Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

SL11R
USB Controller/
16-Bit RISC Processor
Data Sheet

Cypress Semiconductor Corporation • 3901 North First Street • San Jose • CA 95134 • 408-943-2600
Document #: 38-08006 Rev. ** Revised December 3, 2001

Table of Contents

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

List of Figures

List of Tables

License Agreement

Use of this document and the intellectual properties contained herein indicates acceptance of the following License Agreement. If you do not accept the terms of this License Agreement, do not use this document, nor the associated intellectual properties, nor any other material you received in association with this product, and return this document and the associated materials within fifteen (15) days to Cypress Semiconductor Corporation or (CY) or CY's authorized distributor from whom you purchased the product.

1. You can only legally obtain CY's intellectual properties contained in this document through CY or its authorized distributors.
2. You are granted a nontransferable license to use and to incorporate CY's intellectual properties contained in this document into your product. The product may be either for your own use or for sale.
3. You may not reverse-engineer the SL11R or otherwise attempt to discover the designs of SL11R.
4. You may not assign, distribute, sell, transfer, or disclose CY's intellectual properties contained in this document to any other person or entity.
5. This license terminates if you fail to comply with any of the provisions of this Agreement. You agree upon termination to destroy this document, stop using the intellectual properties contained in this document and any of its modification and incorporated or merged portions in any form, and destroy any unused SL11R chips.

Warranty Disclaimer and Limited Liability

Cypress Semiconductor Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Cypress's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Cypress are granted by the Company in connection with the sale of Cypress products, expressly or by implication. Cypress's products are not authorized for use as critical components in life support devices or systems.

SL11R is a trademark of the Cypress Semiconductor Corporation. All other product names are trademarks or registered trademarks of their respective owners.

1.0 Definitions

USB	**U**niversal **S**erial **B**us
SL11R	The SL11R is a Cypress **USB** Controller, which provides multiple functions on a single chip.
QT	**Q**uick stream data **T**ransfer engine, which contains a small set of RISC instructions designed for the **SL11R USB** controller.
QTS	'**QT**' is a naming convention that represents QT Engine utility tools. For example: '**QTS**' indicates all tools, which interface with the RS232 serial interface port.
QTU	QT Engine Tools that interface with the USB port
R/W	Read/Write
PLL	**P**hase **L**ock **L**oop.
PWM	**P**ulse **W**idth **M**odulation
DVC	**D**igital **V**ideo **C**amera
MFU	**M**ulti **F**unction **U**nits
WDT	**W**atch **D**og **T**imer
RAM	**R**andom **A**ccess **M**emory
EPP	**E**nhanced **P**arallel **P**ort: An asynchronous, byte-wide, bidirectional channel controlled by the host device. This mode provides separate address and data cycles over the eight data lines of the interface.
2-wire serial interface	2-wire Serial EEPROM interface.
R0-R15	SL11R Registers: R0-R7 Data registers or general-purpose registers. R8-R14 Address/Data registers, or general-purpose registers. R15 Stack pointer register.
SL11R BIOS	A simulation model similar to 80x86 BIOS

2.0 References

[Ref. 1] SL11R_BIOS

[Ref. 2] SL11R_TOOLS

[Ref. 3] Universal Serial Bus Specification 1.1

3.0 Introduction

3.1 Overview

The SL11R is a low-cost, high-speed Universal Serial Bus (USB) RISC based Controller. It contains a 16-bit RISC processor with built-in SL11R BIOS ROM to greatly reduce firmware development work. Its 2-wire serial EEPROM interface offers low cost storage for USB device configuration and customer's product-specific functions. New functions can be programmed into the 2-wire serial interface by downloading them from a USB Host PC. This unique architecture provides the ability to upgrade products in the field without changing the peripheral hardware.

The SL11R Processor can execute code from either internal ROM/RAM or external ROM and SRAM. The SL11R Programmable bidirectional Data Port supports both DMA and I/O modes. A built in USB port supports data transfers up to 12 MBits/sec which is the maximum USB transfer rate. All USB protocol modes are supported: Isochronous (up to 1024 bytes/packet), Bulk, Interrupt, and Control. The SL11R requires a 3.3-Volt power supply, which can be powered via a USB host PC or a Hub. Suspend/Resume, and Low power modes are available.

The SL11R offers an optimal solution for a variety of peripheral products such as: Scanners, Digital Cameras (Video and Still), Color Printers, Multi-function Units (MFU), Faxes, External Storage devices, Monitors, Connectivity boxes, and other peripherals that traditionally interface via EPP or SCSI to a host PC.

3.2 SL11R Features

- Cypress offers a Development Kit for each of its product lines. These Development Kits include multiple peripheral Mini-port class drivers for Windows 98/ME/2000, firmware source code and demo USB source code for a variety of applications. Also available is an SL11R "C" compiler, debugger, and assembler with a reference demo board.
- 48 MHz 16 bit RISC Processor
- Up to 16 bits of Programmable Bidirectional Data I/O

- Up to 32 bits of General Purpose I/O (GPIO)
- 6Kx8 internal Mask ROM with built-in BIOS in supporting a comprehensive list of interrupt calls (see [Ref. 1] SL11R_BIOS for detailed information). These include USB functions, 2-wire serial interface, and UART and Boot-Up options (Boot-up from 2-wire serial interface or External ROM). Executable code can also run from 8-bit or 16-bit external Memory.
- 3Kx8 internal RAM that can be configured as the USB Ping-Pong buffer for USB DATA0 and DATA1 packets. It also can be used for data and/or code.
- Two-wire serial EEPROM interface port with SL11R BIOS support to allow on-board EEPROM programming
- Flexible Programmable external memory wait-states and a 8/16 data path
- Up to 16-bit address for Extended Memory Interface Port for External SRAM and ROM
- On chip DRAM Controller
- On chip fast EPP Interface
- On chip 8/16-bit DMA data path interface
- Supports 12 MHz/48MHz external crystal or clock
- Executable code or data can be loaded either from the USB port or via the UART port. The code/data is moved to RAM for debugging purposes (using a break point register), or to be programmed via a two-wire serial EEPROM.
- USB Port (12 Mbits/sec), including a built-in USB transceiver. All USB standard protocol modes are supported: Isochronous mode (up to 1024 packet size), Bulk, Interrupt, and Control modes.
- There are four available Endpoints. Each endpoint utilizes a bidirectional DMA port to move data between the Memory and the USB. Data can be sent/received to/from the Data Port Independently.
- Two General Purpose Timers, a Watchdog timer (WDT), four programmable PWM channels, and four Programmable Timing Generator outputs
- Four PWM or Programmable Timing Generator output channels are available. Each channel provides a programmable timing generator sequence that can be used to interface to various CCD, CIS, and CMOS image sensors, or can be used for other types of applications.
- Suspend/Resume and Low Power modes are supported
- UART interface supports from 900 Baud to 115.2K Baud
- USB Generic Mini-Port Driver for WIN98/2000 is available
- Debugger and QT-Assembler are available
- “C” Compiler option available
- Package: 100 LPQFP
- Power requirements 3.3V

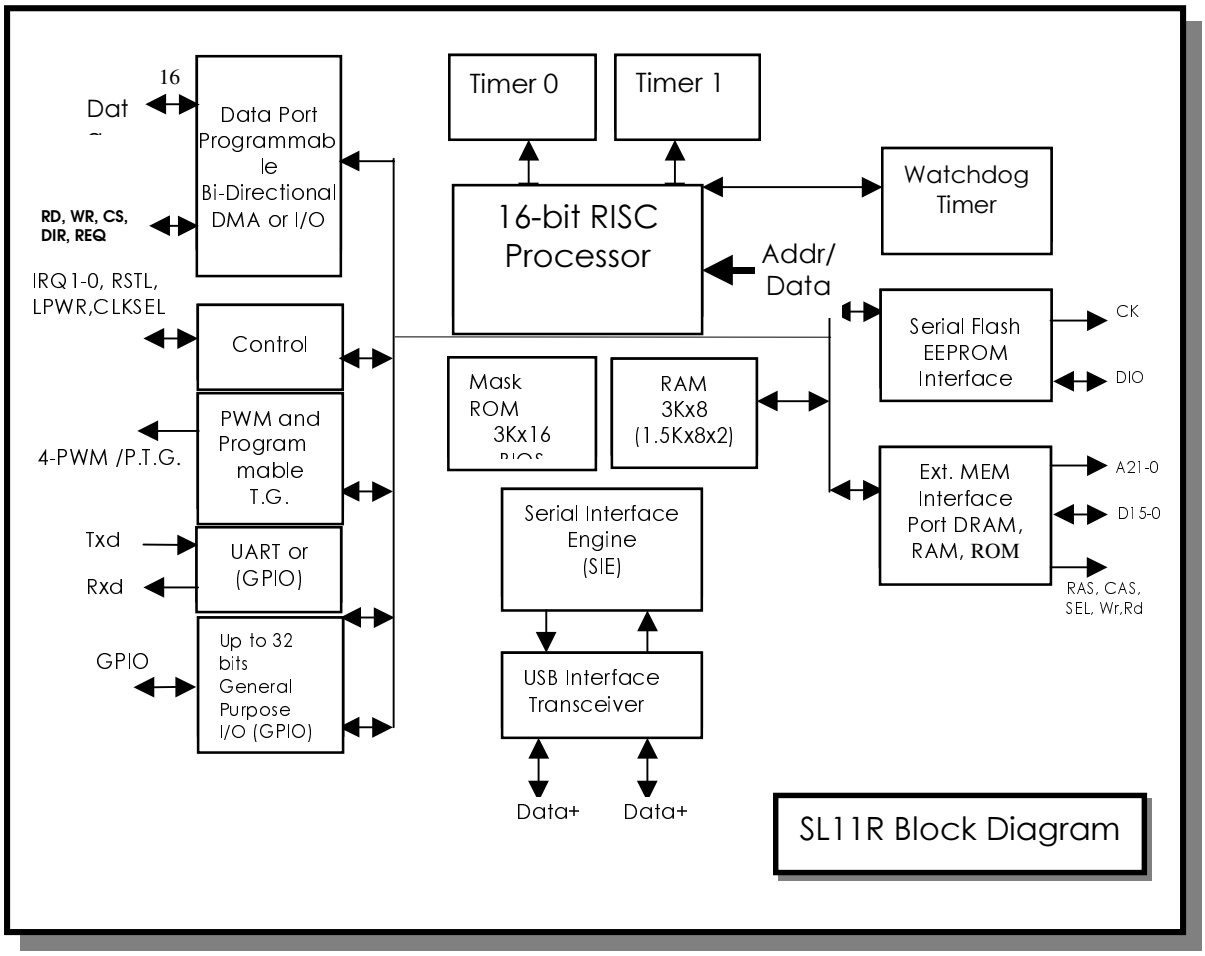

Figure 3-1. SL11R Block Diagram

3.3 SL11R 16-Bit RISC Processor

The SL11R can be used as a general purpose 16 bit embedded processor. It includes a USB interface (Universal Serial Bus) and up to 32 bits of GPIO supporting a variety of functions and modes. The 16-bit main data port can be used in either I/O or DMA bidirectional modes. Also, the SL11R contains 4 PWM channels or four Programmable Time Generator (PTG) signals, a UART, a 2-wire serial EEPROM interface, an additional External DRAM or SRAM interface for extended memory, two Timers, a Watchdog Timer, an internal mask BIOS ROM (3kx16) and an SRAM (3Kx8). The SL11R is optimized to offer maximum flexibility in the implementation of a variety of applications such as: Embedded Digital Video USB controller, USB Scanner controller, USB Cable Modems, Printers, External Storage Devices, MFU, etc.

The SL11R contains a specialized instruction set (RISC) that is highly optimized to provide efficient coding for a variety of applications such as video processing algorithms, Network data packet translation and USB transaction processing. The SL11R includes a simple software interface for all USB transaction processing, which supports Bulk mode (up to 64 Bytes/packet), Isochronous mode (up to 1024 Bytes/packet), all Interrupt and Control modes.

3.4 3Kx16 Mask ROM and BIOS

The SL11R has a built in 3Kx16 Mask ROM that contains the SL11R BIOS. This BIOS ROM provides the software interface for the USB and a boot-up option for a 2-wire serial interface or an external 8/16 EEPROM.

3.5 Internal RAM

The SL11R contains 3K x 8 internal RAM. The RAM can be used for code/program, variables, buffer I/O, DMA data (i.e. Video data), and USB packets. This memory can be accessed by the 16-Bit processor for data manipulation or by the SIE (Serial Interface Engine), which receives or sends USB host data.

3.6 Clock Generator

A 12, 48 MHz external Crystal, or logic-level clock can be used with the SL11R. Two pins, X1 and X2, are provided to connect a low cost crystal circuit to the device. If a logic-level clock is available, it may be connected directly to the X1 pin instead of a crystal.

Register C006 must be configured appropriately depending on the frequency used.

3.7 USB Interface

The SL11R has a built-in SIE and USB transceiver that meet the USB (Universal Serial Bus) specification v1.1. The transceiver is capable of transmitting or receiving serial data at the USB maximum data rate of 12 Mbits/sec. The SL11R Controller supports four endpoints. Endpoint 0 is the default pipe and is used to initialize and manipulate the peripheral device. It also provides access to the peripheral device's configuration information, and supports control transfers. Endpoint 1,2, and 3 support Interrupt transfers, Bulk transfers (up to 64 Bytes/packet), or Isochronous transfers (up to 1024 Bytes/packet size).

3.8 Processor Control Registers

The SL11R provides software control registers that can be used to configure the chip mode, the clock generator, the software breakpoint, and to read the BIOS version.

3.9 Interrupts

The SL11R provides 127 interrupt vectors for it's BIOS software interface (see [Ref. 1] SL11R_BIOS).

3.10 UART Interface

The SL11R has a built-in UART interface, which supports data rates from 900 baud to 115.2K Baud. It can be used as a development port or for other interface requirements. The Cypress development environment for the SL11R chip includes a debugger and assembler. Optional "C" compiler is also available[1]. You can download modified code to internal SRAM and debug it using the built-in Breakpoint register and Breakpoint Interrupt to break on any specified address location.

3.11 *2-wire Serial EEPROM Interface

The SL11R provides an interface to an external serial EEPROM. The interface is implemented using General Purpose I/O signals. A variety of serial EEPROM formats can be supported; currently the BIOS ROM supports a two-wire serial EEPROM. A serial EEPROM can be used to store specific Peripheral USB configuration and value-added functions. In addition, serial EEPROM can be used for field product upgrades.

3.12 External SRAM/DRAM/EPROM Interface

The SL11R provides a multiplexed address port and an 8/16-bit data port. This port can be configured to interface to an external SRAM, EPROM or DRAM. The port provides nRAS; nCASL, nCASH, nDRAMWR and nDRAMOE control signals for data access and refresh cycles to the DRAM.

3.13 General Timers and Watch Dog Timer

The SL11R has two built in programmable timers that can provide an interrupt to the SL11R Engine. On every clock tick which is 1 microsecond the timers decrement. An interrupt occurs when the timer reaches zero. A separate Watchdog timer is also provided to provide a fail-safe mechanism. The Watchdog timer can also interrupt the SL11R processor.

3.14 Special GPIO Functionality for Suspend, Resume and Low Power modes

The SL11R CPU supports suspend, resume and CPU low power modes. The SL11R BIOS assigns GPIO29 for the USB DATA+ line pull-up (this pin can simulate USB cable removal or insertion while the USB power is still applied to the board) and the GPIO20 for controlling the power off function.

3.15 Programmable Pulse/PWM Interface

The SL11R has four built-in PWM output channels available under 8/16-bit DMA mode. Each channel provides a programmable timing generator sequence that can be used to interface to various lines CCD, CIS, CMOS image sensors or can be used for other various applications. This feature is only available in the 8/16-bit DMA Mode.

Note:
1. Contact Cypress for details. (support@scanlogic.com)

3.16 Mailbox and DMA Overview

The Mailbox and DMA protocol use the same data latching and steering logic, so it is important to note that in a system where both mechanisms are used, the system designer must be careful to ensure that one type of transfer is finished before the next is started. Setting bits 1 and 2 to '1' in register 0xC006 enables the Mailbox/DMA interface. All transfers to and from the Data Registers are made through GPIO 0 to 15. Data written into the SL11R is latched into a 16-bit register on the rising edge of nWR (GPIO 17) when ADDR (GPIO 19) is high and nCS (GPIO 18) is low. Data is read out of the SL11R by asserting nRD (GPIO 16) low while ADDR (GPIO 19) is high and nCS (GPIO 18) is low. This also applies when data is written or read during DMA transfers. A functional logic diagram is shown in *Figure 3-2*.

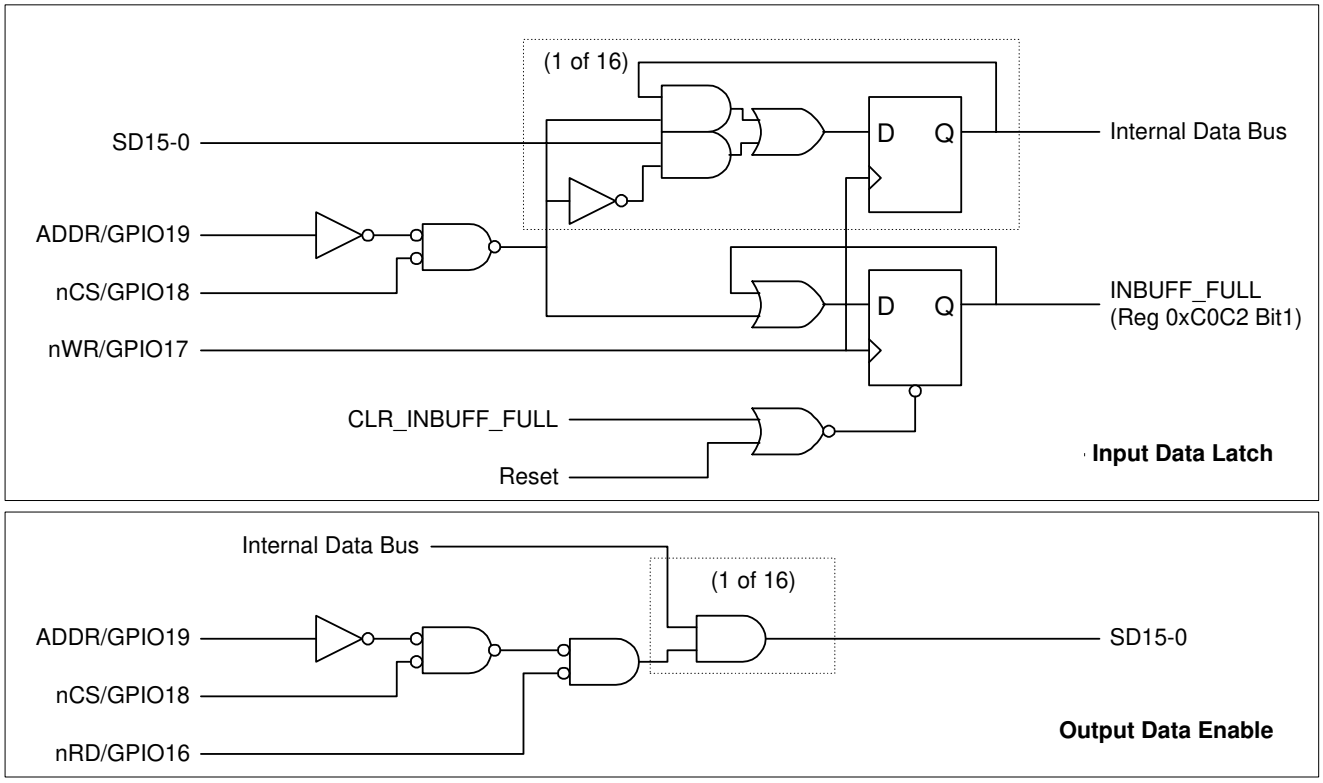

Figure 3-2. Functional Logic Diagram

3.17 Mailbox Interface

The mailbox interface is accessed through three registers:

1. INBUFF Data Register
   (0xC0C4; SL11R Read)
2. OUTBUFF Data Register
   (0xC0C4; SL11R Write)
3. STATUS Register
   (0xC0C2: Read Only)

When data is transferred to the Sl11R through the Mailbox Interface, the external system must perform the transfer based on the values in the status register. When a word is written to the SL11R, the 'IF' (INBUFF FULL) bit in the status register (0xC0C2) is set by the SL11R hardware. This bit must be polled until it is cleared to '0' by the SL11R. This indicates that the word has been accepted by the SL11R and that it is ready for another word. When data is read from the SL11R, the 'OF' (OUTBUFF FULL) bit in the status register will be set by the SL11R when valid data is available. When the data is read by the external system, this bit will be cleared by the SL11R hardware. When a new word is available, the OF bit will again be set.\

3.18 DMA Interface

This interface uses the same hardware as the Mailbox Interface, except that the DMA engine inside the SL11R generates DMA requests to control the transfer. Because the DMA engine handles the transfer of data through the Input and Output buffer, there is no need for the external system to poll the status register. The external system simply waits for a DREQ (GPIO 20) from the SL11R and transfers data. The SL11R DMA engine can transfer data in only one direction at a time. Please note that only 16-bit DMA transfers are supported on the SL11R.

The 22-bit DMA counter is loaded through registers 0xC02C and 0xC02E. This makes up the DMA start address and the location of the first word to be written or read. The 22-bit DMA end address register is loaded through registers 0xC030 and 0xC032. This will be location of the last word written into the SL11R. When reading data out of the SL11R, the end address register should be loaded with the last address to be read from plus two. After these registers are loaded, the DMA control register (0xC0C0) must be loaded with a 0x0007 to enable the DREQ output pin. Lastly, the other DMA control register (0xC02A) is loaded with either a 0x0001, to start DMA transfers into the SL11R, or 0x0003, to start DMA transfers out of the SL11R.

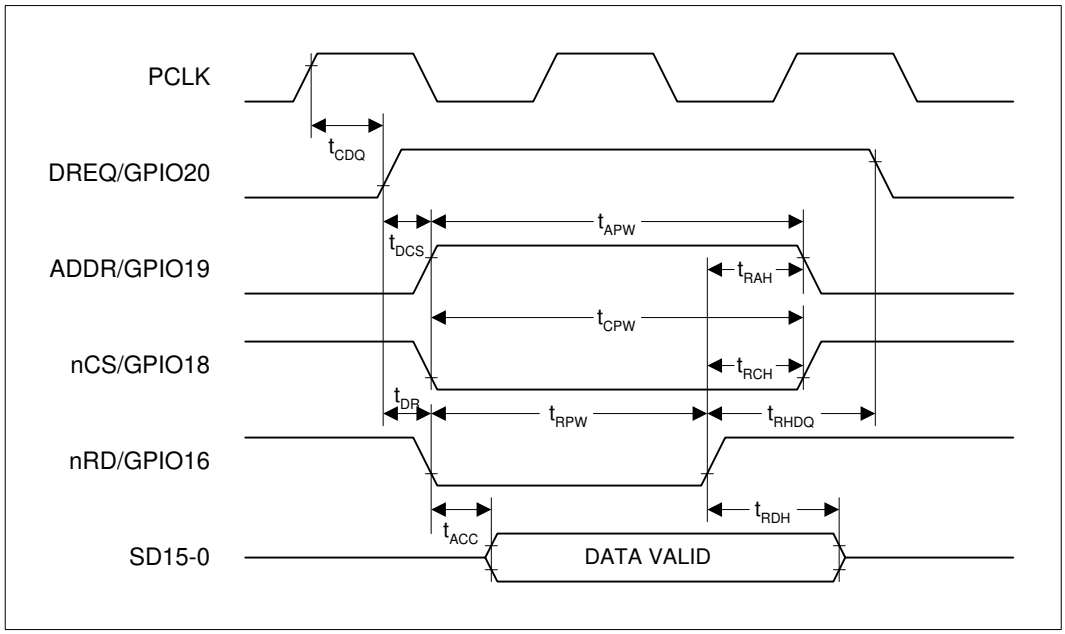

Figure 3-3. Mailbox/DMA Read

Parameter	Description	Min.	Typical	Max.	Unit
t_{CDQ}	PCLK to DREQ high	1		17	ns
t_{APW}	ADDR pulse width	30			ns
t_{CPW}	nCS pulse width	30			ns
t_{RPW}	Read pulse width	30			ns
t_{RAH}	ADDR hold after read high	0			ns
t_{RCH}	nCS hold after read high	0			ns
t_{ACC}	Read access time			25	ns
t_{DCS}	DREQ high to CS low	5			ns
t_{DR}	DREQ high to read low	5			ns
t_{RHDQ}	Read high to DREQ low hold			30	ns
t_{RDH}	Read high to data hold			10	ns

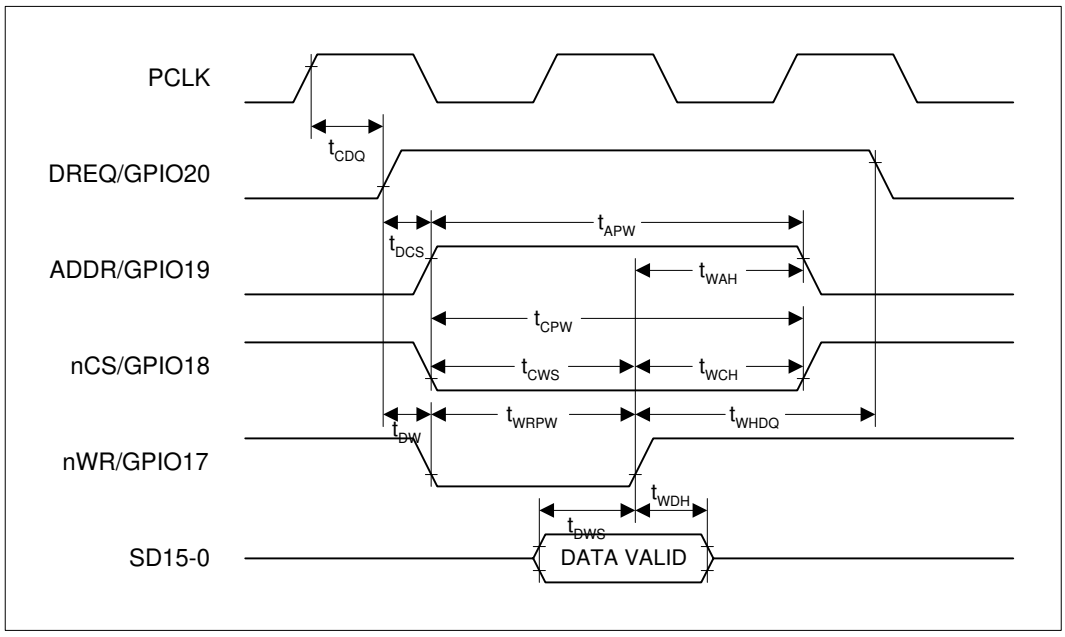

Figure 3-4. Mailbox/DMA Write

Parameter	Description	Min.	Typical	Max.	Unit
t_{CDQ}	PCLK to DREQ high	1		17	ns
t_{APW}	ADDR pulse width	20			ns
t_{CPW}	nCS pulse width	20			ns
t_{CWS}	nCS low to write high setup	10			ns
t_{WRPW}	Write pulse width	10			ns
t_{WAH}	ADDR hold after read high	5			ns
t_{WCH}	NCS hold after read high	5			ns
t_{DWS}	Data setup to write high setup	10			ns
t_{DCS}	DREQ high to CS low	5			ns
t_{WHDQ}	Write high to DREQ low hold			30	ns
t_{DW}	DREQ high to write low	5			ns
t_{WDH}	Write high to data hold	5			ns

3.19 Fast DMA Mode

This mode is currently used by the DVC 8-Bit DMA and 8/16-Bit DMA modes. In the DVC 8-Bit DMA mode, the DMA data path will be 8-bits, which correspond to SD0-SD7. In the 8/16-Bit DMA mode, the DMA data path can be configured to either 8 or 16 bits.

3.20 SL11R Interface Modes

The SL11R has four modes. They are: General Purpose I/O mode, Fast EPP mode, 8-bit DMA mode, and 8/16-bit DMA Mailbox Protocol ports mode. These modes are shared and can be configured under software control.

Note: The UART and 2-wire serial interface I/O pins are fixed in all cases.

3.20.1 General Purpose IO mode (GPIO)

In the GPIO mode, the SL11R has up to 32 General-Purpose IO signals available. However, four pins that are used by the UART and the 2-wire serial interface that cannot be used as GPIO pins. A typical application for this GPIO is the Parallel Port to USB. The SL11R executes at 48MHz, which is fast enough to generate any Parallel Port timing. The SL11R also includes a special mode for EPP timing designed for special devices that have no delay in EPP mode. On any other available General Purpose programmable I/O, the pins can be programmed for peripheral control and/or status.

Note: The Fast DMA and PWM Interfaces are not supported in this mode.

3.20.2 8/16-bit DMA Mode

This Mode includes the Mailbox Protocol and DMA Protocol. The Mailbox Protocol allows asynchronous exchange of data between the external Processor (i.e. DSP or other Microprocessor) and SL11R via SD0-SD15 (GPIO 0-15) which is a bidirectional data port. The DMA Protocol allows large blocks of data to be transferred to or from the SL11R via the 8/16-bit DMA port.

3.20.3 Fast EPP Mode

This mode is designed to interface with a special optimized high-speed EPP interface. In this mode, the SL11R processor has direct access to the EPP control port.

Note: The Fast DMA and PWM Interface are not supported in this mode.

3.20.4 DVC 8-bit DMA Mode

This DVC 8-bit DMA mode is designed to interface with CCD cameras. Camera control and setup is performed through the serial control bus. The SL11R 16-bit processor has direct access to the control port and the camera operation is dependent on commands passed from the USB Host to the SL11R. Raw video data from the CCD Camera is input to the SL11R on the 8-bit video data bus (SD7-SD0) using a combination of clock, control signals and 8-bit DMA.

Note: The PWM Interface is not supported in this mode.

4.0 Interface

4.1 Internal Masked ROM: 0xE800-0xFFFF

The SL11R has a built-in 3Kx16 internal masked ROM that contains software bootstrap code to allow programs in an external 8/16-bit ROM to be executed. The ROM code can also load data from the 2-wire serial interface into internal RAM for execution. In addition, the internal BIOS ROM contains the Interrupt Service Routines (see [Ref. 1] SL11R_BIOS for information) that support the USB, 2-wire serial interface, UART interfaces and Boot-Up options (Boot-up from 2-wire serial interface or External ROM). This SL11R BIOS ROM eases software development of all SL11R interfaces. The SL11R Chip is ready for all the USB enumeration and download/program code.

The SL11R Internal Masked ROM (i.e. SL11R BIOS) is mapped from address 0xE800 to 0xFFFF. On power up or hardware reset, the SL11R processor jumps to the address of 0xFFF0, which contains a long jump to the beginning of the internal ROM of address 0xE800. See *Table 4-1*.

Table 4-1. Internal Masked ROM (SL11R BIOS)

Address	Memory Description
0xE800-0xFFEF	SL11R BIOS code/data space
0xFFF0-0xFFF3	Jump to 0xE800
0xFFF4-0xFFF9	Reserved for future use.
0xFFFA-0xFFFB	ROM BIOS Checksum
0xFFFC-0xFFFD	SL11R BIOS Revision
0xFFFE-0xFFFE	Peripheral Revision
0xFFFF-0xFFFF	QT Engine Instruction Revision

4.2 External ROM: 0xC100-0xE800

The SL11R BIOS ROM reserves addresses from 0xC100 to 0xE800 for external ROM. During BIOS initialization, the SL11R will scan for the signature ID (0xCB36) at location 0xC100. After a valid signature is detected, execution will begin at address 0xC102 (see [Ref. 1] SL11R_BIOS for more information). The signal nXROMSEL is used to enable the external ROM. It is mapped from

0xC100 to 0xE800 by default. However, the Extended Memory Control can be used to configure multiple windows for external ROM set-up.

Note: The Address space from 0x8000-0xC100 can also be used as the external ROM (see the External Memory Control set-up for more detail).

4.3 Internal RAM: 0x0000-0x0BFF

The SL11R contains a 1.5Kx16 internal RAM. This memory is used to buffer video data and USB packets and is accessed by the 16-bit processor and the SIE (Serial Interface Engine). USB transactions are automatically routed to the memory buffer. The 16-bit processor has the ability to set up pointers and block sizes in buffer memory for USB DMA transactions. For example, video data can be read from the camera interface and is sent to the USB port by the internal DMA engine. The SL11R BIOS uses this internal RAM for USB buffers, BIOS variables and user data/code. Executable code or data can reside in multiple locations: internal masked ROM (3Kx16), internal RAM (3Kx8), external ROM and external SRAM. Program code or data can also be loaded to either the internal or the external RAM from the USB port, the RS232 port, or the 2-wire serial interface.

The SL11R Internal RAM is mapped from 0x0000 to 0x0BFF. See internal RAM memory usage in **Table 2** below:

Table 4-2. Internal RAM Memory Usage

Address	Memory Description
0x0000 - 0x00FF	Hardware/Software Interrupts
0x0100 - 0x01FF	Register Banks/USB Control/Software Stack
0x0200 - 0x021F	Hardware Interrupts stack
0x0220 - 0x0343[2]	SL11R BIOS internal buffers & variables
0x0344 - 0x0BFF	User's Programming Space

Note:

2. This address may be changed due to SL11R BIOS revision updates. The new SL11R BIOS may require more internal memory for its variable usage in any new SL11R BIOS.

- The addresses from 0x0000 to 0x00FF are reserved for hardware and software interrupt vectors (see [Ref. 1] SL11R_BIOS for more information).
- Addresses from 0x0100 to 0x01FF are reserved for Internal Register Banks (SL11R register R0-R15 bank 0 and R0-R15 bank 1) and the software stack. Others are reserved for USB Control registers and other read/write control registers.
- Addresses from 0x0200 to 0x021F are reserved for the hardware interrupt stack.
- Addresses from 0x0220 to 0x0343 are available internal RAM for application software. Software can be downloaded via the USB port or UART interface (see [Ref. 1] SL11R_BIOS for more information).

4.4 Clock Generator

The SL11R has an option to use either a 48-MHz or 12-MHz external crystal or oscillator as its clock source. SL11R includes an internal PLL that can be configured by software. At power-up, the SL11R BIOS default configuration sets the processor clock to run at 2/3 of X1 (of the external provided clock).

Example 1 Changing SL11R CPU Speed

The default of the SL11R BIOS assumes a 48MHz input clock, so the SL11R processor clock is (2/3)*48MHz = 32MHz. See example below:

```
mov    [0xC006],0x10    ;clock = 2/3*X1
mov    [0xC008],0       ;CPU clock at 32MHz
```

If the X1 input clock is 48 MHz, then the maximum speed of the SL11R processor can be set at follows:

```
mov    [0xC006],0       ;clock = set up at X1 clock input
mov    [0xC008],0       ;CPU clock at 48MHz
```

If the X1 input clock is 12 MHz, then the maximum speed of the SL11R processor can be set to:

```
mov    [0xC006],0x40    ;clock = 4*X1
mov    [0xC008],0       ;CPU clock at 48MHz
```

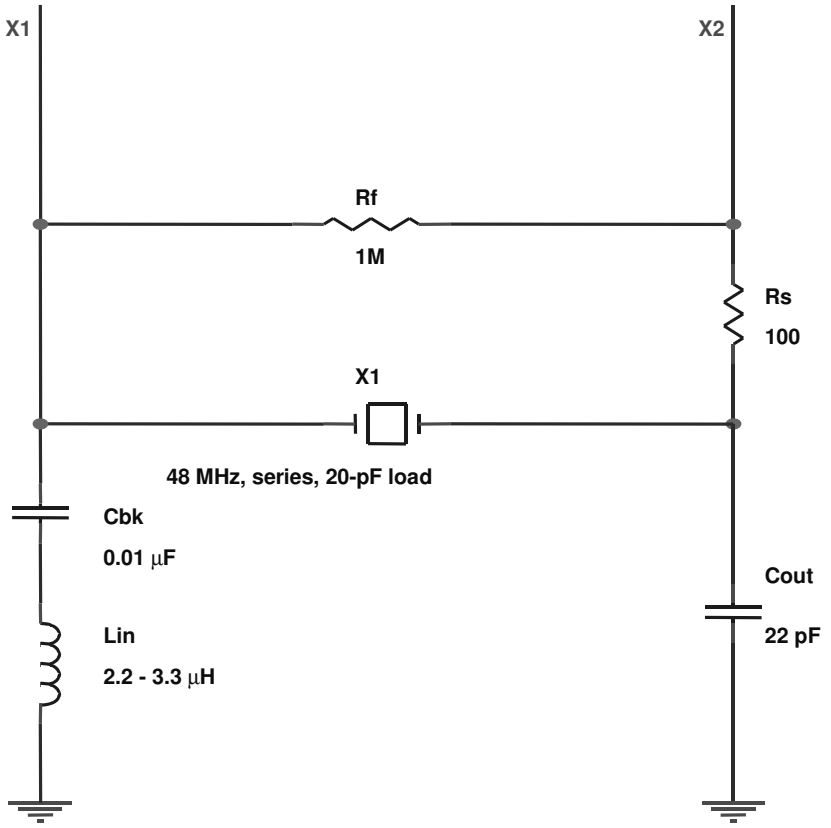

Figure 4-1. 48-MHz Crystal Circuit

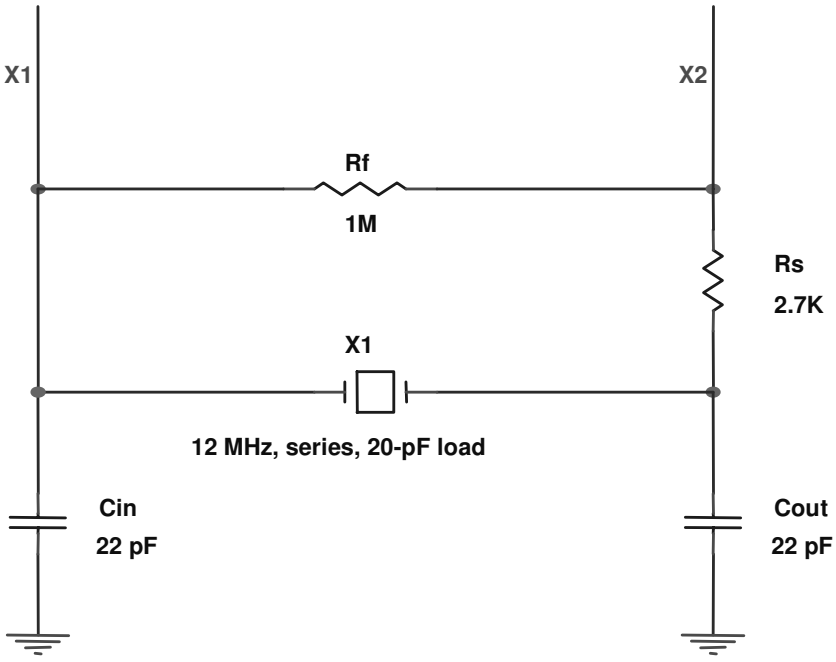

Figure 4-2. 12-MHz Crystal Circuit

Note: You need to set bit C2 =1 from configuration address (0xC006). See section 4.5 for CPU control speed.

4.5 USB Interface

The SL11R has a built-in transceiver that meets the USB specification v1.1. The transceiver connects directly to the physical layer of the USB engine. It is capable of transmitting or receiving serial data at the USB maximum data rate of 12 Mbits/sec. The SL11R has four USB DMA engines for four USB endpoints. Each of the USB DMA engines is independently responsible for its respective USB transaction. The 16-bit processor has the ability to set up pointers and block sizes in buffer memory for USB transactions.

The SL11R Controller contains a number of Registers that provide overall control and status functions for USB transactions. The first set of registers is for control and status functions, while the second group is dedicated to specific endpoint functions. Communication and data flow on the USB is implemented using endpoints. These uniquely identifiable entities are the terminals of communication flow between a USB host and USB devices. Each USB device is composed of a collection of independently operating endpoints. Each endpoint has a unique identifier: the endpoint number. (See USB specification v1.1. Sec 5.3.1)

The SL11R also includes the SL11R BIOS that provides a set of subroutines via interrupt calls for all USB interface functions required to communicate with a USB host (refer to [Ref. 1] SL11R_BIOS for more information). The SL11R BIOS greatly simplifies the firmware/software development cycle.

4.5.1 USB Global Control & Status Register (0xC080: R/W)

The USB Global Control & Status Register allows high-level control and provides status of the USB-DMA engines. The Global Control & Status register bits are defined as follows:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	UA	US	UR	UE

D15-D4	Reserved	
D0	UE	USB Enable = ’1’, Overall USB enable/disable bit
D1	UR	USB Reset = ’1’, USB received Reset command
D2	US	USB SOF = ’1’, USB received SOF command
D3	UA	USB Activity = ’1’, Activity Seen

Notes:

- Suspend state should be entered if there is no activity after 3mS (UA).
- The US and UA bits are automatically cleared after they are read by the SL11R processor.
- D15-D4 are the reserved bits, should be written with zeros.
- The SL11R BIOS will set the UE=1 upon reset.

4.5.2 USB Frame Number Register (0xC082: Read Only)

The Frame Number Register contains the 11-bit ID Number of the last SOF received by the device from the USB Host.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

D15-D11	Reserved	set to all zeros.
D10-D0	S10-S0	SOF ID Number of last SOF Received

Note:

- The SL11R BIOS uses this register to detect USB activity for the internal idle task.

4.5.3 USB Address Register (0xC084: R/W)

Address Register holds the USB address of the device assigned by the Host - initialized to address 0x0000 upon Power up.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	A6	A5	A4	A3	A2	A1	A0

D15-D7	Reserved	set to all zeros
D6-D0	A6-A0	USB Address of device after assignment by Host

Note:

- The SL11R BIOS modifies this register upon receiving the SET_ADDRESS from the host. (See [Ref. 3] Universal Serial Bus Specification 1.1, Chapter 9 for more information)

4.5.4 USB Command Done Register (0xC086: Write Only)

This is the USB command done register. It is only used by the control point (endpoint 0).

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E

D15-D1	Reserved	set to all zeros.
D0	E	Set E=0 for Successful Command Completion Set E=1 for Error Command Completion

Note:

- The SL11R BIOS modifies this register upon command completion on endpoint 0.

4.6 USB Endpoint 0 Control & Status Register (0xC090: R/W)

See the USB Endpoint 3 Control & Status Register for more information.

4.7 USB Endpoint 1 Control & Status Register (0xC092: R/W)

See the USB Endpoint 3 Control & Status Register for more information.

4.8 USB Endpoint 2 Control & Status Register (0xC094: R/W)

See the USB Endpoint 3 Control & Status Register for more information.

4.9 USB Endpoint 3 Control & Status Register (0xC096: R/W)

4.9.1 General Description for All Endpoints from Endpoint 0 to Endpoint 3

The SL11R Controller supports four endpoints. Endpoint 0 is the default pipe and is used to initialize and control the peripheral device. It also provides access to the peripheral device's configuration information, and supports control transfers. Endpoint 1, 2, and 3 support interrupt transfers, bulk transfers up to 64 Bytes/packet, or Isochronous transfers up to 1024 Bytes/packet size.

4.9.2 USB Endpoints Control (For Writing)

Each of the endpoint Control Registers when written have the following functions assigned:

Bit Position	Bit Name	Function
D0	ARM	Allows enabled transfers when set to '1'. Cleared to '0' when transfer is complete
D1	Enable	When set to ‘1’ it allows transfers to this endpoint. When set to '0' USB transactions are ignored. If enable = '1' and Arm = '0', the endpoint will return NAK to USB transmissions.
D2	DIR	When set to ‘1’, It transmits to Host (IN). When '0' receive from Host (OUT)
D3	ISO	When set to '1' It allows Isochronous mode for this endpoint
D4	Stall	When set to ‘1’ It sends Stall in response to next request on this endpoint
D5	Zero Length	When set to ‘1’ It sends a zero length packet
D6-D15	Not Defined	Set to logic ‘0’s

4.9.3 USB Endpoints Status (For Reading)

Reading the Endpoint Status Register provides Status information relative to the packet that has been received or transmitted. The register is defined as follows:

Bit Position	Bit Name	Function
D0	Arm	If '1', the endpoint is armed
D1	Enable	If '1', the endpoint is enabled
D2	DIR	Direction bit. If '1', set to transmit to Host (IN). If '0', set to receive from Host (OUT)
D3	ISO	If '1', isochronous mode selected for this endpoint
D4	Stall	If '1', endpoint will send stall on USB when requested
D5-D12	Not used	Read returns logic '0's
D13	Setup	If '1', a Setup packet has been received
D14	Error	If '1', an error condition occurred on last transaction for this endpoint
D15	Done	If '1', transaction completed. Arm Bit is cleared to '0' when Done Set

Notes:

- Endpoint 0 is set up as a control endpoint. The **DIR** bit is read-only, and indicates the direction of the last completed transaction. If the direction is incorrect, it is the firmware's responsibility to handle the error. On other endpoints, **DIR** bit is written, and if the direction of the transfer does not match the **DIR** bit, then the transaction is ignored.
- At the end of any transfer to an armed and enabled endpoint (with the correct DIR bit), an interrupt occurs, and vectors to a different location depending upon whether an error occurred or not. At the end of this transfer, the corresponding endpoint is disarmed (the Arm bit is cleared), and the DATA0/DATA1 toggle bit is advanced if no error occurred. If a packet is received with an incorrect toggle state, the packet is ignored so that the host will re-send the data.
- The DATA0/DATA1 bit is automatically toggled by the hardware. To reset this DATA0/DATA1 toggle bit to DATA0, the **Enable** on the **D1** bit should be cleared to '0' and then set to '1'.
- When the Zero Length bit **(D5)** is set, the host will receive the zero length USB packet, regardless of the number of bytes in the USB Count register.
- The SL11R BIOS has full control of USB endpoint 0. The SL11R BIOS responds to all numeration from the host. On other endpoints, the SL11R BIOS can be used to control under BIOS interrupt calls (see [Ref. 1] SL11R_BIOS).
- The SL11R BIOS will set all USB Control & Status registers for endpoint 1 through 3 to zero upon receiving the SET_CONFIG command from host. (See [Ref. 3] Universal Serial Bus Specification 1.1**1,** Chapter 9 for more information.)

4.9.4 USB Endpoint 0 Address Register (0x0120: R/W)

This is the pointer to memory buffer location for USB reads and writes to this Endpoint. At the end of any transfer, this register will contain its original value plus the value in the USB Endpoint Count Register.

4.9.5 USB Endpoint 1 Address Register (0x0124: R/W)

See USB Endpoint 0 Address Register (0x0120: R/W)

4.9.6 USB Endpoint 2 Address Register (0x0128: R/W)

See USB Endpoint 0 Address Register (0x0120: R/W)

4.9.7 USB Endpoint 3 Address Register (0x012C: R/W)

See USB Endpoint 0 Address Register (0x0120: R/W)

4.9.8 USB Endpoint 0 Count Register (0x0122: R/W)

This register is used to set the maximum packet size for the USB transfer. At the end of a successful transfer, the USB endpoint Count Register is set to zero.

4.9.9 USB Endpoint 1 Count Register (0x0126: R/W)

See USB Endpoint 0 Count Register (0x0122: R/W)

4.9.10 USB Endpoint 2 Count Register (0x012A: R/W)

See USB Endpoint 0 Count Register (0x0122: R/W)

4.9.11 USB Endpoint 3 Count Register (0x012E: R/W)

See USB Endpoint 0 Count Register (0x0122: R/W)

4.10 Processor Control Registers

The SL11R provides software control registers that can be used to configure the chip mode, clock control, read software version and software breakpoint control.

4.10.1 Configuration Register (0xC006: R/W)

The Configuration Register is used to configure the SL11R into the appropriate mode, and to select a clock multiplier.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	C2	C1	C0	CD	M1	M0	MD

Note: D6-4 and C2-0 are Clock Configuration bits. These bits select the clock source. The clock may come from an outside pin (X1 or X_PCLK) or it may come from the PLL multiplier as indicated in the table.

C2	C1	C0	PCLK	RCLK	OE
0	0	0	X1	X1	0
0	0	1	2/3*X1	X1	0
0	1	0	X_PCLK	X1	0
0	1	1	2/3*X1	X1	1
1	0	0	4*X1	4*X1	0
1	0	1	8/3*X1	4*X1	0
1	1	0	4*X1	4*X1	1
1	1	1	8/3*X1	4*X1	1

D3 CD

If Clock Disable bit = ‘1’, this Clock Configuration register can no longer be modified through software writes. It is a “sticky bit” used to lock the configuration through a write to this bit in the boot prom code.

Note:

- On the SL11R chip set, this bit will be set to zero.
- `There are four modes defined in this documentation: DVC 8-bit DMA mode, Fast EPP mode, 8/16-bit DMA mode and General Purpose IO (GPIO) mode. All modes are pin-compatible.

D2, D1 M1,M0: SL11R modes are selected as shown here:

M1	M0	Mode
0	0	GPIO
0	1	DVC 8-Bit DMA
1	0	Fast EPP
1	1	8/16-Bit DMA

D0 MD

If Mode Disable bit = ‘1’, this Configuration register can no longer be modified through software writes. It is a “sticky bit” used to lock the configuration through a Write to this bit in the boot prom code.

Note:

By default, this bit will be set to zero by the SL11R BIOS.

D15-D7 Reserved should be set to all zeros.

Where:

PCLK is connected to the SL11R processor clock.

RCLK is the resulting clock that connects to other modules (i.e. PWM, USB engine).

OE when **OE**=1, the **X_PCLK** (pin 59) will become an output pin of the **PCLK** value.

Notes:

- When the X1 input pin is fed with a 12 MHz signal, the software should set **C2 to '1'** to enable the PLL.
- **X_PCLK** is a bidirectional pin allowing an additional clock input for PCLK when selected or an observation pin for PCLK when $\overline{OE}$ = '1'.
- The **X_PCLK** can be used as the input clock like X1, but only when mode C2=0, C1=1, C0=0.
- Upon reset, the SL11R BIOS will set this register equal to 0x0010 (i.e. C2=0, C1=0, C0=1, PCLK=X1, RCLK=X1, OE=0, M1-M0=0=GPIO Mode).

4.10.2 Speed Control Register (0xC008: R/W)

The Speed Control Register allows the SL11R processor to operate at a number of speed selections. A four-bit divider (SPD3-0 + 1) selects the speed as shown below. Speed will also depend on the clock multiplier. See Configuration Register (0xC006: R/W) for more information.

D15-D4	D3	D2	D1	D0
0	SPD3	SPD2	SPD1	SPD0

D3-D0 SPD3-SPD0 Speed selection bits

SPD3-0	SL11R Speed
0000	48.00 MHz.
0001	24.00 MHz.
0010	16.00 MHz.
0011	12.00 MHz.
0100	09.60 MHz.
0101	08.00 MHz.
0110	06.86 MHz.
0111	06.00 MHz.
1000	05.33 MHz.
1001	04.80 MHz.
1010	04.36 MHz.
1011	04.00 MHz.
1100	03.69 MHz.
1101	03.42 MHz.
1110	03.20 MHz.
1111	03.00 MHz.

D15-D4 Reserved should be set to all zeros.

Note:

Upon reset, the lowest speed is selected for low power operation. The SL11R BIOS will configure the clock to 24MHz as part of its initialization.

4.10.3 Power Down Control Register (0xC00A: R/W)

During Power down mode, the peripherals are put in a "**pause**" state. All counters and timers stop incrementing and the PWM stops.

D15-D6	D5	D4	D3	D2	D1	D0
0	USB	GPIO	PUD1	PUD0	SUSPEND	HALT

There are two ways to enter power-down mode:

Suspend or **Halt**.

D5 USB Enable restarts on USB transition resulting in device power up.

D4 GPIO Enable restarts on GPIO transition resulting in device power up (See GPIO Interrupt Control Register (0xC01C:R/W)).

D3-D2 PUD1-PUD0 Power Up Delay Selection. Four delays are provided and selected using these select bits. This is time from power up until processor starts executing allowing clock to settle.

PUD1	PUD0	Power-up Delay
0	0	0 milliseconds
0	1	1 milliseconds
1	0	8 milliseconds
1	1	64 milliseconds

D1 SUSPEND To save power, Suspend mode stops all clocks in the SL11R.

This mode ends with a transition on either USB or any Interrupt. It is followed by a delay set in the Power-up delay bit fields.

D0 HALT ends with an interrupt.

D15-D6 Reserved should be set to all zeros.

4.10.4 Breakpoint Register (0xC014: R/W)

The Breakpoint Register holds the breakpoint address. Access to this address causes an INT127.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

D15-0 A15-0 Breakpoint address.

4.11 Interrupts

The SL11R provides 127 interrupt vectors. The first 64 vectors are hardware interrupts and the next 64 are software interrupts (see the [Ref. 1] SL11R_BIOS for more information).

4.11.1 Hardware Interrupts

The SL11R allocates addresses from 0x0000 to 0x003E for hardware interrupts. The hardware interrupt vectors are shown below:

Table 4-3. Hardware Interrupt Table

Interrupt Number	Vector Address	Interrupt Type
0	0x0000	Timer0[3]
1	0x0002	Timer1[4]
2	0x0004	GP IRQ0[4]
3	0x0006	GP IRQ1[4]
4	0x0008	UART Tx[3]
5	0x000A	UART Rx[3]
6	0x000C	Fast DMA Done[4]
7	0x000E	USB Reset
8	0x0010	USB SOF[5]
9	0x0012	USB Endpoint0 No Error[3]
10	0x0014	USB Endpoint0 Error[3]
11	0x0016	USB Endpoint1 No Error
12	0x0018	USB Endpoint1 Error
13	0x001A	USB Endpoint2 No Error
14	0x001C	USB Endpoint2 Error
15	0x001E	USB Endpoint3 No Error
16	0x0020	USB Endpoint3 Error
17	0x0022	8/16-bit DMA Mode Mailbox TX Empty[4]
18	0x0024	8/16-bit DMA Mode Mailbox RX Full[4]
19-63	0x0026- 0x003E	Reserved ♦

Notes:
3. These hardware interrupt vectors are reserved for internal SL11R-BIOS usage. You should not attempt to overwrite these functions.
4. These hardware interrupt vectors are initialized to return on the interrupt.
5. The SOF interrupt is generated when there is an incoming SOF on the USB.

All these vector interrupts are read/write accessible. You can overwrite these default software interrupt vectors by replacing your interrupt service subroutine.

The addresses from 0x0000 to 0x003E are read/write accessible and can be used for variables.

4.11.2 Interrupt Enable Register (0xC00E: R/W)

This is a global hardware interrupt enable register that allows control of the hardware interrupt vectors. The SL11R BIOS default set-up of this register is 0x28 (i.e. USB and UART bits are set).

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	MBX	USB	FDMA	UART	GP	T1	T0

D6	MBX	Mail Box interrupt enable (8/16-bit DMA Mode Only)
D5	USB	USB Interrupt enable
D4	FDMA	Fast DMA Done Interrupt enable
D3	UART	UART Interrupt enable

D2	GP	

General Purpose I/O pins Interrupt enables (see GPIO Interrupt Control Register (0xC01C: R/W))

D1	T1	Timer1 Interrupt Enable
D0	T0	Timer0 Interrupt Enable

4.11.3 GPIO Interrupt Control Register (0xC01C: R/W)

This register defines the polarity of the GPIO interrupt on IRQ1 (GPIO25) and IRQ0 (GPIO24). The **GPIO** bit on the Interrupt Enable Register must be set in order for this register to operate.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	P1	E1	P0	E0

D3	P1	IRQ1 polarity is rising edge if “1”, falling edge if “0”
D2	E1	Enable IRQ1 if set to “1”
D1	P0	IRQ0 polarity is rising edge if “1”, falling edge if “0”
D0	E0	Enable IRQ0 if set to “1”

Note:

The interrupts can be enabled for “Suspend mode” by the power down Register or enabled for interrupts by the Interrupt Enable Register.

4.11.4 Software Interrupts

The SL11R allocates addresses from 0x0040 to 0x00FE for software interrupts. The software interrupt vectors are shown in *Table 4-4*: