P_TimeBaseC_Ctrl]=r1   //Clear TimeBaseC Flag
                            r1 = [P_IOB_Buffer]
                            r1 = r1 xor 0x0004
                            [P_IOB_Data] = r1       // Toggle IOB2 (frequency = 0.5Hz = 1Hz/2 )
L_EndCheckTimeBaseC?:
                    r1=[ P_TimeBaseB_Ctrl]
                    jpl     L_EndCheckTimeBaseB?
                            [P_TimeBaseB_Ctrl]=r1   //Clear TimeBaseB Flag
                            r1 = [P_IOB_Buffer]
                            r1 = r1 xor 0x0002
                            [P_IOB_Data] = r1       // Toggle IOB1 (frequency = 8Hz = 16Hz/2 )
L_EndCheckTimeBaseB?:
                    r1=[ P_TimeBaseA_Ctrl]
                    jpl     L_CheckTimeBaseOverflow?
                            [P_TimeBaseA_Ctrl]=r1   //Clear TimeBaseA Flag
                            r1 = [P_IOB_Buffer]
                            r1 = r1 xor 0x0001
                            [P_IOB_Data] = r1       // Toggle IOB0 (frequency = 64Hz = 128Hz/2 )
                    jmp     L_CheckTimeBaseOverflow?
```

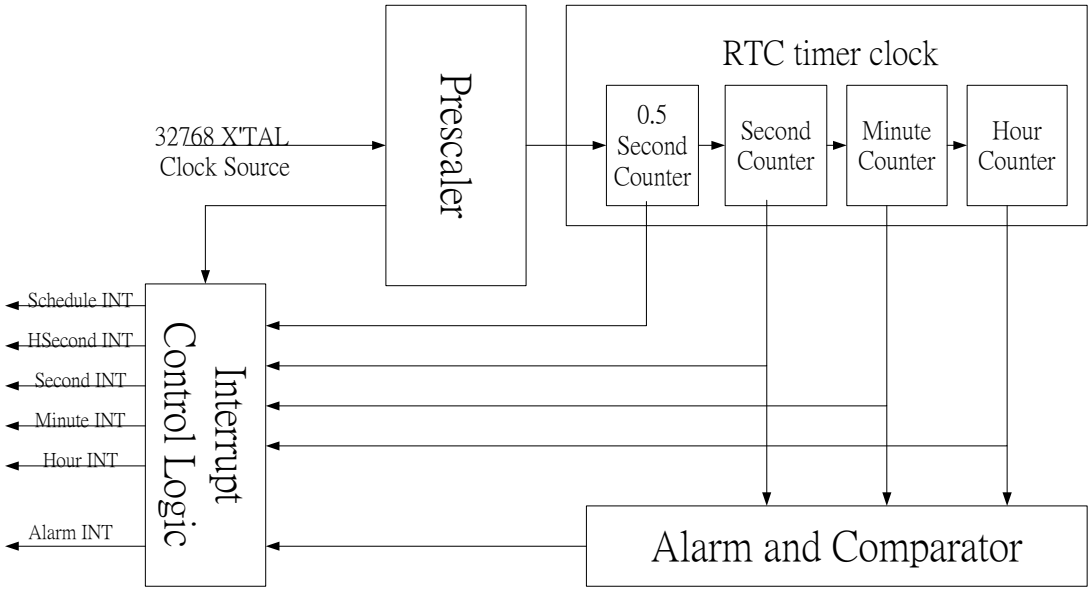
10 Real Time Clock (RTC)

10.1 Introduction

The SPL161001A provides real-time clock (RTC) module, which offers the following:

- Auto-update, up-to-hour clock register
- One alarm comparison registers
- Hour / Minute / Second / Half second interrupts
- Alarm Interrupt
- Scheduler interrupt: 16Hz / 32Hz / 64Hz / 128Hz / 256Hz / 512Hz / 1024Hz / 2048Hz

10.2 RTC Structure and clock source



10.3 Control Registers

Real Time Clock Control Register Summary Table

Name	Address	Description
P_Second	0x7120	Second Register
P_Minute	0x7121	Minute Register
P_Hour	0x7122	Hour Register
P_Alarm_Second	0x7124	Alarm Second Register
P_Alarm_Minute	0x7125	Alarm Minute Register
P_Alarm_Hour	0x7126	Alarm Hour Register
P_RTC_Ctrl	0x7134	HMS / Alarm / Scheduler Control Register
P_RTC_INT_Status	0x7135	HMS / Alarm / Scheduler Interrupt Flag & Clear Register
P_RTC_INT_Ctrl	0x7136	HMS / Alarm / Scheduler Interrupt Control Register

P_Second **0x7120** **Second Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	RTCSEC					
Default	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	RTCSEC	R/W	Real Time Clock Second Setup Register	Only 0x00~0x3B Valid (0~59)

P_Minute **0x7121** **Minute Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	RTCMIN					
Default	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	RTCMIN	R/W	Real Time Clock Minute Setup Register	Only 0x00~0x3B Valid (0~59)

P_Hour **0x7122** **Hour Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	RTCHR				
Default	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:5]			Reserved	
[4:0]	RTCCHR	R/W	Real Time Clock Hour Setup Register	Only 0x00~0x17 Valid (0~23)

P_Alarm_Second **0x7124** **Alarm Second Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	ALMSEC					
Default	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	ALMSEC	R/W	Alarm Second Setup Register	Only 0x00~0x3B Valid (0~59)

P_Alarm_Mintue **0x7125** **Alarm Minute Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	ALMMIN					
Default	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	ALMMIN	R/W	Alarm Minute Setup Register	Only 0x00~0x3B Valid (0~59)

P_Alarm_Hour **0x7126** **Alarm Hour Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	ALMHR				
Default	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:5]			Reserved	
[4:0]	ALMHR	R/W	Alarm Hour Setup Register	Only 0x00~0x17 Valid (0~23)

P_RTC_Ctrl **0x7134** **HMS / Alarm / Schedule Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RTCEN	-	-	-	-	ALMEN	HMSEN	SCHEN	-	-	-	-	SCHSEL			
Default	1	-	-	-	-	0	0	0	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
15	RTCEN	R/W	RTC Module Enable If this bit is set to “1”, this RTC module and Alarm	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
			Module will be enabled. If this bit is cleared to “0”, RTC module is disabled to save power.	
[14:11]			Reserved	
10	ALMEN	R/W	Alarm Function Enable If this bit is set to “1”, this alarm function will be enabled, and vice versa.	0= Disabled 1= Enabled
9	HMSEN		Hour/Minute/Second Function Enable If this bit is set to “1”, this hour/minute/second automatic updating function will be enabled, and vice versa.	0= Disabled 1= Enabled
8	SCHEN	R/W	Scheduler Function Enable If this bit is set to “1”, this scheduler module and will be enabled. If this bit is cleared to “0”, scheduler module is disabled to save power.	0= Disabled 1= Enabled
[7:4]			Reserved	
[3:0]	SCHSEL	R/W	Schedule Time Period Selection These 3 control bits are valid only when SCHEN control bit is set to “1”.	000 = 16Hz 001 = 32Hz 010 = 64Hz 011 = 128Hz 100 = 256Hz 101 = 512Hz 110 = 1024Hz 111 = 2048Hz

P_RTC_INT_Status 0x7135 HMS/ Alarm/ Schedule Interrupt Flag & Clear Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	ALMIEF/C	-	SCHIF/C	-	-	-	-	HRIF/C	MINIF/C	SECIF/C	HSECIF/C
Default	-	-	-	-	-	0	0	0	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
10	ALMIF/C	R/W	Alarm Interrupt Flag / Clear Write “1” to clear the flag This bit is set to “1” by hardware if the alarm interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
9			Reserved	
8	SCHIF/C	R/W	Schedule Interrupt Flag / Clear Write “1” to clear the flag. This bit is set to “1” by hardware if the schedule interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag

Bit	Function	Type	Description	Condition
[7:4]			Reserved	
3	HRIF/C	R/W	Hour Interrupt Flag / Clear Write “1” to clear the flag This bit is set to “1” by hardware if the hour interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
2	MINIF/C	R/W	Minute Interrupt Flag / Clear Write “1” to clear the flag This bit is set to “1” by hardware if the minute interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
1	SECIF/C	R/W	Second Interrupt Flag / Clear Write “1” to clear the flag This bit is set to “1” by hardware if the second interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
0	HSECIF/C	R/W	Half Second Interrupt Flag / Clear Write “1” to clear the flag This bit is set to “1” by hardware if the half-second interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag

P_RTC_INT_Ctrl 0x7136 HMS / Alarm / Schedule Interrupt Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	ALMIEN	-	SCHIEN	-	-	-	-	HRIEN	MINIEN	SECIEN	HSECIEN
Default	-	-	-	-	-	0	0	0	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
10	ALMIEN	R/W	Alarm Interrupt Enable If this bit is set to “1”, and alarm interrupt occurs, hardware will issue an IRQ7 to CPU. If this bit is cleared to “0”, this interrupt will be masked off.	0= Disabled 1= Enabled
9			Reserved	
8	SCHIEN	R/W	Scheduler Interrupt Enable If this bit is set to “1”, and schedule interrupt occurs, hardware will issue an IRQ6 to CPU. If this bit is cleared to “0”, this interrupt will be masked off.	0= Disabled 1= Enabled
[7:4]			Reserved	
3	HRIEN	R/W	Hour Interrupt Enable If this bit is set to “1”, and hour interrupt occurs, hardware will issue an HMS IRQ7 to CPU. If this bit is cleared to “0”, this interrupt will be masked	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
			off.	
2	MINIEN	R/W	Minute Interrupt Enable If this bit is set to “1”, and minute interrupt occurs, hardware will issue an HMS IRQ7 to CPU. If this bit is cleared to “0”, this interrupt will be masked off.	0= Disabled 1= Enabled
1	SECIEN	R/W	Second Interrupt Enable If this bit is set to “1”, and second interrupt occurs, hardware will issue an HMS IRQ7 to CPU. If this bit is cleared to “0”, this interrupt will be masked off.	0= Disabled 1= Enabled
0	HSECIEN	R/W	Half Second Interrupt Enable If this bit is set to “1”, and half second interrupt occurred, hardware will issue an HMS IRQ7 to CPU. If this bit is cleared to “0”, this interrupt will be masked off.	0= Disabled 1= Enabled

10.4 Program Examples

```
_RESET:
                    r1 = 59                     // Setup Current Clock as 07:59:59 (H:M:S)
                    [P_Minute] = r1
                    [P_Second] = r1
                    r1 = 7
                    [P_Hour] = r1
                    r1 = 0                      // Setup Alarm Clock as 08:00:00 (H:M:S)
                    [P_Alarm_Second] = r1
                    [P_Alarm_Minute] = r1
                    r1 = 8
                    [P_Alarm_Hour] = r1
                    r1 = 0x0400                 // Enable Alarm Interrupt only
                    [P_RTC_INT_Ctrl] = r1
                    r1 = 0x8700                 // Enable Alarm, Scheduler and
                    [P_RTC_Ctrl] = r1           // Hour/Minute/Second/Half-Second Module
                    int     irq                 // Enable CPU IRQ function
                    jmp     $                   // Dead Loop

//*******************************
_IRQ7:
                    push        r1,r2 to [sp]
```

```
                    r2 = [P_INT_Status2]
                    r1 = r2 & C_INT_Alarm
                    jz      L_End_Alarm_ISR?
                    r1 = [P_RTC_INT_Status]         // Note that even though that scheduler and
                    [P_RTC_INT_Status]=r1           // Hour/Minute/Second/Half-Second interrupt
                    cmp r1 , 0x050F                 // is not enabled, their interrupt flags can still be
                    jne     L_RTCTestError?         // polled ( if interrupt event occurs,
                    DisplayResultCode   D_OK        // corresponding flag is read as "1" )
L_RTCTestError?:    jmp   L_End_Alarm_ISR?

L_End_Alarm_ISR?:   DisplayResultCode   D_NG

                    pop         r1,r2 from [sp]
                    reti                            // Note that DisplayResultCode is a MACRO
```

11 Audio Output

11.1 DAC/PWM

The SPL161001A supports speech and melody synthesis functions. The sound data can be played back in the sequence of the control functions as designed by user's program. Several algorithms are recommended for sound compression: PCM, LOG PCM, DM and ADPCM. In addition, SUNPLUS provides SUBBAND, CLEP, LPC, HASC, LRC, Wavetable and FM low bitrate (high compression rate) software algorithm. On the other hand, for melody synthesis, the SPL161001A provides a dual tone mode to take advantage of less CPU bandwidths, despite the fact that SUNPLUS has already provided high quality Wavetable and FM solutions.

The SPL161001A provides both PWM and DAC drivers, which can be selected by software. Both PWM and DAC drivers support two channels. Each channel is able to play speech or tone individually. The PWM driver is able to drive speaker or buzzer directly without buffer or amplification circuit. In the DAC mode, the audio driver can be amplified by a bipolar-transistor or by an amplifier to drive a speaker or buzzer.

- Two 10-bit PWM / DAC channels
- PWM mode is able to drive speaker or buzzer directly (without any buffer or amplification circuit)
- Two sound modes - speech and tone
- Each channel has 8 * 10-bit ring buffer (FIFO)
- FIFO empty interrupt
- FIFO full indication flag
- Time sharing mechanism to serve two PWM channels

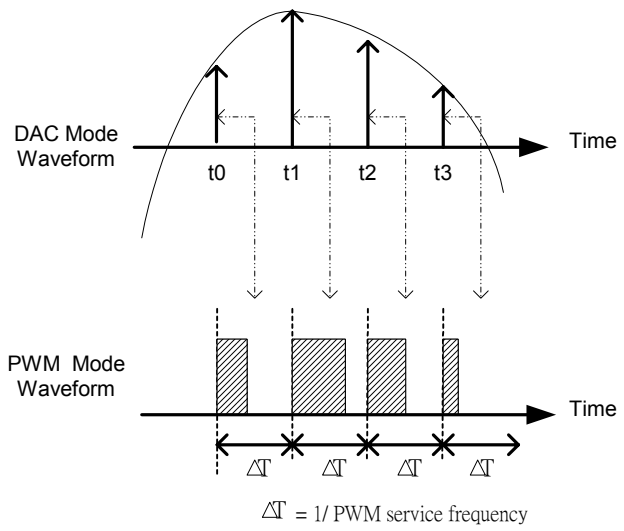
11.2 DAC Operation

In the DAC mode, the DACA current signal is output through AUDP pin, and the DACB is output through AUDB pin. The data of DACA and DACB should be delivered to P_CHA_Data (0x70F1) and P_CHB_Data (0x70F9) registers respectively. External components (some amplification circuits) are needed to drive the speaker. There are several solutions for these external components; for example, simple transistors (8050 BJT), standard OPs (LM324), or SUNPLUS amplifier SPY0012 and SPY0030.

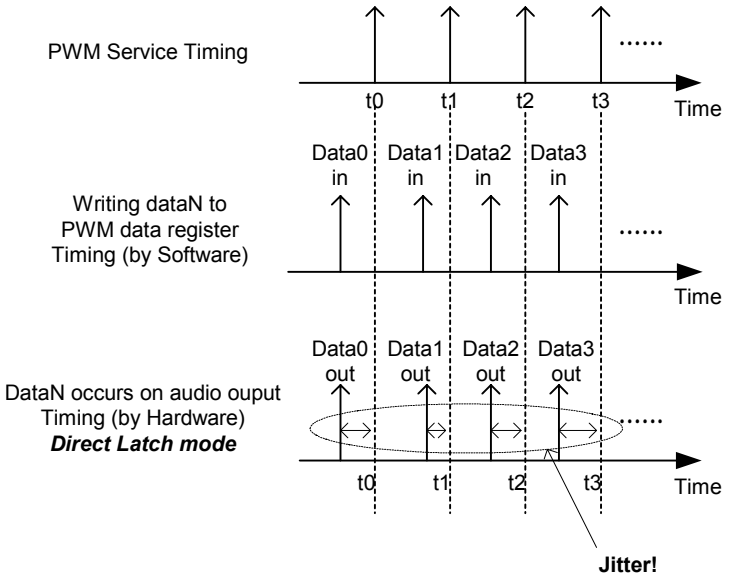
11.3 PWM Operation

The PWM driver can drive speaker or buzzer directly without buffer or amplification circuit. Generally, the PWM is sliced into pieces of time slots. In each time slot, it is further divided into 512 different pulse widths. These pulse widths can represent 10-bit data valuea. Therefore, the PWM driver requires a clock to serve each time slot. In addition, the PWM needs higher frequency clock to slice different widths of pulses. This highest frequency is called the PWM service frequency or service rate.

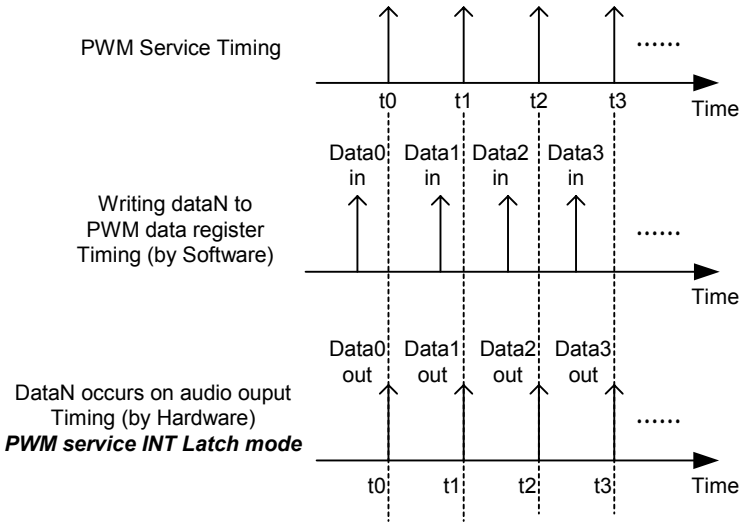
The following diagram is the comparison of timing scheme between DAC and PWM modes.



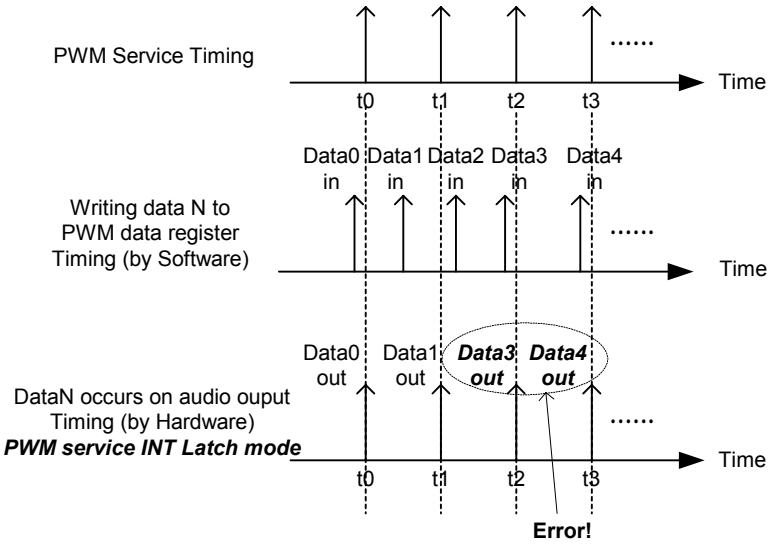
In the PWM mode, if hardware bypasses channel data register directly into the audio output, and once the program writes value to the channel data registers, the jitter will be depicted as in the following diagram because of the different timings on each output. Jitter is a major factor that causes high frequency noise. Note that the PWM service frequency is not the sampling rate of the audio signal. Generally, the PWM service frequency is higher than the sampling rate.



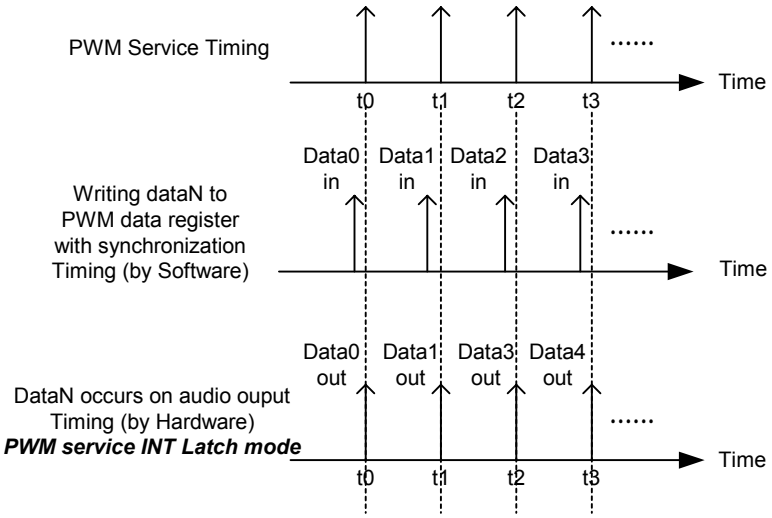
As a result, in the PWM mode, the time of data latch should be consistent with the PWM service time. As depicted in the following figure, the jitter does not occur.



However, it is not guaranteed that no error will occur based on the software control mentioned above. Suppose the timing written into the channel control register comes from different sources other than the clock source of the PWM service frequency, data may be lost. See the following illustration.



SUNPLUS recommends writing PWM data synchronized to the PWM service frequency. That is, the clock source of the timer must be considered while writing it into the channel data control register. It must come from the clock source of the PWM service frequency.



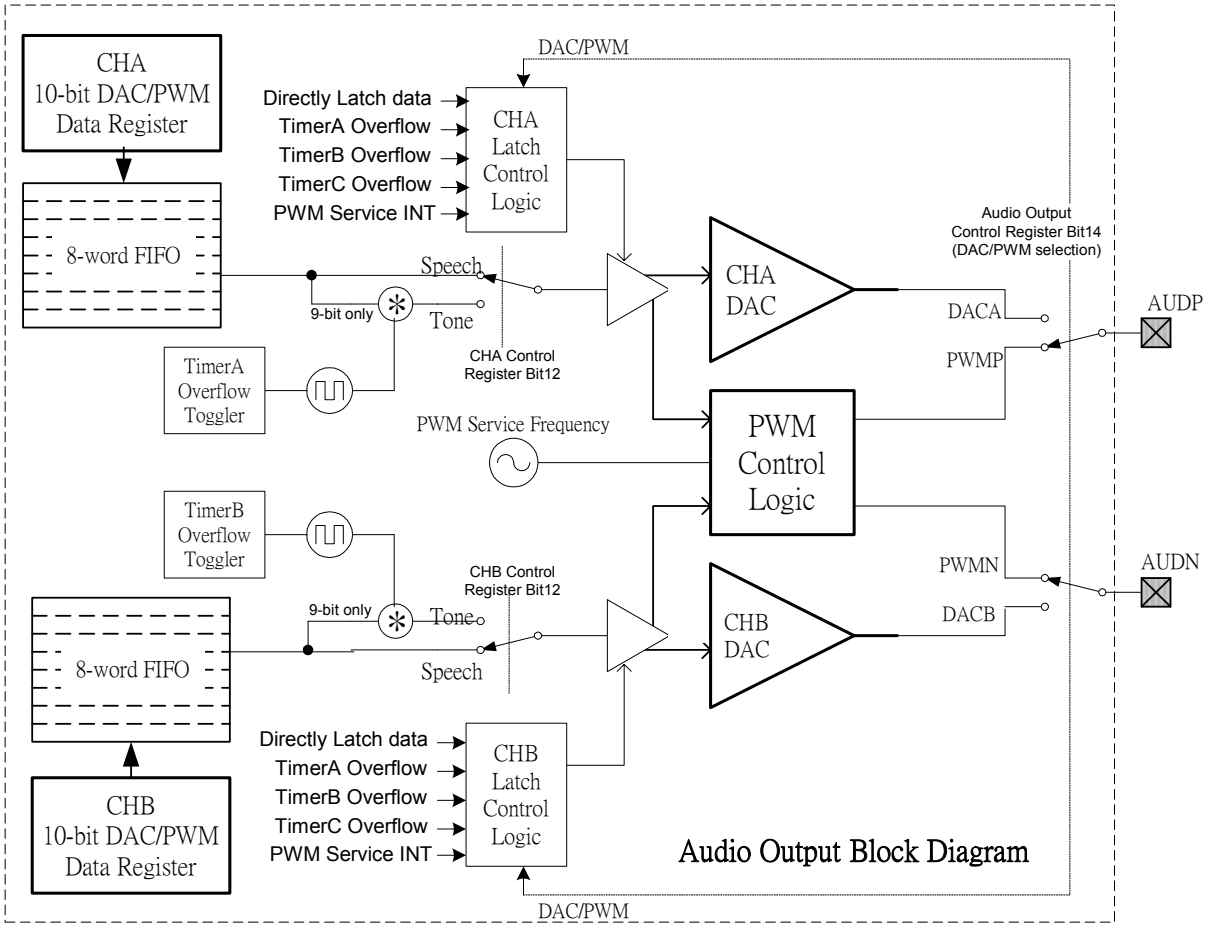
Note that the frequency used write data to the PWM data register is the sampling rate of audio data. Therefore, in order to avoid the above situation, the PWM service frequency should be a multiple of the sampling rate.

In conclusion, ***In PWM Mode, the PWM service frequency should be a multiple of the sampling rate of audio data.***

The maximum available PWM service rate of the SPL161001A is 100.8KHz. Therefore, the following sampling rate will induce better audio performance.

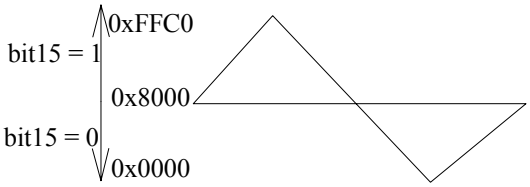
Multiple	2	3	4	5	6	7	8	9	10	11
Timer Preload Value	0xFFFE	0xFFFD	0xFFFC	0xFFFB	0xFFFA	0xFFF9	0xFFF8	0xFFF7	0xFFF6	0xFFF5
Sample Frequency (Hz)	50.4K	33.6K	25.2K	20.16K	16.8K	14.4K	12.6K	11.2K	10.08K	9.16K
Multiple	12	13	14	15	16	17	18	19	20	
Timer Preload Value	0xFFF4	0xFFF3	0xFFF2	0xFFF1	0xFFF0	0xFFEF	0xFFEE	0xFFED	0xFFEC	
Sample Frequency (Hz)	8.4K	7.75K	7.2K	6.72K	6.3K	5.93K	5.6K	5.3K	5.04K	

11.4 Structure

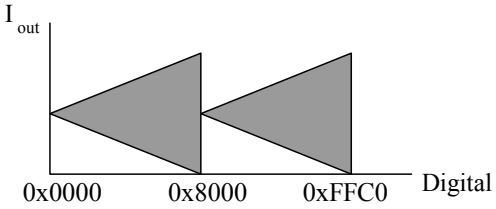


11.5 Speech Mode and Tone Mode

The SPL161001A provides two control bits to select speech mode and tone mode for each channel. In the speech mode, the data written to the data control register is the data in audio output. To playback voice or audio sound, the programmer only needs to obtain the appropriate PCM data and write to the channel data control register with a certain frequency, called the sampling rate, e.g., 8KHz. In most cases, programmers usually use one timer interrupt to serve this regular process with fixed frequency. The way of obtaining PCM data depends on the chosen decompression algorithm. In the tone mode, the audio output value is the production of a square wave and 9 LSB value in the channel data -register. The frequency of the square wave is half the frequency of the timer overflow. Therefore, programmers only need to setup the TimerA and TimerB overflow frequency for each channel, shape the envelop of each channel by modifying channel data value, and the hardware will toggle the tone wave automatically without using any interrupt and without run-time modification on tone envelope and timer overflow frequency. The audio output will generate a continuous tone at a fixed frequency.



DAC value to output relationship on speech mode



DAC value to output relationship on tone mode

11.6 Control Registers

DAC / PWM Control Register Summary Table

Name	Address	Description
P_AudioOut_Ctrl	0x70E0	Audio Output Control Register
P_CHA_Ctrl	0x70F0	Channel A Control Register
P_CHA_Data	0x70F1	Channel A Data Register
P_CHA_FIFO	0x70F2	Channel A FIFO Control Register
P_CHB_Ctrl	0x70F8	Channel B Control Register
P_CHB_Data	0x70F9	Channel B Data Register
P_CHB_FIFO	0x70FA	Channel B FIFO Control Register

P_AudioOut_Ctrl **0x70E0** **Audio Output Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PWMS	PWMDAC	OCS	-	RES	-	-	-	-	-	-	-	-	-	PWMFSEL	
Default	0	0	0	-	0	-	-	-	-	-	-	-	-	-	0	1

Bit	Function	Type	Description	Condition
15	PWMS	R	PWM service channel (read-only)	0= service CHA 1= service CHB
14	PWMDAC	R/W	PWM/DAC selection (write-once)	0= DAC 1= PWM
13	OCS	R/W	DAC Mode Output Current Selection Only available on DAC mode	0 = 3mA 1 = 2mA
12			Reserved	
11	RES	W	Software reset Write 1 to this bit to reset the DAC or PWM driver. It does not reset the register setting.	Write 1 to generate one reset signal
[1:0]	PWMFSEL	R/W	PWM service frequency (default: 00) These two bits are valid only when PWM mode is selected (PWMDAC control bit is set to “1”). *The default PWM service frequency is 50.4KHz when system clock is 51.6096MHz. In this case, to make sampling rate be multiple of service rate (50.4KHz), SUNPLUS recommends the sample rate to be 16.8KHz, 14.4KHz, 12.6KHz, 11.2KHz, 10.08KHz, 9.16KHz, 8.4KHz, 7.75KHz, and 7.2KHz.	00= SYSCLK/ 512 01= SYSCLK/ 1024* 10= SYSCLK/ 2048 11= SYSCLK/ 4096

P_CHA_Ctrl **0x70F0** **CHA DAC/PWM Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FEMIF/C	FEMIEN	CHAEN	TN/SP	-	-	DLSEL		-	-	-	-	-	-	-	-
Default	0	0	0	0	-	-	0	0	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
15	FEMIF/C	R/W	FIFO Empty Interrupt Flag Write “1” to clear the flag This bit is set to “1” by hardware if the FIFO empty interrupt is asserted. CHAFEILV defines the level that FIFO is considered as “empty”	Read 0= Not Occurred Read 1= Occurred Write 0 = No effect Write 1= Clear the flag
14	FEMIEN	R/W	FIFO Empty Interrupt enable	0= Disable

Bit	Function	Type	Description	Condition
			If this bit is set to "1", and if FIFO empty interrupt occurs, hardware will issue an IRQ0 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked. To select between IRQ0 and FIQ, please refer to ***Chapter Interrupt***	1= Enable
13	CHAEN	R/W	CHA Enable To avoid noise in DAC mode, SUNPLUS recommends to fill data 0x0000 into P_CHA_Data right after this bit is enabled. In PWM mode, this action is not needed.	0= Disable 1= Enable
12	TN/SP	R/W	Tone Mode and Speech Mode Selection **Note tone mode 00 is dedicated**	0= Speech Mode 1= Tone Mode
[11:10]			Reserved	
[9:8]	DLSEL	R/W	ChannelA Data Latch Mode Selection. This 2-bit control register is only valid when speech mode is selected. Note that when tone mode is selected, Data Latch Mode will automatically turn to direct mode.	00= Directly 01= After TimerA overflow 10= After TimerB overflow 11= After TimerC overflow
[7:0]			Reserved	

P_CHA_Data **0x70F1** **CHA DAC/PWM Data Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHADATA										-	-	-	-	-	-
Default	1	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
[15:6]	CHADATA	R/W	Channel A Data Register for both PWM Mode and DAC Mode	
[5:0]			Reserved	

P_CHA_FIFO **0x70F2** **CHA FIFO Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FFUL	-	-	-	-	-	-	-	CHAFEILV				CHAFINX			
Default	0	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	FFUL	R	CHA FIFO full flag. This flag is set to "1" by hardware if the FIFO is full. If the flag is "1", any data written via	0= FIFO is not full 1= FIFO is full

Bit	Function	Type	Description	Condition
			P_CHA_Data is invalid.	
[14:8]			Reserved	
[7:4]	CHAFEILV	R/W	CHA FIFO Empty Interrupt Level. The control bits are used to setup FIFO empty interrupt timing that is enabled on FEMIEN of P_CHA_Ctrl register. It defines the number of data left in FIFO to be considered as empty by hardware. The larger value is, the higher frequency of FIFO empty interrupt occurs. The smaller the value is, the less the frequency of an FIFO empty interrupt happening. Consequently, it saves the CPU's computational power.	FIFO Empty Interrupt issue timing 0000= when no data in FIFO 0001= when # of data in FIFO <=1 0010= when # of data in FIFO <=2 0011= when # of data in FIFO <=3 0100= when # of data in FIFO <=4 0101= when # of data in FIFO <=5 0110= when # of data in FIFO <=6 0111= when # of data in FIFO <=7 1000= when # of data in FIFO <=8 1001~1111= not valid
[3:0]	CHAFINX	R	CHA FIFO used (Default = 0000) FIFO is a ring consisted of 8*10bits.	0000= 0 data is in FIFO 0001= 1 data is in FIFO 0010= 2 data is in FIFO 0011= 3 data is in FIFO 0100= 4 data is in FIFO 0101= 5 data is in FIFO 0110= 6 data is in FIFO 0111= 7 data is in FIFO 1000= 8 data is in FIFO 1001~1111= not valid

P_CHB_Ctrl **0x70F8** **CHB DAC/PWM Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FEMIF/C	FEMIEN	CHBEN	TN/SP	-	-	DLSEL		-	-	-	-	-	-	-	-
Default	0	0	0	0	-	-	0	0	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
15	FEMIF/C	R/W	FIFO Empty Interrupt Flag Write "1" to clear the flag This bit is set to "1" by hardware if the FIFO empty interrupt is asserted. CHBFEILV defines the level that FIFO is considered as "empty"	Read 0= Not Occurred Read 1= Occurred Write 0 = No effect Write 1= Clear the flag
14	FEMIEN	R/W	FIFO Empty Interrupt enable If this bit is set to "1", and if FIFO empty interrupt occurs, hardware will issue an IRQ0 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked. To select between IRQ0 and FIQ, please refer to ***Chapter***	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
			Interrupt.	
13	CHBEN	R/W	CHB Enable To avoid noise in DAC mode, SUNPLUS recommends to fill data 0x0000 into P_CHB_Data right after this bit is enabled. In PWM mode, this action is not needed.	0= Disabled 1= Enabled
12	TN/SP	R/W	Tone Mode and Speech Mode Selection **Note tone mode 00 is dedicated.**	0= Speech Mode 1= Tone Mode
[11:10]			Reserved	
[9:8]	DLSEL	R/W	Channel B Data Latch Mode Selection This 2-bit control register is only valid when speech mode is selected. Note that when tone mode is selected, Data Latch Mode will automatically turn to direct mode.	00= Directly 01= After TimerA overflow 10= After TimerB overflow 11= After TimerC overflow
[7:0]			Reserved	

P_CHB_Data **0x70F9** **CHB DAC/PWM Data Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHBDATA										-	-	-	-	-	-
Default	1	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
[15:6]	CHBDATA	R/W	Channel B Data Register for both PWM Mode and DAC Mode	
[5:0]			Reserved	

P_CHB_FIFO **0x70FA** **CHB FIFO Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FFUL	-	-	-	-	-	-	-	CHBFEILV				CHBFINX			
Default	0	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	FFUL	R	CHB FIFO full flag This flag is set to “1” by hardware if the FIFO is full. If the flag is “1”, any data written via P_CHB_Data is invalid.	0= FIFO is not full 1= FIFO is full
[14:8]			Reserved	
[7:4]	CHBFEILV	R/W	CHB FIFO Empty Interrupt Level The control bits are used to setup FIFO empty interrupt timing that is enabled on	FIFO Empty Interrupt issue timing 0000= when no data in FIFO 0001= when # of data in FIFO <=1

Bit	Function	Type	Description	Condition
			FEMIEN of P_CHB_Ctrl register. It defines the number of data left in FIFO to be considered as empty by hardware. The larger value is, the higher frequency of the FIFO empty interrupt happens. The smaller value means the less the frequency of an FIFO empty interrupt occuring. Consequently, it saves the CPU's computational power.	0010= when # of data in FIFO <=2 0011= when # of data in FIFO <=3 0100= when # of data in FIFO <=4 0101= when # of data in FIFO <=5 0110= when # of data in FIFO <=6 0111= when # of data in FIFO <=7 1000= when # of data in FIFO <=8 1001~1111= not valid
[3:0]	CHBFINX	R	CHB FIFO used (Default = 0000) FIFO is a ring consisted of 8*10bits.	0000= 0 data is in FIFO 0001= 1 data is in FIFO 0010= 2 data is in FIFO 0011= 3 data is in FIFO 0100= 4 data is in FIFO 0101= 5 data is in FIFO 0110= 6 data is in FIFO 0111= 7 data is in FIFO 1000= 8 data is in FIFO 1001~1111= not valid

11.7 Mute level

In the PWM mode, the mute level is 0x8000. In the DAC mode, the mute level is 0x0000. In order to make the sound completely silent, SUNPLUS recommends programmers should make both channelA and channelB audio output value to the mute level. In addition, to completely shut down the audio output module in the SPL161001A, both channelA and channelB enable control bit (P_CHA_Ctrl.bit13 and P_CHB_Ctrl.bit13) should also be clear to "0".

11.8 Ramp Up and Ramp Down

The middle value of the speech file is 0x8000 (first PCM data of speech file usually starts with middle level, 0x8000). The mute level in the DAC mode is 0x0000. Therefore, in order to avoid unexpected sound caused by changing audio output value rapidly (from 0x0000 to 0x8000), some smooth methods, known as ramp up and ramp down, should be used.

Ramp up is the operation that changes from mute level to middle level (0x8000). Ramp down is the operation that changes from middle level (0x8000) to mute level. The implementation of ramp up and ramp down is used to achieve the target value by gradually adding/subtracting a constant slope from the start value.

11.9 Program Examples (Playing a PCM Speech)

```
                    r1 = 0xD007                         // Setup PLL Frequency as 51.6096MHz
                    [P_Clock_Ctrl] = r1
                    r1 = 0x4000                         // Setup PWM Mode with Service Rate
                    [P_AudioOut_Ctrl] = r1              // 100.8KHz (=51.6096MHz / 512)
                    r1 = 0x0030                         // Setup Audio Channel A, B FIFO Empty
                    [P_INT_Priority1]= r1               // Interrupt as FIQ (Fast Interrupt)

                    r1 = 65536-(100800/16800)           // Setup TimerB, C Overflow Frequency
                    [P_TimerB_Preload] = r1             // as 16.8KHz
                    [P_TimerC_Preload] = r1
                    r1 = 0xE008                         // Setup PWM Service Rate as TimerB,C
                    [P_TimerB_Ctrl] = r1                // Clock Source
                    [P_TimerC_Ctrl] = r1

                    r1 = 0x0003                         // Setup FIFO Empty Level = 3
                    [P_CHA_FIFO] = r1
                    [P_CHB_FIFO] = r1
                    r1 = 0xE200                         // Setup TimerB overflow rate as audio
                    [P_CHA_Ctrl] = r1                   // output rate of PWM Channel A
                    r1 = 0xE300                         // Setup TimerC overflow rate as audio
                    [P_CHB_Ctrl] = r1                   // output rate of PWM Channel B
                    fiq     on
                    ........
                    ........
                    ........
//******************************
_FIQ:


                    push r1,r4 to [sp]
                    r4 = [P_INT_Status1]
                    r1 = r4 & C_INT_AUDAFIFOEmpty
                    jz      L_EndServiceCHAInterrupt?
L_ServiceCHAInterrupt?:
                    [P_CHA_Ctrl] = r1                   // Clear Channel A FIFO Empty Interrupt
                    r1 = [P_CHA_Ctrl]
                    ReadResourceData      D_Sound1      // Fetch PWM Data
                    [P_CHA_Data] = r1                   // and output the PWM data to CHA FIFO
                    ReadResourceData      D_Sound1      //   for 5 times
                    [P_CHA_Data] = r1
                    ReadResourceData      D_Sound1
```

```
                    [P_CHA_Data] = r1
                    ReadResourceData    D_Sound1
                    [P_CHA_Data] = r1

                    ReadResourceData    D_Sound1
                    [P_CHA_Data] = r1
L_EndServiceCHAInterrupt?:
                    r1 = r4 & C_INT_AUDBFIFOEmpty
                    jz    L_EndServiceCHBInterrupt?
L_ServiceCHBInterrupt?:                             // Clear Channel B FIFO Empty Interrupt
                    [P_CHB_Ctrl] = r1
                    r1 = [P_CHB_Ctrl]               // Fetch PWM Data
                    ReadResourceData    D_Sound2    // and output the PWM data to CHB FIFO
                    [P_CHB_Data] = r1               //   for 5 times
                    ReadResourceData    D_Sound2
                    [P_CHB_Data] = r1
                    ReadResourceData    D_Sound2
                    [P_CHB_Data] = r1
                    ReadResourceData    D_Sound2
                    [P_CHB_Data] = r1
                    ReadResourceData    D_Sound2
                    [P_CHB_Data] = r1

                    pop r1,r4 from [sp]
                    reti                            // Note: ReadResourceData is MACRO
L_EndServiceCHBInterrupt?:
```

Theoretically, the interrupt service routine of FIQ should be as short as possible. This is because FIQ plays an important role like NMI (Non-Maskable Interrupt). The above FIQ interrupt service routine is only for a demonstration that depicts the operation and function of control registers. This sample code may not be a perfect example for playing speech.

12 LCD

12.1 Introduction

The LCD controller in the SPL161001A has the following features:

- Supports standard STN LCD drivers interface.
- LCD buffer shares with system memory.
- Built-in frame rate control for gray-level display
- Supports standard 1/ 4/ 8-bit LCD driver Interface
- Supports monochrome, 2 gray levels, 4 gray levels, and 16 gray levels.
- Supports virtual display screen up to 1024 x 1024 for hardware horizontal / vertical scrolling capability.

12.2 LCD Control Pin Configuration

Name	I/O	Description
LCDFM	O	Frame modulation signal (dedicated)
LCDFP	O	Frame Rate signal (dedicated)
LCDLP	O	Line Scan signal (dedicated)
LCDCP	O	Shifting clock signal (dedicated)
LCDD0	O	LCD Data 0 (dedicated)
LCDD1	O	LCD Data 1 (shared with GPIO PortD4)
LCDD2	O	LCD Data 2 (shared with GPIO PortD5)
LCDD3	O	LCD Data 3 (shared with GPIO PortD6)
LCDD4	O	LCD Data 4 (shared with GPIO PortA3)
LCDD5	O	LCD Data 5 (shared with GPIO PortA2)
LCDD6	O	LCD Data 6 (shared with GPIO PortA1)
LCDD7	O	LCD Data 7 (shared with GPIO PortA0)

LCDFM, LCDFP, LCDLP, and LCDCP are synchronous signals used for transferring LCD display data from the SPL161001A to SPLD/SPLC LCD drivers. The LCDD[7:0] is a data port for transferring LCD data.

12.3 LCD Buffer

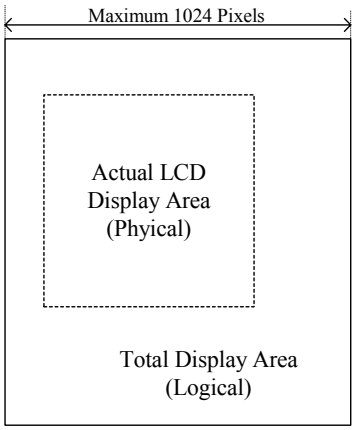
The SPL161001A allows programmers to locate LCD buffer inside entire SUNPLUS 16-bit CPU 4M-word addressing range except internal ROM (0x08000 ~ 0x47FFF). In other words, programmers are able to define any area within the 4M-word addressing field as an LCD buffer. Certainly, the area,

defined as a LCD buffer, can be SRAM, ROM, or Flash. The LCD buffer can also be used as working RAM. The control registers, P_LCD_Buffer_HighAdr and P_LCD_Buffer_LowAdr, define the start address of the LCD buffer. In addition, different LCD pages (LCD buffer) can be easily switched by programming the LCD start address register.

The programmer can define more than one LCD buffer in the system. In this case, the switch operation between different LCD pages can speed up frame updating. Please note that when the LCD buffer is switched from high speed SRAM to slow speed ROM, it is necessary to properly modify in the LCD buffer wait state. For detail criterion about LCD buffer wait cycle setup, refer to the following ***Section: Control Register.***

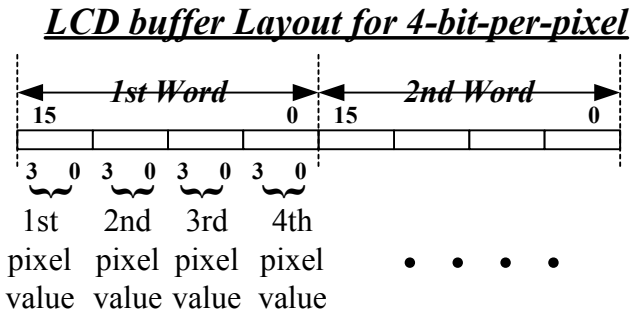
LCD buffer size is determined by two control registers, P_LCD_Common and P_LCD_Segment. The maximum number of common lines (horizontal line) to be supported is 320 pixels and the maximum segment line (vertical line) is also 320 pixels as well. In most cases, the size of LCD buffer equals to { (P_LCD_Common +1) x (P_LCD_Segment +1) x bpp}}}}}}} bits, where bpp represents bits-per-pixel.

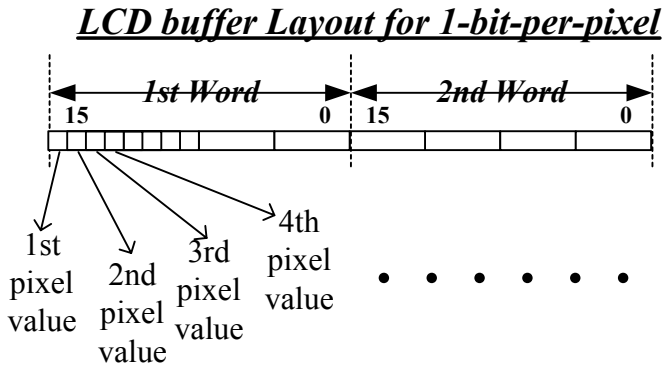
The SPL161001A also supports a virtual page function that allows programmers to define a virtual LCD buffer larger than actual viewable size (actual LCD size). In this case, the programmer can easily scroll actual LCD display screen horizontally and vertically on the full virtual page by simply defining P_LCD_Buffer_Offset control register and modifying the following two control registers, P_LCD_Buffer_HighAdr and P_LCD_Buffer_LowAdr. The maximum width in a virtual page is 1024 pixels, as depicted in the following diagram.



12.4 LCD Buffer Layout

The SPL161001A offers two modes of LCD buffer layout: 4-bit per pixel for 16 gray-level display and 1-bit per pixel for mono display. In the gray-level mode, 0b0000 represents the lightest (white) gray-level while 0b1111 represents the darkest (black) gray-level.





12.5 Control Registers

LCD Control Register Summary Table

Name	Address	Description
P_LCD_Setup	0x7180	LCD setup register
P_LCD_Clock	0x7181	LCD clock register
P_LCD_Segment	0x7182	LCD segment number register
P_LCD_Common	0x7183	LCD common number register
P_LCD_Buffer_LowAdr	0x7184	LCD start address register (A0~A15)
P_LCD_Buffer_HighAdr	0x7185	LCD start address register (A16~A21)
P_LCD_Buffer_Offset	0x7186	LCD virtual page offset register
P_LCD_Timing_Ctrl	0x7187	LCD control signal timing register
P_LCD_Frame_Ctrl	0x7188	LCD frame modulation control register

P_LCD_Ctrl **0x7180** **LCD Setup Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FPIF/C	FPIEN	LCDEN	SELF	-	-	BUSW		-	-	-	-	LCDBUFW			
Default	0	0	0	0	-	-	0	0	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
15	FPIF/C	R/W	LCD FP Signal Interrupt Flag. This bit is set to “1” by hardware if the FP interrupt is asserted. When FP signal is at rising edge and FPIEN is set to “1”, the FP interrupt is issued (FPIF=1). It informs CPU that a new frame is beginning. And CPU may update some display data to a new frame.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
14	FPIEN	R/W	LCD FP Signal Interrupt Enable If this bit is set to “1” and if FP interrupt is generated, hardware will issue an IRQ5 or FIQ to CPU. If this bit is cleared to “0”, the interrupt will be masked off. To select between IRQ5 and FIQ, please refer to ***Chapter Interrupt.***	0= Disabled 1= Enabled
13	LCDEN	R/W	LCD Interface Enable If this bit is set to “1”, LCD interface is enabled. Or, LCD interface is disabled. Setting LCDEN control bit to “1” will enable LCD controller. This control bit should be remained as “1” whenever the LCD interface is active. When this control bit is cleared to “0”, all output signals will stay in LOW state.	0= Disabled 1= Enabled
12	SELF	R/W	Self Refresh mode If the external LCD driver(s) involves the built-in memory and supports the self-refresh mode, the LCD interface can be configured to self-refresh mode. The LCD driver shows the last display data and SPL161001A outputs FM, FP, LP signals only. The CP and LCD Data signal will remain at ground state.	0= Disabled 1= Enabled
[11:10]			Reserved	
[9:8]	BUSW	R/W	LCD hardware data bus width configuration If “01” is set, PortD[6:4] plus dedicated LCDD0 pin to form a 4-bit LCD data bus, and PortD[6:4] cannot be used as GPIO. If “10” is set, PortD[6:4] and PortA[3:0] plus	00= 1-bit (LCDD0 Valid) 01= 4-bit (LCDD[3:0] Valid) 10= 8-bit (LCDD[7:0] Valid) 11= Reserved

Bit	Function	Type	Description	Condition
			LCDD0 pin to form an 8-bit LCD data bus, and both PortD[6:4] and PortA[3:0] cannot be used as GPIO.	
[7:4]			Reserved	
[3:0]	LCDBUFW	R/W	LCD Buffer Wait Cycle Setup This control register is to set the LCD buffer wait cycle. LCD buffer can be either external memory or internal memory.	(2+LCDBUFW) x SYSCLK_cycle Range= 0 ~ 15

The LCD controller can either fetch display data from external or from internal memory. While accessing some slow external memory, the LCD controller executes some wait states if the access time of external memory is longer than two times of system clock cycles. In other words, the following criterion must be satisfied.

External memory access time < (LCDBUFW + 2) * system clock cycle

According to the memory data sheet, access time is under specified voltage conditions. However, the lower operating voltage is, the longer access time it takes. Therefore, making sure criterion given above will always be satisfied for all operating voltage ranges.

Please note that when the LCD buffer is switched from high speed SRAM to slow speed ROM or FLASH, it is necessary to appropriately modify the LCD buffer wait state .

P_LCD_Clock 0x7181 LCD Clock Generation Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	LCDCLK									
Default	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	LCDCLK	R/W	Pixel Clock Divider	Range= 0~ 1023

LCDCLK[9:0] = SYSCLK / (LCDCOM+1) / (LCDSEG+1) / Frame-rate - 2

where

LCDCOM is number of common - 1, LCDSEG is number of segment - 1

Frame-rate is the frequency of frame pulse signal (FP).

In general, for mono display, frame rate is approximately 60Hz. For gray level or color display, frame rate can be about 180~90Hz.

For example:

If in a 160x160 gray-level color solution and system clock is 51.6096MHz,

We obtain:

51609600 / 160 / 160 / 180 – 2 = 9.2

Therefore, LCDCLK[9:0] can be set as 9 or 10 in decimal.

Frame rate = 51609600 / 160 / 160 / (9 +2) = 183.3Hz (if LCDCLK[9:0]=8)

Frame rate = 51609600 / 160 / 160 / (10 +2) = 168Hz (if LCDCLK[9:0]=9)

If in a 320x240 mono solution and system clock is 29.4912MHz,

We obtain:

29491200 / 320 / 240 / 60 – 2 = 4.4

Therefore, LCDCLK[9:0] can be set as 4 or 5, even 3 in decimal for higher frame rate.

Frame rate = 29491200 / 320 / 240 / (3 + 2) = 76.8 Hz (if LCDCLK[9:0]=2)

Frame rate = 29491200 / 320 / 240 / (4 + 2) = 64 Hz (if LCDCLK[9:0]=3)

Frame rate = 29491200 / 320 / 240 / (5 + 2) = 54.6 Hz (if LCDCLK[9:0]=4)

Important Note: To decrease power consumption and save CPU computational power on external memory device, system designers can try to minimize LCD frame rate as small as possible until LCD is not flicking.

P_LCD_Segment 0x7182 LCD Segment Number Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	LCDSEG								
Default	-	-	-	-	-	-	-	0	0	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
[8:0]	LCDSEG	R/W	LCD Panel Segment Number Register LCDSEG[3:0] is fixed to “1111” in binary.	Range= 15 ~ 319 LCDSEG[3:0] should be “1111”

This register defines the segment number of LCD panels that are equals to LCDSEG[9:0]+1. It must be a multiple of 16. The maximum segment number is 320 and minimum segment number is 16. Therefore, LCDSEG[3:0] should be equal to “1111” in binary. Attempt to write “0” to any of LCDSEG[3:0] will have no effect. In other words, LCDSEG[3:0] is read only.

P_LCD_Common 0x7183 LCD Common Number Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	LCDCOM								
Default	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
[8:0]	LCDCOM	R/W	LCD Panel Common Number Register	Range= 0~319

LCDCOM[8:0] control register defines the vertical size of the LCD panel. The actual size (number of common) is LCDCOM[8:0]+1. The maximum common number is 320 and minimum common number is 1.

P_LCD_Buffer_LowAdr 0x7184 LCD Buffer Address A15 ~ A0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LCDBUFAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LCDBUFAL	R/W	LCD Buffer Address [15:0]	

P_LCD_Buffer_HighAdr 0x7185 LCD Buffer Address A22~A16

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	LCDBUFAH					
Default	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0

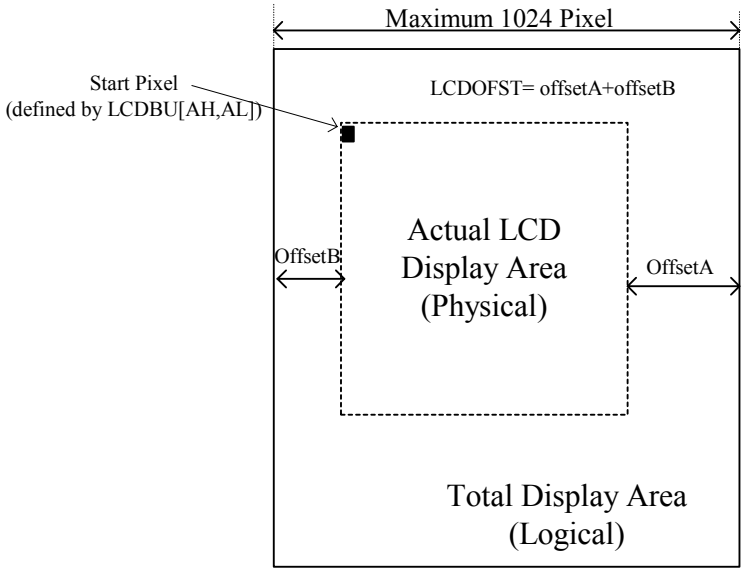
Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	LCDBUFAH	R/W	LCD Buffer Address [21:16]	

LCDBUFAH and LCDBUFAL construct a 22-bit addressing register to define the start address of the LCD buffer. The 22-bit addressing register is able to accomplish the entire 16-bit CPU view, at a total of a 4M-word addressing area.

P_LCD_Buffer_Offset 0x7186 LCD Offset Size

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	LCDOFST									
Default	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	LCDOFST	R/W	LCD Virtual Page Offset (Unit: pixel). The value given here must be the multiple of 16, i.e. the bit3~bit0 must be all 0s.	



Note: LCDOFST+LCDSEG+1 should smaller than 1024.

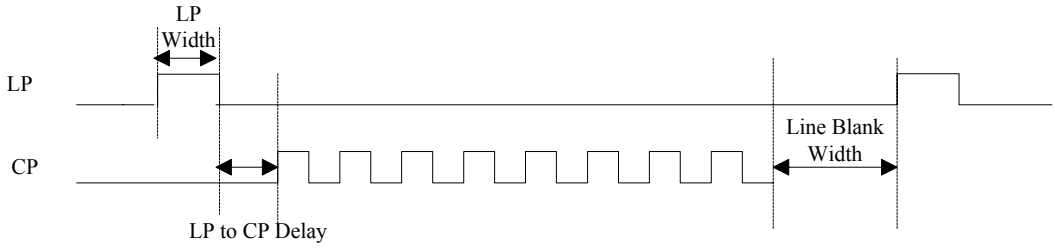
P_LCD_Buffer_Offset and P_LCD_Buffer_Segment control registers setup the horizontal size of a virtual page. Note that the maximum of a virtual page is 1024 pixels. To move the actual display area horizontally or vertically, programmers have to modify P_LCD_Buffer_HighAdr and P_LCD_Buffer_LowAdr control registers.

P_LCD_Timing_Ctrl 0x7187 LCD Control Signal Timing Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	LBVL				LPW				LPCPD			
Default	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:8]	LBVL	R/W	Line Blank Width Please refer to following timing diagram	T= (LBVL+1) x CLCPCLK Range= 0~15
[7:4]	LPW	R/W	LP Pulse Width Please refer to following timing diagram	T= (LVW+1) x CLCPCLK Range= 0~15
[3:0]	LPCPD	R/W	LP to CP Delay Please refer to following timing diagram	T= (LPCPD+1) x CLCPCLK Range= 0~15

Note: Generally, this register does not need to be changed. The purpose of this register is to adjust the LCD control signals for special LCD drivers.

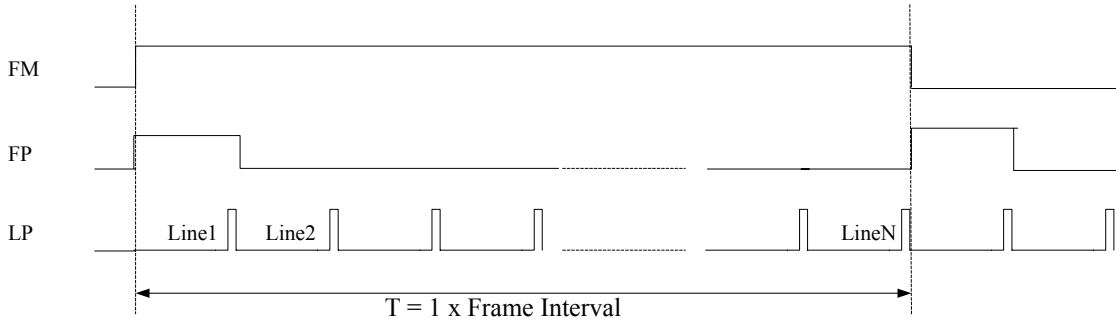


P_LCD_Frame_Ctrl **0x7188** **LCD Frame Modulation Control Register**

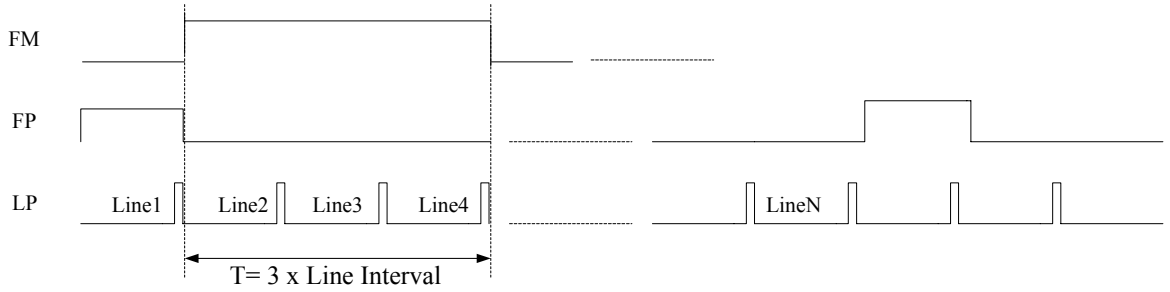
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BCMOD	-	-	-	-	-	-	-	MVAL							
Default	0	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	BCMOD	R/W	LCD Frame Modulation Type Please refer to following diagram	0= B type (Typical) 1= C type
[14:8]			Reserved	
[7:0]	MVAL	R/W	Define the frequency of frame modulation when C type is in active. Be sure that the (no. of common) and (MVAL+1) are not	Range= 0 ~ 255

For B Type Frame Modulation Mode (BCMOD=0), FM toggles at each FP signal.



In the C type frame modulation mode (BCDMOD=1), FM changes its state for each (MAL[7:0]+2) LP signal. For example, if MVAL[7:0]=1, FM signal changes its state each three LP signals. See the following diagram for reference. If MVAL[7:0]+2 equals the number of common, the C type FM signal is the same as B type FM signal.



P_LCD_Buffer_Ctrl 0x7189 LCD Buffer Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	OVIF/C	-	-	-	-	-	-	-	-	-	-	-	-	BPP	-	-
Default	0	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-

Bit	Function	Type	Description	Condition
15	OVIF/C	R/W	This bit is set to “1” by hardware if the LCD DMA operation overflow interrupt is asserted.	Read 0= Not Occurs Read 1= Happened Write 1= Clear the flag
[14:3]			Reserved	
2	BPP	R/W	Bit Per Pixel	0= 1 bit/pixel 1= 4 bits/pixel
[1:0			Reserved	

12.6 Operation during Wait/Halt/Standby & Wakeup Procedure

In wait mode, the LCD is able to remain functioning even when CPU is turned off because the SPL161001A keeps the system clock activated. On the other hand, the system clock is shutdown in both the halt and standby modes making the LCD unable to display in such modes. Therefore, Sunplus recommends setting the LCDEN control bit to “0” before entering the halt mode and standby mode. After the SPL161001A wakes up from those modes, set the LCDEN control bit to “1” again. Simply, The programmer simplyturns on/off the LCD panel via the configuration of the LCDEN control bit. The SPL161001A will automatically perform the LCD power on/off procedures in order to avoid unexpected lines occurring on the LCD panel. On the other hand, Sunplus recommends setting the SYSCLK to be from OSC, which is of lower frequency, when system enters wait mode to save power.

12.7 LCD Image Resource File Tooling

Sunplus offers some tools for running on your computer to transfer BMP files to LCD buffer format. The tools are as follows: PO4COLOR.EXE, SWAPWORD.EXE, and POBMP.EXE.

****

<Mono BMP File> to <1-Bit-Per-Pixel Format of LCD Buffer>(2 Mono)

****

Conversion Batch File

```
pobmp /f2 %1.bmp /d /o
swapword %1.put %.raw /o /b4
```

Note: there is no header in the .raw file

****

<256 Color BMP File> to <4-Bit-Per-Pixel Format of LCD Buffer>(16 Gray)

****

Conversion Batch File

```
po4color b %1.bmp %1.bin /o /b-4 /x
swapword %1.bin %1.raw /o /b4
```

Note: The first word of .raw represents the number of picture width and the second word of .raw means the number of picture height. The image raw data starts from third word.

12.8 Program Examples

160 (Segment) x 160 (Common) 16 gray-level, 4-bit interface; LCD buffer start address = 0x030000

```
.DEFINE     SEGMENT     160
.DEFINE     COMMON      160

            r1 = 0x0381
            [P_MCS0_Ctrl] = r1
            r1 = (51690600/SEGMENT/COMMON/180-2)
            [P_LCD_Clock] = r1
            r1 = 5
            [P_LCD_Buffer_HighAdr] = r1
            r1 = 0
            [P_LCD_Buffer_LowAdr] = r1
            [P_LCD_Buffer_Offset] = r1
            [P_LCD_Timing_Ctrl] = r1
```

```
[P_LCD_Frame_Ctrl] = r1

r1 = SEGMENT -1
[P_LCD_Segment] = r1
r1 = COMMON -1
[P_LCD_Common] = r1                    // 8-bit-per-pixel (256 Colors)

r1 = 0x0004                            // Data width 4-bit
[P_LCD_Buffer_Ctrl] = r1               // External LCD buffer access time 50ns
r1 = 0x2101
[P_LCD_Ctrl] = r1
retf
```

Note that the display pattern can be modified ONLY when the LCDEN control bit is cleared to 0 (P_LCD_Ctrl.bit13=0).

12.9 LCD Driver Selection Guide

SPLC56X Series

SUNPLUS provides the SPLC56X as a new LCD driver series for color or high-resolution LCD products. The following table depicts the comparison table for the SPLC56X series. For more details, please refer to the SPLC56X data sheet.

Body	Channel	Segment	Common	Maximum Voltage	Interface	Package Form
SPLC560	160	160 0	0 160	30V	4/8 bit[2]	Cold bump /TCP
SPLC562	240	240 0	0 240	30V	4/8 bit[2]	Cold bump /TCP
SPLC563	320	320	0	5.5V	4/8 bit[2]	Cold bump /TCP
SPLC564[1]	240	0	240	43V	-	Cold bump /TCP

[1] the SPLC564 is a pure common LCD driver. It should always be paired with the SPLC563 because they both are of the APT driving scheme. In other words, neither the SPLC563 nor the SPLC564 cooperate with the SPLC560 or SPLC562.

[2] The criterion used to select between the 4-bti and 8-bit LCD driver interface is a trade off between the number of I/O pad connections and power consumption (8-bit consumes less power than 4-bit LCD interface).

SPLD Series

Like the SPL13X, the SPL161001A can also cooperate with the SPLD Series. Please refer to the following comparison table.

Body	Channel	Segment	Common	Maximum Voltage	Interface	Package Form
SPLD1080	80	80 48 32 0	0 32 48 80	12V	1 bit	COB
SPLD803	240	160	80	16V	1 / 4 bit	COG
SPLD804	288	240 176 160 192	128 128 120 96	18V	4 / 8 bit	COG

* SPLD801B also provides 4-gray-level solution.

Example:

160(Segment) x 128 (Common) 16 Gray Level STN LCM

160(Segment) x 128 (Common) LCD Driver is required

Therefore, the LCD driver configuration will be a single chip solution, that is, one SPLD804 only.

Note that, unlike the SPLD801B 4-gray-level solution, the LCD controller built into the SPL161001A provides a frame rate control function to perform up to 16 gray levels in the STN LCD. In other words, to achieve a gray-level display in the SPL161001A, a specific LCD driver (like the SPLD801) is not required in the system design.

SPLCX01 Series

SUNPLUS also provides two LCD drivers that are built into the LCD controller and the LCD buffer. They are controlled by memory access interface. The SPL161001A can control this type of chip by way of accessing any one of the chip select signals (CS[3..0]). Unfortunately, because the LCD display is done by the SPLCX01 built-in LCD controller, this configuration can not perform gray level display. On the other hand, the LCD controller inside the SPL161001A is disabled in this configuration, and it does not consume CPU computational power.

Body	Channel	Segment	Common	Maximum Voltage	Interface	Package Form
SPLC501	197	132	65	12V	MPU	Cold bump
SPLC701*	174	120	54	12V	MPU	Cold bump

* The SPLC701 also has a built-in Mask ROM that contains a bitmap of 8192 fonts (12x11 bitmap).

Example:

132(Segment) x 65 (Common) Mono STN LCM

132(Segment) x 65 (Common) LCD Driver is required

Therefore, the LCD driver configuration will be a single chip solution, that is, one SPLC501 only.

List of Difference Between the LCD Driver Series

	SPLC56X series	SPLD series	SPLCX01 series
Interface to SPL161001A	LCD Driver Interface	LCD Driver Interface	Memory Access Bus
Internal DC-DC converter	Only SPLC564 built-in	All built-in	All built-in
Internal LCD bias resistor	Not supported	Only SPLD801 SPLD803, SPLD804 built-in	All built-in
Internal LCD Brightness Control	Not supported	Only SPLD801 SPLD803, SPLD804 built-in	All built-in

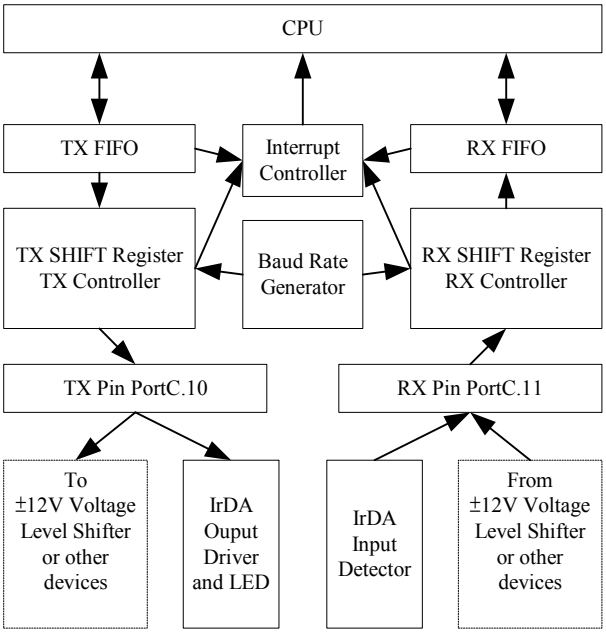
13 UART / IrDA Interface

13.1 Introduction

The UART/ IrDA module built into the SPL161001A performs serial-to-parallel conversion on data received from an external device and it also performs parallel-to-serial conversion on data transmitted to the external device. The transmission and reception paths are individually buffered with internal FIFO memories, allowing up to 16 bytes. This module provides the following features.

- Programmable baud rate.
- Data width can be 5, 6, 7, or 8 bits.
- Parity can be even, odd or disabled.
- Stop bit width can be 1 or 2 bits.
- Individual 8-bit transmit and 11-bit (8-bit data + 3-bit status) receive FIFOs (16 First-in & first-out buffer).
- 16 byte-deep transmit and receive FIFO, respectively.
- Programmable FIFO depth, 1-byte minimum.
- Programmable UART or IrDA SIR input /output (PortC.10 and PortC.11).
- Programmable transmit and receive FIFO interrupt.
- Supports receiving timeout interrupt.
- Frame, Parity, break error detection during reception (error information also saved in FIFO).
- Overrun detection.
- Complies with IrDA SIR physical layer specification.
- Supports high data rate (a division of the system clock).
- Supports normal 3/16 and low power (1.41us) bit duration
- Programmable IrDA TX and RX latency
- Programmable IrDA TX and RX signal polarity
- Supports Loop Back Testing for system diagnosis or mass product testing

13.2 Structure and Block Diagram

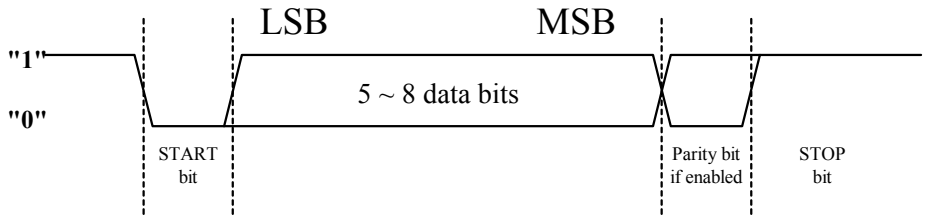


The SPL161001A contains a module that manipulates both UART and IrDA signals. It is impossible to operate transmitting and receiving functions with the UART and IrDA at the same time, since they share the same TX and RX pins. However, it is possible to implement both UART and IrDA communication functions in one product if certain time-sharing methods are applied to the port.

The IrDA interface, compatible with SIR (Serial Infrared) level IrDA, accompanied with UART Interface, is built into the SPL161001A. The SPL161001A and external IrDA transceiver module, connected via PortC10 and PortC11, implement IrDA physical layer. In addition, this built-in half-duplex IrDA interface also provides programmable latency and programmable signal pulse duration.

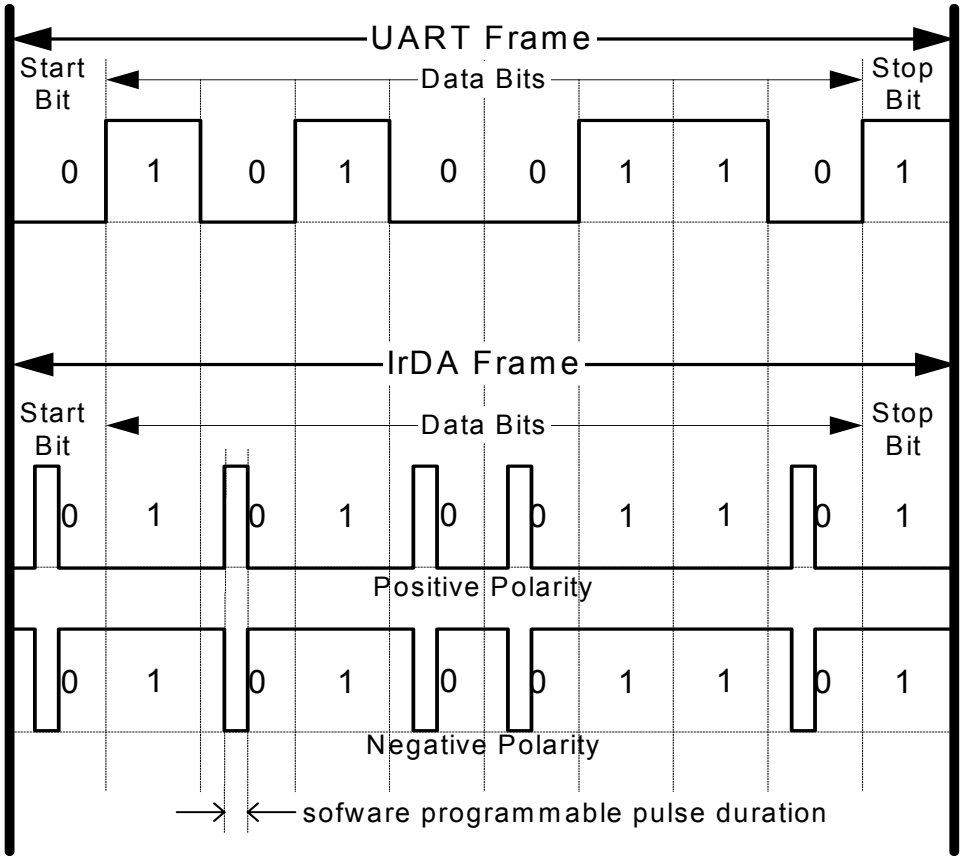
13.3 UART / IrDA SIR Frame Scheme

UART character frame is depicted in the following diagram.



Note that “1” and “0” mean logical level, representing VCC and GND in SPL161001A; on the other hand, “1” and “0” mean –12V and +12V in RS232 compatible line.

IrDA SIR character frame is depicted in the following diagram.



Before starting programming, please make sure that the signal polarity of chosen IrDA module is correct.

According to IrDA physical layer specifications, the upper and lower limits of pulse width duration are different in various signal rates. The following table is the relative specifications for the IrDA SIR physical layer. Programmers must ensure that IrDA SIR specifications are met.

Signal rate	Modulation	Rate tolerance (% of rate)	Pulse duration (Minimum)	Pulse duration (Normal)	Pulse duration (Maximum)
2400 bit/s	RZI	±0.87	1.41 us	78.13 us	88.55 us
9600 bit/s	RZI	±0.87	1.41 us	19.53 us	22.13 us
19200 bit/s	RZI	±0.87	1.41 us	9.77 us	11.07 us
38400 bit/s	RZI	±0.87	1.41 us	4.88 us	5.96 us
57600 bit/s	RZI	±0.87	1.41 us	3.26 us	4.34 us
115200 bit/s	RZI	±0.87	1.41 us	1.63 us	2.23 us

13.4 UART/IrDA Control Pin Configuration

Name	I/O	Description
URX	I	UART/IrDA Reception Pin (shared with PortC11)
UTX	O	UART/IrDA Transmission Pin (shared with PortC10)

13.5 Control registers

UART/IrDA Control Register Summary Table

Name	Address	Description
P_UARTIrDA_Data	0x7100	UART/ IrDA Data Register
P_UARTIrDA_RXStatus	0x7101	UART Reception Status (Error Flag) Register
P_UARTIrDA_Ctrl	0x7102	UART/ IrDA Control Register
P_UARTIrDA_BaudRate	0x7103	UART/ IrDA Baud Rate Setup Register
P_UARTIrDA_Status	0x7104	UART/ IrDA Interrupt Register
P_IrDA_Ctrl	0x7108	IrDA Control Register

P_UARTIrDA_Data **0x7100** **UART / IrDA Data Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	OE	BE	FE	PE	UARTDATA							
Default	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
11	OE	R	Overrun Error (Ready-only) the same as **P_UART_RXStatus bit 3** This bit is set to 1 if data is received and the FIFO is full.	Read 0= Not Occurred Read 1= Occurred
10	BE	R	Break Error (Ready-only) the same as **P_UART_RXStatus bit 2** This bit is set to “1” if a break condition is detected indicating that the received data input is held LOW for more than a full-word transmission time (defined as start, data, parity and stop bits)	Read 0= Not Occurred Read 1= Occurred
9	PE	R	Parity Error (Ready-only) the same as **P_UART_RXStatus bit 1** This bit is set to “1” if the parity of the received data character does not match the parity selected in PSEL Control bit	Read 0= Not Occurred Read 1= Occurred
8	FE	R	Frame Error (Ready-only) the same as **P_UART_RXStatus bit 0** This bit is set to “1” if the received character does not have a valid stop bit (a valid stop bit is 1 bit)	Read 0= Not Occurred Read 1= Occurred
[7:0]	UARTDATA	R/W	UART/IrDA Data Read/Write Register If the FIFO is enabled, the data byte is extracted and a 3-bit status (break, frame and parity) is pushed into the 11-bit wide receiving FIFO. This control register represents the entry to transmitting/receiving FIFO. If the FIFO is disabled, this control register represents the transmitting/receiving hold register.	

Note: Read-only error flags in bit[11..8] has the same function as control register bits located in P_UART_RXStatus.bit[3..0].

P_UARTIrDA_RXStatus 0x7101 UART Reception Error Flag Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	OE	BE	PE	FE
Default	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
3	OE	R/W	Overrun Error This bit is set to “1” if data is received and the FIFO is full.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear this Error Flag
2	BE	R/W	Break Error This bit is set to “1” if a break condition is detected indicating that the received data input is held LOW for more than a full-word transmission time (defined as start, data, parity and stop bits)	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear this Error Flag
1	PE	R/W	Parity Error This bit is set to “1” if the parity of the received data character does not match the parity selected in PSEL Control bit	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear this Error Flag
0	FE	R/W	Frame Error This bit is set to “1” if the received character does not have a valid stop bit (a valid stop bit is 1 bit)	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear this Error Flag

Received status is read from P_UARTIrDA_RXStatus control register. The status information corresponds to the data character read from P_UARTIrDA_Data control register prior to reading P_UARTIrDA_RXStatus control register. Writing “1” to corresponding register bit will clear the frame, parity break and overrun error.

Note that, the received data byte must be read first from P_UARTIrDA_Data before reading the corresponding error status from P_UARTIrDA_RXStatus. This read sequence cannot be reversed since the status register P_UARTIrDA_RXStatus is updated only when a read operation is performed on the P_UARTIrDA_Data control register.

P_UARTIrDA_Ctrl **0x7102** **UART / IrDA Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RXIE	TXIE	RTIE	UEN	RST	UCE	SLT	-	-	WLSEL		FEN	SBSEL	PSEL	PEN	SB
Default	0	0	0	0	0	0	0	-	-	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	RXIE	R/W	Receiving Interrupt Enable **For FIFO is enabled (16-byte deep,** FEN=1): If this bit is set to “1”, and the data number in receive FIFO is more or equal to 8, hardware will issue an IRQ3 or FIQ to CPU. **For FIFO is disabled (1-byte deep,** FEN=0**):** If this bit is set to “1”, and the receive buffer is just held one new-coming data, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to “0”, this receiving interrupt will be masked. To select between IRQ3 and FIQ, please refer to ***Chapter Interrupt***.	0= Disabled 1= Enabled
14	TXIE	R/W	Transmitting Interrupt Enable **For FIFO is enabled (16-byte deep,** FEN=1): If this bit is set to “1”, and the data number in transmit FIFO is less or equal to 8, hardware will issue an IRQ3 or FIQ to CPU. **For FIFO is disable (1-byte deep,** FEN=0**):** If this bit is set to “1”, and the transmit buffer is empty, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to “0”, this receiving interrupt will be masked. To select between IRQ3 and FIQ, please refer to ***Chapter Interrupt***.	0= Disabled 1= Enabled
13	RTIE	R/W	Receiving Timeout Interrupt Enable If this bit is set to “1”, and when the receive FIFO is not empty and no further data is received over a 32-bit period, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to “0”, this receiving interrupt will be masked. To select between IRQ3 and FIQ, please refer to ***Chapter Interrupt***.	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
12	UEN	R/W	UART Enable If this bit is set to “1”, the UART Interface is enabled. Data transmission and reception occurs for either UART signals or IrDA SIR signals based on the setting of Control bit IEN (P_IrDA_Ctrl.bit9). For example, to enable IrDA control module, both UEN and IEN bit should be set to “1”, to enable UART control module, UEN should be set to “1”, but IEN should clear to “0”. Note that when UART/IrDA module is enabled, IOC11(URX) is input floating and IOC10(UTX) is buffer output.	0= Disabled 1= Enabled
11	RST		Reset the UART/IrDA module	Write “1” to generate a single internal reset pulse to reset the UART/IrDA module
10	UCE		Enable UART/IrDA clock generator The internal UART/IrDA clock is not generated to save power until it is enabled.	0= Disabled 1= Enabled
9	SLT	R/W	Self-Loop Test Enable For system diagnosis or mass product testing, UART/IrDA self-loop test is an easy way to test IC internal logic, PCB layout and external components at one time. To enable self-loop of UART/IrDA, this control bit must be set to “1”. In normal operation, please clear this bit to “0”	0= Disabled 1= Enabled
[8:7]			Reserved	
[6:5]	WLSEL	R/W	Word Length Definition Indicate number of data bits transmitted or received in a frame.	00= 5 bits 01= 6 bits 10= 7 bits 11= 8 bits
4	FEN	R/W	FIFO Buffer Enable/Disable Setting this bit to “1” will enable the 16-byte deep FIFO buffer on both transmitting and receiving operation. When clear this bit to “0”, the FIFO becomes 1-byte deep hold registers.	0= Disabled 1= Enabled
3	SBSEL	R/W	Stop Bit Size Selection When this bit is set to “1”, two stop bits are	0= 1 Stop Bit 1= 2 Stop Bit

Bit	Function	Type	Description	Condition
			transmitted at the end of the frame. The receiving logic does not check for two stop bits being received.	
2	PSEL	R/W	Parity Selection If this bit is set to “1”, even parity generation and checking are performed during transmission and reception, which checks for an even number of 1s in data and parity bits. When cleared to “0” the odd parity is performed to check for an odd number of 1s. This bit has no effect when parity is disabled by PEN Control bit is cleared to “0”	0= Odd Parity (if PEN= 1) 1= Even Parity (if PEN= 1)
1	PEN	R/W	Parity Enable If this bit is set to “1”, parity checking and generation is enabled, else parity is disabled and no parity bit added to the data frame.	0= Disabled 1= Enabled
0	SB	R/W	Send Break If this bit is set to “1”, a low level is continuously outputted on the TX output pin after completing the current character transmission. This bit must be asserted for at least one complete frame period in order to generate a break condition. The transmit FIFO contents remain ineffective during a break condition. For normal usage, this bit must be cleared to “0”.	0= Normal Operation 1= Send Break Signal

P_UARTIrDA_BaudRate 0x7103 UART / IrDA Baud Rate Setup Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	UARTBUD															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	UARTBUD	R/W	UART/ IrDA Baud Rate Divisor	

The Baud Rate Divisor value is calculated as follows

UARTBUD = [SYSCLK / (16 X Baud Rate)] –1

For example, if the system clock is 51.6096MHz, to obtain 115200bps Baud rate, the divisor must be set to

UARTBUD = [51609600 / (16 X 115200)] –1 = 27 = 0x001B

Baud Rate Setting Table for System Clock =51.6096 MHz

Programmable divisor (value on UARTBUD)	Available UART Baud Rate (Bit Per Second)	Available IrDA Baud Rate (Bit Per Second)
0x0006	460,800	-
0x000D	230,400	-
0x001B	115,200	115,200*
0x0029	76,800	-
0x0037	57,600	57,600
0x0053	38,400	38,400
0x006F	28,800	28,800
0x00A7	19,200	19,200
0x00DF	14,400	-
0x014F	9,600	9,600
0x053F	2,400	2,400
0xA7F	1,200	-

*According to IrDA SIR specifications, the maximum IrDA baud rate is 115200bps.

P_UARTIrDA_Status 0x7104 **UART / IrDA Status Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RXIF	TXIF	RTIF	-	-	-	-	-	TXEF	RXFF	TXFF	RXEF	BY	-	-	-
Default	0	0	0	-	-	-	-	-	1	0	0	1	0	-	-	-

Bit	Function	Type	Description	Condition
15	RXIF*	R	Receive Interrupt Flag **For FIFO is enabled (16-byte deep,** FEN=1**):** This bit is set to “1” by hardware if the receive interrupt enable bit is set to “1”, and the data number in receive FIFO is more or equal to 8. This bit is cleared to “0” by hardware if data number in receive FIFO is less than 8. In other word, receive interrupt is cleared by reading data from the receive FIFO until it becomes less than 8 data. **For FIFO is disabled (1-byte deep,** FEN =0**):** This bit is set to “1” by hardware if the receive interrupt enable bit is set to “1”, and the data is received completely. The receive interrupt is cleared by performing a single read of receive data register (P_UARTIrDA_Data).	If FIFO is enabled: Read 0= no. in RX FIFO < 8 Read 1= no. in RX FIFO >= 8 If FIFO is disabled: Read 0= RX Data is not Ready Read 1= RX Data is Ready

Bit	Function	Type	Description	Condition
14	TXIF*	R	Transmit Interrupt Flag **For FIFO is enabled (16-byte deep,** FEN=1): This bit is set to "1" by hardware if the transmitting interrupt enable bit is set to "1", and the data number in transmit FIFO is less or equal to 8. This bit is cleared to "0" by hardware if data number in transmit FIFO is more than 8. In other word, transmit interrupt is cleared by filling data to transmit FIFO until it becomes more than 8 data. **For FIFO is disabled (1-byte deep,** FEN=0**):** This bit is set to "1" by hardware if the transmitting interrupt enable bit is set to "1", and there is no data present in the transmitter. This flag is cleared to "0", by performing a single write to the transmit data register (P_UARTIrDA_Data).	If FIFO is enabled: Read 0= no. in TX FIFO > 8 Read 1= no. in TX FIFO <= 8 If FIFO is disabled: Read 0= TX Buffer is not Ready Read 1= TX Buffer is Ready
13	RTIF*	R	Receive Timeout Interrupt Flag This bit is set to "1" by hardware if the receiving timeout interrupt is asserted. (When receive FIFO is not empty and no further data is received over a 32-bit period). This interrupt flag is cleared to "0" by hardware when the FIFO becomes empty through reading all the data (or by reading the holding register)	Read 0= Not Occurred Read 1= Occurred
[12:8]			Reserved	
7	TXEF	R	Transmit FIFO Empty Flag The meaning of this bit depends on the state of the FEN control bit. **For FIFO is enabled (16-byte deep,** FEN=1): This bit is set to "1" when the transmit FIFO is empty. **For FIFO is disabled (1-byte deep,** FEN=0**):** This bit is set to "1" when the transmitting buffer is empty. Note that this flag is read-only, hardware will set or clear this flag automatically.	If FIFO is enabled: Read 0= no. in TX FIFO > 0 Read 1= no. in TX FIFO = 0 If FIFO is disabled: Read 0= TX buffer is not empty Read 1= TX buffer is empty
6	RXFF	R	Receive FIFO Full Flag The meaning of this bit depends on the state of the FEN control bit. **For FIFO is enabled (16-byte deep,** FEN=1):	If FIFO is enabled: Read 0= no. in RX FIFO < 16 Read 1= no. in RX FIFO = 16 If FIFO is disabled:

Bit	Function	Type	Description	Condition
			This bit is set to "1" when the receive FIFO is full. **For FIFO is disabled (1-byte deep,** FEN=0**):** This bit is set to "1" when the receiving buffer is full. Note that this flag is read-only, hardware will set or clear this flag automatically.	Read 0= RX buffer is not full Read 1= RX buffer is full
5	TXFF	R	Transmit FIFO Full Flag The meaning of this bit depends on the state of the FEN control bit. **For FIFO is enabled (16-byte deep,** FEN=1): This bit is set to "1" when the transmit FIFO is full. **For FIFO is disabled (1-byte deep,** FEN=0**):** This bit is set to "1" when the transmitting buffer is full. Note that this flag is read-only, hardware will set or clear this flag automatically.	If FIFO is enabled: Read 0= no. in TX FIFO < 16 Read 1= no. in TX FIFO = 16 If FIFO is disabled: Read 0= TX buffer is not full Read 1= TX buffer is full
4	RXEF	R	Receive FIFO Empty Flag The meaning of this bit depends on the state of the FEN control bit. **For FIFO is enabled (16-byte deep,** FEN=1): This bit is set to "1" when the receive FIFO is empty **For FIFO is disabled (1-byte deep,** FEN=0**):** This bit is set to "1" when the receiving buffer is empty. Note that this flag is read-only, hardware will set or clear this flag automatically.	If FIFO is enabled: Read 0= no. in RX FIFO > 0 Read 1= no. in RX FIFO = 0 If FIFO is disabled: Read 0= RX buffer is not empty Read 1= RX buffer is empty
3	BY	R	BUSY If this bit is set to "1", the UART or IrDA module is busy in transmitting data. This bit remains set until the complete byte, including all the stop bits, has been sent from the shift register. Note that this flag is read-only, hardware will set or clear this flag automatically.	0= Ready 1= Busy
[2:0]			Reserved	

*The above three interrupt flags are combined into a single interrupt flag on P_INT_Status1.bit11 by an OR logic. Programmers should determine which interrupt occurs from this control register (P_UARTIrDA_Status.bit[15..13]).

** Also note that TX interrupt or RX interrupt flag is “clear” by the write and read operation on the corresponding FIFOs, and they are not cleared by any specified control bit.

P_IrDA_Ctrl **0x7108** **IRDA Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TXTXLT		TXRXLT		TPOL	RPOL	IEN	ILP	IrDAPD							
Default	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:14]	TXTXLT	R/W	IrDA SIR Transmit-to-Transmit Latency Setup IrDA transmit-to-transmit latency is the time interval starting from the time of transmitter stopped sending light pulses to the time of transmitter starts sending first light pulse of next transmitting data.	00= 0-bit period 01= 2-bit period 10= 4-bit period 11= 8-bit period
[13:12]	TXRXLT	R/W	IrDA SIR Transmit-to-Receive Latency Setup[1] Because IrDA receiver will receive its own signals during transmitting process. A proper latency is necessary to prevent incorrect response. IrDA receiver latency is the time interval starting from the time of transmitter stopped sending light pulses to the time of receiver guaranteed ready to receive data.	00= 4-bit period 01= 8-bit period 10= 16-bit period 11= 31-bit period
11	TPOL	R/W	IrDA Transmission Polarity Selection Please refer to waveform depicted on ***Section: UART/IrDA SIR Frame Scheme*** for detail.	0: Positive Polarity 1: Negative Polarity
10	RPOL	R/W	IrDA Reception Polarity Selection Please refer to waveform depicted on ***Section: UART/IrDA SIR Frame Scheme*** for detail.	0: Positive Polarity 1: Negative Polarity
9	IEN	R/W	IrDA SIR Enable If this bit is set to “1”, the IrDA SIR encoder is enabled. This bit has no effect if the UART is not enabled by UEN Control bit set to “1”. When the IrDA SIR encoder is enabled, IrDA data is transmitted and received on PortC10 and PortC11, respectively. These two pins cannot be used as GPIO any more.	0= Disabled 1= Enabled
8	ILP	R/W	IrDA SIR Low Power Mode Selection	0= low level pulse width is fixed to

Bit	Function	Type	Description	Condition
			If this bit is cleared to "0", low level data bits are transmitted as an active high pulse with a width of 3/16th of the bit period. If this bit is set to "1", low level data bits are transmitted with a pulse width which is 3 times of the period input signal defined in IrDAPD control bits, regardless of the selected bit rate. Note that setting this bit consumes less power, but the transmission distance may be shorter.	3/16th of corresponding bit period 1= low level pulse width depends on the setting in IrDAPD control bits
[7:0]	IrDAPD	R/W	IrDA Power Modulation Selection These control bits are valid only when ILP control bit is set to "1". This is an 8-bit read/write register that stores the low-power counter divisor value which generates the IrDA pulse signal by dividing down from the 3.6864MHz. Refer to following formula[2] for selection	

[1]Due to its optical nature, the IrDA transceiver module will receive its own signals during the transmitting process, thus a proper latency is necessary to prevent incorrect response. IrDA receiver latency is the time interval starting from the time transmitter stops to sending infrared pulses to the time receiver guaranteed ready to receive data.

[2]Formula:

$$\text{Divisor} = (3686400 / F_{IrIPBaud16})-1$$

where $F_{IrIPBaud16}$ is nominally 1.8432MHz.

The divisor must be chosen so that 1.42MHz < FIrLPBaud16 < 2.12MHz which results in a low power pulse duration of 1.41~2.11us (three times the period of IrLPBaud16).

For example, to set the IrDA pulse width 2.0us, first calculate the FIrLPBaud16 = 2.0us = 0.5MHz.
So, divisor = (3686400/500000) – 1 = 6.3728.
So the IrDAPD value can be 6 or 7. Then the IrDA pulse width is approximately 2.0us.

The minimum frequency of IrLPBaud16 ensures that pulses less than one period of IrLPBaud16 are rejected. However, pulses greater than 1.4us are accepted as valid pulses. Note that the smaller the pulse width is, the less the power consumption is, but the transmission distance will be shorter. Programmers have to look up the specifications of the IrDA tranceiver module to do the proper setting.

13.6 Program Examples

This sample code is a self-loop test of the UART module. To make it works, IOC11 and IOC10 should be interconnected.

```
                    r1 = 0x0001
                    [P_UARTIrDA_BaudRate] = r1      // Baud Rate = 115,200 bps
                    r1 = 0x9070
                    [P_UART_Ctrl] = r1              //UART Enable, N-8-1, FIFO Enable
                    r4 = 0x0050
                    [P_UARTIrDA_Data] = r4
                    r4 + = 1                        //1st TX FIFO data = 0x50
                    r1 = 7
L_FillRXFIFO?:
                    [P_UARTIrDA_Data] = r4          //2nd, 3rd, 4th, 5th, 6th, 7th, 8th TX FIFO data
                    r1 = r1 – 1                     // = 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57
                    r4 = r1 + 1
                    jnz     L_FillRXFIFO?
L_WaitRX_FIFO_Empty_INT?:                           // If data number of RX FIFO is more than or
                    r1 = [P_UARTIrDA_Status]        // equal to 8, RX interrupt flag will be set to "1"
                    jpl     L_WaitRX_FIFO_Empty_INT?

                    r4 = 0x50
L_Wait_RX_FIFO_Empty?:                              // Verify all RX FIFO data
                    r1 = [P_UARTIrDA_Status]
                    r1 = r1 & 0x0010
                    jnz     L_UARTSelfLoopTestOK?
                    r3 = [P_UARTIrDA_Data]
                    cmp    r3,r4
                    jne     L_UARTSelfLoopTestError?
                    r4 = r4 + 1
                    Jmp    L_Wait_RX_FIFO_Empty?
L_UARTSelfLoopTestOK?:
                    ……
                    ……
                    ……
L_UARTSelfLoopTestError?:
                    ……
                    ……
                    ……
```

14 Serial Peripheral Interface (SPI)

14.1 Introduction

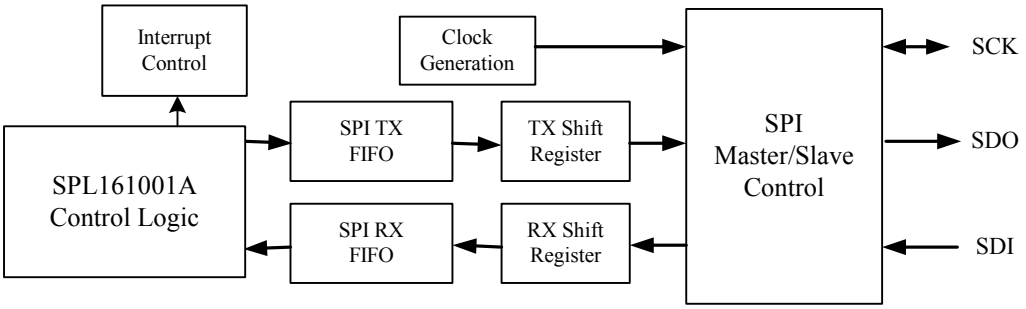
A Serial Peripheral Interface (SPI) controller is built into the SPL161001A series to facilitate communicating with other devices and components. There are three control signals on SPI, SPICLK (SCK), SPIDI (SDI), and SPIDO (SDO). These signals are shared with PortB11,PortB10 and PortB9. While the SPI module is enabled by corresponding control bit, these three pins cannot be GPIOs. In other words, any setting on corresponding GPIO control register will have no effect.

The SPI provides the following features:

- Supports single byte and consecutive bytes transferring.
- Supports overrun error indication
- Supports transmitting / receiving interrupt request
- Supports 16-byte deep transmit / receive FIFO buffers
- Programmable FIFO interrupt level
- Programmable phase and polarity of master clock
- Selectable data sampling time (end or middle of clock period)
- Programmable master SCK clock frequency: System Clock / 4, /8, /16, /32, /64, /128

14.2 Structure

The following is a function diagram of the SPI module:



14.3 SPI Control Pin Configuration

Name	I/O	Description
SPICK	I/O	Serial Peripheral Interface, Clock Pin (Shared with GPIO PortB11)
SPIDO	O	Serial Peripheral Interface, Data Out Pin (Shared with GPIO PortB10)
SPIDI	I	Serial Peripheral Interface, Data In Pin (Shared with GPIO PortB9)

14.4 Master Mode

As in the master mode, the shifting clock (SPICLK) is generated by the SPL161001A. There are two control bits to control the clock phase and polarity. The transmission starts immediately from a register write to the SPIBUF control register.

After software writes one byte through the SPIBUF control register, the data is latched into its internal transmitting buffer. If the shift register is not shifting data, the data will be loaded to the shift register and start transmitting at the next SCK phase. On the other hand, if the shift register is busy shifting data (SPITXBF flag is set), the new data will not be shifted out until the present byte has been shifted out.

The SPL161001A provides both transmit and receive FIFOs. If the transmit FIFO declines below a specified level, an interrupt will be asserted. Similarly, if the receive FIFO exceeds a specified level, an interrupt will be asserted. The FIFO interrupt levels are programmable.

The SPI shifts the data from MSB to LSB through the SDO pin. The 8-bit data is shifted out after 8 SCK cycles. At the same time, the data is also shifted through the slave device SDI pin. When each 8-bit transfer is completed, the SPIIF flag bit will be set. In addition, an SPI interrupt will be generated if the SPIIRQEN bit is set.

In contrast, while the SPL161001A receives one byte successfully, the received data will be latched into the receiving buffer. At that same time, the SPIIF flag bit will be set and an SPI interrupt will be issued to CPU if the SPIIREQEN is enabled.

The following diagram depicts the timing scheme in the SPI master mode for different operation types (polarity control bit equals “1” or “0”, phase control bit equals “1” or “0”). Note that for better sampling accuracy, the sample strobe is automatically set according to the clock phase.

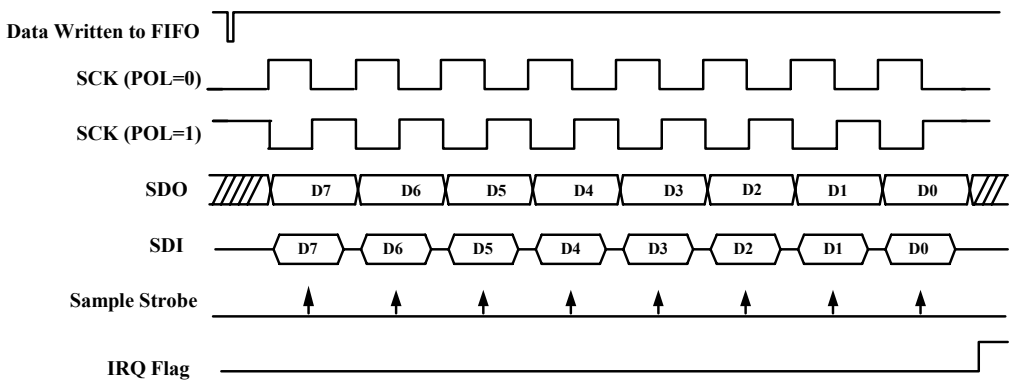


Fig1. Master Mode, PHASE=0

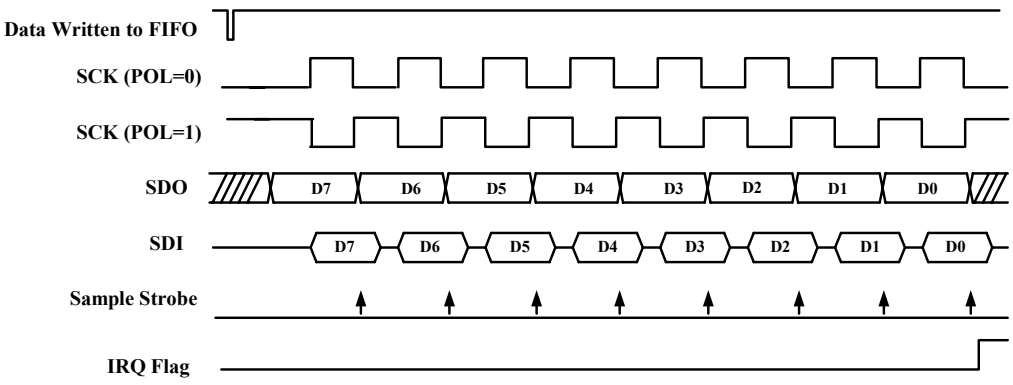


Fig2. Master Mode, PHASE=1

The SPI driving scheme might vary with different SPI slave devices. In order to make the reliable transfer between the SPL161001A and a slave device, please check the timing scheme of the SPI slave device first, and configure the correct SPI driving scheme in the SPL161001A (P_SPI_Ctrl.bit[5,4,3]).

14.5 Control Registers

SPI Control Register Summary Table

Name	Address	Description
P_SPI_Ctrl	0x7140	SPI Control Register
P_SPI_TXStatus	0x7141	SPI Transmitting Status Register
P_SPI_TXData	0x7142	SPI Transmitting Buffer Register
P_SPI_RXStatus	0x7141	SPI Status Receiving Register
P_SPI_RXData	0x7142	SPI Receiving Buffer Register

P_SPI_Ctrl **0x7140** **SPI Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SPIEN	-	-	-	RST	-	-	MOD	-	-	SCKPHA	SCKPOL	-	SCKSEL		
Default	0	-	-	-	0	-	-	0	-	-	0	0	-	0	0	0

Bit	Function	Type	Description	Condition
15	SPIEN	R/W	SPI enable If this bit is set to “1”, PortB[11:9] becomes SPI Interface. These pins cannot be used as GPIO. Therefore, any further corresponding setting on GPIO PortB[11:9] will be no effect. In Master mode: When SPI is enabled, IOB11 and IOB10 are buffer output and IOB9 is floating input.	0= Disabled 1= Enabled
[14:12]			Reserved	
11	RST	W	SPI Module Reset	Write “1” to this bit to generate an internal reset pulse to reset the SPI module. The register settings will not be reset.
[10:9]			Reserved	
8	MOD	R/W	SPI Mode	0= Master mode 1= Slave mode
[7:6]			Reserved	
5	SCKPHA	R/W	SPI clock phase	Refer to timing scheme in previous section
4	SCKPOL	R/W	SPI clock polarity	Refer to timing scheme in previous section
3			Reserved	
[2:0]	SCKSEL	R/W	Master mode clock selection	000= SYSCLK / 4 001= SYSCLK/ 8 010= SYSCLK/ 16

Bit	Function	Type	Description	Condition
				011= SYSCLK/ 32 100= SYSCLK / 64 101= SYSCLK / 128 110 = SYSCLK / 128 111 = SYSCLK / 128

P_SPI_TXStatus **0x7141** **SPI Transmission Status Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TXIF	TXIEN	-	-	-	TXSTRT	TXFLEV					TXFLAG				
Default	0	0	-	-	-	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	TXIF	R/W	SPI TX Interrupt flag. In Master Mode: This bit is set to “1” by hardware when the transmit interrupt level is reached. Write “1” to this control bit will clear this flag.	Read 0= Not Occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
14	TXIEN	R/W	SPI TX Interrupt Enable In Master Mode: If this bit is set to “1”, and SPI interrupt (when 8-bit TX data has been shifted out through SDO) occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to “0”, this interrupt will be masked. To select between IRQ3 and FIQ, please refer to ***Chapter Interrupt***.	0 = Disabled 1 = Enabled
[13:11]			Reserved	
10	TXSTRT	W	Transmission start If the SPI is enabled, only after this bit is written “1” the communication will start.	Master mode: 1= start transmission 0= no effect Slave mode: 1= start waiting Master CLOCK 0= no effect
[9:5]	TXFLEV	R/W	TX FIFO interrupt level If the contents in the TX FIFO (TXFLAG) ≦ TXFLEV, TXIF will be asserted. If the TX interrupt is enabled, the interrupt will be issued.	Valid values are 0x0001 to 0x1000.
[4:0]	TXFLAG	R	TX FIFO content level TXFLAG value shows the number of bytes in the TX FIFO. If TXFLAG is 0x0000, the FIFO is empty. If TXFLAG is 0x1000, the FIFO is full.	

P_SPI_TXData **0x7142** **SPI Transmission Buffer Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	TXDATA							
Default																

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	TXDATA	R/W	Write data to SDO pin	

P_SPI_RXStatus **0x7143** **SPI Reception Status Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RXIF	RXIEN	STDC	MTDC	FUL	FERR	RXFLEV					RXFLAG				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	RXIF	R/W	SPI RX Interrupt flag. In Master Mode: This bit is set to “1” by hardware when the 8-bit RX data has been shifted in through SDI. Writing “1” to this control bit will clear this flag.	Read 0= Not Occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
14	RXIEN	R/W	SPI RX Interrupt Enable In Master Mode: If this bit is set to “1”, and SPI interrupt (when 8-bit RX data has been shifted in through SDI) occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to “0”, this interrupt will be masked. To select between IRQ3 and FIQ, please refer to ***Chapter Interrupt***.	0 = Disabled 1 = Enabled
13	STDC	R/W	TX Don’t-care for Slave Mode If this bit is written 1 and some data is written to TX FIFO, the data will be sent. If the FIFO is empty, “0x00” will be automatically sent.	1: TX don’t care 0: Normal
12	MTDC	R/W	TX Don’t-care for Master Mode If this bit is written 1 and some data is written to TX FIFO, the data will be sent. If the FIFO is empty, “0x00” will be automatically sent.	1: TX don’t care 0: Normal

Bit	Function	Type	Description	Condition
11	FUL	R	FIFO full If the RX FIFO is full, this flag will be set. If the FIFO is read, this flag will be clear.	1: Full 0: Not full
10	FERR	R/W	FIFO overwritten If the RX FIFO is full, and there is a next data coming in, the last input data will be overwritten and this flag will be set. Write 1 to clear this bit.	Read 1: Overwritten Read 0: Not overwritten Write 1 to clear
[9:5]	RXFLEV	R/W	RX FIFO interrupt level If the contents in the RX FIFO (RXFLAG) ≧ RXFLEV, RXIF will be asserted. If the RX interrupt is enabled, the interrupt will be issued.	Valid values are 0x0001 to 0x1000.
[4:0]	RXFLAG	R	RX FIFO content level RXFLAG value shows the number of bytes in the RX FIFO. If RXFLAG is 0x0000, the FIFO is empty. If RXFLAG is 0x1000, the FIFO is full.	

P_SPI_RXData **0x7144** **SPI Reception Buffer Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	RXDATA							
Default																

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	RXDATA	R/W	Read data from SDI pin	

14.6 Program Examples

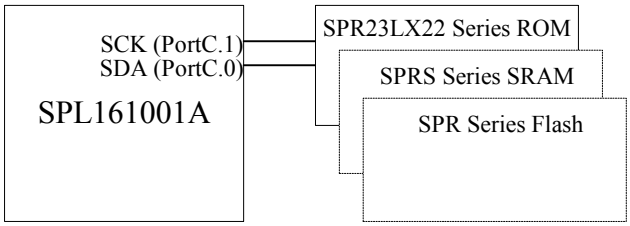
```
F_SPISelfLoopTest:
                    Int off
                    r1=0x0080
                    [P_SPI_Ctrl]=r1          //Enable SPI
                    r1=0xA000
                    [P_SPI_TXStatus]=r1      //Clear Flag
                    [P_SPI_RXStatus]=r1
L_CheckTXBufferFull?:
                    r1 = [P_SPI_TXStatus]
                    r1 = r1 and 0x1000
                    jnz    L_CheckTXBufferFull?
                    r2 = 0x0055
```

```
                    [P_SPI_TXData]=r2                   // Output 0x55 to SPI
L_WaitSPIRXReady?:
                    r1=[P_SPI_RXStatus]
                    jpl     L_WaitSPIRXReady?
                    cmp   r2,[P_SPI_RXData]
                    jne   L_Error?
                    DisplayResultCode       D_OK
                    retf
L_Error?:
                    DisplayResultCode       D_NG
                    Retf                                //Note that DisplayResultCode is a MACRO
```

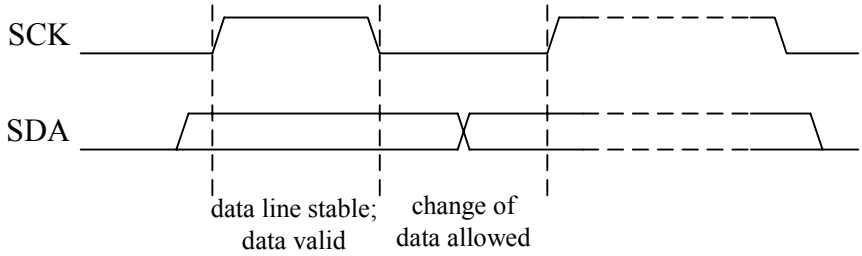
15 Serial Interface

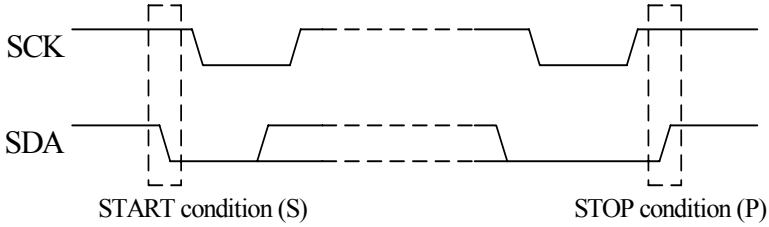
15.1 Introduction

To easily connect Sunplus serial memory products (SRAM, ROM, Flash), the SPL161001A implements a serial interface. The serial interface consists of two lines: SCK (shared with PortC.1) and SDA (shared with PortC.0).

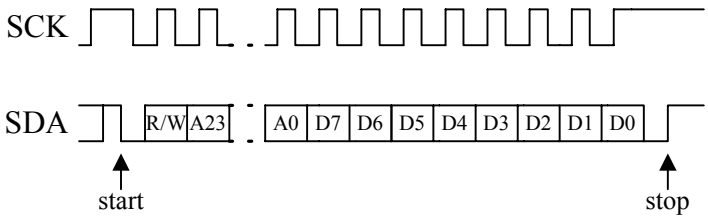


The timing diagrams of serial interface are as the following.

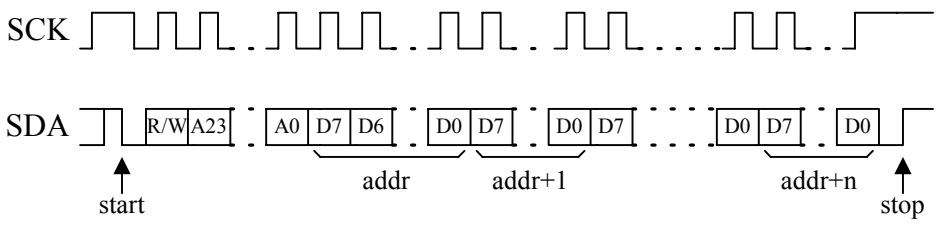




The single-byte read/write timing diagram is as follows:



The continuous read/write timing diagram is as follows:



15.2 Control Registers

Serial Interface Control Register Summary Table

Name	Address	Description
P_SIO_Ctrl	0x71C0	Serial I/F Control Register
P_SIO_AddrML	0x71C1	Serial I/F Address Middle/Low Byte Register
P_SIO_AddrH	0x71C2	Serial I/F Address High Byte Register
P_SIO_Start	0x71C3	Serial I/F Start Transmission
P_SIO_Stop	0x71C4	Serial I/F Stop Transmission
P_SIO_Write_EN	0x71C5	Serial I/F Write Enable
P_SIO_Read_EN	0x71C6	Serial I/F Read Enable

P_SIO_Ctrl **0x71C0** **Serial I/F Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	SEN	MOD	RWB	SEL1	SEL0	AS1	AS0
Default	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
6	SIOEN	R/W	Serial Interface enable If this bit is written “1”, the PortC.1 and PortC.0 is no longer GPIO.	1: Enable 0: Disable
5	MODE	R/W	Serial Interface mode selection	1: Write 0: Read
4	RWBIT	R/W	Selection of whether the Read/Write bit is applied If this bit is written “0”, the R/W bit will be applied in the transmission packer. Also, all Sunplus’ memory products need the R/W bit to distinguish from reading or writing.	1: No R/W bit is applied 0: R/W bit is applied

Bit	Function	Type	Description	Condition
[3:2]	CLKSEL	R/W	SCK clock frequency selection	00: SYSCLK/16 01: SYSCLK/4 10: SYSCLK/8 11: SYSCLK/32
[1:0]	ADDRSE	R/W	Address length selection Different Sunplus' memory products may require different address length. Please check the memory products' datasheet for proper setting.	00: Address=16 bit(A15~A0) 01: No Address 10: Address=8 bit(A7~A0) 11: Address=24 bit(A23~A0)

Please refer to the following tables for SUNPLUS serial memory product address lengths.

	Model Number
8-bit	N/A
16-bit	SPRS256
24-bit	SPRS512, SPR1024, SPR4096, SPR23L822, SPR23L1622

P_SIO_AddrML **0x71C1** **Serial I/F Address Middle/Low Byte Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	Middle/Low Byte of the Address															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	AML	R/W	Middle byte and Low byte of Serial Interface Address	

P_SIO_AddrH **0x71C2** **Serial I/F Address High Byte Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	High Byte of the Address															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]		R/W	High byte of Serial Interface Address	

P_SIO_Start **0x71C3** **Serial I/F Start Transmission**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	Write Any Value to Start Transmission															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	START	W	Write any value to this port to start transmission	Write any value

P_SIO_Stop **0x71C4** **Serial I/F Stop transfer**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	Write Any Value to Stop Transmission															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	STOP	W	Write any value to this port to stop transmission	Write any value

P_SIO_Write_EN **0x71C5** **Serial I/F Write Enable**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	WEN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	WEN	W	Write the data to this register, and the data will be transmitted from SDA (PortC.0).	

P_SIO_Read_EN **0x71C6** **Serial I/F Read Enable**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	REN															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	REN	R	Read the data received from SDA (PortC.0).	

15.3 Program Example

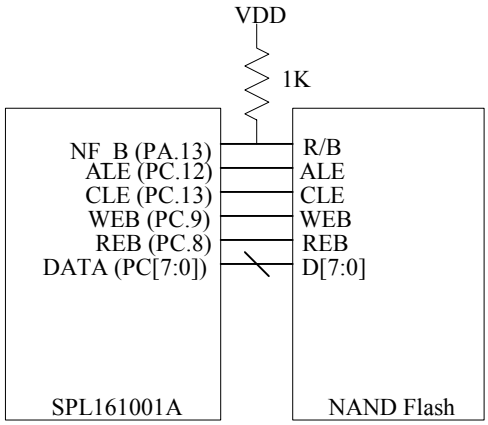
```
F_EraseTest:    r1=0x0068          // Erase SPR1024 and check erase data
                [P_SIO_Ctrl]=r1    // Serial interface enable, write mode, R/W
                r1 = 0x6000        // applied, SYSCLK/8, 16-bit address
                [P_SIO_AddrML] = r1
```

```
                    [P_SIO_Start] = r1
                    r1 = 0x0080
                    [P_SIO_Write_EN] = r1
L_EraseWait?:
                    r1 = [P_SIO_Ctrl]
                    jpl     L_EraseWait?
                    [P_SIO_Ctrl] = r1
                    [P_SIO_Stop] = r1

                    r1 = 0x004B
                    [P_SIO_Ctrl] = r1
                    r1 = 0x0000                 // Serial interface enable, write mode, R/W
                    [P_SIO_AddrML] = r1         // applied, SYSCLK/8, 24-bit address
                    r1 = 0x080
                    [P_SIO_AddrH]=r1

                    DS = 3
                    r2 = 0x00
                    [P_SIO_Start] = r1
L_ReadWait?:
                    r1 = [P_SIO_Ctrl]
                    jpl     L_ReadWait?
                    [P_SIO_Ctrl] = r1
                    r1 = [P_SIO_Read_EN]
                    r1 &= 0x0ff
                    DS:[r2++] = r1
                    cmp   r2, 0x8000
                    jne     L_ReadWait?
                    [P_SIO_Stop] = r1
                    jmp    $
```

16 NAND Flash Interface

16.1 Introduction

As a mass storage device, NAND-gate flash becomes more and more popular nowadays for its large capacity and relatively low price. To extend the SPL161001A application field, a NAND flash interface is incorporated. The SPL161001A provides easy-to-use control registers to generate read/write signals to access NAND flash. In addition, a DMA channel is dedicated to NAND Flash access to speed up data movement. A sample connection diagram between the SPL161001A and NAND flash is depicted as follows:



The NAND Flash interface provides the following features:

- Control and data pins shared with GPIO
- Programmable 8-bit or 16-bit mode
- Programmable setup/hold timing for accessing
- Automatic ECC calculation and verification

16.2 Control Registers

NAND Flash Interface Control Registers Summary

Name	Address	Description
P_NF_Ctrl	0x7050	NAND Flash Control Register
P_NF_Command	0x7051	NAND Flash Command Register
P_NF_Address	0x7052	NAND Flash Address Register
P_NF_DataW	0x7053	NAND Flash Data Write Register
P_NF_DataR	0x7054	NAND Flash Data Read Register
P_NF_Data	0x7055	NAND Flash Data Register
P_ECC_Ctrl	0x7056	ECC Control Register
P_ECC_LPRL	0x7057	ECC Line parity LSB Register

Name	Address	Description
P_ECC_LPRH	0x7058	ECC Line parity MSB Register
P_ECC_CPR	0x7059	ECC Column parity Register
P_ECC_LPRCKL	0x705A	ECC Line parity Check LSB Register
P_ECC_LPRCKH	0x705B	ECC Line parity Check MSB Register
ECCCPCKR	0x705C	ECC Column parity Check Register
P_ECC_Err0	0x705D	ECC Error Flag0
P_ECC_Err1	0x705E	ECC Error Flag1
P_DMA_AddrL	0x7380	DMA Start Address LSB
P_DMA_AddrH	0x7381	DMA Start Address MSB
P_DMA_Size	0x7382	DMA SIZE
P_DMA_Write	0x7383	DMA Write Action
P_DMA_Read	0x7384	DMA Read Action
P_DMA_WriteCtrl	0x7385	DMA Write Control
P_DMA_ReadCtrl	0x7386	DMA Read Control

P_NF_Ctrl **0x7050** **NAND Flash Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NFBF	8/16	NFEN	-	-	NFCTRL										
Default	0	0	0	-	-	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	NFBF	R	Busy status (NF_B)	1: Ready 0: Busy
14	NF8_16	R/W	Access NAND flash memory by 8 bit or 16 bit data bus	0: 8 bit 1: 16 bit
13	NFEN	R/W	NAND flash enable control	0: Disable 1: Enable
[12:11]			Reserved	
[10:0]	NFCTRL	R/W	Adjustable setup/hold timing Except tMEM, all the setup timings are SYSCLK_cycle X [Value+1]; tMEM is SYSCLK_cycle X [Value+2]. For example, if bit[10:8] are set to 0x111, then the tMEM = SYSCLK_cycle x 8.	[10:8]: tMEM [7:6]: tRH, REBhigh pulse [5:4]: tRA, access time [3:2]: tWH, WEB high pulse [1:0]: tWP, WEB low pulse

P_NF_Command **0x7051** **NAND Flash Command Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NFCMD															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	NFCMD	W	Write Command. Write “Command value” to the resister, and NAND flash interface will write this command value to NAND flash memory.	[15:0]: Command value For example: 00H: Read block A area 01H: Read block B area 50H: Read block C area 80H, 10H: Page programming 70H: read status 90H: read ID FFH: reset 60H,D0H: Block erasing

P_NF_Address **0x7052** **NAND Flash Address Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NFADDR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	NFADDR	W	Write Address instruction Write “Address value” to the resister, and NAND FLASH interface will write this Address value to NAND FLASH memory	[15:0]: Address value

P_NF_DataW **0x7053** **NAND Flash Data Write Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NFDATA															
Default																

Bit	Function	Type	Description	Condition
[15:0]	NFDATA	W	Write Data instruction. Write “Data value” to the resister, and NAND flash interface will write this Data value to NAND flash memory.	[15:0]: Data value

P_NF_DataR **0x7054** **NAND Flash Data Read Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NFREAD															
Default																

Bit	Function	Type	Description	Condition
[15:0]	NFREAD	W	Read data instruction Write any value to this register to inform NAND FLASH interface start reading NAND flash memory. After that, the read register ($7055) should be executed to get the data.	

P_NF_Data **0x7055** **NAND Flash Data Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NFD															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	NFD	R	Read data from NAND FLASH memory. This register should be read after **P_NF_DataR** is written to obtain data	B[15:0]: NAND flash data to be read

P_ECC_Ctrl **0x7056** **ECC Control Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CKP	ERST
Default																

Bit	Function	Type	Description	Condition
[15:2]			Reserved	
1	CKP	W	Write “1” to check parity (Line or Column) once. The Error Flag will show on $71D7 and &71D8	1: Check once 0: no effect
0	ERST	W	Reset ECC. The reset action must be done before any data input to the ECC module.	1: Reset 0: not Reset

Note that SPL161001A internal ECC calculation logic is always functioning. It will calculate any value that appears on the NAND flash data bus, including read data, written data, and commands. In addition, because of the nature of ECC logic, only after exactly 512 bytes (256 words) of data is input to the ECC logic, the ECC calculation result is valid. Hence, Sunplus recommends the resetting of ECC (write 1 to P_ECC_Ctrl’s bit 0 BEFORE consecutively read or write 512 bytes) and then retrieve the results immediately after that.

P_ECC_LPRL **0x7057** **ECC Line parity LSB Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRL	R	The ECC line parity register (LSB)	

P_ECC_LPRH **0x7058** **ECC Line parity MSB Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRH															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRH	R	The ECC line parity register (MSB)	

P_ECC_CPR **0x7059** **ECC Column parity Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CPR															
Init	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	CPR	R	The ECC column parity register	

P_ECC_LPRCKL **0x705A** **ECC Line parity Check LSB Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRCKL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRCKL	R/W	The ECC line parity check register (LSB)	

P_ECC_LPRCKH **0x705B** **ECC Line parity Check MSB Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRCKH															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRCKH	R/W	The ECC line parity check register (MSB)	

ECCCPCKR **0x705C** **ECC Column parity Check Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	CPRCK	R/W	The ECC column parity check register	

P_ECC_Err0 **0x705D** **ECC Error Flag0**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	ERR0												
Init	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:13]			Reserved	
[12:0]	ERR0	R	The ECC error flag for the first 256-byte in the 512-byte frame. Two-bit errors are not recoverable, while one-bit errors can be recovered by finding the error byte and the error bit and then reverse it.	Bit12 = 1: there is a two-bit error Bit11 = 1: there is a one-bit error Bit[10:8]: error bit number Bit[7:0]: error byte number in the first 256-byte

P_ECC_Err1 **0x705E** **ECC Error Flag1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	ERR1												
Init	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:13]			Reserved	
[12:0]	ERR1	R	The ECC error flag for the second 256-byte in the 512-byte frame. Two-bit errors are not recoverable, while one-bit errors can be recovered by finding the error byte and the error bit and then reverse it.	Bit12 = 1: there is a two-bit error Bit11 = 1: there is a one-bit error Bit[10:8]: error bit number Bit[7:0]: error byte number in the second 256-byte

P_DMA_AddrL **0x7380** **DMA Start Address LSB**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DSAL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	DSAL	R/W	DMA start address (LSB)	

P_DMA_AddrH **0x7381** **DMA Start Address MSB**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	DSAH					
Init	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	DSAH	R/W	DMA start address (MSB)	

P_DMA_Size **0x7382** **DMA SIZE**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	DSIZE									
Init	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	DSIZE	R/W	DMA SIZE	

P_DMA_Write **0x7383** **DMA Write Action**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	WACT															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	WACT	W	Write this port to activate DMA WRITE action	Write any value

P_DMA_Read **0x7384** **DMA Read Action**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RACT															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	RACT	W	Write this port to activate DMA READ action	Write any value

P_DMA_WriteCtrl **0x7385** **DMA Write Control**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DWIF	WIE	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Init	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
15	DWIF	R	DMA Write OK interrupt Flag If DMA	Write '1' to clear
14	WIE	R/W	DMA Write OK interrupt enable	0: Disable 1: Enable
[13:0]			Reserved	

P_DMA_ReadCtrl **0x7386** **DMA Read Control**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DRIF	RIE	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Init	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Function	Type	Description	Condition
15	DRIF	R	DMA Read OK interrupt Flag	Write '1' to clear
14	RIE	R/W	DMA Read OK interrupt enable	0: Disable 1: Enable
[13:0]			Reserved	

16.3 Program Example

16.3.1 8-bit Mode Without DMA

```
F_WriteNAND_Byte:
        r1=0x2100+C_NFTIMING        // Set timings according to NAND flash spec
        [P_NF_Ctrl]=r1
        r1=0x80
        [P_NF_Command]=r1           // Send command "0x80"
        r1=0x0000
        [P_NF_Address]=r1           // Send address
        r1=0x0000                   // The three address bytes consists the start
        [P_NF_Address]=r1           // address; in this case, the first page
        r1=0x0000
```

```
                    [P_NF_Address]=r1
                    r1=0x1
                    [P_ECC_Ctrl]=r1          // Reset ECC logic
                    bp=0
L_NF_Write_Not_Finished:                     // Write 512 bytes consecutively
                    r1=[bp]                  // Data will first be written to "A" area and then
                    [P_NF_DataW]=r1          // "B" area
                    bp+=1
                    cmp bp, 0x0200
                    jne L_NF_Write_Not_Finished

                    r2=[P_ECC_LPRL]
                    r3=[P_ECC_LPRH]
                    r4=[P_ECC_CPR]           // Read ECC result
                    r1=r2 and 0x00ff
                    [P_NF_DataW]=r1          // Write ECC result 6 bytes (to "C" area)
                    r1=r2 and 0xff00
                    r1=r1 lsr 4
                    r1=r1 lsr 4
                    [P_NF_DataW]=r1
                    r1=r3 and 0x00ff
                    [P_NF_DataW]=r1
                    r1=r3 and 0xff00
                    r1=r1 lsr 4
                    r1=r1 lsr 4
                    [P_NF_DataW]=r1
                    r1=r4 and 0x00ff
                    [P_NF_DataW]=r1
                    r1=r4 and 0xff00
                    r1=r1 lsr 4
                    r1=r1 lsr 4
                    [P_NF_DataW]=r1

                    r1=0x10                  // Send command "0x10"
                    [P_NF_Command]=r1
                    call F_CheckNANDBusy
                    call F_ReadNANDStatus_Byte
                    retf
F_ReadNANDStatus_Byte:                      // Read NAND flash status
                    r1=0x70
                    [P_NF_Command]=r1
                    [P_NF_DataR]=r1
                    r1=[P_NF_Data]
                    retf
```

```
F_CheckNANDBusy:                                    // Check the status of NAND flash if busy
                    r1=[P_NFCTRL]
                    r1=r1 and 0x8000
                    cmp r1,0x8000
                    jne F_CheckNANDBusy
                    retf
```

16.3.2 16-bit Mode With DMA

```
F_ReadNAND_Word_DMA:                                //read data
                    r1=0x6000+C_NFTIMING
                    [P_NF_Ctrl]=r1
                    r1=0x00
                    [C_DMAFlag]=r1
                    irq on

                    r1=0x4000
                    [P_DMA_ReadCtrl]=r1
                    r1=0x1000
                    [P_DMA_AddrL]=r1
                    r1=0x0
                    [P_DMA_AddrH]=r1
                    r1=255
                    [P_DMA_Size]=r1

                    r1=0x00
                    [P_NF_Command]=r1               //command
                    r1=C_NF_ADDR2
                    [P_NF_Address]=r1
                    r1=C_NF_ADDR3
                    [P_NF_Address]=r1
                    call F_CheckNANDBusy
                                                    // reset ECC
                    r1=0x1
                    [P_ECC_Ctrl]=r1
L_Read_Not_Finish:  r1=0
                    [P_DMA_Read]=r1
                    r1=[0x1000]
                    [0x800]=r2
                    r1=[C_DMAFlag]
                    cmp r1,0x9999
                    jne L_Read_Not_Finish
```

```
                                                    //read ecc result
                r2=[P_ECC_LPRL]
                r3=[P_ECC_LPRH]
                r4=[P_ECC_CPR]
                r1=0x02                             //check parity
                [P_ECC_Ctrl]=r1

                r2=[P_ECC_Err0]
                r3=[P_ECC_Err1]
                bp=0x1200
L_Read_C_Area:
                [P_NF_DataR]=r1
                r1=[P_NF_Data]
                [bp++]=r1
                cmp bp,0x1208
                jne L_Read_C_Area

                bp=0
                r1=0x1000
L_Verify:
                r2=[bp]
                r3=[r1]
                cmp r2,r3
                jne L_Read_Error
                retf
L_Read_Error:
                jmp $
_IRQ1:
                int off
                r1=[P_INT_Status1]
                r1=r1 and 0x0008
                cmp r1, 0x0008
                je L_IRQ_NF_Write

                r1=[P_INT_Status1]
                r1=r1 and 0x0004
                cmp r1, 0x0004
                je L_IRQ_NF_Read
L_IRQ_NF_Write:
                r1=[P_DMA_WriteCtrl]
                [P_DMA_WriteCtrl]=r1
                r1=0x8888
                [C_DMAFlag]=r1
                reti
```

```
L_IRQ_NF_Read:
                    r1=[P_DMA_ReadCtrl]
                    [P_DMA_ReadCtrl]=r1
                    r1=0x9999
                    [C_DMAFlag]=r1
                    reti
```

17 Miscellaneous

17.1 Introduction

For easy programming, the SPL161001A offers bit, nibble, and byte operations. In addition, the SPL161001A also supports data segment update control register. Modifying data segment (DS) in Status Register (SR) by writing data to this control register will save several CPU cycles.

17.2 Specified Register

Miscellaneous Register Summary Table

Name	Address	Description
P_Byte_Swap	0x71F0	Byte Swap
P_Nibble_Swap	0x71F1	Nibble Swap
P_TwoBit_Swap	0x71F2	Two-Bit Swap
P_Bit_Reverse	0x71F3	Bit Reverse

P_Byte_Swap **0x71F0** **Byte Swap**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	BYTESWAP	R/W	Byte Swap Write [B15:B0] to this control register and read this control register to obtain [B7:B0,B15:B8]	

P_Nibble_Swap **0x71F1** **Nibble Swap**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	NIBSWAP	R/W	Nibble swap Write [B15:B0] to this control register and read this control register to obtain [B11:B8,B15:B12, B3:B0,B7:B4]	

P_TwoBit_Swap **0x71F2** **2-Bit Swap**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	2BSWAP	R/W	Two bits swap Write [B15:B0] to this control register and read this control register to get [B13:B12,B15:B14, B9:B8, B11:B10, B5:B4, B7:B6,B1:B0,B3:B2]	

P_Bit_Reverse **0x71F3** **Bit Reverse**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	BITREV	R/W	Bit reverse Write [B15:B0] to this control register and read this control register to get [B0:B15]	

17.3 Program Example

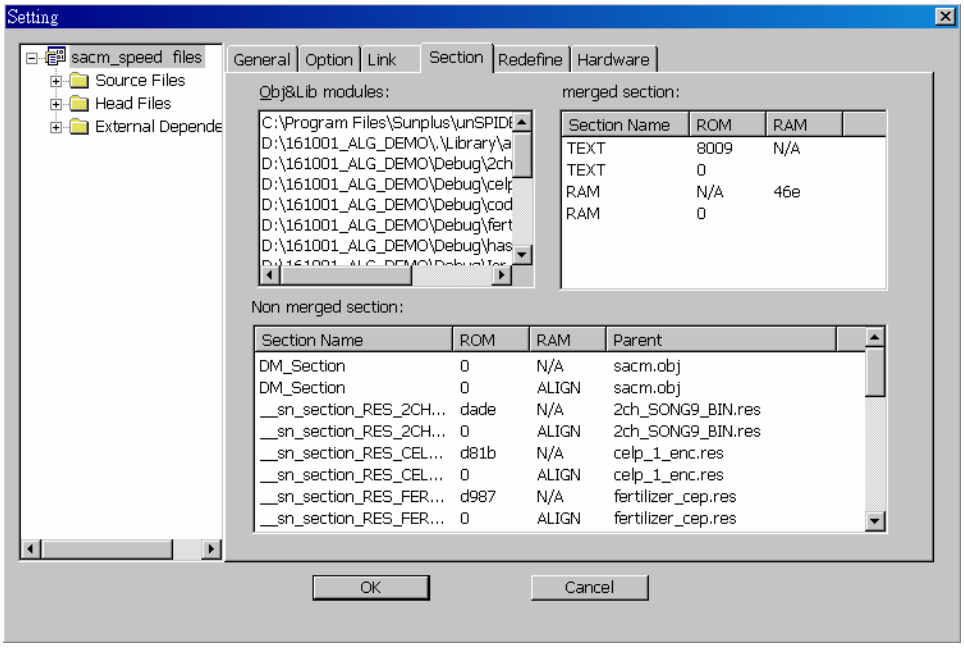
```
r1 = 0xFA50
[P_Byte_Swap]=r1
[P_Nibble_Swap]=r1
[P_Bit_Reverse]=r1
r1 = 0x6996
[P_TwoBit_Swap]=r1

r1= [P_Byte_Swap]
cmp   r1,0x50FA                    // Verify Byte Swap
jne   L_OtherTestError?
r2=[P_Nibble_Swap]
cmp   r2,0xAF05                    // Verify Nibble Swap
jne   L_OtherTestError?
r3=[P_Bit_Reverse]
```

```
                    cmp   r3,0x0A5F                    // Verify Bit Reverse
                    jne   L_OtherTestError?

                    r4=[P_TwoBit_Swap]
                    cmp   r4,0x9669                    // Very 2-bit Reverse
                    jne   L_OtherTestError?
                    ……
L_OtherTestError?:
                    ……
```

18 Link Your Program

u'nSP IDE allocates the program and resource in two ways: fixed sections for resource and program. The way to setup a start address is as follows:

project→setting…→section



Note: Since the setting will override the others, programmer must avoid the location overlap of program and resource.

After a program is compiled and linked, a .lik file is produced. The .lik file includes the following information: obj, resource, and the definition of section link. We can allocate all resource and section location in the .lik fie manually. See the following for details:

Options: m37

Obj: "D:\SPL161001\BOARDTEST\DEBUG\RESOURCE.OBJ"

Obj: "D:\SPL161001\BOARDTEST\DEBUG\BOARDTEST.OBJ"

Obj: "D:\SPL161001\BOARDTEST\DEBUG\ISR.OBJ"

Obj: "D:\SPL161001\BOARDTEST\DEBUG\SPLLIB.OBJ"

Obj: "D:\SPL161001\BOARDTEST\DEBUG\STARTUP.OBJ"

Obj: "D:\SPL161001\BOARDTEST\DEBUG\A38_16K.RES"

Obj: "D:\SPL161001\BOARDTEST\DEBUG\A32_16K.RES"

Locate: main in "boardtest.obj" at **911B**
;Align: main in "boardtest.obj" with 0
Locate: CODE in "SPLLIB.obj" **after main**
;Align: CODE in "SPLLIB.obj" with 0
Locate:_RES_A38_16K in "A38_16K.res" at **10000**
;Align: _RES_A38_16K in "A38_16K.res" with 0
Locate:_RES_A32_16K in "A32_16K.res" **after _RES_A38_16K**
;Align: _RES_A32_16K in "A32_16K.res" with 0
Output: ".\Debug\boardtest.S37"

Set Manually

This method is able to avoid the overlap of resource and program when they are modified.

19 Appendix

19.1 Commonly Used Abbreviations

Although all abbreviations indicated here are upper cases, lower cases are also accepted for naming flexibility. However, the first letter should be capitalized. For example, “BUF” = “Buf”, “CH0” = “Ch0”, “CLK”=”Clk”, etc.

Abbreviation	Description
ADR	Address
BUF	Buffer
CHAR	Character
CHA, CHB, etc	CHA, CHB, etc.
CLK	Clock
CMP	Compare
CNT	Counter
COM	Common (starts with “0”, e.g., COM0, COM1…)
DACA, DACB, etc.	DACA, DACB etc.
DEC	Decrease
DISP	Display
ECLK	External clock
ERR	Error
FG	Flag
FLOAT	Float state
FUNC	Function
INC	Increase
INT	Interrupt
DEFAULT	Initialization
LVD	Low Voltage Detection
LVPD	Low Voltage Power Down
LVR	Low Voltage Reset
MAX	Maximum
MIN	Minimum
MUTE	Mute
NMI	Non-masked-interrupt
ODN	Open drain NMOS (sink)
ODP	Open drain PMOS (send)
PortA, PortB	I/O PortA, IO PortB, etc
PH	Pull high resistor
PL	Pull low resistor
PWM	Pulse-Width Modulation
R/W	Readable/writable
ROSC	R-oscillator

Abbreviation	Description
RTC	Real time clock
SEG	Segment (starts with “0”, e.g., SEG0, SEG1…)
SLP	Sleep
SPCH	Speech
STR	String
TMA, TMB, etc	TimerA, TimerB, etc.
TMP	Temporary
TONE	Tone
UART	Universal Asynchronized Receiver Transmitter
VOL	Volume
WAKE	Wakeup
WDG	Watchdog
X’TAL or XTAL	Crystal oscillator

19.2 Control Register Mapping List (by function)

0000_{16}~$1FFF_{16}$	8KW SRAM
2000_{16}~$6FFF_{16}$	Reserved
7000_{16}~$701F_{16}$	System Control
7020_{16}~$704F_{16}$	Memory Control
7050_{16}~$705F_{16}$	NAND Flash Interface Control
7060_{16}~$709F_{16}$	I/O Port Control
$70A0_{16}$~$70AF_{16}$	Interrupt Control
$70B0_{16}$~$70BF_{16}$	Time Base Control
$70C0_{16}$~$70DF_{16}$	Timer Control
$70E0_{16}$~$70FF_{16}$	Audio Output Control
7100_{16}~$711F_{16}$	UART / IrDA Control
7120_{16}~$713F_{16}$	RTC Control
7140_{16}~$715F_{16}$	SPI Control
7160_{16}~$717F_{8}$	Reserved
7180_{16}~$719F_{8}$	LCD Control
$71A0_{16}$~$71AF_{16}$	Reserved
$71B0_{16}$~$71BF_{16}$	Reserved
$71C0_{16}$~$71CF_{16}$	Sunplus Serial Interface Control
$71D0_{16}$~$71EF_{16}$	Reserved
$71F0_{16}$~$71FF_{16}$	Miscellaneous
7200_{16}~$72FF_{16}$	Reserved
7300_{16}~$7FFF_{16}$	Reserved
7380_{16}~$738F_{16}$	NAND Flash DMA Control

19.3 Control Registers Summary

System Control Register Summary Table

Name	Address	Description
P_BodyID	0x7000	Body Identification Number Register
P_BodyInfo	0x7001	Body Information
P_Reset_Flag	0x7006	Reset Event Flag Register
P_Clock_Ctrl	0x7007	System Clock Control Register
P_LVR_Ctrl	0x7008	Low Voltage Reset Control Register
P_LVD_Ctrl	0x7009	Low Voltage Detection Control Register
P_Watchdog_Ctrl	0x700A	Watchdog Control Register
P_Watchdog_Clear	0x700B	Watchdog Clear Register
P_WAIT	0x700C	Wait Mode Entrance Register
P_HALT	0x700D	Halt Mode Entrance Register
P_SLEEP	0x700E	Sleep Mode Entrance Register

Memory Control Register Summary Table

Name	Address	Description
P_MCS0_Ctrl	0x7020	Chip Selection 0 Memory Device Control Register
P_MCS1_Ctrl	0x7021	Chip Selection 1 Memory Device Control Register
P_MCS2_Ctrl	0x7022	Chip Selection 2 Memory Device Control Register
P_MCS3_Ctrl	0x7023	Chip Selection 3 Memory Device Control Register
P_MA16_Disable	0x7030	Memory Address 16 Disable (GPIO PortD7 Enable) Write-once Register
P_MA17_Disable	0x7031	Memory Address 17 Disable (GPIO PortD8 Enable) Write-once Register
P_MA18_Disable	0x7032	Memory Address 18 Disable (GPIO PortD9 Enable) Write-once Register
P_MA19_Disable	0x7033	Memory Address 19 Disable (GPIO PortD10 Enable) Write-once Register
P_MA20_Disable	0x7034	Memory Address 20 Disable (GPIO PortD11 Enable) Write-once Register
P_MA21_Disable	0x7035	Memory Address 21 Disable (GPIO PortD12 Enable) Write-once Register
P_MA22_Disable	0x7036	Memory Address 22 Disable (GPIO PortD13 Enable) Write-once Register
P_MCS0_Disable	0x7040	Chip Selection 0 Disable (GPIO PortD3 Enable) Write-once Register
P_MCS1_Disable	0x7041	Chip Selection 1 Disable (GPIO PortD2 Enable) Write-once Register
P_MCS2_Disable	0x7042	Chip Selection 2 Disable (GPIO PortD1 Enable) Write-once Register
P_MCS3_Disable	0x7043	Chip Selection 3 Disable (GPIO PortD0 Enable) Write-once Register
P_MRE_Disable	0x7048	Memory Read Enable Signal Disable (GPIO PortC15 Enable) Write-once Register
P_MWE_Disable	0x7049	Memory Write Enable Signal Disable (GPIO PortC14 Enable) Write-once Register
P_Madr_Status	0x704A	Memory Address Signal Enable Status Register
P_MCtrl_Status	0x704B	Memory Control Signal (CS[3:0], WE, RE) Enable Status Register
P_InMem_Ctrl	0x704C	Internal Memory Device Control Register

I/O Port Control Register Summary Table

Name	Address	Description
P_IOA_Data	0x7060	I/O PortA Data Register
P_IOA_Buffer	0x7061	I/O PortA Buffer Register
P_IOA_Dir	0x7062	I/O PortA Direction Register
P_IOA_Attrib	0x7063	I/O PortA Attribute Register
P_IOA_Latch	0x7064	I/O PortA Latch Register for Wakeup
P_IOB_Data	0x7068	I/O PortB Data Register
P_IOB_Buffer	0x7069	I/O PortB Buffer Register
P_IOB_Dir	0x706A	I/O PortB Direction Register
P_IOB_Attrib	0x706B	I/O PortB Attribute Register
P_IOB_Latch	0x706C	I/O PortB Latch Register for Wakeup
P_IOC_Data	0x7070	I/O PortC Data Register
P_IOC_Buffer	0x7071	I/O PortC Buffer Register
P_IOC_Dir	0x7072	I/O PortC Direction Register
P_IOC_Attrib	0x7073	I/O PortC Attribute Register
P_IOD_Data	0x7078	I/O PortD Data Register
P_IOD_Buffer	0x7079	I/O PortD Buffer Register
P_IOD_Dir	0x707A	I/O PortD Direction Register
P_IOD_Attrib	0x707B	I/O PortD Attribute Register

Global Interrupt Control Register Summary Table

Name	Address	Description
P_INT_Status1	0x70A0	Interrupt Status Register 1
P_INT_Status2	0x70A1	Interrupt Status Register 2
P_INT_Priority1	0x70A4	Interrupt Priority Register 1
P_INT_Priority2	0x70A5	Interrupt Priority Register 2
P_MINT_Ctrl	0x70A8	Miscellaneous Interrupt Control Register

Timer Control Register Summary Table

Name	Address	Description
P_TimerA_Ctrl	0x70C0	TimerA Control Register
P_TimerA_CCP_Ctrl	0x70C1	TimerA Capture / Comparison / PWM Control Register
P_TimerA_Preload	0x70C2	TimerA Preload Register
P_TimerA_CCP_Reg	0x70C3	TimerA Capture / Comparison / PWM Register
P_TimerA_UpCount	0x70C4	TimerA up-count value
P_TimerB_Ctrl	0x70C8	TimerB Control Register
P_TimerB_CCP_Ctrl	0x70C9	TimerB Capture / Comparison / PWM Control Register
P_TimerB_Preload	0x70CA	TimerB Preload Register
P_TimerB_CCP_Reg	0x70CB	TimerB Capture / Comparison / PWM Register
P_TimerB_UpCount	0x70CC	TimerB up-count value
P_TimerC_Ctrl	0x70D0	TimerC Control Register
P_TimerC_Preload	0x70D2	TimerC Preload Register
P_TimerC_UpCount	0x70D4	TimerC up-count value

Timebase Control Register Summary Table

Name	Address	Description
P_TimeBaseA_Ctrl	0x70B0	TimeBaseA Control Register
P_TimeBaseB_Ctrl	0x70B1	TimeBaseB Control Register
P_TimeBaseC_Ctrl	0x70B2	TimeBaseC Control Register
P_TimeBase_Reset	0x70B8	TimeBase Counter Reset Register

Real Time Clock Control Register Summary Table

Name	Address	Description
P_Second	0x7120	Second Register
P_Minute	0x7121	Minute Register
P_Hour	0x7122	Hour Register
P_Alarm_Second	0x7124	Alarm Second Register
P_Alarm_Minute	0x7125	Alarm Minute Register
P_Alarm_Hour	0x7126	Alarm Hour Register
P_RTC_Ctrl	0x7134	HMS / Alarm / Scheduler Control Register
P_RTC_INT_Status	0x7135	HMS / Alarm / Scheduler Interrupt Flag & Clear Register
P_RTC_INT_Ctrl	0x7136	HMS / Alarm / Scheduler Interrupt Control Register

DAC / PWM Control Register Summary Table

Name	Address	Description
P_AudioOut_Ctrl	0x70E0	Audio Output Control Register
P_CHA_Ctrl	0x70F0	Channel A Control Register
P_CHA_Data	0x70F1	Channel A Data Register
P_CHA_FIFO	0x70F2	Channel A FIFO Control Register
P_CHB_Ctrl	0x70F8	Channel B Control Register
P_CHB_Data	0x70F9	Channel B Data Register
P_CHB_FIFO	0x70FA	Channel B FIFO Control Register

LCD Control Register Summary Table

Name	Address	Description
P_LCD_Setup	0x7180	LCD setup register
P_LCD_Clock	0x7181	LCD clock register
P_LCD_Segment	0x7182	LCD segment number register
P_LCD_Common	0x7183	LCD common number register
P_LCD_Buffer_LowAdr	0x7184	LCD start address register (A0~A15)
P_LCD_Buffer_HighAdr	0x7185	LCD start address register (A16~A21)
P_LCD_Buffer_Offset	0x7186	LCD virtual page offset register
P_LCD_Timing_Ctrl	0x7187	LCD control signal timing register
P_LCD_Frame_Ctrl	0x7188	LCD frame modulation control register

UART/IrDA Control Register Summary Table

Name	Address	Description
P_UARTIrDA_Data	0x7100	UART/ IrDA Data Register
P_UARTIrDA_RXStatus	0x7101	UART Reception Status (Error Flag) Register
P_UARTIrDA_Ctrl	0x7102	UART/ IrDA Control Register
P_UARTIrDA_BaudRate	0x7103	UART/ IrDA Baud Rate Setup Register
P_UART_Status	0x7104	UART/ IrDA Interrupt Register
P_IrDA_Ctrl	0x7108	IrDA Control Register

SPI Control Register Summary Table

Name	Address	Description
P_SPI_Ctrl	0x7140	SPI Control Register
P_SPI_TXStatus	0x7141	SPI Transmitting Status Register
P_SPI_TXData	0x7142	SPI Transmitting Buffer Register
P_SPI_RXStatus	0x7141	SPI Status Receiving Register
P_SPI_RXData	0x7142	SPI Receiving Buffer Register

Serial Interface Control Register Summary Table

Name	Address	Description
P_SIO_Ctrl	0x71C0	Serial I/F Control Register
P_SIO_AddrML	0x71C1	Serial I/F Address Middle/Low Byte Register
P_SIO_AddrH	0x71C2	Serial I/F Address High Byte Register
P_SIO_Start	0x71C3	Serial I/F Start Transmission
P_SIO_Stop	0x71C4	Serial I/F Stop Transmission
P_SIO_Write_EN	0x71C5	Serial I/F Write Enable
P_SIO_Read_EN	0x71C6	Serial I/F Read Enable

NAND Flash Interface Control Registers Summary

Name	Address	Description
P_NF_Ctrl	0x7050	NAND Flash Control Register
P_NF_Command	0x7051	NAND Flash Command Register
P_NF_Address	0x7052	NAND Flash Address Register
P_NF_DataW	0x7053	NAND Flash Data Write Register
P_NF_DataR	0x7054	NAND Flash Data Read Register
P_NF_Data	0x7055	NAND Flash Data Register
P_ECC_Ctrl	0x7056	ECC Control Register
P_ECC_LPRL	0x7057	ECC Line parity LSB Register
P_ECC_LPRH	0x7058	ECC Line parity MSB Register
P_ECC_CPR	0x7059	ECC Column parity Register
P_ECC_LPRCKL	0x705A	ECC Line Parity Check LSB Register
P_ECC_LPRCKH	0x705B	ECC Line Parity Check MSB Register
P_ECC_CPCKR	0x705C	ECC Column Parity Check Register
P_ECC_Err0	0x705D	ECC Error Flag0
P_ECC_Err1	0x705E	ECC Error Flag1
P_DMA_AddrL	0x7380	DMA Start Address LSB
P_DMA_AddrH	0x7381	DMA Start Address MSB
P_DMA_Size	0x7382	DMA SIZE
P_DMA_Write	0x7383	DMA Write Action

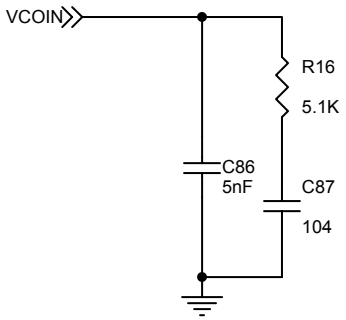
Name	Address	Description
P_DMA_Read	0x7384	DMA Read Action
P_DMA_WriteCtrl	0x7385	DMA Write Control
P_DMA_ReadCtrl	0x7386	DMA Read Control

Miscellaneous Register Summary Table

Name	Address	Description
P_Byte_Swap	0x71F0	Byte Swap
P_Nibble_Swap	0x71F1	Nibble Swap
P_TwoBit_Swap	0x71F2	Two-Bit Swap
P_Bit_Reverse	0x71F3	Bit Reverse

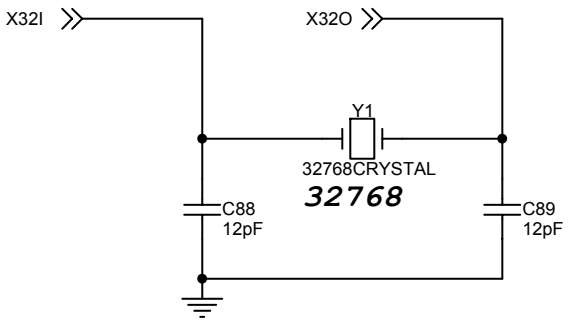
19.4 System Clock Usage Guide

The following components must be placed as near as possible to the SPL161001A while designing layout. In addition, the value of components should not be changed.



19.5 32768Hz Crystal Usage Guide

The following components must be placed as closed as possible to the SPL161001A while designing layout. The value of components should not be changed.

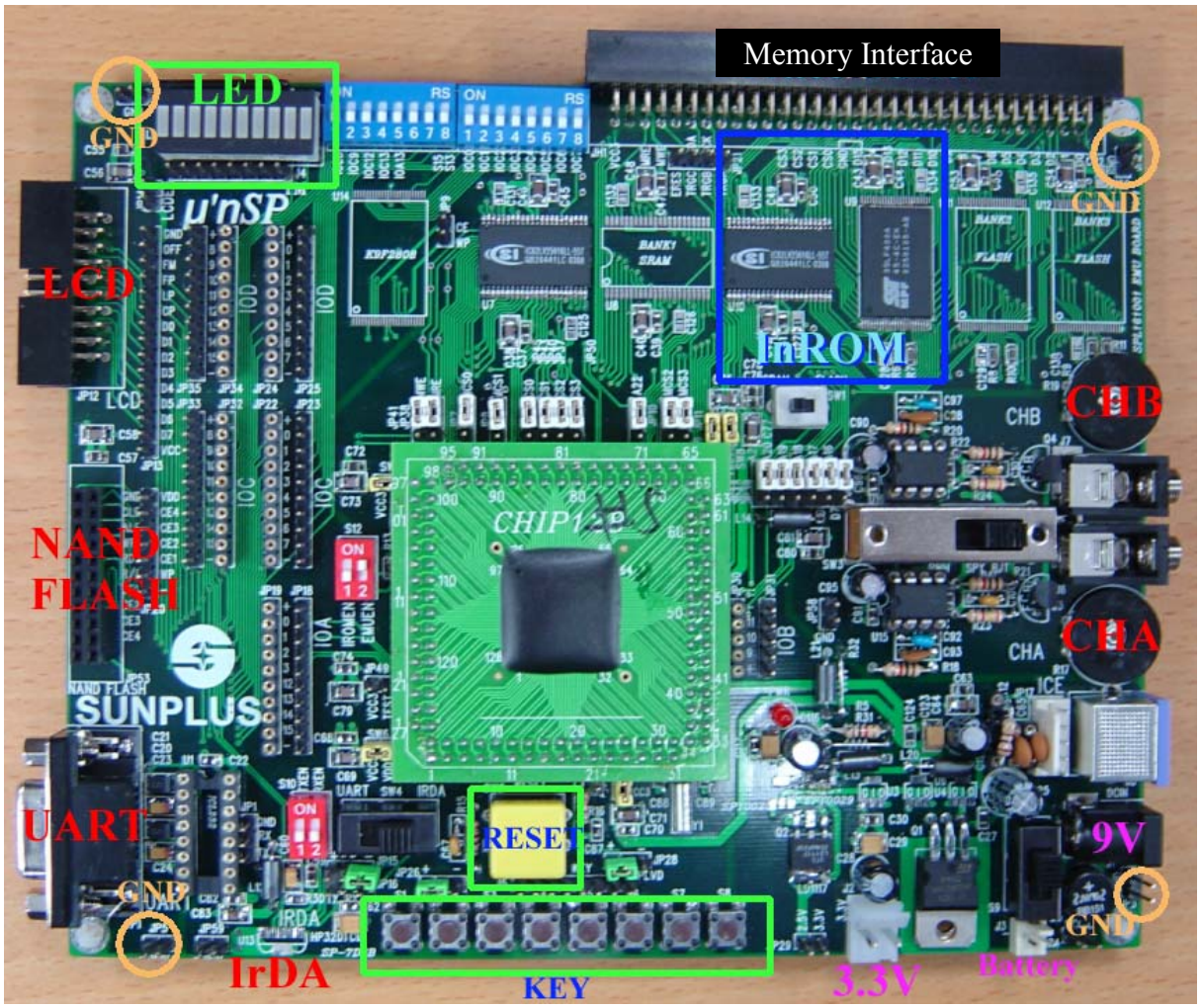


19.6 Development (Emulation) Board Configuration

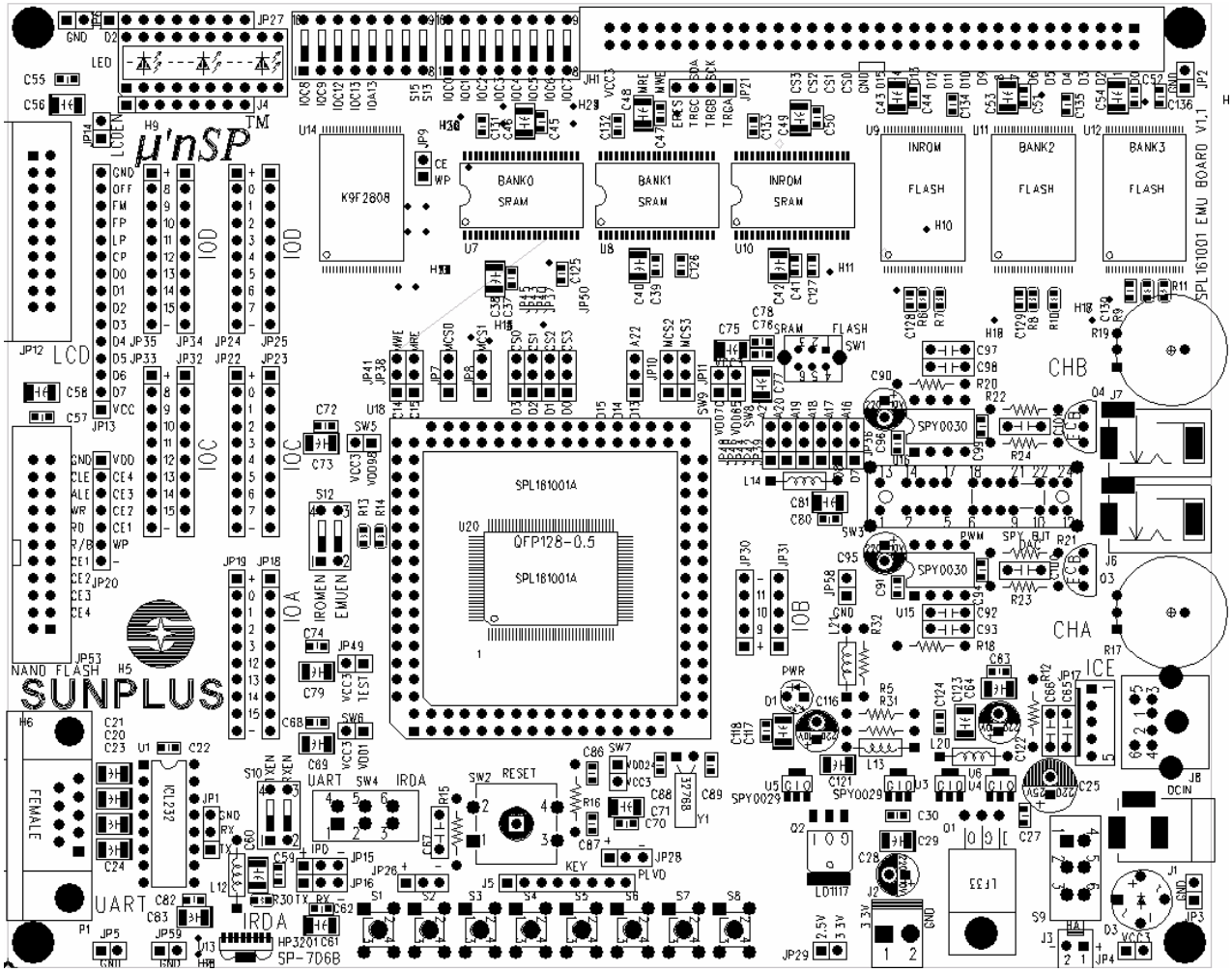
This appendix gives a brief overview of the SPL161001A development (emulation) board and details on how to use this board to emulate various products.

I. Connection

II. Emulation Board



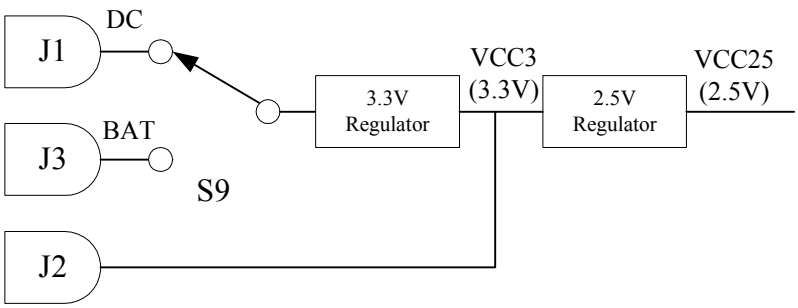
III. Emulation Board Top View



19.6.1 Power Adjustment & Selection

There are three ways to supply DC power for the SPL161001A emulation board. If the board is powered, LED (indicated as D1, labeled with PWR) will be turned on.

- **J1** Adapter DC Input, on-board regulators are activated.
  Input voltage level should be higher than DC 4.0V.
- **J2** Direct Power Input, bypass on-board 3.3V regulator (LF33).
  Input voltage level should be from DC 2.4V ~ 3.6V (IC Operation Voltage).
  The LD1117 (2.5V regulator) is still activated.
- **J3** Battery DC Input, on-board regulators are activated**.**
  Input voltage level should be larger than DC 4.0V.



Note that **S9** Switch is to select between two DC inputs: J1 and J3.

19.6.2 The Dual Power System

As mentioned in previous chapters, the SPL161001A provides dual power input to save power consumption. Pins VDDC and VDDP can be connected to a lower voltage (2.5V) for lower power. If SW7 and SW9 on the emulation board are short, VDDC and VDDP are connected to 2.5V power. If some other voltage levels are needed for VDDC and VDDP, simply remove the jumpers and connect them to that specific power. Note that VDDC and VDDP must have the same voltage level for power consumption reasons.

19.6.3 Reset

The switch SW2 (labeled as ***RESET***) resets the system manually and initializes the SPL161001A emulation board.

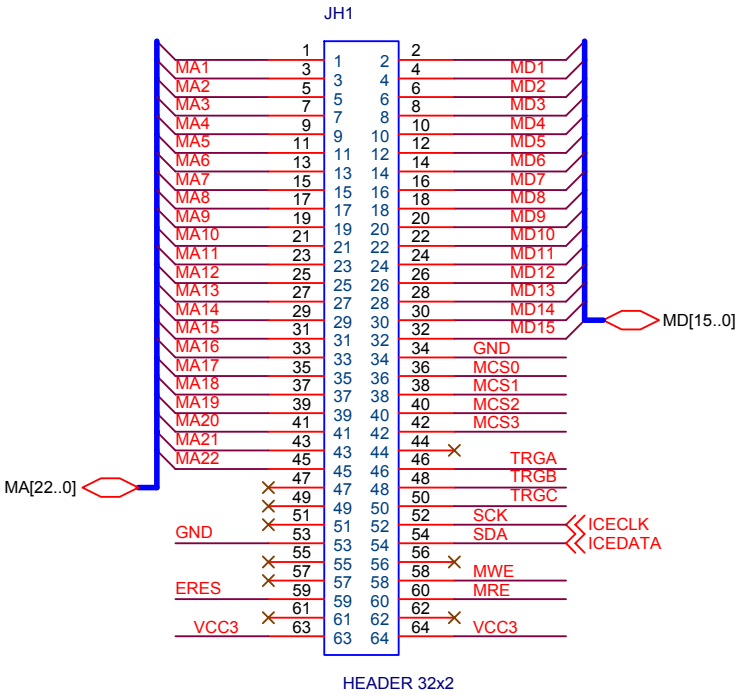
19.6.4 ICE

JP17 (labeled with ***ICE***) is the connector for SUNPLUS ICE Probe. The J8 jack is reserved for future ICE Probe interface. If your ICE probe interface is for J8, simply plug it in J8 and leave JP17 unconnected.

19.6.5 Memory

There are six memory footprints on the emulation board: 3 SRAM and 3 flash memories. There are 2 memories used to emulate internal mask ROM: 128/256K-word SRAM (faster download speed) and 128/256K-word Flash memory (standalone execution capability). Both are used as 128K-word memory for emulating the SPL161001A. These two memories can be selected by switch SW1. Generally, SRAM is used for ICE downloading. In addition, CS0 and CS1 can be connected to 256K-word SRAM via Jumper (JP7, JP8). Similarly, CS2 and CS3 can be connected to 2M-word flash via Jumper (JP10, JP11). Certainly, if CS0 and CS1 configurations are not SRAM, or CS2 and CS3 configurations are not flash, the programmer can disable the CE signal for SRAM or flash (normally pulled-high) with the jumpers. That is, the programmer can design the layout based on external memory requirements.

The pin-layout of the Memory Cartridge Interface is shown in the following schematic. Note that the pin-layout is no longer compatible with that of the SPL16256A emulation board.



The SPL161001A allows the memory control signal of CS0, CS1, CS2, CS3 and A21, A20, A19, and A18 to be set to GPIO. When it is changed to PortD's GPIO, Jumper JP37(CS3), JP40(CS2), JP43(CS1), JP45(CS0), JP48(MA21), JP46(MA20), JP44(MA19), and JP42(MA18) must be properly setup.

19.6.6 NAND Flash Interface

JP53 is a header for NAND Flash connections. The User can layout a NAND Flash daughter board to fit this header. The pins of JP53 connect to corresponding shared I/O pins on the SPL161001A directly. For example, the R/B pin on the JP53 is connected to PortA.13. JP20 contains the control pins to NAND Flash, CE and WP. Also, a NAND Flash footprint (U14) is reserved on the SPL161001A emulation board which fits the pin layout of SAMSUNG's K9F28 series. If U14 is used, please note DIP switches S13 and S15. Only when the S13 and S15 are turned on, the pins of U14 are connected to the corresponding shared I/O pins of the SPL161001A. For example, if S15 pin 5 is not turned on, the R/B pin of U14 is not connected to PortA.13. In addition, CE and WP of U14 are connected separately to JP9 so that users can use arbitrary I/O to control them.

19.6.7 Audio Output

J6 (labeled as CHA) and J7 (labeled as CHB) are phone-jacks to external mono speaker.

The SW3 Switch can select between three audio output modes: PWM, SPY and BJT.

Switch Selection (SW3)	Description	Audio Output Mode
PWM	Should be selected if SPL161001A audio output mode is configured as PWM mode (direct-driven). Users can connect a speaker on either CHA phonejack or CHB phonejack.	PWM Mode
SPY	If SPL161001A is configured as DAC mode and SPY is selected, two audio output pins (AUDA and AUDB) are amplified by SUNPLUS OPs, named SPY0030. Users can modify the gain of the amplifiers for CHA and CHB by simply changing the value of corresponding potential resistor (R6 and R8).	DAC Mode
BJT	If SPL161001A is configured as DAC mode and BJT is selected, two audio output pins (AUDA and AUDB) are amplified by BJTs (8050)	DAC Mode

To configure audio output mode between DAC and PWM, please refer to ***Chapter Audio***.

19.6.8 Others Input and Output

I/O Ports:

PortA (referenced as JP18, JP19, JP28 and JP29, labeled with ***IOA***)

PortB (referenced as JP20, JP21, JP30 and JP31, labeled with ***IOB***)

PortC (referenced as JP22, JP23, JP32 and JP33, labeled with ***IOC***)

PortD (referenced as JP24, JP25, JP34 and JP35, labeled with ***IOD***)

LCD Driver Interface Header

Referenced as JP12 and JP13, labeled with ***LCD.***

JP14, which is connected to the LCDEN (DISPLAYOFF#) pin of external LCM, is used to enable the LCM. Please connect any pin of JP14 to VCC. In addition, users can connect one designated GPIO pin to JP14 to turn the LCM on or off. Refer to the LCD schematic diagram for detailed pin assignment.

UART Interface Header

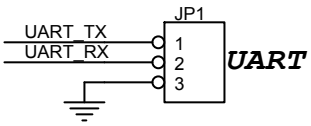
Referenced as JP1, labeled with ***UART.***

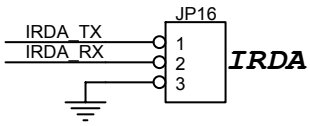
Note that, if the UART interface is enabled, both TXEN and RXEN DIP switches on S10 should be turned on. Also, SW4 should be switched to the UART side. The SPL161001A emulation board also has a built-in RS232 level-shifter IC (HIN232) with female output header (referenced as P1).

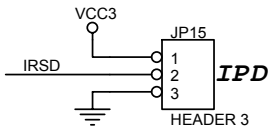
IrDA Interface Header

Referenced as JP16, labeled with ***IRDA.***

Note that, if the IrDA interface is enabled, both TXEN and RXEN DIP switches that are on S10 should be turned on. Also, the SW4 should be switched to the IrDA side. Also, the SPL161001A emulation board reserves a footprint of an IrDA transceiver module (HP3201). This IC is optional, and not soldered onto the PCB. The JP15 is a header for the IRSD pin of the IrDA transceiver module. The user can connect this pin to a GPIO for power-down control.



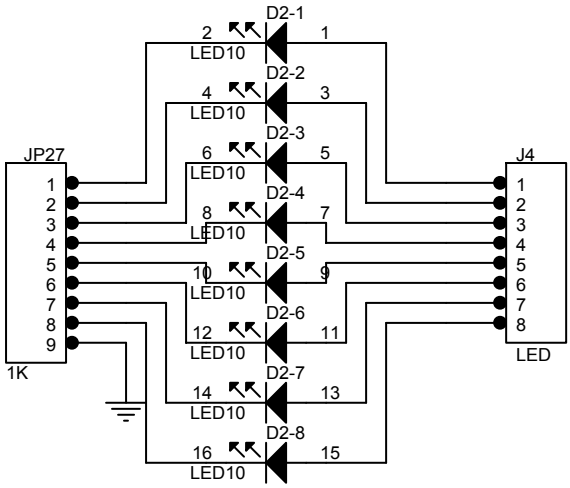




On-board LED indicator Header

Referenced as J4, labeled with ***LED.***

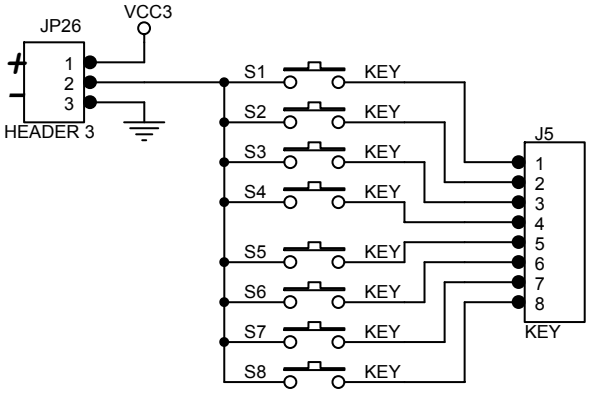
This LED array is for general-purpose indicator(s), and connecting any one pin on J4 to high voltage will turn on corresponding LED.



On-board Key Array Header

Referenced as J5, labeled with ***KEY.***

This key array is for general-purpose input(s). Each pin on J5 connects to one side of each key individually, and the other sides of all keys are all interconnected together into one common pin in the middle of JP26. This common pin can be easily shorted to high voltage or low voltage via a single jumper.



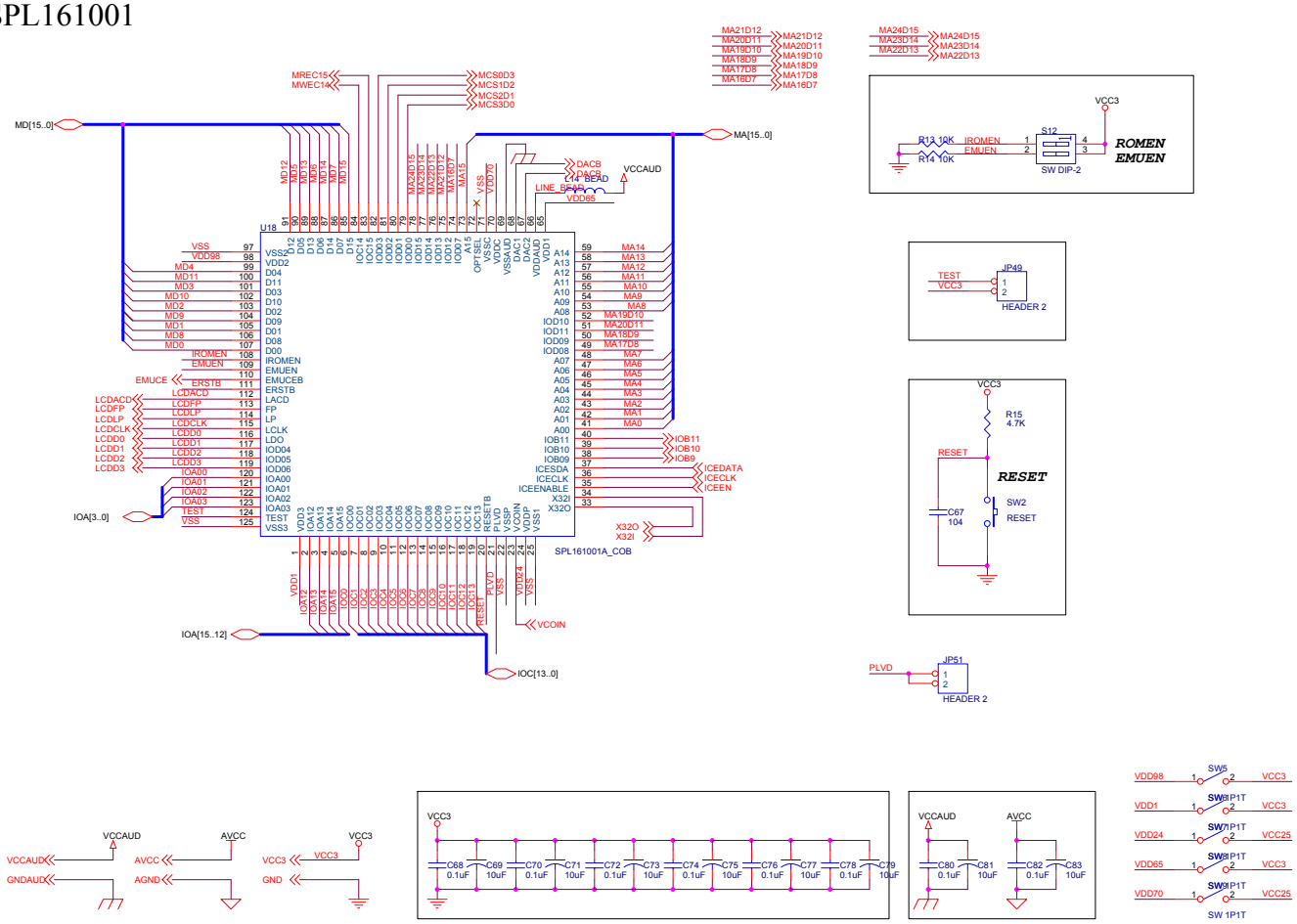
LVD Header

Referenced as JP51, both pins of JP51 connect to PLVD on the SPL161001A. The user can connect the JP51 to a voltage source which is to be detected, or connect it to GND to simulate low-voltage condition.

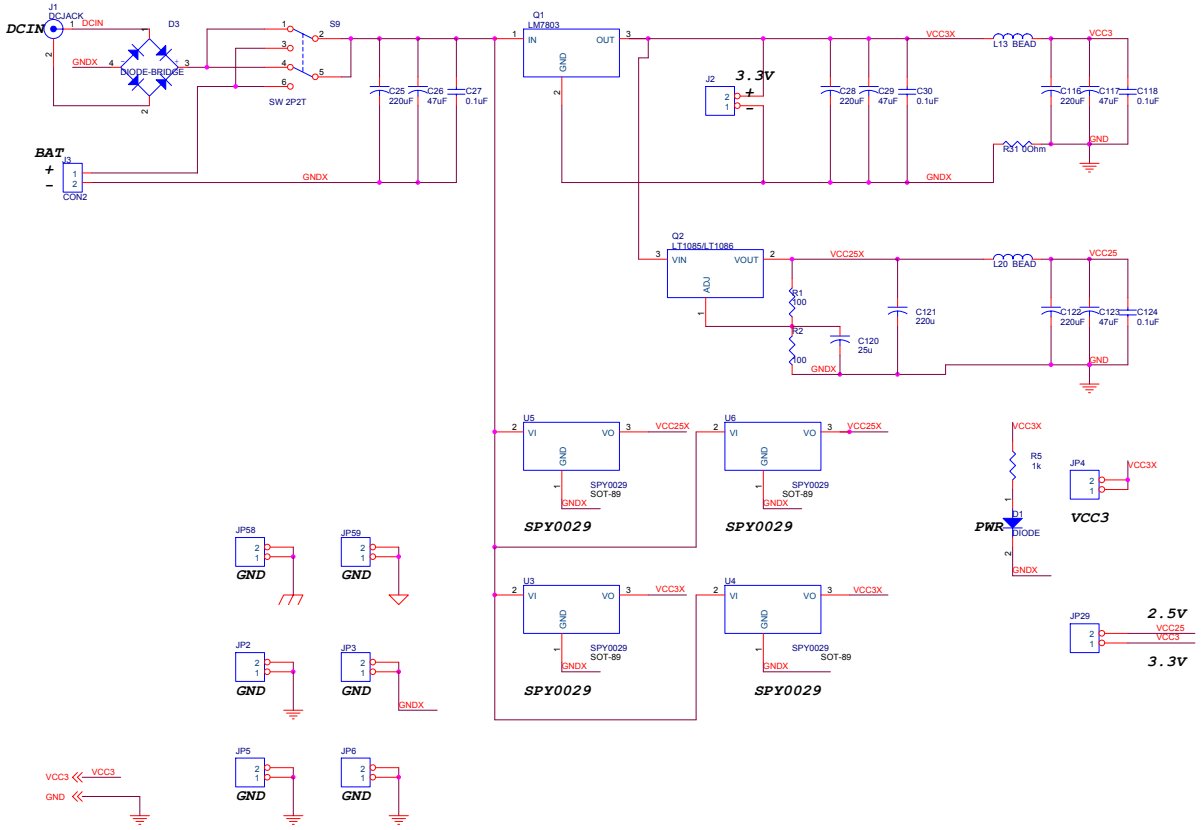
19.7 Development (Emulation) Board Schematic

A. MAIN (Emulation chip)
B. Power
C. Clock
D. ICE
E. GPIO (General-Purposed Input & Output)
F. Memory Array
G. Audio Output
H. LCD Interface
I. UART Interface
J. IrDA Interface
K. NAND Flash Interface

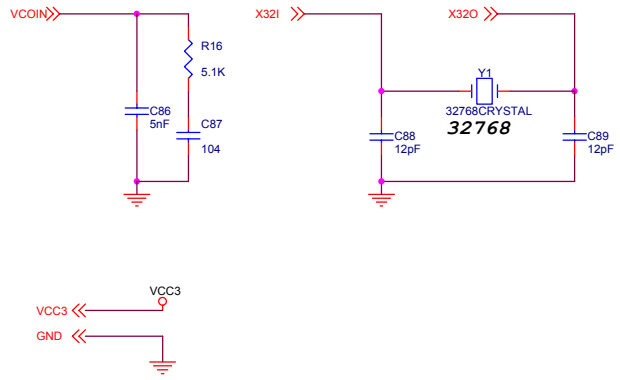
SPL161001


U18
SPL161001A_COB
MD[15..0]
MA[15..0]
IOA[3..0]
IOA[15..12]
IOC[13..0]
VCCAUD
VCOIN
VCC3
ROMEN
EMUEN
SW DIP-2
JP49
HEADER 2
RESET
SW2
JP51
PLVD
VCCAUD
AVCC
GNDAUD
AGND
GND
SW5
SW 1P1T

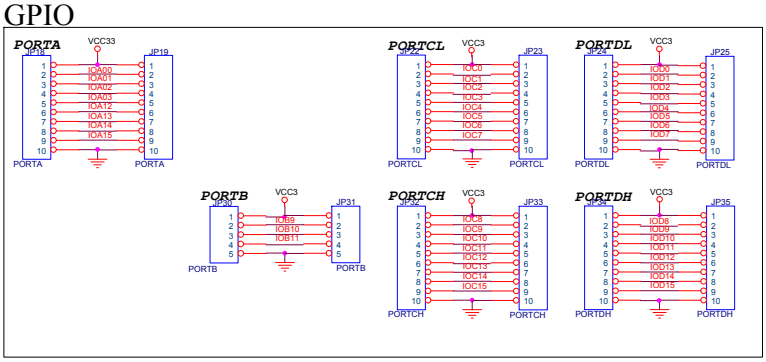
POWER

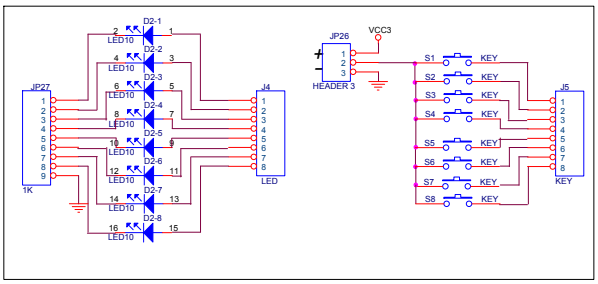

DCIN
J1
DCJACK
D3
DIODE-BRIDGE
GNDX
S9
SW 2P2T
C25
220uF
C26
47uF
C27
0.1uF
Q1
LM7803
IN
OUT
GND
J2
3.3V
C28
220uF
C29
47uF
C30
0.1uF
VCC3X
L13 BEAD
VCC3
C116
220uF
C117
47uF
C118
0.1uF
R31 0Ohm
BAT
J3
CON2
Q2
LT1085/LT1086
VIN
VOUT
ADJ
VCC25X
R1
100
R2
100
C120
25u
C121
220u
L20 BEAD
VCC25
C122
220uF
C123
47uF
C124
0.1uF
U5
U6
U3
U4
VI
VO
SPY0029
SOT-89
VCC25X
VCC3X
R5
1k
D1
DIODE
PWR
JP4
VCC3
JP58
JP59
JP2
JP3
JP5
JP6
GND
JP29
2.5V
VCC25
VCC3
3.3V

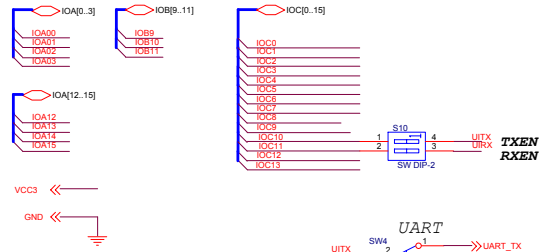
CLOCK

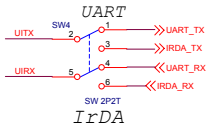

VCOIN
R16
5.1K
C86
5nF
C87
104
X32I
X32O
Y1
32768CRYSTAL
32768
C88
12pF
C89
12pF
VCC3
GND

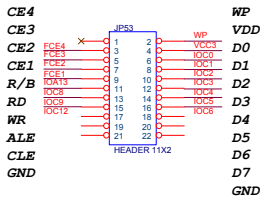
GPIO

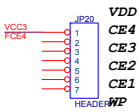

PORTA
PORTCL
PORTDL
PORTB
PORTCH
PORTDH
VCC3
VCC33

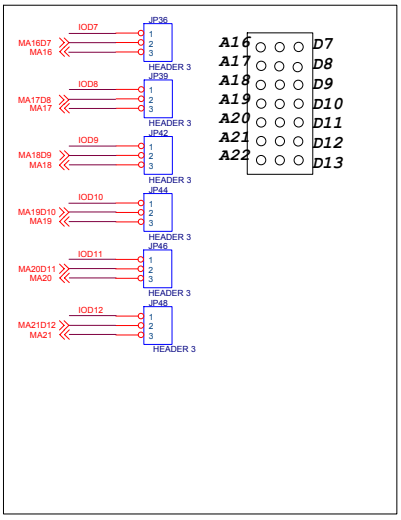

D2-1
D2-2
D2-3
D2-4
D2-5
D2-6
D2-7
D2-8
LED10
JP27
1K
J4
LED
JP26
HEADER 3
VCC3
S1
S2
S3
S4
S5
S6
S7
S8
KEY
J5

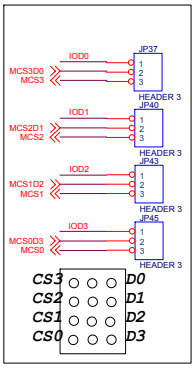

IOA[0..3]
IOB[9..11]
IOC[0..15]
IOA[12..15]
S10
SW DIP-2
UITX
UIRX
TXEN
RXEN
VCC3
GND

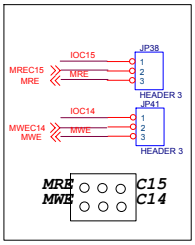

UART
SW4
UITX
UIRX
UART_TX
IRDA_TX
UART_RX
IRDA_RX
SW 2P2T
IrDA

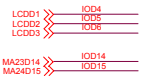

CE4
CE3
CE2
CE1
R/B
RD
WR
ALE
CLE
GND
JP53
HEADER 11X2
WP
VCC3
D0
D1
D2
D3
D4
D5
D6
D7
GND


VCC3
FCE4
JP20
HEADER
VDD
CE4
CE3
CE2
CE1
WP


IOD7
IOD8
IOD9
IOD10
IOD11
IOD12
HEADER 3
A16
A17
A18
A19
A20
A21
A22
D7
D8
D9
D10
D11
D12
D13

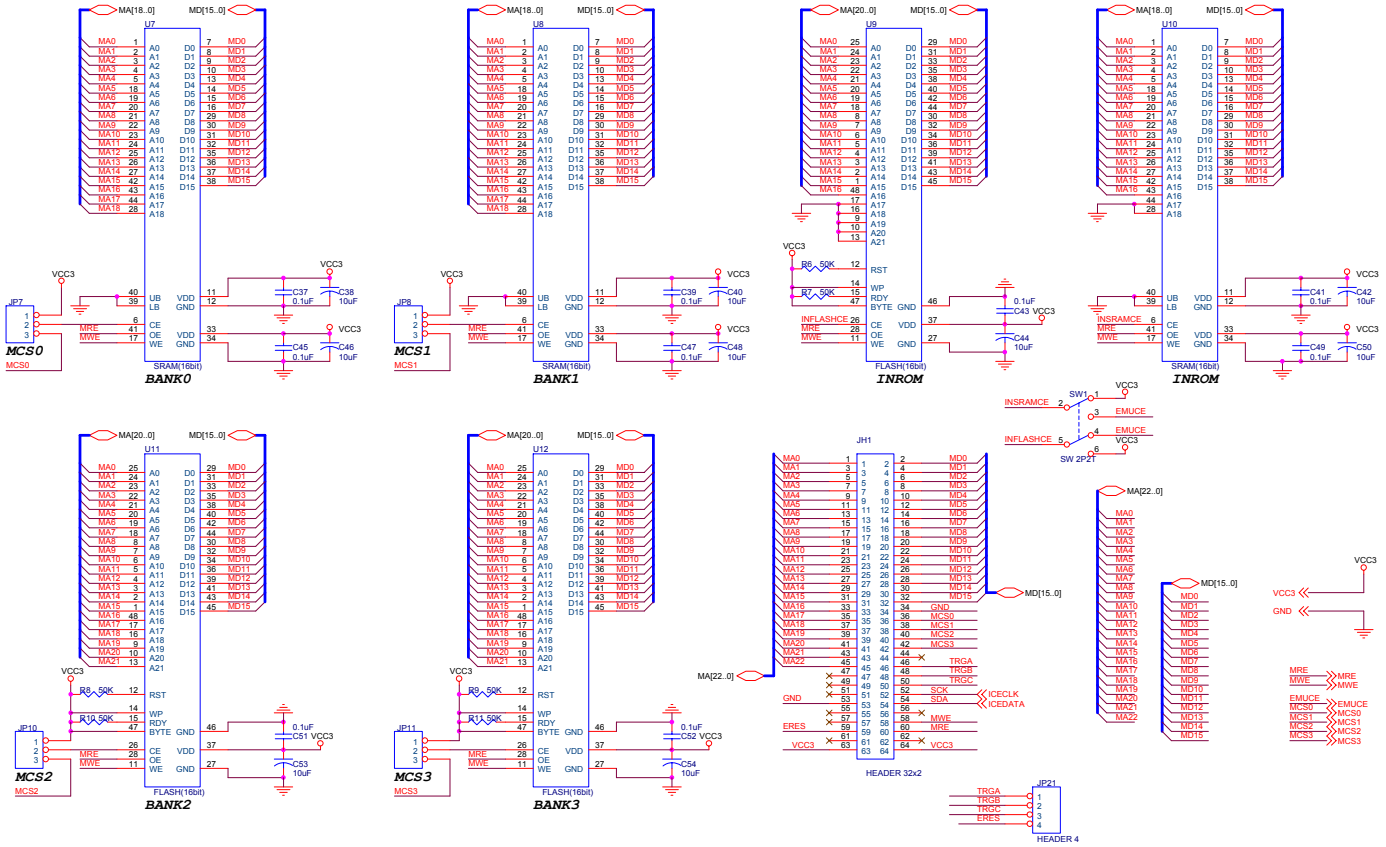

IOD0
IOD1
IOD2
IOD3
HEADER 3
CS3
CS2
CS1
CS0
D0
D1
D2
D3


IOC15
IOC14
HEADER 3
MRE
MWE
C15
C14

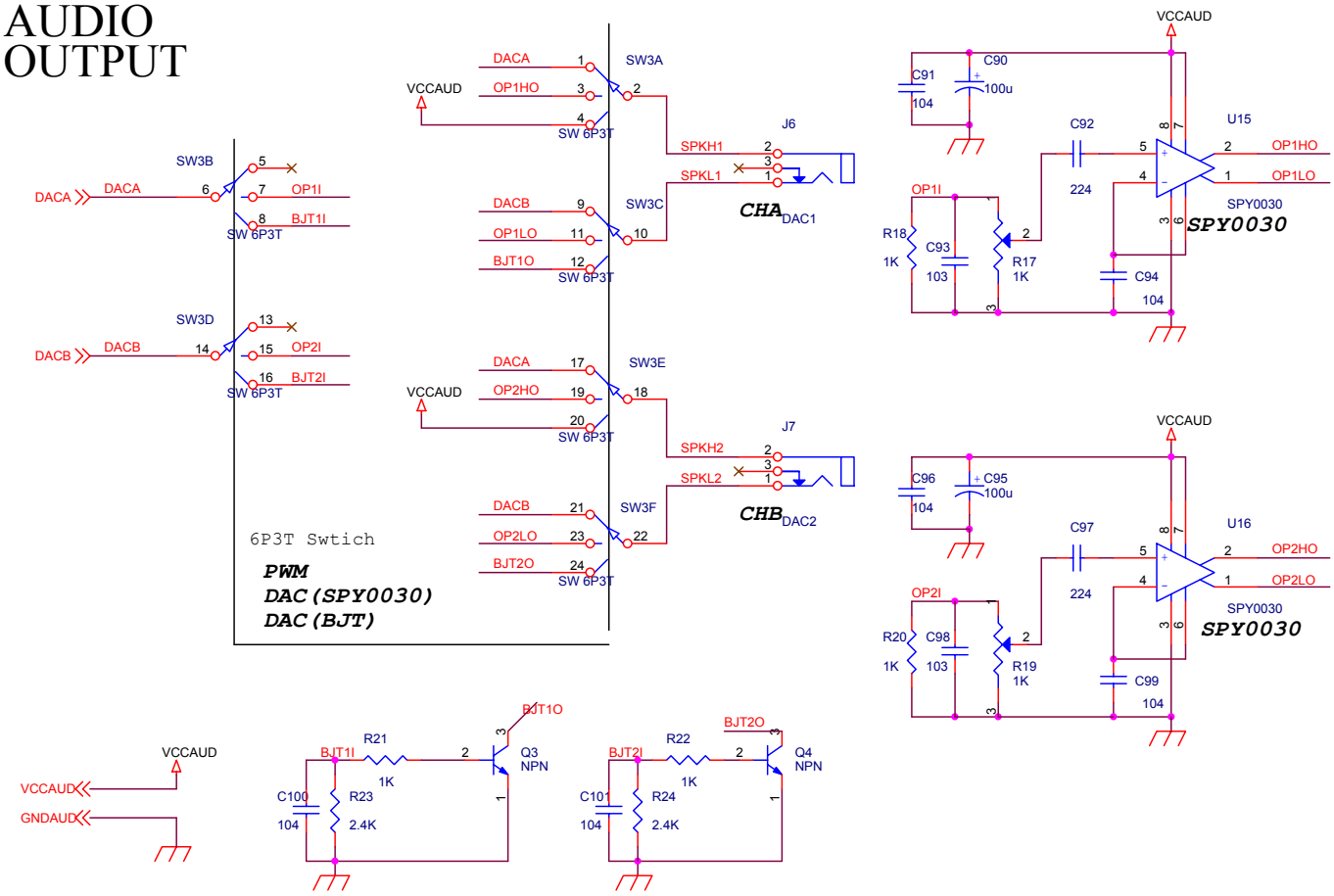

LCDD1
LCDD2
LCDD3
IOD4
IOD5
IOD6

MA23D14
MA24D15
IOD14
IOD15

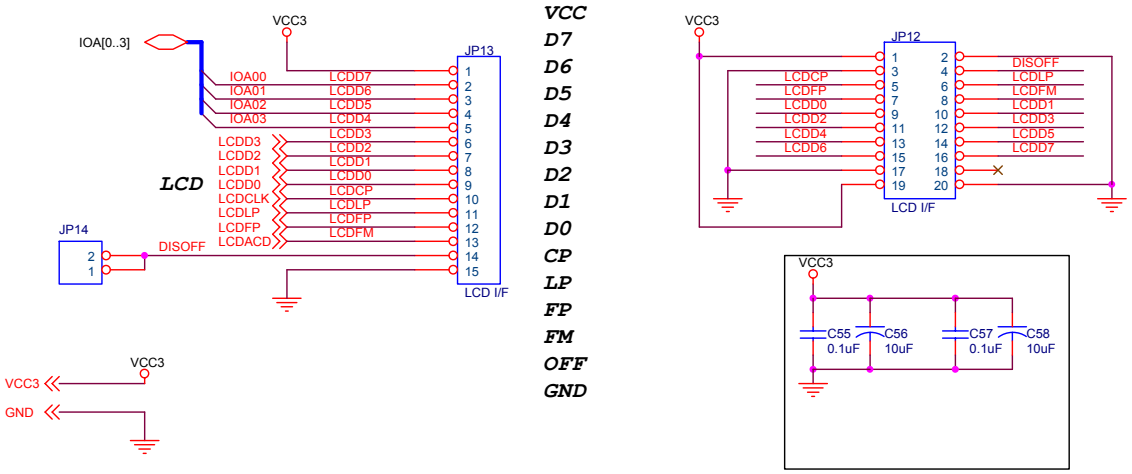
MEMORY


U7
SRAM(16bit)
BANK0
MCS0
U8
SRAM(16bit)
BANK1
MCS1
U9
FLASH(16bit)
INROM
U10
SRAM(16bit)
INROM
U11
FLASH(16bit)
BANK2
MCS2
U12
FLASH(16bit)
BANK3
MCS3
JH1
HEADER 32x2
JP21
HEADER 4
SW1
SW 2P2T
MA[18..0]
MD[15..0]
MA[20..0]
MA[22..0]
VCC3
INSRAMCE
INFLASHCE
EMUCE
ICECLK
ICEDATA
TRGA
TRGB
TRGC
ERES
MRE
MWE
MCS0
MCS1
MCS2
MCS3

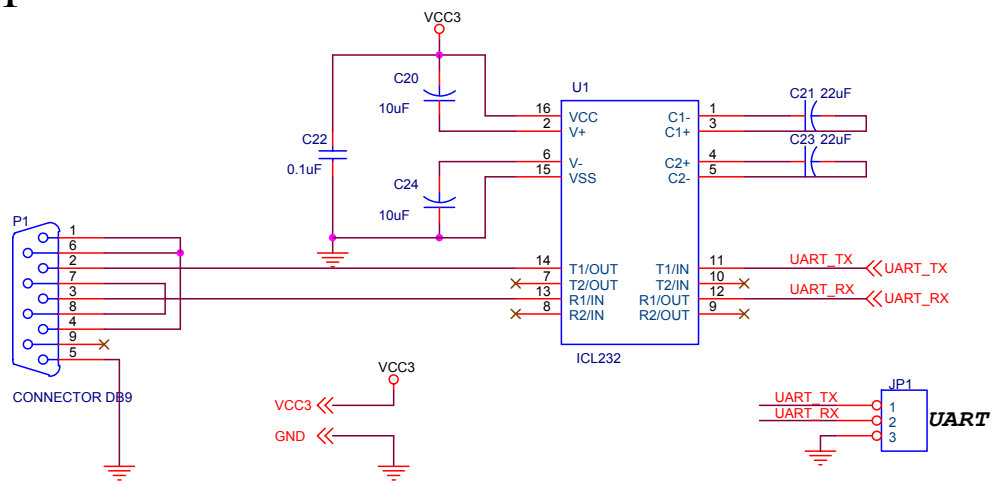
AUDIO OUTPUT



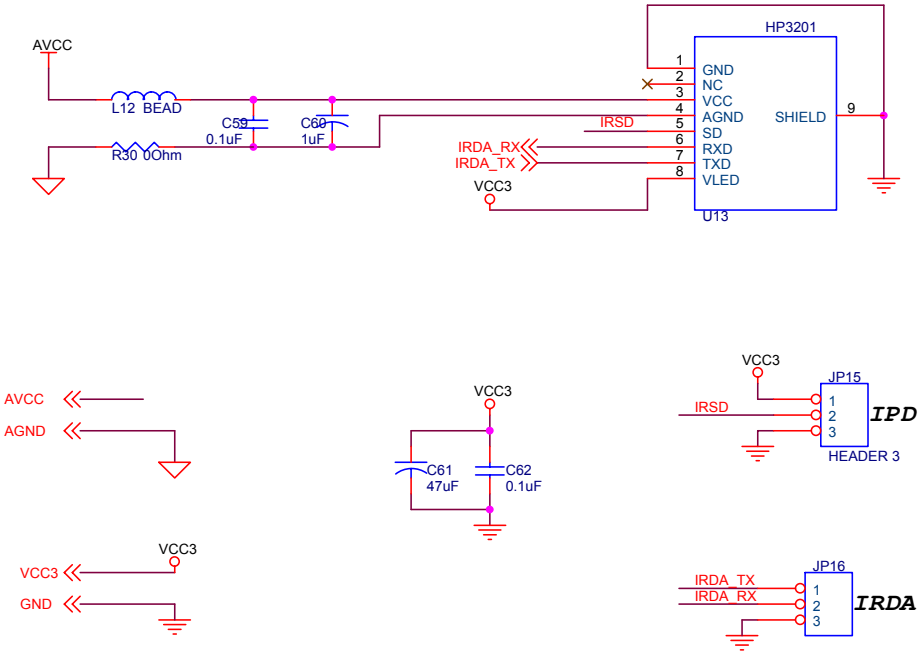
LCD


VCC3
IOA[0..3]
JP13
IOA00
IOA01
IOA02
IOA03
LCDD7
LCDD6
LCDD5
LCDD4
LCDD3
LCDD2
LCDD1
LCDD0
LCDCP
LCDLP
LCDFP
LCDFM
LCD
LCDD3
LCDD2
LCDD1
LCDD0
LCDCLK
LCDLP
LCDFP
LCDACD
JP14
DISOFF
LCD I/F
VCC3
VCC3
GND
VCC
D7
D6
D5
D4
D3
D2
D1
D0
CP
LP
FP
FM
OFF
GND
VCC3
JP12
LCDCP
LCDFP
LCDD0
LCDD2
LCDD4
LCDD6
DISOFF
LCDLP
LCDFM
LCDD1
LCDD3
LCDD5
LCDD7
LCD I/F
VCC3
C55 0.1uF
C56 10uF
C57 0.1uF
C58 10uF

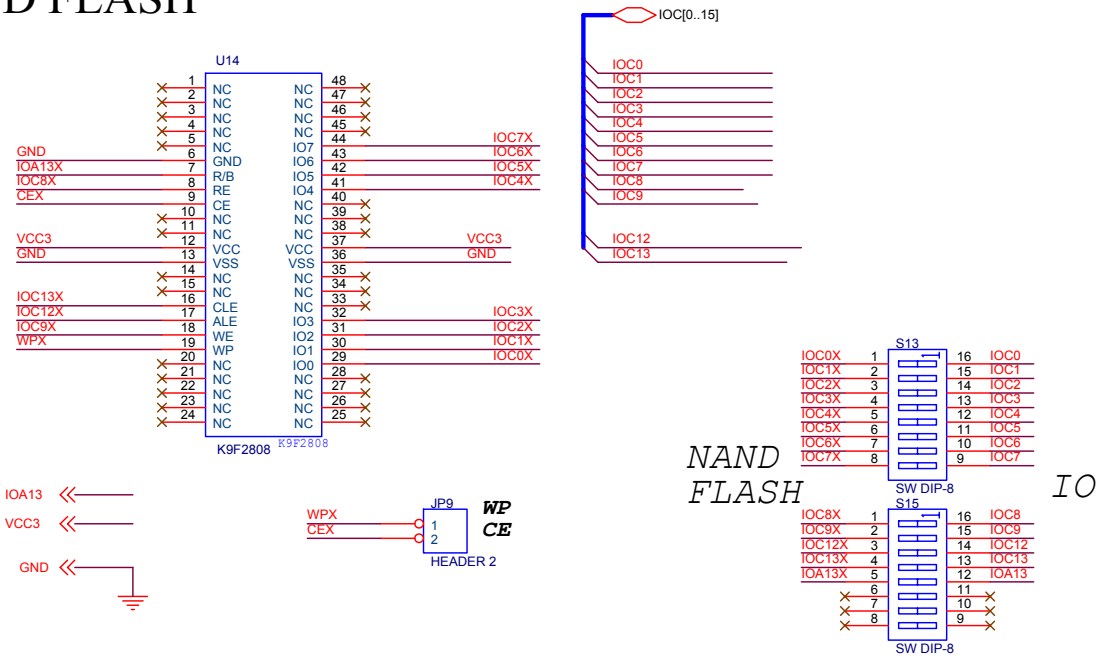
UART


VCC3
C20
10uF
C22
0.1uF
C24
10uF
U1
VCC
V+
V-
VSS
C1-
C1+
C2+
C2-
C21 22uF
C23 22uF
T1/OUT
T2/OUT
R1/IN
R2/IN
T1/IN
T2/IN
R1/OUT
R2/OUT
UART_TX
UART_RX
ICL232
P1
CONNECTOR DB9
VCC3
GND
JP1
UART
UART

IrDA


AVCC
L12 BEAD
R30 0Ohm
C59
0.1uF
C60
1uF
HP3201
GND
NC
VCC
AGND
SD
RXD
TXD
VLED
SHIELD
IRSD
IRDA_RX
IRDA_TX
VCC3
U13
AVCC
AGND
VCC3
C61
47uF
C62
0.1uF
VCC3
JP15
IPD
HEADER 3
VCC3
GND
IRDA_TX
IRDA_RX
JP16
IRDA

NAND FLASH


U14
K9F2808
IOC[0..15]
IOC0
IOC1
IOC2
IOC3
IOC4
IOC5
IOC6
IOC7
IOC8
IOC9
IOC12
IOC13
GND
IOA13X
IOC8X
CEX
VCC3
GND
IOC13X
IOC12X
IOC9X
WPX
IOC7X
IOC6X
IOC5X
IOC4X
VCC3
GND
IOC3X
IOC2X
IOC1X
IOC0X
IOA13
VCC3
GND
WPX
CEX
JP9
HEADER 2
WP
CE
S13
SW DIP-8
S15
SW DIP-8
NAND FLASH
IO

19.8 Header File for C or Assembly Language

Note that the syntax of definition are not exactly the same for both c and assembly header files.

C Header File Example (SPL16.h)

```
#define    P_System_Ctrl_Base        0x7000
#define    P_Memory_Ctrl_Base        0x7020
#define    P_IOPort_Ctrl_Base        0x7060
#define    P_Interrupt_Ctrl_Base     0x70A0
#define    P_TimeBase_Ctrl_Base      0x70B0
#define    P_Timer_Ctrl_Base         0x70C0
#define    P_AudioOuput_Ctrl_Base    0x70E0
#define    P_UARTIrDA_Ctrl_Base      0x7100
#define    P_RTC_Ctrl_Base           0x7120
#define    P_SPI_Ctrl_Base           0x7140
#define    P_LCD_Ctrol_Base          0x7180
#define    P_Miscellaneous_Base      0x71F0
#define    P_NANDFlash_Base          0x7050
#define    P_NANDFlash_DMA_Base      0x7380
```

Assembly Header File Example (SPL16.inc)

```
//////////////////////////////////////////////////////////////////////////////
//  Definitions for Peripheral Control Register
//////////////////////////////////////////////////////////////////////////////
P_System_Ctrl_Base:        .VDEF    0x7000
P_Memory_Ctrl_Base:        .VDEF    0x7020
P_IOPort_Ctrl_Base:        .VDEF    0x7060
P_Interrupt_Ctrl_Base:     .VDEF    0x70A0
P_TimeBase_Ctrl_Base:      .VDEF    0x70B0
P_Timer_Ctrl_Base:         .VDEF    0x70C0
P_AudioOuput_Ctrl_Base:    .VDEF    0x70E0
P_UARTIrDA_Ctrl_Base:      .VDEF    0x7100
P_RTC_Ctrl_Base:           .VDEF    0x7120
P_SPI_Ctrl_Base:           .VDEF    0x7140
P_LCD_Ctrol_Base:          .VDEF    0x7180
P_Miscellaneous_Base:      .VDEF    0x71F0
P_NANDFlash_Base           .VDEF    0x7050
P_NANDFlash_DMA_Base       ,VDEF    0x7380

//////////////////////////////////////////////////////////////////////////////
//  System Control Registers
//      Base Address Start from 0x7000
```

```
////////////////////////////////////////////////////////////////////////////
P_BodyID:                                   .VDEF   P_System_Ctrl_Base+0x00
P_Reset_Flag:                               .VDEF   P_System_Ctrl_Base+0x06
P_Clock_Ctrl:                               .VDEF   P_System_Ctrl_Base+0x07
P_LVR_Ctrl:                                 .VDEF   P_System_Ctrl_Base+0x08
P_LVD_Ctrl:                                 .VDEF   P_System_Ctrl_Base+0x09
P_Watchdog_Ctrl:                            .VDEF   P_System_Ctrl_Base+0x0A
P_Watchdog_Clear:                           .VDEF   P_System_Ctrl_Base+0x0B
P_WAIT:                                     .VDEF   P_System_Ctrl_Base+0x0C
P_HALT:                                     .VDEF   P_System_Ctrl_Base+0x0D
P_SLEEP:                                    .VDEF   P_System_Ctrl_Base+0x0E


////////////////////////////////////////////////////////////////////////////
//    Memory Control Registers
//          Base Address Start from 0x7020
////////////////////////////////////////////////////////////////////////////
P_MCS0_Ctrl:                                .VDEF   P_Memory_Ctrl_Base+0x00
P_MCS1_Ctrl:                                .VDEF   P_Memory_Ctrl_Base+0x01
P_MCS2_Ctrl:                                .VDEF   P_Memory_Ctrl_Base+0x02
P_MCS3_Ctrl:                                .VDEF   P_Memory_Ctrl_Base+0x03
P_MA16_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x10
P_MA17_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x11
P_MA18_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x12
P_MA19_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x13
P_MA20_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x14
P_MA21_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x15
P_MA22_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x16
P_MCS0_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x20
P_MCS1_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x21
P_MCS2_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x22
P_MCS3_Disable:                             .VDEF   P_Memory_Ctrl_Base+0x23
P_MRE_Disable:                              .VDEF   P_Memory_Ctrl_Base+0x28
P_MWE_Disable:                              .VDEF   P_Memory_Ctrl_Base+0x29
P_MAdr_Status:                              .VDEF   P_Memory_Ctrl_Base+0x2A
P_MCtrl_Status:                             .VDEF   P_Memory_Ctrl_Base+0x2B
P_InMEM_Ctrl:                               .VDEF   P_Memory_Ctrl_Base+0x2C

////////////////////////////////////////////////////////////////////////////
//    I/O Port Control Registers
//          Base Address Start from 0x7060
////////////////////////////////////////////////////////////////////////////
P_IOA_Data:                                 .VDEF   P_IOPort_Ctrl_Base+0x00
P_IOA_Buffer:                               .VDEF   P_IOPort_Ctrl_Base+0x01
```

```
P_IOA_Dir:                          .VDEF   P_IOPort_Ctrl_Base+0x02
P_IOA_Attrib:                       .VDEF   P_IOPort_Ctrl_Base+0x03
P_IOA_Latch:                        .VDEF   P_IOPort_Ctrl_Base+0x04
P_IOB_Data:                         .VDEF   P_IOPort_Ctrl_Base+0x08
P_IOB_Buffer:                       .VDEF   P_IOPort_Ctrl_Base+0x09
P_IOB_Dir:                          .VDEF   P_IOPort_Ctrl_Base+0x0A
P_IOB_Attrib:                       .VDEF   P_IOPort_Ctrl_Base+0x0B
P_IOB_Latch:                        .VDEF   P_IOPort_Ctrl_Base+0x0C
P_IOC_Data:                         .VDEF   P_IOPort_Ctrl_Base+0x10
P_IOC_Buffer:                       .VDEF   P_IOPort_Ctrl_Base+0x11
P_IOC_Dir:                          .VDEF   P_IOPort_Ctrl_Base+0x12
P_IOC_Attrib:                       .VDEF   P_IOPort_Ctrl_Base+0x13
P_IOD_Data:                         .VDEF   P_IOPort_Ctrl_Base+0x18
P_IOD_Buffer:                       .VDEF   P_IOPort_Ctrl_Base+0x19
P_IOD_Dir:                          .VDEF   P_IOPort_Ctrl_Base+0x1A
P_IOD_Attrib:                       .VDEF   P_IOPort_Ctrl_Base+0x1B

////////////////////////////////////////////////////////////////////////////
//    Interrupt Control Registers
//          Base Address Start from 0x70A0
////////////////////////////////////////////////////////////////////////////
P_INT_Status1:                      .VDEF   P_Interrupt_Ctrl_Base+0x00
.DEFINE     C_INT_EXTA              0x0001;
.DEFINE     C_INT_EXTB              0x0002;
,DEFINE     C_INT_NFREADOK          0x0004;
.DEFINE     C_INT_NFWRITEOK         0x0008
.DEFINE     C_INT_AUDAFIFOEmpty     0x0010;
.DEFINE     C_INT_AUDBFIFOEmpty     0x0020;
.DEFINE     C_INT_ADCAutoSample     0x0080;
.DEFINE     C_INT_FramePulse        0x0200;
.DEFINE     C_INT_SPI               0x0400;
.DEFINE     C_INT_UARTIrDA          0x0800;
.DEFINE     C_INT_Key               0x8000;
P_INT_Status2:                      .VDEF   P_Interrupt_Ctrl_Base+0x01
.DEFINE     C_INT_HMS               0x0001;
.DEFINE     C_INT_Alarm             0x0002;
.DEFINE     C_INT_Schedule          0x0004;
.DEFINE     C_INT_LVD               0x0080;
.DEFINE     C_INT_TimeBaseA         0x0100;
.DEFINE     C_INT_TimeBaseB         0x0200;
.DEFINE     C_INT_TimeBaseC         0x0400;
.DEFINE     C_INT_TimerA            0x1000;
.DEFINE     C_INT_TimerB            0x2000;
.DEFINE     C_INT_TimerC            0x4000;
```

```
P_INT_Priority1:                    .VDEF   P_Interrupt_Ctrl_Base+0x04
P_INT_Priority2:                    .VDEF   P_Interrupt_Ctrl_Base+0x05
P_MINT_Ctrl:                        .VDEF   P_Interrupt_Ctrl_Base+0x08
////////////////////////////////////////////////////////////////////////
//  TimeBase Control Registers
//      Base Address Start from 0x70B0
////////////////////////////////////////////////////////////////////////
P_TimeBaseA_Ctrl:                   .VDEF   P_TimeBase_Ctrl_Base+0x00
P_TimeBaseB_Ctrl:                   .VDEF   P_TimeBase_Ctrl_Base+0x01
P_TimeBaseC_Ctrl:                   .VDEF   P_TimeBase_Ctrl_Base+0x02
P_TimeBase_Reset:                   .VDEF   P_TimeBase_Ctrl_Base+0x08

////////////////////////////////////////////////////////////////////////
//  Timer Control Registers
//      Base Address Start from 0x70C0
////////////////////////////////////////////////////////////////////////
P_TimerA_Ctrl:                      .VDEF   P_Timer_Ctrl_Base+0x00
P_TimerA_CCP_Ctrl:                  .VDEF   P_Timer_Ctrl_Base+0x01
P_TimerA_Preload:                   .VDEF   P_Timer_Ctrl_Base+0x02
P_TimerA_CCP_Reg:                   .VDEF   P_Timer_Ctrl_Base+0x03
P_TimerA_CCP_Reg2:                  .VDEF   P_Timer_Ctrl_Base+0x04
P_TimerB_Ctrl:                      .VDEF   P_Timer_Ctrl_Base+0x08
P_TimerB_CCP_Ctrl:                  .VDEF   P_Timer_Ctrl_Base+0x09
P_TimerB_Preload:                   .VDEF   P_Timer_Ctrl_Base+0x0A
P_TimerB_CCP_Reg:                   .VDEF   P_Timer_Ctrl_Base+0x0B
P_TimerB_CCP_Reg2:                  .VDEF   P_Timer_Ctrl_Base+0x0C
P_TimerC_Ctrl:                      .VDEF   P_Timer_Ctrl_Base+0x10
P_TimerC_Preload:                   .VDEF   P_Timer_Ctrl_Base+0x12

////////////////////////////////////////////////////////////////////////
//  Audio Output Control Registers
//      Base Address Start from 0x70E0
////////////////////////////////////////////////////////////////////////
P_AudioOut_Ctrl:                    .VDEF   P_AudioOuput_Ctrl_Base+0x00
P_CHA_Ctrl:                         .VDEF   P_AudioOuput_Ctrl_Base+0x10
P_CHA_Data:                         .VDEF   P_AudioOuput_Ctrl_Base+0x11
P_CHA_FIFO:                         .VDEF   P_AudioOuput_Ctrl_Base+0x12
P_CHB_Ctrl:                         .VDEF   P_AudioOuput_Ctrl_Base+0x18
P_CHB_Data:                         .VDEF   P_AudioOuput_Ctrl_Base+0x19
P_CHB_FIFO:                         .VDEF   P_AudioOuput_Ctrl_Base+0x1A

////////////////////////////////////////////////////////////////////////
//  UART / IrDA Control Registers
//      Base Address Start from 0x7100
```

```
////////////////////////////////////////////////////////////////////////////
P_UARTIrDA_Data:                    .VDEF   P_UARTIrDA_Ctrl_Base+0x00
P_UART_RXStatus:                    .VDEF   P_UARTIrDA_Ctrl_Base+0x01
P_UARTIrDA_Ctrl:                    .VDEF   P_UARTIrDA_Ctrl_Base+0x02
P_UARTIrDA_BaudRate:                .VDEF   P_UARTIrDA_Ctrl_Base+0x03
P_UARTIrDA_Status:                  .VDEF   P_UARTIrDA_Ctrl_Base+0x04
P_IrDA_Ctrl:                        .VDEF   P_UARTIrDA_Ctrl_Base+0x08


////////////////////////////////////////////////////////////////////////////
//  RTC Control Regsiters
//      Base Address Start from 0x7120
////////////////////////////////////////////////////////////////////////////
P_Second:                           .VDEF   P_RTC_Ctrl_Base+0x00
P_Minute:                           .VDEF   P_RTC_Ctrl_Base+0x01
P_Hour:                             .VDEF   P_RTC_Ctrl_Base+0x02
P_Alarm_Second:                     .VDEF   P_RTC_Ctrl_Base+0x04
P_Alarm_Minute:                     .VDEF   P_RTC_Ctrl_Base+0x05
P_Alarm_Hour:                       .VDEF   P_RTC_Ctrl_Base+0x06
P_RTC_Ctrl:                         .VDEF   P_RTC_Ctrl_Base+0x14
.DEFINE     C_RTCEN                 0x8000
.DEFINE     C_ALMEN                 0x0400
.DEFINE     C_HMSEN                 0x0200
.DEFINE     C_SCHEMEN               0x0100
.DEFINE     C_16HzSchd              0x0000
.DEFINE     C_32HzSchd              0x0001
.DEFINE     C_64HzSchd              0x0002
.DEFINE     C_128HzSchd             0x0003
.DEFINE     C_256HzSchd             0x0004
.DEFINE     C_512HzSchd             0x0005
.DEFINE     C_1024HzSchd            0x0006
.DEFINE     C_2048HzSchd            0x0007
P_RTC_INT_Status:                   .VDEF   P_RTC_Ctrl_Base+0x15
P_RTC_INT_Ctrl:                     .VDEF   P_RTC_Ctrl_Base+0x16
.DEFINE     C_HSecondINT            0x0001
.DEFINE     C_SecondINT             0x0002
.DEFINE     C_MinuteINT             0x0004
.DEFINE     C_HourINT               0x0008
.DEFINE     C_ScheduleINT           0x0100
.DEFINE     C_AlarmINT              0x0400

////////////////////////////////////////////////////////////////////////////
//  SPI Control Registers
```

```
//    Base Address Start from 0x7140
////////////////////////////////////////////////////////////////////////////
P_SPI_Ctrl:                                     .VDEF    P_SPI_Ctrl_Base+0x00
P_SPI_TXStatus:                                 .VDEF    P_SPI_Ctrl_Base+0x01
P_SPI_TXData:                                   .VDEF    P_SPI_Ctrl_Base+0x02
P_SPI_RXStatus:                                 .VDEF    P_SPI_Ctrl_Base+0x03
P_SPI_RXData:                                   .VDEF    P_SPI_Ctrl_Base+0x04


////////////////////////////////////////////////////////////////////////////
//    LCD Control Registers
//          Base Address Start from 0x7180
////////////////////////////////////////////////////////////////////////////
P_LCD_Ctrl:                                     .VDEF    P_LCD_Ctrol_Base+0x00
P_LCD_Clock:                                    .VDEF    P_LCD_Ctrol_Base+0x01
P_LCD_Segment:                                  .VDEF    P_LCD_Ctrol_Base+0x02
P_LCD_Common:                                   .VDEF    P_LCD_Ctrol_Base+0x03
P_LCD_Buffer_LowAdr:                            .VDEF    P_LCD_Ctrol_Base+0x04
P_LCD_Buffer_HighAdr:                           .VDEF    P_LCD_Ctrol_Base+0x05
P_LCD_Buffer_Offset:                            .VDEF    P_LCD_Ctrol_Base+0x06
P_LCD_Timing_Ctrl:                              .VDEF    P_LCD_Ctrol_Base+0x07


////////////////////////////////////////////////////////////////////////////
//P_Miscellaneous_Base
//          Base Address Start from 0x71F0
////////////////////////////////////////////////////////////////////////////
P_Byte_Swap:                                    .VDEF    P_Miscellaneous_Base+0x00
P_Nibble_Swap:                                  .VDEF    P_Miscellaneous_Base+0x01
P_TwoBit_Swap:                                  .VDEF    P_Miscellaneous_Base+0x02
P_Bit_Reverse:                                  .VDEF    P_Miscellaneous_Base+0x03


////////////////////////////////////////////////////////////////////////////
//P_NANDFlash_Base
//          Base Address Start from 0x7050
////////////////////////////////////////////////////////////////////////////
P_NF_Ctrl:                                      .VDEF    P_NANDFlash_Base+0x00
P_NF_Command:                                   .VDEF    P_NANDFlash_Base+0x01
P_NF_Address:                                   .VDEF    P_NANDFlash_Base+0x02
P_NF_DataW:                                     .VDEF    P_NANDFlash_Base+0x03
P_NF_DataR:                                     .VDEF    P_NANDFlash_Base+0x04
P_NF_Data:                                      .VDEF    P_NANDFlash_Base+0x05
P_ECC_Ctrl:                                     .VDEF    P_NANDFlash_Base+0x06
```

```
P_ECC_LPRL:                     .VDEF   P_NANDFlash_Base+0x07
P_ECC_LPRH:                     .VDEF   P_NANDFlash_Base+0x08
P_ECC_CPR:                      .VDEF   P_NANDFlash_Base+0x09
P_ECC_LPRCKL:                   .VDEF   P_NANDFlash_Base+0x0a
P_ECC_LPRCKH:                   .VDEF   P_NANDFlash_Base+0x0b
P_ECC_CPCKR:                    .VDEF   P_NANDFlash_Base+0x0c
P_ECC_Err0:                     .VDEF   P_NANDFlash_Base+0x0d
P_ECC_Err1:                     .VDEF   P_NANDFlash_Base+0x0e

////////////////////////////////////////////////////////////////////////////
//P_NANDFlash_DMA_Base
//        Base Address Start from 0x7380
////////////////////////////////////////////////////////////////////////////
P_NF_DMA_AddrL:                 .VDEF   P_NANDFlash_DMA_Base+0
P_NF_DMA_AddrH:                 .VDEF   P_NANDFlash_DMA_Base+1
P_NF_DMA_Size:                  .VDEF   P_NANDFlash_DMA_Base+2
P_NF_DMA_Write:                 .VDEF   P_NANDFlash_DMA_Base+3
P_NF_DMA_Read:                  .VDEF   P_NANDFlash_DMA_Base+4
P_NF_DMA_WriteCtrl:             .VDEF   P_NANDFlash_DMA_Base+5
P_NF_DMA_ReadCtrl:              .VDEF   P_NANDFlash_DMA_Base+6

////////////////////////////////////////////////////////////////////////////
// End of include file
////////////////////////////////////////////////////////////////////////////
```

19.9 Differences Between Emulation Chip and Real Chip

1. If the LCD Buffer is external (on CS0 or CS1 or CS2 or CS3), not on internal SRAM, real chip does not have CPU performance downgrade issue. However, in the EMU chip, CPU performance will downgrade if LCD interface is enabled. In other words, for the same software procedure, the real chip runs faster than the EMU chip if the LCD interface is enabled. Therefore, do NOT use CPU cycle to make very important and accuracy-demanded timing.

For example:

```
//**************************
//**************************
F_Delay:
        r1 = 0
L_Loop?:
        r1 = r1 - 1
        jnz     L_Loop?
        retf
//**************************
//**************************
```

If, in EMU chip, this delay is 100ms,
but after you get your new real chip code,
This delay might be 150ms!

2. On a real chip, we can set MA16 as GPIO IOD7, MWE as GPIO IOC14, and MRE as GPIO IOC15. But on the EMU chip, we cannot set above 4 memory control pins as GPIOs. Sunplus recommends never setting MA16, MWE, and MRW as GPIO.

3. Extra Wait Cycle of Internal Mask ROM can be "+1" while system clock is set to 51.6096 MHz. But for an emulation chip, Internal Mask Wait Cycle should be set to "+2", that is, 0x704C (P_InMem_Ctrl) should be set to 0x0002. Sunplus recommends always setting the extra wait cycle to “2” for internal mask ROM.

4. On a real chip, Internal mask ROM can not be an LCD buffer. But on the emulation chip, internal mask ROM can be an LCD Buffer. Note that this important difference might cause failure to release code of the real chip.

19.10 Differences Between SPL16256A and SPL161001A

1. The SPL161001A has a built-in oscillator to save power when there is no need of high system clock frequency.
2. Low voltage reset is always enabled in the SPL16256A, but it can be disabled in the SPL161001A by writing P_LVR_Ctrl.15.
3. The SPL161001A has a NAND Flash memory interface with DMA channel. The interface includes ECC logic to gain better data reliability.
4. The LCD interface of the SPL161001A only supports up to 16 gray-level display.
5. The internal SRAM size of the SPL161001A is 8K word, and the internal ROM size is 128K words.
6. The SPI interface in the SPL161001 supports a 16-byte-deep FIFO.
7. The SPL16100A has a Sunplus Serial Interface to utilize Sunplus serial memory products.

19.11 CPU Performance Downgrading Issue

LCD Display and System Performance

The SPL161001A can use internal SRAM or external memory as display buffer. If the LCD buffer is configured on external memory (the display data should be read from external memory) and running program is on the other external memory, these two memories will share the same system bus. Therefore, CPU performance is downgraded.

As depicted in the following table, there are six memory-access conditions in the SPL161001A. When the LCD and CPU access the same region of memory, the performance of the CPU is downgraded. In this situation, SUNPLUS recommends taking the following action to improve performance:

(1) Copy the instructions into SRAM, and execute it in internal SRAM.
(2) Configure high-speed-required algorithm into internal ROM (such as speech algorithm).

	LCD uses internal SRAM as data buffer	**LCD uses external memory as data buffer**
Software Program On Internal RAM	CPU performance downgraded	CPU performance not downgraded
Software Program On Internal ROM	CPU performance not downgraded	CPU performance not downgraded
Software Program On External memory	CPU performance not downgraded	CPU performance downgraded

If it is difficult to take the action mentioned above, the user should take care concerning the calculation of the occupied data bus bandwidth by LCD. The factors include the wait cycle for external memory, the mono/gray mode, the size of the LCD panel, and the frame rate of mono/gray display in the LCD panel.

The Formula for Downgrading Factor

The percentage of occupied bus bandwidth by LCD interface equals to:

LCDBPP x (2+LCD_WAIT) x 12 / (192 x N)

where N = SYSCLK / (X-size x Y-size x LCD Frame rate)
N equals to the value P_LCD_Clock + 2

LCDBPP = display mode = 1- or 4-bit per pixel

LCD_WAIT = extra wait cycle of LCD buffer for enough external memory accessing time

This value is the same with the value in P_LCD_Ctrl.b3~b0

SYSCLK = PLL clock frequency or OSC clock frequency

X-size = the width of LCD panel

This value is the same with the value of P_LCD_Segment+1

Y-size = the height of LCD panel

This value is the same with the value of P_LCD_Common+1

LCD Frame rate = the enough frequency for mono/gray display

For example:

PLL clock = 51.6096 MHz

LCD_WAIT= 1 (It means the external memory access time is less than 3 PLL clock cycle.)

X-size x Y-size x Frame rate	N	BPP=1	BPP=4
320 * 240 * 168	4	4.6875%	18.75%
320 * 240 * 134.4	5	3,75%	15%
160 * 160 * 134.4	15	1.25%	5%

19.12 Audio Output Components Selection Guide

DAC Output RC circuit vs. frequency response

For DAC audio output, an external audio driver circuit is required. The bandwidth of the audio driver circuit can affect the perceived audio performance. The user can adjust the bandwidth according to the reference table below. R is the effective resistance of the circuit and C is the effective capacitance. Please refer to the SPL161001A application circuit for details.

For example, in SPL161001A Emulation board V1.1:

BJT Amplifier

If R21(R22)=1Kohm, R23(R24)=2.4Kohm

C100 (C101) = 0.047uF for F3dB=4 KHz ;

C100 (C101) = 0.01uF for F3dB=20 KHz

SPY0030 Amplifier

If R17(R19)=1Kohm, R18(R20)=1Kohm

C93 (C98) = 0.1uF for F3dB=3 KHz ;

C93 (C98) = 0.02uF for F3dB=15 KHz

F_{3dB}(3dB frequency)		R (ohm)															
		100	120	180	220	330	390	470	510	560	680	750	820	910	1000	1100	1500
C(uF)	0.0047	338.628	282.19	188.127	153.922	102.614	86.8276	72.0485	66.3976	60.4693	49.7982	45.1504	41.2961	37.2118	33.8628	30.7843	22.5752
	0.01	159.155	132.629	88.4195	72.3432	48.2288	40.809	33.8628	31.2069	28.4205	23.4052	21.2207	19.4092	17.4896	15.9155	14.4686	10.6103
	0.02	79.5775	66.3146	36.1716	36.1716	24.1144	20.4045	15.6034	15.6034	14.2103	11.7026	9.70458	9.70458	8.74478	7.95775	5.30517	5.30517
	0.047	33.8628	28.219	18.8127	15.3922	10.2614	8.68276	7.20485	6.63976	6.04693	4.97982	4.51504	4.12961	3.72118	3.38628	3.07843	2.25752
	0.1	15.9155	13.2629	8.84195	7.23432	4.82288	4.0809	3.38628	3.12069	2.84205	2.34052	2.12207	1.94092	1.74896	1.59155	1.44686	1.06103
	0.22	7.23432	6.0286	4.01907	3.28833	2.19222	1.85495	1.53922	1.41849	1.29184	1.06387	0.96458	0.88223	0.79498	0.72343	0.65767	0.48229
	0.47	3.38628	2.8219	1.88127	1.53922	1.02614	0.86828	0.72048	0.66398	0.60469	0.49798	0.4515	0.41296	0.37212	0.33863	0.30784	0.22575
	1	1.59155	1.32629	0.88419	0.72343	0.48229	0.40809	0.33863	0.31207	0.28421	0.23405	0.21221	0.19409	0.1749	0.15916	0.14469	0.1061

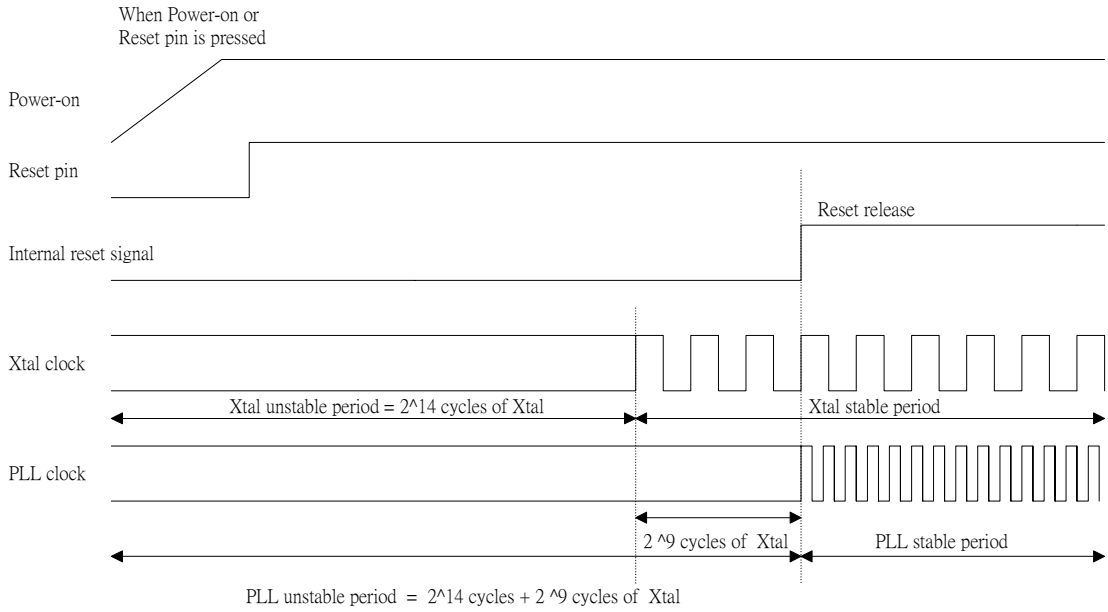
	The frequency range, 5K~4 KHz is for speech application and marked in yellow cells.
	The frequency range, 15K~25K Hz is for Audio application and marked in blue cells..

19.13 32768 Crystal and PLL Power-on Stable Time

When powered on, the SPL161001A will remain in the reset state. After 16384 cycles from the 32768Hz crystal, the reset state will released in about a time of 0.5 seconds.

The reference clock of the PLL is from the 32768Hz crystal oscillator. When the 32768Hz clock is stable, it still takes 512 cycles of the 32768Hz crystal to make the PLL clock stable.

The following diagram shows the detail waveform from power-on to PLL stable period.



Note: While changing PLL clock by control register (P_Clock_Ctrl), the CPU clock will change to 32768HZ for 512 cycles temporarily. So user must take care about real-time applications by counting system clock.

19.14 Reset Type

CPU reset

When Low-Voltage-Reset, watchdog timeout reset CPU, watchdog mode protect reset, power-saving mode protect, or memory write-protect reset occur, it only resets the CPU. The peripheral is not affected by the above reset mechanism, and remains in its original state. The reset flag can be read out to judge which reset happened.

System reset

When Power-on reset or watchdog timeout reset system occur, the SPL161001A (including CPU and all peripherals) is reset to the initial state excluding SRAM. All the flags return to the initial state including reset flag. therefore, the reset flag must be recorded to SRAM to avoid disappearance. In other words, CPU reset is the subset of system reset.

Watchdog timeout reset

When Watchdog timeout reset is activated, the value of 0xAXX5 must be written to the Watchdog clear register within the period that the user defines. Otherwise, CPU reset occurs. Also note that reset target of watchdog-timeout can be selected between CPU and system by software programming (1-bit control register).

Watchdog mode protection reset

When Watchdog timeout reset is activated, the wrong value (not Axx5) is written to the Watchdog clear register, and CPU reset occurs.

Power saving mode protection reset

The SPL161001A defines three power saving modes, including the Wait, Halt, and Standby modes. When entering power saving mode, the Wait, Halt, or Standby mode control register must be written to the corresponding value. If the wrong value is written into the control register, CPU reset occurs.

Memory write-protection reset

The memory applied to SPL161001A can be set as read-only to protect its contents. When any write-action happens, the CPU resets.

Low voltage reset

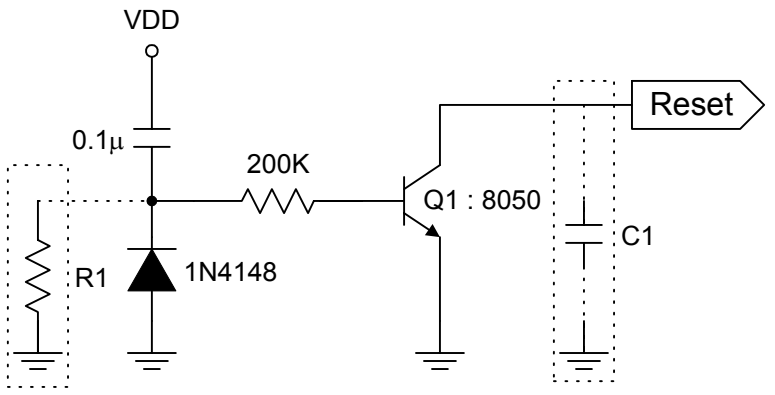
When the operation voltage is lower than 2.5V (or 2.3V), the Low voltage reset mechanism will activate to reset the CPU. The LVR takes the 32768Hz clock source as a sampling clock to save power. Therefore, when the SPL161001A enters the standby mode, LVR is disabled automatically.

Power on reset

When power is supplied to the SPL161001A from 0V, power-on reset occurs. There are some opportunities that the power-on reset does not have due to unstable operation voltage. The Super-reset mechanism is necessary to avoid this condition.

Super-reset circuit

The application circuit is shown below. It detects the variance of operating voltage and keeps the RESET signal in a low state when operating voltage is changing from ground to stable state. When the operating voltage is in a stable state, the Reset signal releases from '0' to '1'.



Note : When the reset circuit applied to the application is more sensitive to generate the reset signal , user can add the resistor (R1) and capacitor (C1) to reduce the sensitivity of the reset signal .

19.15 Important Note for the Setup of Memory Access Time

The SPL161001A provides flexible adjustment of memory access time, including internal ROM, SRAM, register, and external memory. Through the adjustment of memory access time and flexible clock selection, the user can fine-tune the system performance. For example, when the PLL clock is 51.6096MHz, it has 19.3ns per clock cycle. In this case, when accessing the external memory, which has a minimum access time of 55ns, it takes 3 clock cycles to supply enough time for memory accessing.

There are seven independent control registers utilized to set the memory accessing time in the SPL161001A, which include P_InMEM_Ctrl (0x704C),P_MCS0_Ctrl (0x7020), P_MCS1_Ctrl (0x7021), P_MCS2_Ctrl (0x7022), P_MCS3_Ctrl (0x7023), and P_LCD_Ctrl.[3..0] (0x7180.b3~b0).

P_InMEM_Ctrl must be set to fit the access time of the internal SRAM, register, internal ROM or the external memory that try to emulate internal ROM. In general, internal SRAM and peripheral control register is fast enough to use a 2 cycle clock regardless of PLL clock selection. As for internal ROM or the external memory that tries to emulate internal ROM, it takes 60ns access time.

While external memory is used, user must check the datasheet of memory to make sure that there is enough access time. The load on the PCB and operatiing voltage must be seriously considered.

When the LCD interface is activated, P_LCD_Ctrl.[3..0] must be appropriately programmed.

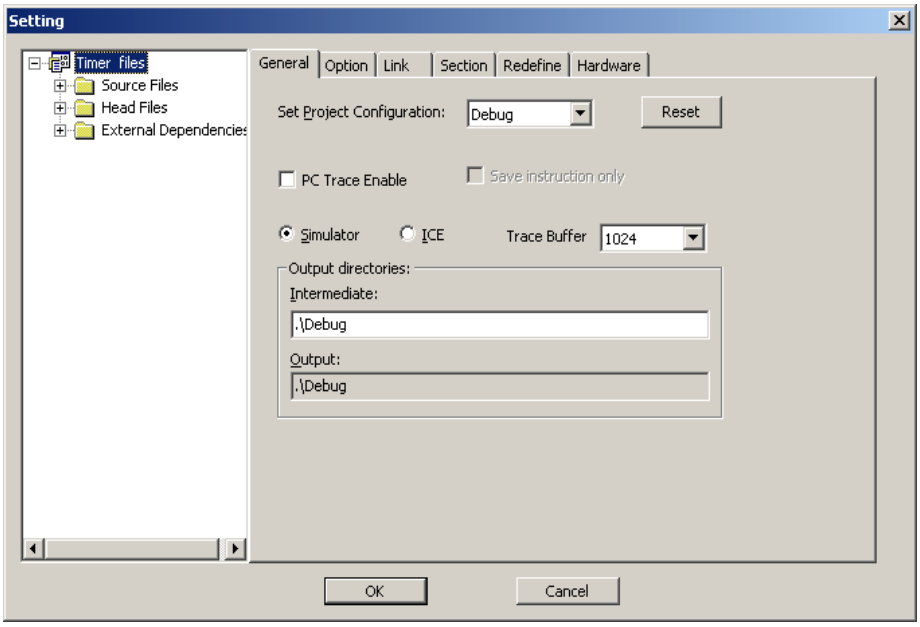
19.16 Project Setting on IDE

For most projects, options are set at project level. Options can be setup for different objects that are created by either system or user. A project can display the structure of options for every object. The options given for the project level also apply to all files in the project.

Set Option for Project

1. Open a project or create a project.
2. Click [Project]→[Setting] to display "Setting" dialog box.
3. Click on each label for further setting.

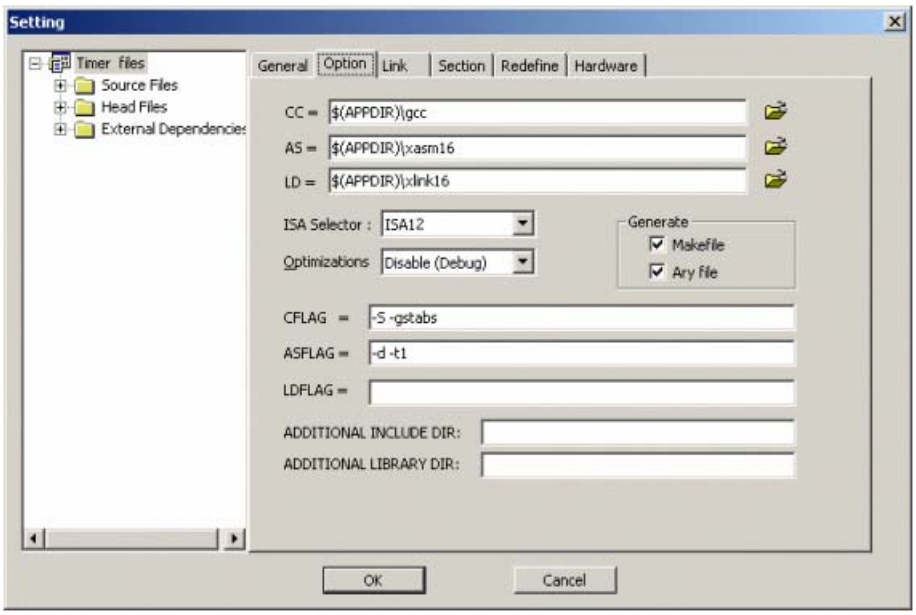
[General]



1. If simulator is chosen, all data will be stored in buffer.
2. Set Project Configuration: Select which version of the project is to be used, debug or release.
3. Simulator /ICE: select u'nSP® IDE running mode. If ICE mode is selected, you should connect an emulation board to your computer through a parallel port.
4. PC track enable: enable PC trace function (make sure the ice board has PC trace capability during ICE mode).
5. Save instruction only: If checked only instruction fetch will be saved in PC trace buffer, otherwise, it will save all memory read/write in PC trace buffer.
6. Tracer buffer size: Specifies the number of bytes used to store operation records.

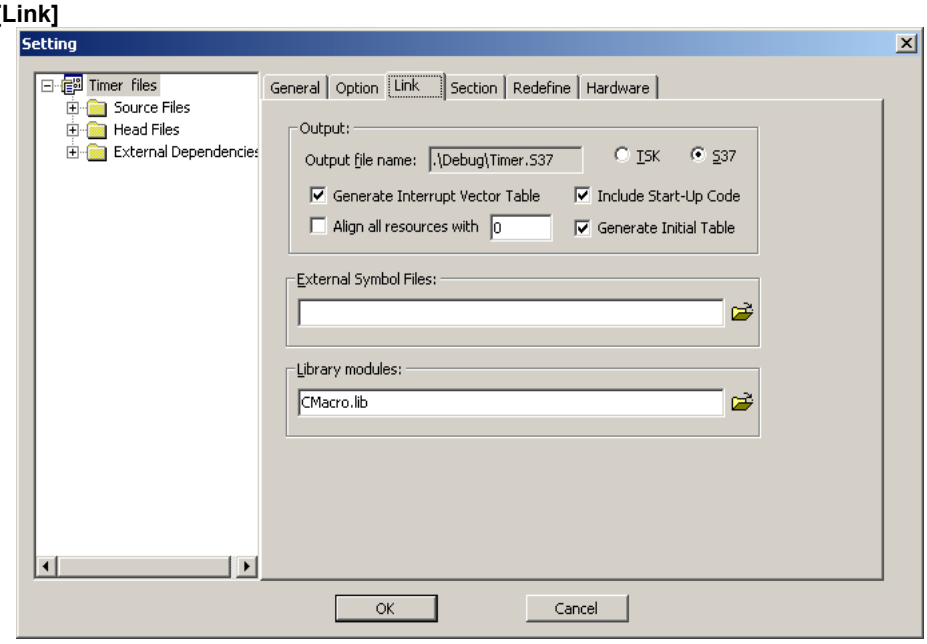
7. Intermediate: Specify directory for intermediate files. Intermediate files normally are generated during compilation.
8. Output: Display the terminal file directory. Normally, it is the same as the intermediate directory.
9. Reset: Reset default configuration.

[Option]



1. CC: Specify the C Compiler location.
2. AS: Specify the Assembler location.
3. LD: Specify the linker location.
4. CFLAG: Specify the C Compiler Operation FLAGS.
5. ASFLAG: Specify the assembler Operation FLAGS.
6. LDFLAG: Specify the linker Operation FLAGS.
7. Optimizations: Select the Optimization Type you want. The optimization flags will be changed automatically.
8. ISA Selector: Select different instruction set of unSP (ISA1.0, ISA1.1, ISA1.2 etc).
9. Makefile: Check if auto updating the makefile.
10. Ary file: Check if auto updating the array file.
11. Additional include dir: Set include file directory.
12. Additional library dir: Set library file directory.

[Link]



1. Output file name: Specify the output file name.

TSK/S37: Types of object file (binary/ASCII (Motorola S37)). Before users define the type of output file, users should <check> both Makefile and Ary file at Option TAB.

2. Generate Interrupt Vector Table:

Uncheck it, if users don't want to produce Interrupt Vector Table in the project output.

3. Include Start-Up Code:

Uncheck it, if users don't want to produce the default start-up code in the project output.

4. Align all resource with:

Check it and input the align base, if users want to align all resource with a specify align base..

5. Generate Initial Table:

Uncheck it, if users don't need an initial table in the project output.

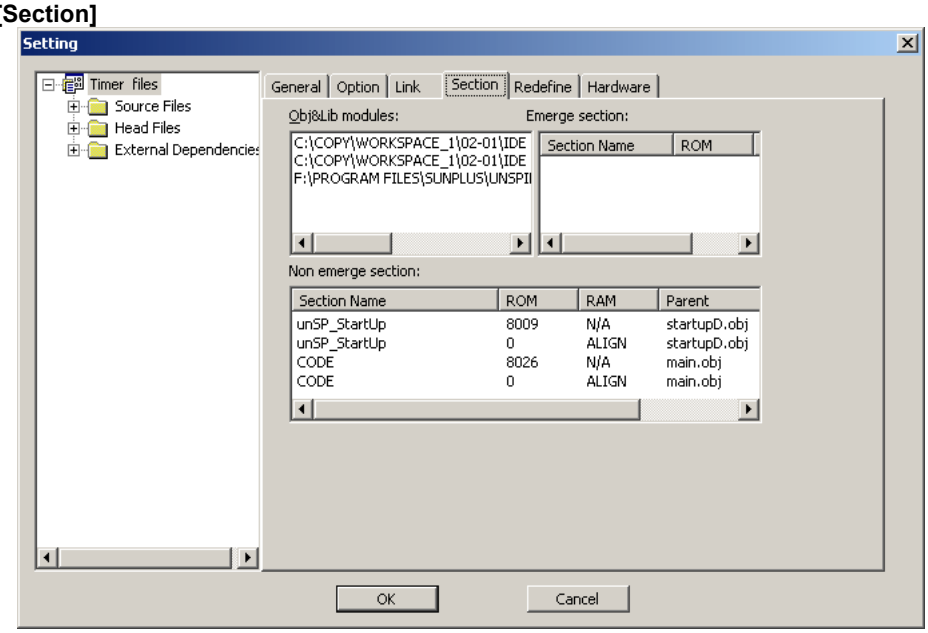
6. External Symbol Files:

Input the other symbol files (*.sym) needed for reference link in the current project.

7. Library modules:

Specify and show all library-modules included in the current project.

[Section]



1. Obj & Lib modules:

   Show all object and library modules in the current project.

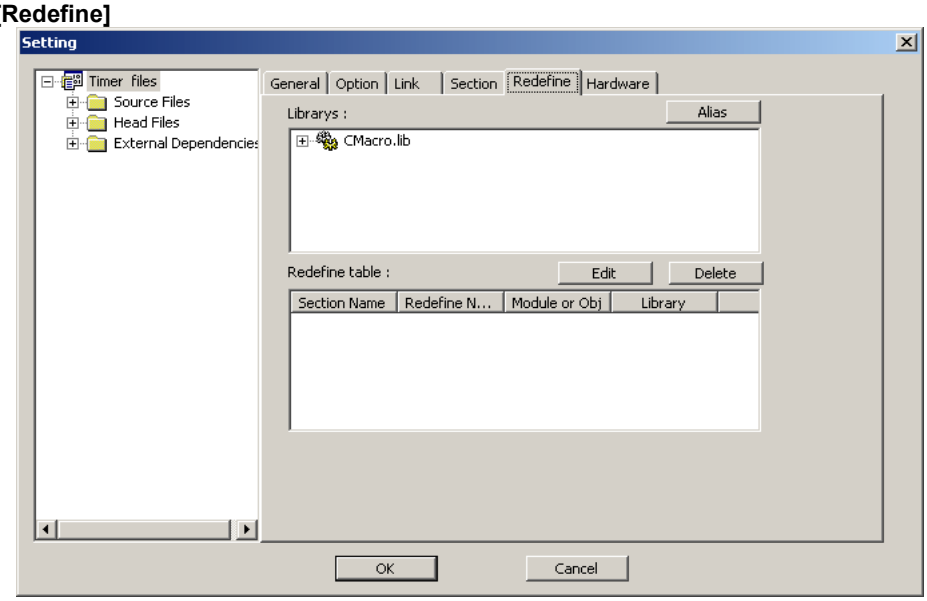
2. Emerge section:

   List the emerged segments in the current project.

3. Non-emerged section:

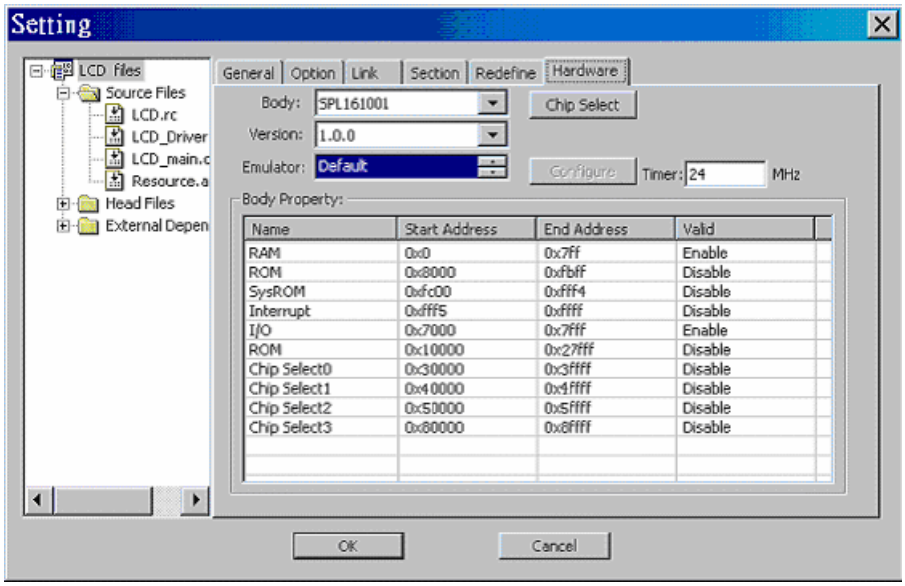
   List the non-emerged sections in current project. You can change the address or align base of these sections by double clicking on the ROM field, and the specified sections will be located at the proper addresses with aligned base after re-linking this project.

[Redefine]



1. Alias: Select a section from the list window of libraries and redefine it as another name.
2. Edit: Edit the selected item in the Redefine table list window. Equal to double click on the item in the Redefine table list.
3. Delete: Delete a selected item in the Redefine table list window.

[Hardware]



1. Body:

   Select a body. The linker and simulator are based on the body description to link and simulate.

   Note

   1. The SPL161001A includes two body selections: SPL161001A and SPL161001A_Flash. The main difference is that SPL161001A_Flash is capable of downloading flash to internal and external memory. In contrast, the SPL161001A does not support the flash downloading function. To download flash in external memory, programmer must select the memory type to “flash” by project → setting → hardware → chip select.
   2. After programmer changes body select, project must be rebuilt.

2. Emulator:

   Select the external device emulator for selected IC. The emulator is a DLL specified at CPT file of correspond body.

3. Timer:

   Set clock frequency of Simulator and Emulator.

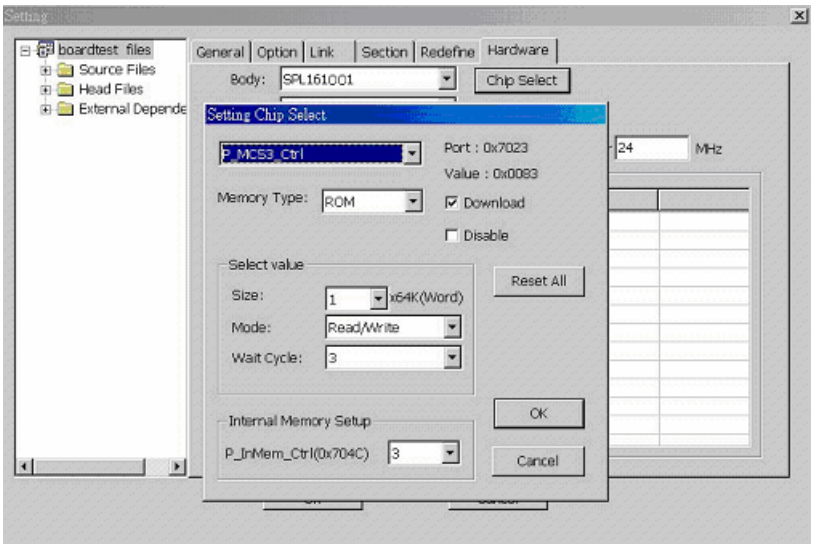
4. Configure:

   Set emulator.

5. Body property:

   Show the memory mapping.

[Chip Select]



Chip Select is the setting of ICE downloading onto EMU board with four external memories (CS0, CS1, CS2 and CS3). Programmer must set it up before downloading program onto EMU board.

Note: This setting is only a reference for downloading program from ICE to EMU board. Programmers must setup properly based on their needs.

1. Memory Chip Select:

   Select which chip to be setup.

2. Memory Type:

   Which type of memory is used, e.g. RAM, ROM, Flash; In download box, check disable if this chip is not used. Therefore, IDE will not download into this memory while re-downloading. The disable sequence must be CS3, CS2, CS1 then CS0.

3. Select value:

   a. Size: setup size of the chip.

      The starting address of the SPL161001A CS# is determined by the previous memory size (CS0 is fixed in 0x30000). Therefore, the starting address of next memory must be appointed. For example, to configure CS0→ 64K-word ROM, CS1 → 128K-word flash, CS2→ 128K-word flash, CS3 → 256K-word flash. That is,

      Set CS0 size to “1” (start address 0x30000)
      Set CS1 size to “2” (start address 0x40000)
      Set CS2 size to “2” (start address 0x60000)

Set CS3 size to “4” (start address 0xA0000)

b. Mode:

Define the memory operation type: Read/Write or Read-Only

c. Wait cycle:

The ICE download speed is 10MHz, i.e. CPU cycle = 100ns. If programmer uses lower access-speed memory, it will have additional wait cycle to prevent download failure. In general, the extra wait cycle can be set as “0”. This is because that access time is seldom larger than 200ns (2 base CPU cycle.

4. Internal Memory Setup:

Setup the extra wait cycle for downloading into the memory that tries to emulate internal Mask ROM.