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

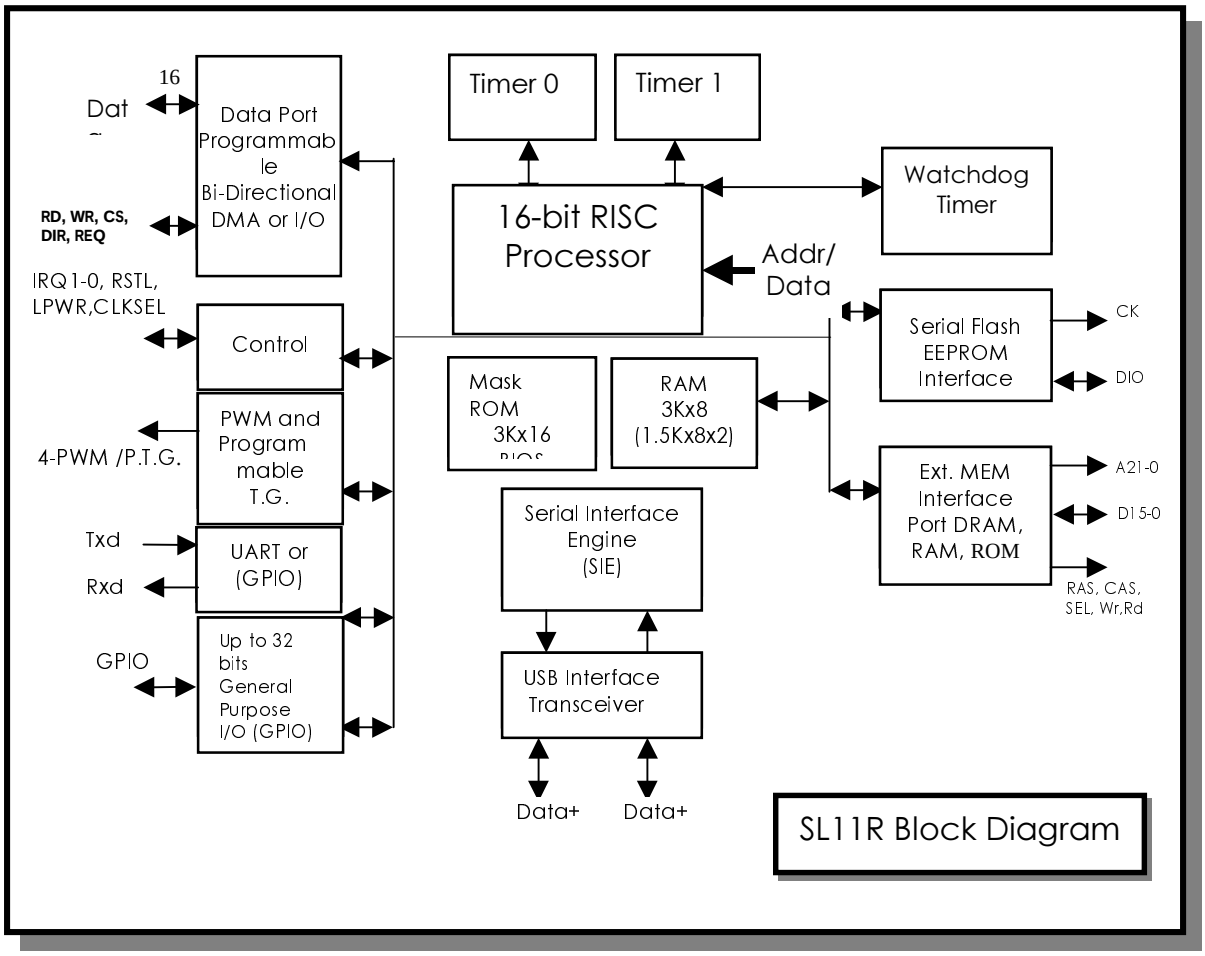


Figure 3-1. SL11R Block Diagram

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*.

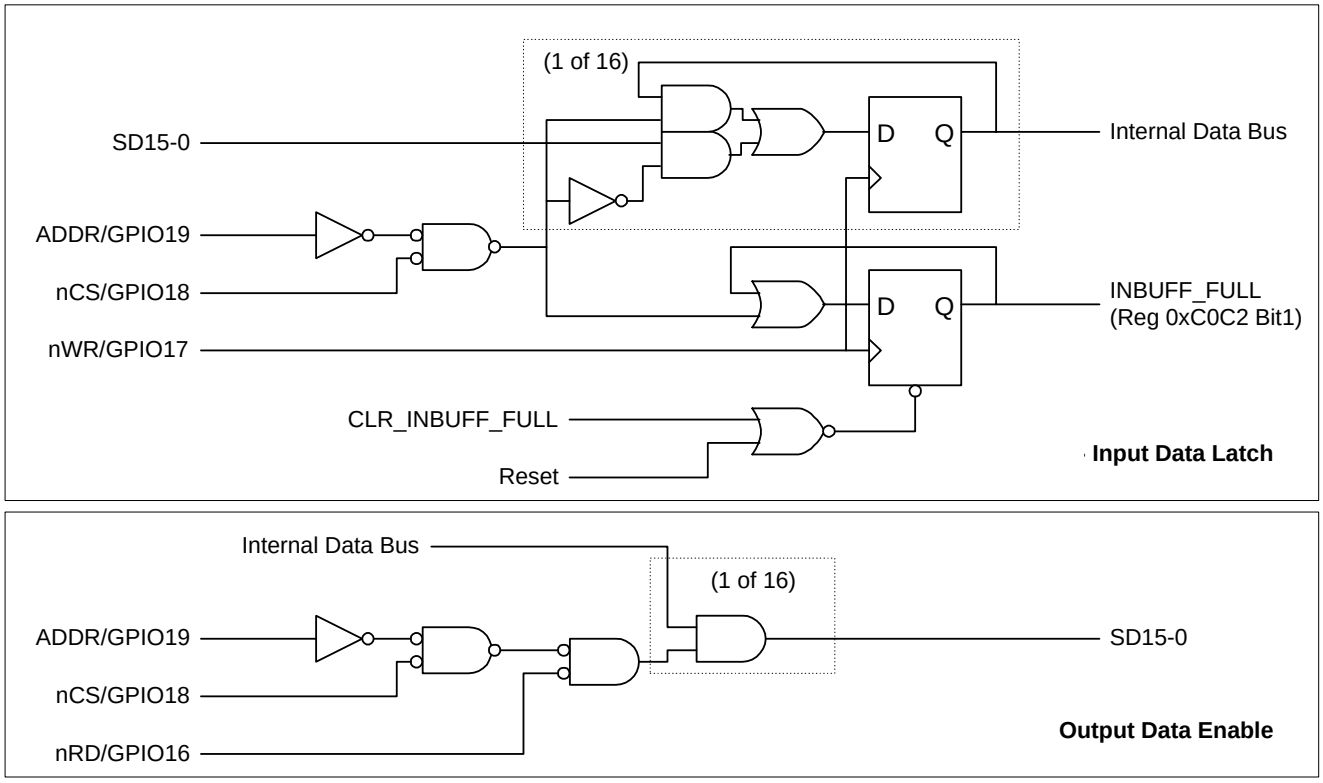


Figure 3-2. Functional Logic Diagram

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.

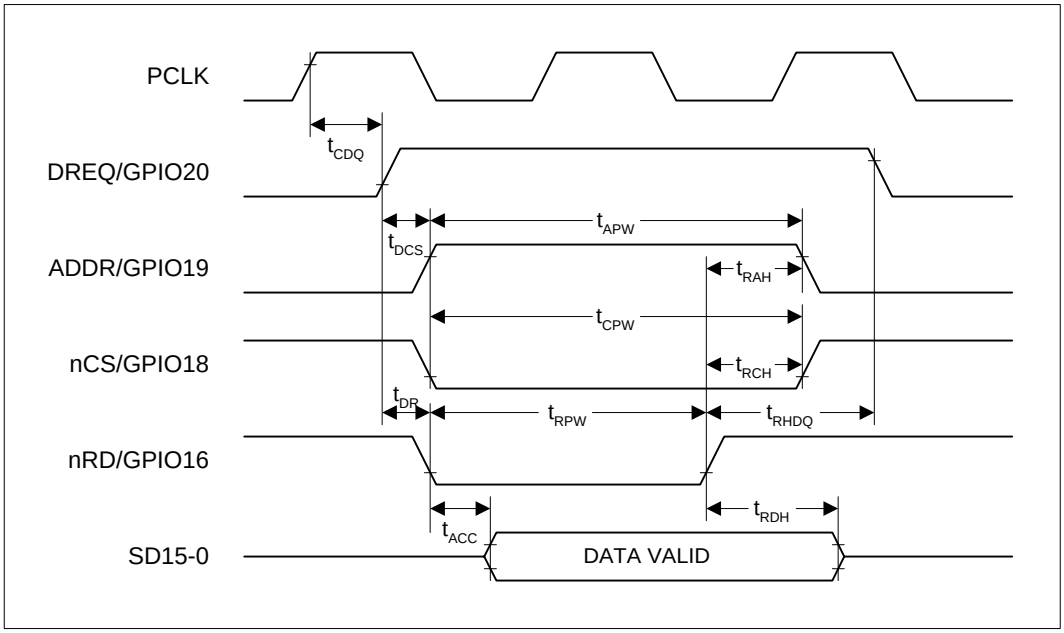


Figure 3-3. Mailbox/DMA Read

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

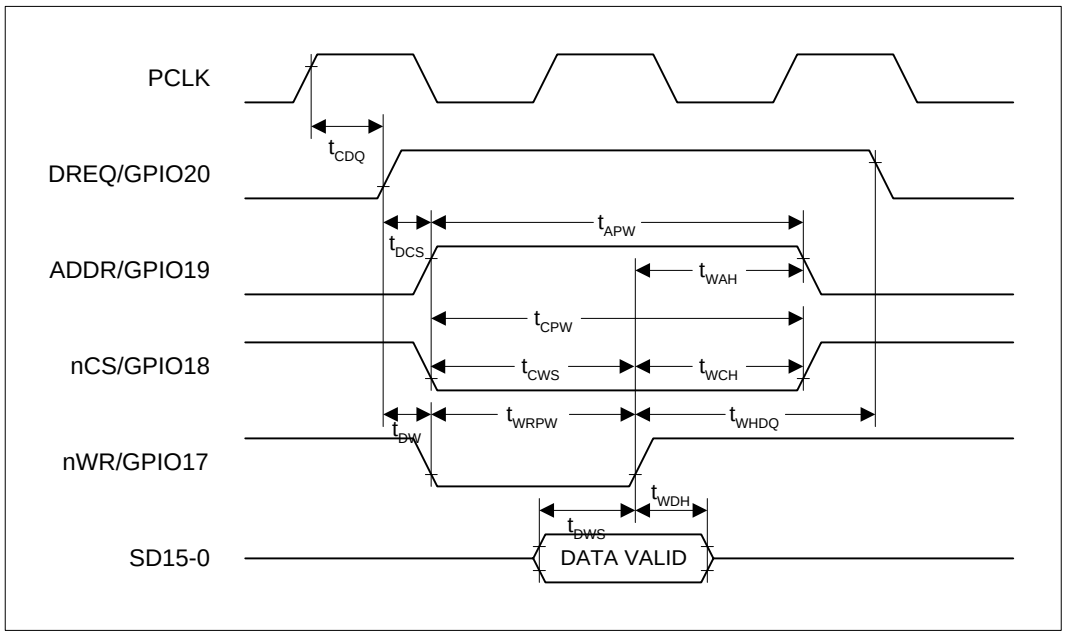


Figure 3-4. Mailbox/DMA Write

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
```

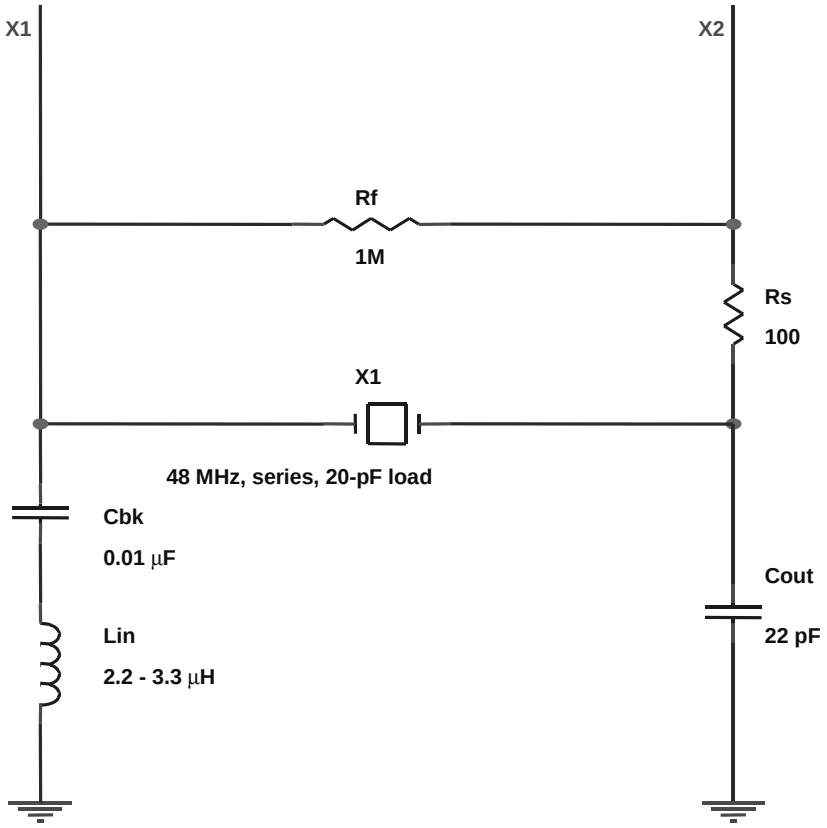


Figure 4-1. 48-MHz Crystal Circuit

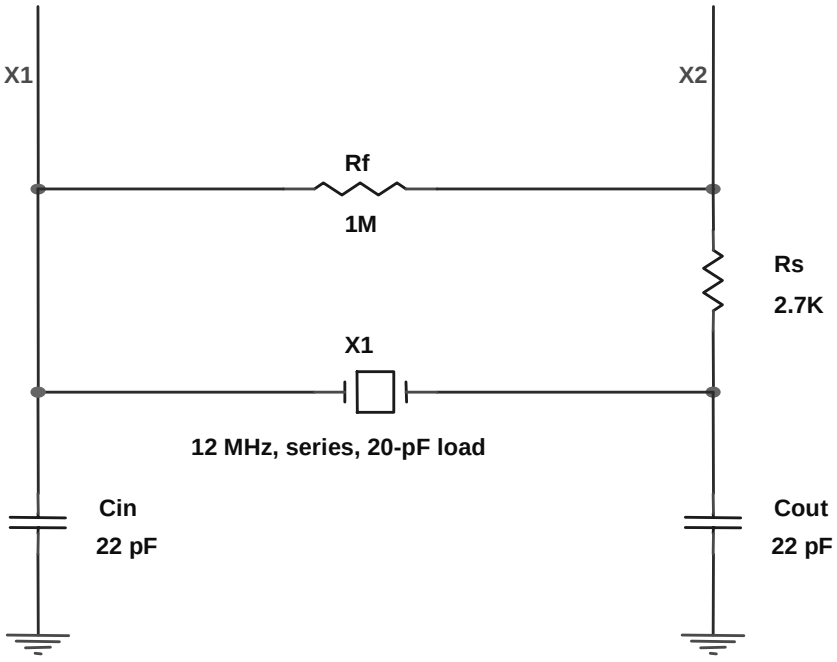


Figure 4-2. 12-MHz Crystal Circuit

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 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*:

Table 4-4. Software Interrupt Table

Interrupt Number	Vector Address	Interrupt Type
64 (0x40)	0x0080	2-wire serial interface_INT[6]
65 (0x41)	0x0082	Reserved for future extension of other Serial EEPROM
66 (0x42)	0x0084	UART_INT[6]
67 (0x43)	0x0086	SCAN_INT[6]
68 (0x44)	0x0088	ALLOC_INT[6]
69 (0x45)	0x008A	Data: start of free memory. Default=0x200[7]
70 (0x46)	0x008C	IDLE_INT
71 (0x47)	0x008E	IDLER_INT
72 (0x48)	0x0090	INSERT_IDLE_INT
73 (0x49)	0x0092	PUSHALL_INT[6]
74 (0x4a)	0x0094	POPALL_INT[6]
75 (0x4b)	0x0096	FREE_INT[6]
76 (0x4c)	0x0098	REDO_ARENA[6]
77 (0x4d)	0x009A	HW_SWAP_REG[6]
78 (0x4e)	0x009C	HW_REST_REG[6]
79 (0x4f)	0x009E	SCAN_DECODE_INT
80 (0x50)	0x00A0	USB_SEND_INT[6]
81 (0x51)	0x00A2	USB_RECEIVE_INT[6]
82 (0x52)	0x00A4	Reserved
83 (0x53)	0x00A6	USB_STANDARD_INT
84 (0x54)	0x00A8	Data: Standard loader vector. Defaut=0[7]
85 (0x55)	0x00AA	USB_VENDOR_INT
86 (0x56)	0x00AC	Data: USB_Vendor loader. Default = 0xff[7]
87 (0x57)	0x00AE	USB_CLASS_INT
88 (0x58)	0x00B0	Data: USB_Class_Loader. Default = 0[7]
89 (0x59)	0x00B2	USB_FINISH_INT
90 (0x5a)	0x00B4	Data: Device Descriptor. Default = Cypress Device Desc[7]
91 (0x5b)	0x00B6	Data: Configuration Desc. Default = Cypress Configuration[7]
92 (0x5c)	0x00B8	Data: String Descriptor. Default = Cypress String Desc.[7]
93 (0x5d)	0x00BA	USB_PARSE_CONFIG_INT
94 (0x5e)	0x00BC	USB_LOADER_INT
95 (0x5f)	0x00BE	USB_DELTA_CONFIG_INT
96 (0x60)	0x00C0	USB_PULLUP_INT
97 - 104	0xC2-0xD0	Reserved for future addition secondary USB Port
105 (0x69)	0x00D2	POWER_DOWN_SUBROUTINE
106-109	0xD4-0xDA	Reserved for future secondary USB Port
110-124	0xDE-0xF8	User's ISR or internal peripheral interrupt
125-127	0xFA-0xFE	Reserved for the Debugger

Notes:

6. These software vectors are reserved for the internal SL11R-BIOS. The user should not overwrite these functions.
7. These vectors are used as the data pointers. The user should not execute code (i.e. **JMP** or **INT**) to these vectors. See [Ref. 1] SL11R_BIOS for more information.

All these vector interrupts are read/write accessible. User can overwrite these default software interrupt vectors by replacing the user's interrupt service subroutine.

4.12 UART Interface.

The SL11R Controller UART port supports a range of baud rates from 900 Baud up to 115.2K Baud. Baud Rate selection is made in the UART Control Register. Buffer status can be monitored in the UART Status Register. Transmit and receive data is written or read from the UART data register. The UART timers are independent of the general-purpose timers. The UART will cause "edge trigger" type interrupts when the receive buffer becomes FULL or the transmit buffer becomes EMPTY.

The SL11R BIOS uses the UART port for the software debugging process. It is recommended that the user include this interface in their hardware design. A simple 4-pin header may be used to connect to a serial cable equipped with a MAX202 transceiver.

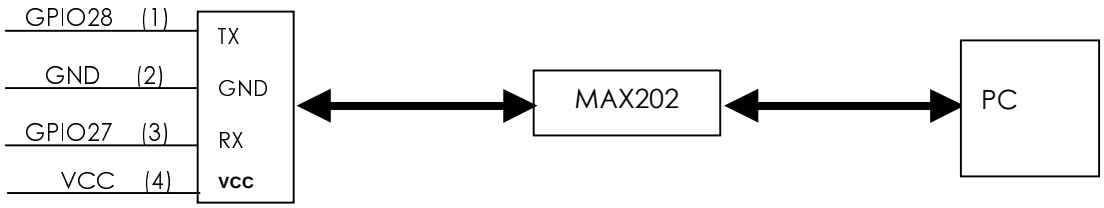


Figure 4-3. UART Port Connection

The SL11R BIOS uses GPIO28 for data transmit (TX) and GPIO27 for data receive (RX). These two pins cannot be used for any other purpose.

Note: On reset, the SL11R BIOS will configure the UART to operate at 14,400 baud. Other parameters are: 1 stop bit, 8 data bits, no parity.

4.12.1 UART Control Register (0xC0E0: R/W)

The SL11R allocates two General Purpose I/O signals for the UART function. They are GPIO28 (UART_TXD) and GPIO27 (UART_RXD). On reset, the SL11R BIOS will default this register to the value of 0x000b (i.e. UART Enable and Baud = 14.4K Baud).

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	DIV8	B2	B1	B0	E

D15-D5	Reserved bits	Set to all zeros.
D4	DIV8	
		Acts as a pre-scaler if set to '1', divides the clock by 8 before generating the UART clock.
D3-1	B2-0	

Baud Rate	Selector Bits.	with /8 Prescaler
000	115.2K Baud	14.4K Baud
001	57.6K Baud	7.2K Baud
010	38.4K Baud	4.8K Baud
011	28.8K Baud	3.6K Baud
100	19.2K Baud	2.4K Baud
101	14.4K Baud	1.8K Baud
110	9.6K Baud	1.2K Baud
111	7.2K Baud	0.9K Baud

D0	E	Enable UART when set = ’1’. When ’0’ UART pins are GPIO.

4.12.2 UART Status Register (0xC0E2: Read Only)

This register is used by the SL11R BIOS to detect the UART status function via RXF and TXE flags.

D15-D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	RXF	TXE

D15-D2	Reserved bits	Set to all zeros.
D1	RXF	Receive Buffer Full Flag.
D0	TXE	Transmit Buffer Empty Flag. Set to ‘1’ when data moves from buffer to output shift register.

Note:

No error detection for received data is supported.

4.12.3 UART Transmit Data Register (0xC0E4: Write Only)

This register is used by the SL11R BIOS to send data to the host.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

D7-D0	TR7-0	UART Transmit Data

4.12.4 UART Receive Data Register (0xC0E4: Read Only)

This register is used by the SL11R BIOS to receive data from the host.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

D7-D0	RD7-0	UART Receive Data.

4.13 Serial EEPROM Interface (2-wire serial 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 the two-wire serial EEPROM type. The serial EEPROM can be used to store specific Peripheral USB configuration and add on value functions. It can also be used for field product upgrades

The SL11R BIOS uses an interrupt to read and write from/to an external serial EEPROM. The recommended serial EEPROM device is a 2-Wire Serial CMOS EEPROM (AT24CXX Device Family). Currently, the SL11R BIOS Revision 1.1 allows reading/writing to/from EEPROM, up to 2K Bytes (16K bits), 2-wire serial interface device (i.e. AT24C16).

The user’s program and USB vendor/device configuration can be programmed and stored into the external EEPROM device. On power up the content of the EEPROM will be downloaded into RAM and may be executed as code or used as data, or both. The advantage of the 2-wire serial interface/EEPROM interface is the space and cost saving when compared to using an external 8-bit PROM/EPROM.

The SL11R BIOS uses two GPIO pins, GPIO31 and GPIO30 to interface to an external serial EEPROM (see *Figure 4-4*):

- GPIO31 is connected to the Serial Clock Input (SCL).
- GPIO30 is connected to the Serial Data (SDA).
- We recommend you add a 5K to 15K pull-up resistor on the Data line (e.g. GPIO30).

- Pin 1 (A0), Pin 2 (A1), Pin 3 (A2), Pin 4 (GND) and Pin 7 Write Protect) are connected to Ground.

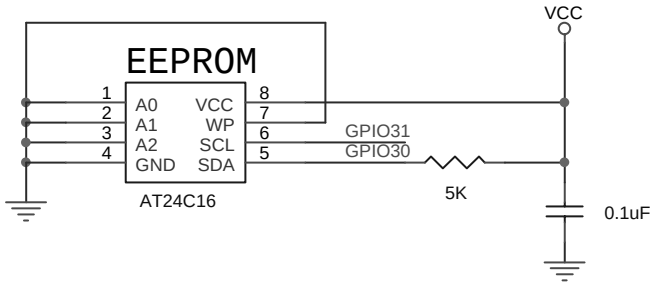


Figure 4-4. 2-Wire Serial Interface 2K-byte Connection

The current SL11R BIOS only support up to a 2Kbyte serial EEPROM. To read and write to a device that is larger than 2Kbytes, the SL11R-BIOS requires additional serial EEPROM to be connected as shown in *Figure 4-5*.

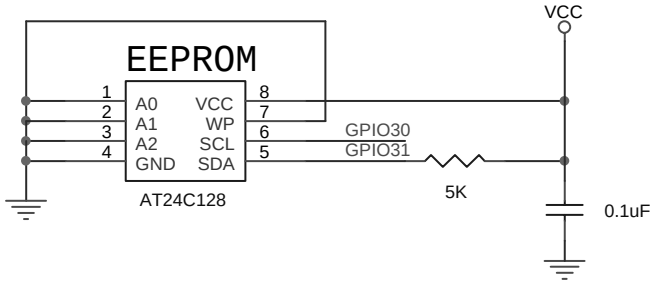


Figure 4-5. 2-Wire Serial Interface 16K Connection

In this example, the SL11R BIOS will first access the (small) program residing on **IC1** serial EEPROM, and then it will access the second **IC2 EEPROM** (see [Ref. 1] SL11R_BIOS for more information).

4.14 External SRAM, EPROM, DRAM

The SL11R has a multiplexed address port and 16-bit data port. These interface signals are provided to interface to an external SRAM, ROM or DRAM. The DRAM port provides RAS, CAS, RD and WR control signals for data access and refresh cycles to the DRAM. At boot up stage, the SL11R BIOS configures the SL11R for external SRAM and serial EEPROM. In addition, the external memory interface is set up as 16-bit and 7 wait states for both external SRAM and EEPROM. The DRAM controller needs to be set up by the user.

Example 2 SL11R extended memory setup:

```
internal_rom_start:
    mov     [0xC03A],0x0077     ;set 16-bit ROM & 7 wait
    cmp     [0xC100],0xCB36     ;check for special pattern in external ROM
    je      0xC102              ;if it's there, jump to it
    mov     [0xC006],0x10       ;2/3 clock
    mov     [0xC008],1          ;at 24 MHz
    mov     [0xC03E],3          ;extra wait state for ROM and Debug
    cmp     [0xC100],0xC3B6     ;external ROM has 0xC3B6 as first 16 bits
    je      xrom_ok
    cmp     b[0xC100],0xB6      ;check 0xc3b6 for 8-bit ROM
    jne     xrom_ok
    or      [0xC03A],0x80       ;set for 8-bit ROM
```

```
xrom_ok:
        mov     [0xC00],0xC3B6    ;check 0xC3B6 for 16-bit RAM
        cmp     [0xC00],0xC3B6
        je      xram_ok
        or      [0xC03A],8        ;set for 8-bit external RAM
xram_ok:
```

Note:

The external memory devices can be 8 or 16 bits wide, and can be programmed to have up to 7 wait-states. External SRAM/PROM requires one wait state.

4.14.1 Memory Control Register (0xC03E: R/W)

This register provides control of Wait States for the internal RAM and ROM.

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	RA	RO	DB

D2	RA	If ‘1’, one-wait state for internal RAM is added
D1	RO	If ‘1’, one-wait state for internal ROM is added
D0	DB	If ‘1’, DEBUG mode is enabled. Internal address bus is echoed to external address pins.

4.14.2 Extended Memory Control Register (0xC03A: R/W)

This register provides control of Wait States for the external SRAM/DRAM/EPROM.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	RM	EM3	EM2	EM1	EM0	RO3	RO2	RO1	RO0	RA3	RA2	RA1	RA0

D12	RM	ROM Merge. If ‘1’, nXROMSEL is active if nXMEMSEL is active.
D11	EM3	Extended Memory Width ('0' = 16, '1' = 8)
D10-8	EM2-0	Extended Memory Wait states (0 - 7)
D7	RO3	External ROM Width ('0' = 16, '1' = 8)
D6-4	RO2-0	External ROM wait states (0 - 7)
D3	RA3	External RAM Width ('0' = 16, '1' = 8)
D2-0	RA2-0	External RAM Wait States (0 - 7)

Note:

The default Wait State setting on power up or reset is 7 wait states.

4.14.3 Extended Page 1 Map Register (0xC018: R/W)

This register contains the Page 1 high order address bits. These bits are always appended to accesses to the Page 1 Memory mapped space. The default is 0x0000.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	A21	A20	A19	A18	A17	A16	A15	A14	A13

If Bit A21 is ‘1’, Page 1 reads/writes will access external DRAM

If bit A21 is ‘0’ Page 1 reads/writes will access some other external area (SRAM, ROM or peripherals). nXMEMSEL will be the external Chip Select for this space.

D8-0 A21-13

Page 1 high order address bits. The address pins on A21-A13 will reflect the content of this register when SL11R accesses the address 0x8000-0x9FFF.

4.14.4 Extended Page 2 Map Register (0xC01A: R/W)

This register contains the Page 2 high order address bits. These bits are always appended to accesses to the Page 2 Memory mapped space. The default is 0x0000.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	A21	A20	A19	A18	A17	A16	A15	A14	A13

If Bit A21 is ‘1’, Page 2 reads/writes will access external DRAM

If bit A21 is ‘0’

Page 2 reads/writes will access some other external area (SRAM, ROM or peripherals) and nXMEMSEL will be the external Chip Select for this space.

D8-0 A21-13

Page 2 high order address bits. The address pins on A21-A13 will reflect the content of this register when SL11R access the address 0xA000-0xBFFF.

4.14.5 DRAM Control Register (0xC038: R/W)

A multiplexed address port and 16-bit data port are provided to interface to an external 256Kx16 or a 1Megx16 EDO DRAM. The port provides nRAS, nCASL, nCASH, nDRAMWR and nDRAMOE control signals for data access and refresh cycles to the DRAM. This register is designed to control the DRAM interface.

D15-D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	DT	PE	RE

D2 DT DRAM Turbo, Enable when set = '1'. Uses 1 clock for CAS instead of 2.

D1 PE DRAM Page Mode Enable when set = '1'.

D0 RE DRAM Refresh Enable when set = '1'.

Note:

- Most of EDO and Page mode DRAM can be used as long as the CAS signal is issued before the RAS signal.
- Page mode access allows multiple CAS addresses to be issued within 1 Row address. The Page really corresponds to the Row. Once the Row address has been accessed, any accesses to that Page can be made without issuing the Row address again. Only the Column address is necessary. This allows for faster read and write accesses to the same page.

4.14.6 Memory Map

The total memory space allocated by the SL11R is 64K-bytes. Program, data, and I/O space are contained within a 64K-byte address space. The program code or data can be stored in either external RAM or external ROM.

The SL11R allows extended data (video) to be stored on an external EDO DRAM. The entire (video image) data can be transferred via DMA directly to DRAM without software intervention. The total DMA size can be up-to 2M-bytes. The SL11R processor can access DRAM data via the address space from **0x8000** to **0xBFFF**.

The SL11R Controller provides a 16-bit Memory interface that can support a wide variety of external DRAM, RAM and ROM devices. The SL11R Controller memory space is byte addressable and is divided as follows:

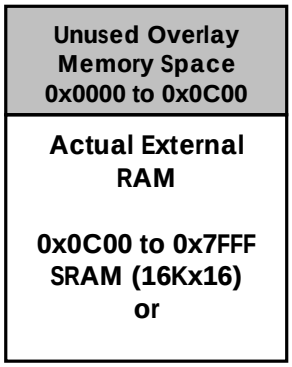
Table 4-5. Memory Map

Function	Address
Internal RAM	0x0000 – 0x0BFF
External RAM	0x0C00 – 0x7FFF[8]
Extended Page 1/DRAM	0x8000 – 0x9FFF
Extended Page 2/DRAM	0xA000 – 0xBFFF
Memory Mapped Registers	0xC000 – 0xC0FF
External ROM	0xC100 – 0xE7FF[9]
Internal ROM	0xE800 – 0xFFFF

Each External memory space can be 8 or 16 bits wide, and can be programmed to have up to 7 wait-states.

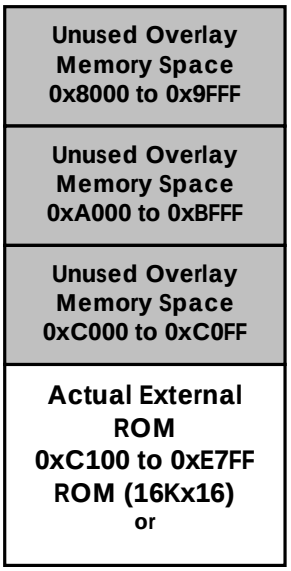
Notes:

8. The External RAM address from 0x0000 to 0x0C00 will not be accessible from the SL11R processor. This is an overlay memory space between internal RAM and external RAM. The addressable external RAM will occupy from 0x0C00-0x7FFF, which is 29K-byte. The signal name nXRAMSEL on SL11R-pin56 will be active when the CPU access address from 0x0C00 to 0x7FFF.

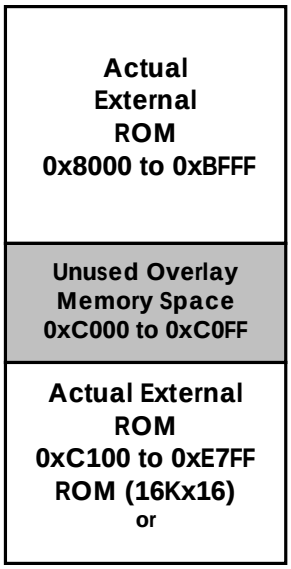


9. When bit 12 (ROM Merge Bit) of the Extended Memory Controller Register at address 0xC03A is '0', then the External ROM address space will be mapped from 0xC100 to 0xE7FF. The address from 0x8000 to 0xC100 and the address from 0xE800 to 0xFFFF are the overlay memory spaces. The actual total size of the external ROM will be (0xE800-0xC100), which is 9.75K-byte. The signal nXROMSEL on the SL11R (pin57) will be active when the CPU accesses the address from 0xC100 to 0xE7FF. The signal nXMEMSEL on the SL11R (pin58) will be active when the CPU accesses the address from 0x8000 to 0xBFFF. When bit 12 (ROM Merge Bit) of the Extended Memory Controller Register at address 0xC03A is '1', then the External ROM address space will be mapped into these windows: 0x8000 to 0xBFFF and 0xC100 to 0xE7FF. The address from 0xC000 to 0xC100 and the address from 0xE800 to 0xFFFF are the overlay memory spaces. The actual total size of the external ROM will be (0xC000-0x8000) and (0xE800-0xC100), which is 16K-byte + 9.75K-bytes, or 25.75K.

Bit 12 (ROM Merge) of the Extended Memory Controller Register = 0



Bit 12 (ROM Merge) of the Extended Memory Controller Register = 1



4.15 General Timers and Watch Dog Timer

The SL11R Controller has two built in programmable timers that can provide an interrupt to the SL11R Engine. The timers decrement on every microsecond clock tick. An interrupt occurs when the timer reaches zero.

4.15.1 Timer 0 Count Register (0xC010: R/W)

The SL11R BIOS uses the timer 0 for time-out function and power down mode. At the end of the power up, the SL11R BIOS disables the timer 0 interrupt. If you wish to use timer 0 for power down function, see the [Ref. 1] SL11R_BIOS for more information.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

D15-0	T15-0	Timer Count value.

4.15.2 Timer 1 Count Register (0xC012: R/W)

The SL11R timer 1 is for user applications. The SL11R BIOS does not use this timer.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

D15-0	T15-0	Timer Count value

4.15.3 Watchdog Timer Count & Control Register (0xC00C: R/W)

The SL11R provides a Watchdog timer to monitor certain activities. The Watchdog timer can also interrupt the SL11R processor. The default value of this register is 0x0000.

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	WT	TO1	TO0	ENB	EP	RC

D5	WT	Watchdog Time-out occurred.		
D4-3	TO1-0	Time-out Count:	00	01 milliseconds
			01	04 milliseconds
			10	16 milliseconds
			11	64 milliseconds
D2	EP	Enable Permanent WD timer. If set =’1’		
		WD timer is always enabled. Cleared only on Reset.		
D1	ENB	Enable WD Timer operation when =’1’.		
D0	RC	Reset Count. When set = ’1’.		

Notes:

- You must assert Reset Count (RC), before time-out occurs to avoid Watchdog trigger
- The Watchdog Timer overflow causes an internal processor reset. The Processor can read the WT bit after exiting reset to determine if the WT bit is set. If it is set, a watchdog time-out occurred.
- The WT value will be cleared on the next external reset.

4.16 Special GPIO Function 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 USB power is still applied to the circuit) and the GPIO20

for controlling power off function. The GPIO20 can be used for device low power mode; it will remove power from the peripherals in suspend mode. Once USB power is restored, the power to the peripherals may be enabled. The SL11R BIOS will execute the pull up interrupt upon power-up. To use this feature, the GPIO29 pin must be connected to the DATA+ line of the USB connector (see Figure below). For more information about this function, see the [Ref. 1] SL11R_BIOS.

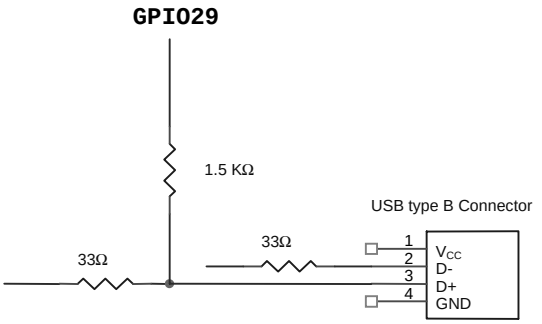


Figure 4-6. Special GPIO Pull-up Connection Example

4.17 Programmable Pulse/PWM Interface

The SL11R Controller supports four Programmable Digital Pulse output channels. These channels can also be used for Pulse Width Modulation (PWM) operation. Operation is directed by the PWM Control Register, Maximum Count Register, and the individual Start and Stop Counter Registers. These are provided for each of the four output channels. To set up PWM operation, the Maximum Count Register is set to the desired maximum count value. Then the start and stop value for each channel is written with the required values. The start and stop values are chosen to achieve the desired pulse widths during each cycle. When the channels are disabled (by the Control Register), the associated I/O pins revert to GPIO use.

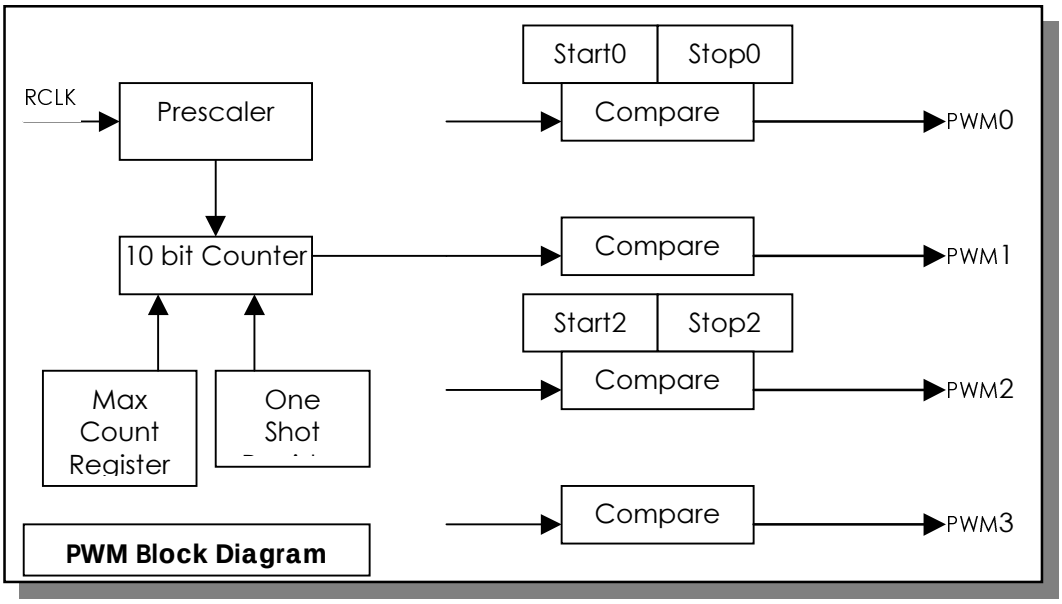


Figure 4-7. PWM Block Diagram

Note: The **RCLK** is the resulting clock (see the Speed Control Register (0xC006: R/W)).

4.17.1 PWM Control Register (0xC0E6: R/W)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
ST	0	0	0	SC2	SC1	SC0	OS	P3	P2	P1	P0	EN3	EN2	EN1	EN0

D15	ST	Start Bit. Set to ’1’ to begin operation. ’0’ stops operation
D14-12	Reserved	always ’0’ ’s.
D11-9	SC2-0	Prescaler value selection

SC2-0	Freq.
000	48.00 MHz
001	24.00 MHz
010	06.00 MHz
011	01.50 MHz
100	375 kHz
101	93.80 kHz
110	23.40 kHz
111	05.90 kHz

D8	OS	Enable One Shot Mode for PWM channels. One Shot mode runs the number of counter cycles set in the PWM cycle count register and then stops. The default is continuous repeat.
D7-D4	P3-0	Individual Polarity bits for channels 3 - 0. ’1’ is active high or rising edge pulse.
D3-D0	EN3-0	Individual Enable bits for channels 3 - 0. ’1’ enables.

Notes:

- If not enabled, i.e., if set is '0', the pins become GPIO. To force the outputs to ‘0’ or ‘1’:
- If start register = stop register, then output stays at “0”
- If stop register > Max count register, then output stays at “1”

4.17.2 PWM Maximum Count Register (0xC0E8: R/W)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

D15-10	Reserved	always '0' 's.
D9-0	C9-C0	Maximum Count Value.

4.17.3 PWM Channel 0 Start Register (0xC0EA: R/W)

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

D15-10	Reserved	always '0' 's.
D9-0	S9-S0	Start Count for PWM Channel 0.

4.17.4 PWM Channel 0 Stop Register (0xC0EC: R/W)

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

D15-10 Reserved always ’0’ ’s.
D9-0 S9-S0 Stop Count for PWM Channel 0.

4.17.5 PWM Channel 1 Start Register (0xC0EE: R/W)

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

D15-10 Reserved always ’0’ ’s.
D9-0 S9-S0 Start Count for PWM Channel 1.

4.17.6 PWM Channel 1 Stop Register (0xC0F0: R/W)

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

D15-10 Reserved always ’0’ ’s.
D9-0 S9-S0 Stop Count for PWM Channel 1.

4.17.7 PWM Channel 2 Start Register (0xC0F2: R/W)

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

D15-10 Reserved always ’0’ ’s.
D9-0 S9-S0 Start Count for PWM Channel 2.

4.17.8 PWM Channel 2 Stop Register (0xC0F4: R/W)

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

D15-10 Reserved always ’0’ ’s.
D9-0 S9-S0 Stop Count for PWM Channel 2.

4.17.9 PWM Channel 3 Start Register (0xC0F6: R/W)

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

D15-10 Reserved always ’0’ ’s.
D9-0 S9-S0 Start Count for PWM Channel 3.

4.17.10 PWM Channel 3 Stop Register (0xC0F8: R/W)

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

D15-10 Reserved always ’0’ ’s.

D9-0 S9-S0 Stop Count for PWM Channel 3.

4.17.11 PWM Cycle Count Register (0xC0FA: R/W)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

D15-0 C15-0

Number of cycles to run in one-shot mode (0-64K)

The OS bit in the PWM Control Register must be set.

Note: Number of OS Cycles to run = C+1. Example for 1 Cycle, set C=2

4.18 Fast DMA Mode

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

4.18.1 DMA Control Register (0xC02A: R/W)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	TSZ	DIR	DMA

External device data presented to S15-SD0/SD7-SD0 is automatically written into the RAM of the SL11R, under fast DMA control. The DMA must be enabled in the DMA Control and Address register.

D2 TSZ Transfer Size. 8 bit when set to ’1’, 16-bit when set to ’0’

D1 DIR

DMA Direction. When set to ’0’, data transfers from Peripheral to Memory. When set to '1’ Memory to Peripheral.

D0 DMA DMA Enabled when set to '1'. Bit clears to ‘0’ when DMA is done.

Note for DVC 8-Bit DMA mode:

Set Transfer Size to 16 bits for the DVC 8-Bit DMA mode.

Set DMA Direction for ***Peripheral to Memory*** for DVC 8-Bit DMA mode.

4.18.2 Low DMA Start Address Register (0xC02C: R/W)

This register contains the low order word of the starting DMA address.

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-A0 Low 16 Bits of DMA address

4.18.3 High DMA Start Address Register (0xC02E: R/W)

This register contains the high order word of the starting DMA address.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	A21	A20	A19	A18	A17	A16

D5-D0 A21-A16 High Address bits for DMA start address

Note:

A21 = 1 the starting memory address will be in the DRAM. The A21 bit in the High DMA Stop Address register must match this bit.

4.18.4 Low DMA Stop Address Register (0xC030: R/W)

This register contains the low order word of the stopping SL11R memory address. This is the last DMA address in memory. DMA will stop when this address is reached and if the **FDMA** bit in the Interrupt Enable Register (0xC00E: R/W) is enabled, an interrupt will also be generated.

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-A0 Low 16 Bits of the DMA stop address

4.18.5 High DMA Stop Address Register (0xC032: R/W)

This register contains the high order word of the stopping SL11R memory address. This is the last DMA address in memory. DMA will stop when this address is reached.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	A21	A20	A19	A18	A17	A16

D5-D0 A21-A16 High Address bits for DMA stop address.

Note:

A21 = 1 the stopping memory address will be in the DRAM. The A21 bit must be matched on the High DMA Start Address register.

5.0 SL11R Interface Modes

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

Note: *The UART and 2-wire serial interface IO pins are fixed in all SL11R Interface modes.*

5.1 General Purpose IO mode (GPIO)

In GPIO mode, the SL11R has up to 32 general-purpose I/O signals available. However, there are 4 pins used by the UART and the 2-wire serial interface that cannot be used as the GPIO pins. A typical application for this GPIO is Parallel Port to USB. The SL11R executes at 48MHz -- 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 the EPP mode. Other available General Purpose I/O pins can be programmed for peripheral control and/or status, etc.

When the SL11R interface is in GPIO mode, a number of GPIO pins are used to support the parallel interface. The remaining pins can be used for GPIO or if desired, some can be configured for special functions.

The following registers are used for all pins configured as GPIO. The outputs are enabled in the I/0 Control registers. Note that the output Data can be read back via the Output Data Register even if the outputs are not enabled.

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

5.1.1 I/O Control Register 0 (0xC022: R/W)

This register controls the input/output direction of the GPIO data pins from GPIO15 to GPIO0. When any bit of this register set to one, the corresponding GPIO data pin becomes an output pin. When any bit of this register is set to zero, the corresponding GPIO data pin becomes an input pin.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

D15-0 E15-0 Enable individual outputs, GPIO 15-0. Logic '1' enables.

5.1.2 I/O Control Register 1 (0xC028: R/W)

This register controls the input/output direction of the GPIO data pins from GPIO31 to GPIO16. When any bit of this register set to one, the corresponding GPIO data pin becomes an output pin. When any bit of this register is set to zero, the corresponding GPIO data pin becomes an input pin.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16

D15-0 E31-16 Enable individual outputs, GPIO 31-16. Logic '1' enables.

5.1.3 Output Data Register 0 (0xC01E: R/W)

This register controls the output data of the GPIO data pins from GPIO15 to GPIO0.

Note: A read of this register reads back the last data written, not the data on pins configured as input (see below).

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
O15	O14	O13	O12	O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1	O0

D15-0 O15-0 Output Pin Data

5.1.4 Output Data Register 1 (0xC024: R/W)

This register controls the output data of the GPIO data pins from GPIO31 to GPIO16.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
O31 15	O30 14	O29 13	O28 12	O27 11	O26 10	O2 59	O2 48	O2 37	O2 26	O2 15	O2 04	O1 93	O1 82	O1 7	O1 60

D15-0 O31-16 Output Pin Data

5.1.5 Input Data Register 0 (0xC020: Read only)

This register reads the input data of the GPIO data pins from GPIO15 to GPIO0.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

D15-0 I15-0 Input Pin data

5.1.6 Input Data Register 1 (0xC026: Read only)

This register reads the input data of the GPIO data pins from GPIO31 to GPIO16.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16

D15-0 I31-16 Input Pin data

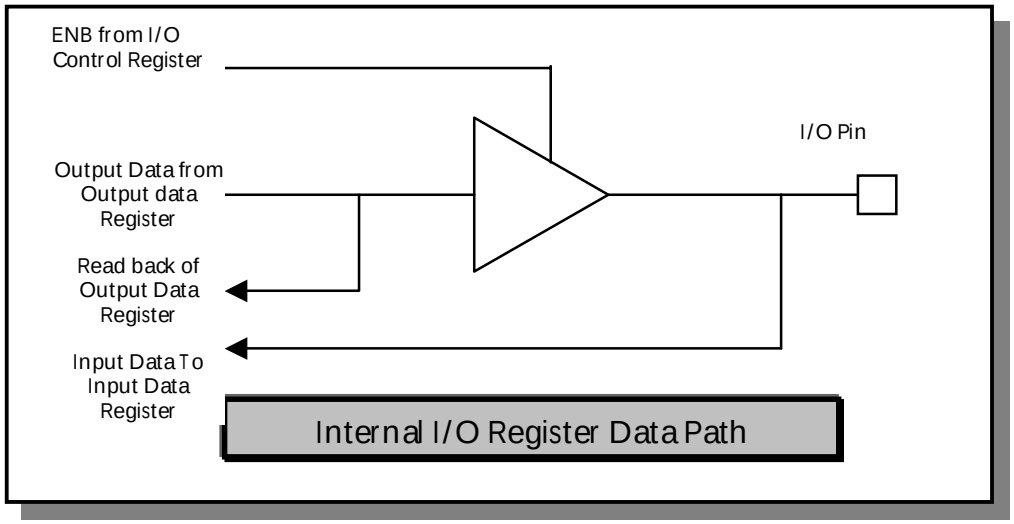


Figure 5-1. GPIO Mode Block Diagram

5.2 8/16-bit DMA Mode

This mode includes the **Mailbox Protocol** and **DMA Protocol**. The **Mailbox Protocol** allows asynchronous exchange of data between an external Processor (i.e. DSP or Microprocessors) and the SL11R, via the bidirectional data port SD15-SD0 (GPIO15-0). The DMA Protocol allows the large data blocks to be transferred to or from SL11R memory devices via the 8/16-bit DMA port.

The SL11R has four built-in PWM output channels available in the 8/16-bit DMA mode. Each channel provides a programmable timing generator sequence which can be used to interface to various line CCD, CIS, CMOS image sensors or can be used for other types of applications (see Programmable Pulse/PWM Interface for more detail of controlling these PWM functions).

Note: Any other unused IO pins can be used as the GPIO pins under control of the General Purpose IO mode (GPIO).

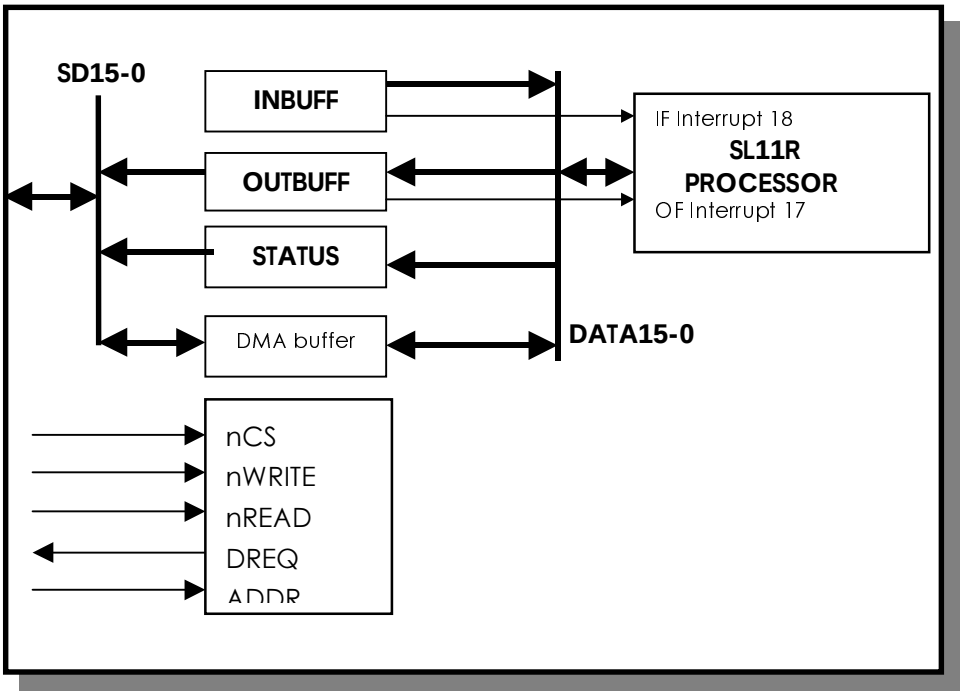


Figure 5-2. 8/16-bit DMA Mode Block Diagram

5.2.1 Mailbox Protocol

The physical interface for the Mailbox is shared with the DMA data path on the SD15-SD0 bus. When accessing the Mailbox INBUFF & OUTBUFF registers, the ADDR pin should be driven high. The ADDR pin should be driven low to access the Mailbox STATUS register. The external processor and SL11R can both access the INBUFF, OUTBUFF & STATUS Mailbox registers. The SL11R includes two interrupt vectors for this Mailbox Protocol. Whenever the external Processor accesses the Mailbox, the associated interrupt will be generated.

Note:

- To enable the Mailbox interrupt, the bit **MBX** in the Register 0xC00E must be enabled.
- The external processor cannot access the Mailbox while DMA is in progress.

5.2.2 INBUFF Data Register (0xC0C4: R/W)

The external processor will write to this register with the ADDR signal set to one and the SL11R will read this register after receiving the interrupt (if the MBX interrupt is enabled in the Register 0xC00E).

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

D15-0 D15-0 Data from input Mailbox

5.2.3 OUTBUFF Data Register (0xC0C4: R/W)

The SL11R will write to this register and the external processor will read from this register with the ADDR signal set to one. The SL11R will receive an interrupt after the external processor finished reading (if the MBX interrupt is enabled in the Register 0xC00E).

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

D15-0 D15-0 Data for Output Mailbox

5.2.4 STATUS Register (0xC0C2: Read Only)

The external processor can read the STATUS of the OUTBUFF and INBUFF Status bits from this output buffer. The external ADDR pin should be driven to low when reading this STATUS register.

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	IF	OF

D1	IF	INBUFF Full
D0	OF	OUTBUFF Full

Note:

The SL11R also can access this register.

5.2.5 DMA Protocol

The physical interface for the DMA is shared with the Mailbox protocol on the SD15-SD0 bus. If the DREQ (DMA Request Enable) bit is set, this enables SL11R DMA cycles to or from the external device (scanner, printer, camera, modem or etc.). The DREQ is asserted by the SL11R when data is ready to be sent or received. When the external device is ready to send Data, it asserts the **nWRITE** signal. Data must be available at this point. If the external device is ready to accept data, it asserts the **nREAD** signal.

The DMA mode can be used to move large amounts of data to or from a variety of peripherals such as Scanners, Printers, Cable Modems, External Storage devices, and others. For example for a DVC, video data from the camera can be moved via DMA to an internal memory buffer for subsequent transfer to the USB host. This data can be transferred to the host via the USB DMA engine (i.e. no SL11R Processing is involved, since the USB has its own DMA engine).

Users can program 8 bit or 16 bit DMA transfers in either direction; Peripheral to SL11R or SL11R to Peripheral. A control register (0xC02A) sets the DMA bus width, direction and DMA enable and four further registers control the DMA start and end addresses (see Fast DMA Mode, section 2.14). Furthermore, if the **FDMA** bit in Register 0xC00E is enabled, an interrupt will be issued to indicate the DMA operation is complete.

5.2.6 DMA Control Register (0xC0C0: R/W)

Before setting this register, the Low DMA Start Address (0xC02C), High DMA Start Address (0xC02E), the Low DMA Stop Address (0xC030) and the High DMA Stop Address (0xC032) must be configured.

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	DREQ	D1	D0

D2	DREQ	External DREQ DMA Enable, if set to ‘1’, the SL11R can DMA to or from the external device (scanner or printer) by asserting the DREQ signal when data is requested or ready to send.
D1	D1	Set to ‘1’
D0	D0	Set to ‘1’

5.3 Fast EPP Mode

This interface is designed to interface with a specially optimized high-speed EPP interface. The SL11R processor has direct access to the EPP control port.

The EPP function has four transfer modes: Data Write, Data Read, Address Write and Address Read. Strobe signals **nDTSRB** and **nASTRB** are generated by the SL11R for data or address operations respectively. The signal **nWRITE** indicates read or write as described below.

Note:

- The Fast DMA and PWM Interface will not be supported in this mode.
- Any other unused IO pins can be used as the GPIO pins under control of theGeneral Purpose IO mode (GPIO).

5.3.1 EPP Data Register (0xC040: R/W)

Writing to this register results in the generation of a pulse on the **nDTSRB** output pin, with the **nWRITE** output pin low, and the register data being driven onto the SD7-SD0 Data bus.

Reading from this register results in the generation of a pulse on the **nDTSRB** output pin, with the **nWRITE** output pin high, and data being read from the SD7-SD0 Data bus.

D7	D6	D5	D4	D3	D2	D1	D0
SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

D7-D0	SD7-SD0	EPP data

5.3.2 EPP Address Register (0xC044: R/W)

Writing to this register results in the generation of a pulse on the **nASTRB** output pin, with the **nWRITE** output pin low, and the register data being driven onto the SD7-SD0 Data bus.

Reading from this register results in the generation of a pulse on the **nASTRB** output pin, with the **nWRITE** output pin high, and data being read from the SD7-SD0 Data bus.

D7	D6	D5	D4	D3	D2	D1	D0
A7	A6	A5	A4	A3	A2	A1	A0

D7-D0	A7-A0	Device and register address value from the SD7-SD0 Data Bus.

5.3.3 EPP Address Buffer Read Register (0xC046: Read Only)

Reading this register returns existing data from Read Buffer to the I/O processor. The **nASTRB** will not be asserted.

D7	D6	D5	D4	D3	D2	D1	D0
A7	A6	A5	A4	A3	A2	A1	A0

D7-D0	A7-A0	Read Data

5.3.4 EPP Data Buffer Read Register (0xC042: Read Only)

Reading this register returns existing data from Read Buffer to the I/O processor. The **nDTSRB** will not be asserted

D7	D6	D5	D4	D3	D2	D1	D0
SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

D7-D0	SD7-SD0	EPP data from the SD7-SD0 data bus.

5.3.5 EPP Status Data Register (0xC04E: R/W)

This register is used to read the actual status signal from GPIO9-8. The P9 pin will be set to the value in D7 when written. GPIO8 and GPIO9 are not affected by writing to this register.

D7	D6	D5	D4	D3	D2	D1	D0
P9 (GPIO21)	0	0	0	0	0	INTR	WAIT

D7	P9	Value output on GPIO21.
D1	INTR	from the GPIO8
D0	WAIT	line from GPIO9.

5.3.6 EPP P_REG Register (0xC050: R/W)

The SL11R has a set of eight pins labeled P1-P8, which are general-purpose output pins. The functionality of each bit is selected in the respective Control Register. The SL11R 16-bit processor has access to these pins through a read write, which is defined in the P_REG register. The bit assignments are noted in the following table.

D7	D6	D5	D4	D3	D2	D1	D0
P8 (GPIO15)	P7 (GPIO14)	P6 (!GPIO13)	P5 (GPIO20)	P4 (GPIO19)	P3 (GPIO18)	P2 (GPIO17)	P1 (GPIO16)

D7-D0 P8-P1 Set to logic ‘1’ or ‘0’ for corresponding P output pin

Note:

A write to this register causes the SL11R to write this out the corresponding GPIO pins (Except P6). The output value of the P6 will be the complement of the value written.

5.3.7 Serial Interface Registers

The SL11R supports subset of an industry standard SPI serial interface, which provides the interface to serial interface device like Multi-Media or Memory Stick interface.

5.3.7.1 Serial Interface Control & Status Register Address C048H

The Serial Interface port of the SL11R can be used to control certain aspects of the interface to serial Memory Flash devices. There are cycles (Address, Write or Read) are implemented on this mode.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	B

D7-1 Not Defined.

D0 B - Busy. When Set = '1', indicates a cycle is in progress.

Cleared to '0' when the cycle has completed. No new cycles are allowed until Busy bit = '0'.

5.3.7.2 Serial Interface Address Register Address C04AH

D7	D6	D5	D4	D3	D2	D1	D0
A7	A6	A5	A4	A3	A2	A1	A0

A7-A0: Address value. Writing a value to this register results in an address cycle on the serial bus. The Busy bit in the status register (C048H) goes to '1' (high) until cycle has completed.

5.3.7.3 Serial Interface Data Write Register Address C04CH

D7	D6	D5	D4	D3	D2	D1	D0
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0

TD7-0: Data to be transmitted. Writing to this register initiates a write cycle on the serial data bus. The busy bit in the status register (C048H) goes to '1' (high) until the cycle has completed.

5.3.7.4 Serial Interface Data Read Register Read OnlyAddress C04CH

D7	D6	D5	D4	D3	D2	D1	D0
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

RD7-0: Read Data. Reading this register initiates a read cycle on the serial bus and sets the Busy bit = ’1’ in the Status Register (C048H). Read data is valid when the Busy bit in the status register is cleared to ’0’.

Notes:

The clock rate for the serial operation is 12MHz base on the processor clock at 48MHz. To change the clock rate user can change the processor clock rate via the register C008H.

A delay is needed between back to back serial accesses to allow serial shifting to occur.

GPIO11 will be used as the DATAS.

GPIO26 will be used as the CLKS.

GPIO25 will be used as the nENS (Can use this as the interrupt IRQ1).

The register C006H need to select this mode to make the interface work.

5.4 DVC 8-bit DMA Mode

This 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 Camera is input to the SL11R on the 8-bit video data bus (SD7-SD0) using a combination of clock and control signals and 8 bit DMA. The signals include a clock (MCK0), Field Index (FI), Sync and blanking signals (SYNC, PBLK), and Drive signals (VD and HD). The DMA Engine is used to transfer the image from the 8-bit bus (SD7-SD0) to external DRAM port. The software uses the Fast DMA configuration registers.

Note:

- The PWM Interface is not supported in this mode.
- Any other unused IO pins can be used as GPIO pins under control of the GPIO mode.

5.4.1 Video Status Register(0xC06E: Read Only)

The camera sends an 8 bit data bus with a number of control signals. The SL11R uses the following control signals to acquire the video data:

MCK0	From Camera	Pixel Clock = 9.534965MHz
FI	From Camera	Field Index. Contains 1 Vertical sweep + Blank
VD	From Camera	Active During Vertical Sweep.
PBLK	From Camera	Active during Horizontal sweep.
SD7-SD0	From Camera	CCD Video data.
N_RST	To Camera	Active Low Reset

The first 8 bytes of Data is discarded from the video stream starting at the assertion of each PBLK. After PBLK is de-asserted, 7 more bytes of data are taken. The Video Control and Status Register allows the acquisition of video images and provides power and reset controls for the camera.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	VRST	P-CONT	0	F	SC

D7-5	Reserved	Always = '0'
D4	VRST	Video Reset. '0' Resets Camera (See Note).
D3	P-CONT	When set = '1' powers up camera.
D2	Reserved	Always = '0'.

D1	F	When set = ’1’ capture even then odd field else when ’0’ capture even field only.
D0	SC	Start Capture When set = ’1’, clears to ’0’ when done.

Note:

After P-CONT is set high to provide camera power, the Video Reset must be held low a minimum of 100 milliseconds.

5.4.2 Camera Serial Interface Registers

The Serial Interface port of the SL11R is used to pass control information to the camera and to communicate with serial EPROM1, and EPROM2 (EPROM 1 is used to store camera default data). The Serial Port Control Register determines if the next operation will be a write cycle to the camera or the EEPROM, or if the operation will be a read cycle of the EPROM. When an EPROM or camera cycle is required the device must be selected in the Control Register, and an EEPROM address location must be written to the Serial Interface Address Register. When a Read Cycle is selected, writing the Busy Bit begins the cycle, and when completed, the valid returned data can be read in the Serial Interface Data Read Register.

The SL11R has a dedicated serial bus to support the camera and EPROM. On power up, an SL11R signal, AEEP, is driven high, (AEEP connects to the Camera DSP device), the SL11R Camera serial bus is three-stated, which allows the Camera DSP to read the camera default data directly from the EEPROM. (During this time the DSP drives the Serial Clock to the EEPROM and the CCS select line to the SL11R, EEP1 is driven high from the SL11R). After a timed interval, the AEEP line is driven low and the SL11R can now communicate to the camera via the Camera Serial Port.

5.4.3 Serial Interface Control & Status Register (0xC068: R/W)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	M1	M0	Busy

D7-3	Reserved	0	
D2-D1	M1-M0	Mode Setting:	11 = AEEP MODE 10 = EPROM 2 01 = EPROM1 00 = DSP PROCESSOR
D0	Busy	When Set = ’1’, initiates a cycle to selected target. Cleared to ’0’ when the cycle has completed. No new cycles are allowed until Busy bit = ’0’.	

5.4.4 Serial Interface Address Register (0xC06A: Write Only)

D8	D7	D6	D5	D4	D3	D2	D1	D0
P2	P1	P0	A5	A4	A3	A2	A1	A0

D8-6	P2- P0	Command: 101 = Write, 110 = read
D5-0	A5-A0	EEPROM or Camera Register address.

5.4.5 Serial Interface Data Write Register (0xC06C: Write Only)

When data is transmitted, setting the Busy bit in the Control Status Register initiates transmission.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

D15-0	T15-0	Data to be transmitted.

5.4.6 Serial Interface Data Read Register (0xC06C: Read Only)

Read cycle data from the selected device in Control Status Register, and address specified in the Address Register. Data is valid after a Read Cycle has been initiated by setting the Busy bit in the Status Register. Data is valid after Busy bit clears to '0'.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0

D15-0 R15-0 Read Data from the SD7-SD0 Data Bus.

5.4.7 I/O Address Map

I/O Address Map

Function	Address	Mode
USB Endpoint 0 Address Register	0x0120	R/W
USB Endpoint 0 Count Register	0x0122	R/W
USB Endpoint 1 Address Register	0x0124	R/W
USB Endpoint 1 Count Register	0x0126	R/W
USB Endpoint 2 Address Register	0x0128	R/W
USB Endpoint 2 Count Register	0x012A	R/W
USB Endpoint 3 Address Register	0x012C	R/W
USB Endpoint 3 Count Register	0x012E	R/W
Configuration Register	0xC006	R/W
Speed Control Register	0xC008	R/W
Power Down Control Register	0xC00A	R/W
Watchdog Timer Count & Control Register	0xC00C	R/W
Interrupt Enable Register	0xC00E	R/W
Timer 0 Count Register	0xC010	R/W
Timer 1 Count Register	0xC012	R/W
Breakpoint Register	0xC014	R/W
Extended Page 1 Map Register	0xC018	R/W
Extended Page 2 Map Register	0xC01A	R/W
GPIO Interrupt Control Register	0xC01C	R/W
Output Data Register 0	0xC01E	R/W
Input Data Register 0	0xC020	Read Only
I/O Control Register 0	0xC022	R/W
Output Data Register 1	0xC024	R/W
Input Data Register 1	0xC026	Read Only
I/O Control Register 1	0xC028	R/W
DMA Control Register	0xC02A	R/W
Low DMA Start Address Register	0xC02C	R/W
High DMA Start Address Register	0xC02E	R/W

I/O Address Map (continued)

Function	Address	Mode
Low DMA Stop Address Register	0xC030	R/W
High DMA Stop Address Register	0xC032	R/W
DRAM Control Register	0xC038	R/W
Extended Memory Control Register	0xC03A	R/W
Memory Control Register	0xC03E	R/W
EPP Data Register	0xC040	R/W
EPP Data Buffer Read Register	0xC042	Read Only
EPP Address Register	0xC044	R/W
EPP Address Buffer Read Register	0xC046	Read Only
EPP Status Data Register	0xC04E	R/W
EPP P_REG Register	0xC050	R/W
Serial Interface Control & Status Register	0xC068	R/W
Serial Interface Address Register	0xC06A	Write Only
Serial Interface Data Write Register	0xC06C	Write Only
Serial Interface Data Read Register	0xC06C	Read Only
Video Status Register	0xC06E	Read Only
USB Global Control & Status Register	0xC080	R/W
USB Frame Number Register	0xC082	Read Only
USB Address Register	0xC084	R/W
USB Command Done Register	0xC086	Write Only
USB Endpoint 0 Control & Status Register	0xC090	R/W
USB Endpoint 1 Control & Status Register	0xC092	R/W
USB Endpoint 2 Control & Status Register	0xC094	R/W
USB Endpoint 3 Control & Status Register	0xC096	R/W
DMA Control Register	0xC0C0	R/W
STATUS Register	0xC0C2	Read Only
INBUFF Data Register	0xC0C4	R/W
OUTBUFF Data Register	0xC0C4	R/W
UART Control Register	0xC0E0	R/W
UART Status Register	0xC0E2	Read Only
UART Transmit Data Register	0xC0E4	Write Only
UART Receive Data Register	0xC0E4	Read Only
PWM Control Register	0xC0E6	R/W
PWM Maximum Count Register	0xC0E8	R/W
PWM Channel 0 Start Register	0xC0EA	R/W
PWM Channel 0 Stop Register	0xC0EC	R/W

I/O Address Map (continued)

Function	Address	Mode
PWM Channel 1 Start Register	0xC0EE	R/W
PWM Channel 1 Stop Register	0xC0F0	R/W
PWM Channel 2 Start Register	0xC0F2	R/W
PWM Channel 2 Stop Register	0xC0F4	R/W
PWM Channel 3 Start Register	0xC0F6	R/W
PWM Channel 3 Stop Register	0xC0F8	R/W
PWM Cycle Count Register	0xC0FA	R/W

6.0 Physical Connection

6.1 Package Type

100 PQFP.

6.2 GPIO and 8/16-Bit DMA Modes—Pin Assignment and Description

GPIO and 8/16-Bit DMA Modes—Pin Assignment and Description

Pin Name	Pin No.	GPIO Pins	Pin Type	GPIO & 8/16-bit DMA modes Pin Chip Revision 1.1
VDD	1		Power	+3.3 VDC Supply
D0	2		Bidir	External Memory Data Bus, Data0
D1	3		Bidir	External Memory Data Bus, Data1
D2	4		Bidir	External Memory Data Bus, Data2
D3	5		Bidir	External Memory Data Bus, Data3
D4	6		Bidir	External Memory Data Bus, Data4
D5	7		Bidir	External Memory Data Bus, Data5
D6	8		Bidir	External Memory Data Bus, Data6
D7	9		Bidir	External Memory Data Bus, Data7
D8	10		Bidir	External Memory Data Bus, Data8
D9	11		Bidir	External Memory Data Bus, Data9
D10	12		Bidir	External Memory Data Bus, Data10
D11	13		Bidir	External Memory Data Bus, Data11
GND	14		GND	Digital ground.
X1	15		Input	External 48-MHz Crystal or Clock Input.
X2	16		Output	External crystal output. No connection when X1 is used for clock input
VDD	17		Power	+3.3 VDC Supply
D12	18		Bidir	External Memory Data Bus, Data12
D13	19		Bidir	External Memory Data Bus, Data13
D14	20		Bidir	External Memory Data Bus, Data14
D15	21		Bidir	External Memory Data Bus, Data15
A20	22		Output	External Memory Address Bus, A20
A19	23		Output	External Memory Address Bus, A19
A18	24		Output	External Memory Address Bus, A18

GPIO and 8/16-Bit DMA Modes—Pin Assignment and Description (continued)

Pin Name	Pin No.	GPIO Pins	Pin Type	GPIO & 8/16-bit DMA modes Pin Chip Revision 1.1
A17	25		Output	External Memory Address Bus, A17
A16	26		Output	External Memory Address Bus, A16
A15	27		Output	External Memory Address Bus, A15
A14	28		Output	External Memory Address Bus, A14
A13	29		Output	External Memory Address Bus, A13
A12	30		Output	External Memory Address Bus, A12
A11	31		Output	External Memory Address Bus, A11
A10	32		Output	External Memory Address Bus, A10
A9	33		Output	External Memory Address Bus, A9
A8	34		Output	External Memory Address Bus, A8
A7	35		Output	External Memory Address Bus, A7
A6	36		Output	External Memory Address Bus, A6
A5	37		Output	External Memory Address Bus, A5
A4	38		Output	External Memory Address Bus, A4
A3	39		Output	External Memory Address Bus, A3
GND	40		GND	Digital ground.
A2	41		Output	External Memory Address Bus, A2
A1	42		Output	External Memory Address Bus, A1
A0	43		Output	External Memory Address Bus, A0
TEST	44		Input	No Connection, MFG test only Note: SL11RB NC=48 MHz, GND=12 MHz
nWRL	45		Output	Active LOW, write to lower bank of External SRAM
nWRH	46		Output	Active LOW, Write to upper bank of External SRAM
nRD	47		Output	Active LOW, Read from External SRAM or ROM
nRESET	48		Input	Master Reset. SL11R Device active LOW reset input.
nRAS	49		Output	Active low, DRAM Row Address Select
VDD	50		Power	+3.3 VDC Supply
VDD	51		Power	+3.3 VDC Supply
nCASL	52		Output	Active LOW, DRAM Column Low Address Select
nCASH	53		Output	Active LOW, DRAM Column High Address Select
nDRAMOE	54		Output	Active LOW, DRAM Output Enable
nDRAMWR	55		Output	Active LOW, DRAM Write
nXRAMSEL	56		Output	Active LOW, select external SRAM (16 bit)
nXROMSEL	57		Output	Active LOW, select external ROM
nXMEMSEL	58		Output	Active LOW, select external Memory bus, external SRAM, DRAM, ROM or any memory mapped device
X_PCLK	59		Bidir	See register 0xC006 for more information
SECLK	60	GPIO31	Bidir	SECLK, Serial EEPROM clock, or GPIO31
SEDO	61	GPIO30	Bidir	SEDO, Serial flash EPROM Data, or GPIO30 This pin requires a 5-kΩ pull-up.
USB_PU	62	GPIO29	Bidir	Turn on/off D+ Pull Up Resistor, or GPIO29
UART_TXD	63	GPIO28	Output	UART Transmit Data (out), or GPIO28

GPIO and 8/16-Bit DMA Modes—Pin Assignment and Description (continued)

Pin Name	Pin No.	GPIO Pins	Pin Type	GPIO & 8/16-bit DMA modes Pin Chip Revision 1.1
GND	64		GND	Digital ground.
GND	65		GND	Digital ground.
UART_RXD	66	GPIO27	Input	UART Receive Data (in), or GPIO27
PWR_OFF	67	GPIO26	Bidir	This signal can be used for device low-power mode, it will turn off or disable external powers to the peripheral in suspend mode. Once USB power is resumed, external power can be enabled again
IRQ1 (in)	68	GPIO25	Bidir	GPIO25, or IRQ1 (in) interrupts the SL11R processor
PWM3, or IRQ0 (in)	69	GPIO24	Bidir	IRQ0 (in) interrupts the SL11R processor or PWM. See the PWM Control register set-up for more information
PWM2	70	GPIO23	Bidir	Same as above or GPIO23
PWM1	71	GPIO22	Bidir	Same as above or GPIO22
PWM0	72	GPIO21	Bidir	Same as above or GPIO21
DREQ	73	GPIO20	Bidir	DMA Request Enable. DREQ indicates that SL11R is ready to accept or send data from/to an external device. DREQ along with nCS, nWRITE and nREAD bits are the DMA handshake signals for the main SDATA port, or GPIO20.
ADDR	74	GPIO19	Bidir	ADDR =1, Read/Write data from the INBUF/OUTBUFF, ADDR=0 read data from the STATUS register, or GPIO19
VDD	75		Power	+3.3 VDC Supply
nCS	76	GPIO18	Bidir	CS (in) Active LOW, Selects the bidirectional SDATA Port, or GPIO18
nWRITE	77	GPIO17	Bidir	Active input LOW signal used to indicate write data transfers to the general bidirectional SD15-0 Data Port. Signal is driven high for read transfers to the SL11R, or GPIO17
nREAD	78	GPIO16	Bidir	Active LOW input signal used to indicate read data transfers from the general bidirectional SD15-0 Data Port. Or GPIO16
GND	79		GND	Digital ground.
SD15	80	GPIO15	Bidir	Main bidirectional SDATA port bit 15, or GPIO15
SD14	81	GPIO14	Bidir	Main bidirectional SDATA port bit 14, or GPIO14
SD13	82	GPIO13	Bidir	Main bidirectional SDATA port bit 13, or GPIO13
SD12	83	GPIO12	Bidir	Main bidirectional SDATA port bit 12, or GPIO12
SD11	84	GPIO11	Bidir	Main bidirectional SDATA port bit 11, or GPIO11
SD10	85	GPIO10	Bidir	Main bidirectional SDATA port bit 10, or GPIO10
SD9	86	GPIO9	Bidir	Main bidirectional SDATA port bit 9, or GPIO9
VDD1	87		Power	USB +3.3 VDC Supply.
DATA+	88		Bidir	USB Differential DATA Signal High Side.
DATA-	89		Bidir	USB Differential DATA Signal Low Side.
GND1	90		GND	USB Digital Ground.
SD8	91	GPIO8	Bidir	Main bidirectional SDATA port bit 8, or GPIO8
SD7	92	GPIO7	Bidir	Main bidirectional SDATA port bit 7, or GPIO7
SD6	93	GPIO6	Bidir	Main bidirectional SDATA port bit 6, or GPIO6
SD5	94	GPIO5	Bidir	Main bidirectional SDATA port bit 5, or GPIO5
SD4	95	GPIO4	Bidir	Main bidirectional SDATA port bit 4, or GPIO4
SD3	96	GPIO3	Bidir	Main bidirectional SDATA port bit 3, or GPIO3
SD2	97	GPIO2	Bidir	Main bidirectional SDATA port bit 2, or GPIO2
SD1	98	GPIO1	Bidir	Main bidirectional SDATA port bit 1, or GPIO1

GPIO and 8/16-Bit DMA Modes—Pin Assignment and Description (continued)

Pin Name	Pin No.	GPIO Pins	Pin Type	GPIO & 8/16-bit DMA modes Pin Chip Revision 1.1
SD0	99	GPIO0	Bidir	Main bidirectional SDATA port bit 0, or GPIO0
VDD	100		Power	+3.3 VDC Supply

6.3 Fast EPP Pin Assignment and Description

Fast EPP Pin Assignment and Description

Pin Name	Pin No.	GPIO pins	Pin Type	GPIO & Fast EPP Pin Chip Revision 1.1
VDD	1		Power	+3.3 VDC Supply
D0	2		Bidir	External Memory Data Bus, Data0
D1	3		Bidir	External Memory Data Bus, Data1
D2	4		Bidir	External Memory Data Bus, Data2
D3	5		Bidir	External Memory Data Bus, Data3
D4	6		Bidir	External Memory Data Bus, Data4
D5	7		Bidir	External Memory Data Bus, Data5
D6	8		Bidir	External Memory Data Bus, Data6
D7	9		Bidir	External Memory Data Bus, Data7
D8	10		Bidir	External Memory Data Bus, Data8
D9	11		Bidir	External Memory Data Bus, Data9
D10	12		Bidir	External Memory Data Bus, Data10
D11	13		Bidir	External Memory Data Bus, Data11
GND	14		GND	Digital ground.
X1	15		Input	External 48-MHz Crystal or Clock Input.
X2	16		Output	External crystal output. No connection when X1 is used for clock input
VDD	17		Power	+3.3 VDC Supply
D12	18		Bidir	External Memory Data Bus, Data12
D13	19		Bidir	External Memory Data Bus, Data13
D14	20		Bidir	External Memory Data Bus, Data14
D15	21		Bidir	External Memory Data Bus, Data15
A20	22		Output	External Memory Address Bus, A20
A19	23		Output	External Memory Address Bus, A19
A18	24		Output	External Memory Address Bus, A18
A17	25		Output	External Memory Address Bus, A17
A16	26		Output	External Memory Address Bus, A16
A15	27		Output	External Memory Address Bus, A15
A14	28		Output	External Memory Address Bus, A14
A13	29		Output	External Memory Address Bus, A13
A12	30		Output	External Memory Address Bus, A12
A11	31		Output	External Memory Address Bus, A11
A10	32		Output	External Memory Address Bus, A10
A9	33		Output	External Memory Address Bus, A9
A8	34		Output	External Memory Address Bus, A8
A7	35		Output	External Memory Address Bus, A7

Fast EPP Pin Assignment and Description (continued)

Pin Name	Pin No.	GPIO pins	Pin Type	GPIO & Fast EPP Pin Chip Revision 1.1
A6	36		Output	External Memory Address Bus, A6
A5	37		Output	External Memory Address Bus, A5
A4	38		Output	External Memory Address Bus, A4
A3	39		Output	External Memory Address Bus, A3
GND	40		GND	Digital ground.
A2	41		Output	External Memory Address Bus, A2
A1	42		Output	External Memory Address Bus, A1
A0	43		Output	External Memory Address Bus, A0
TEST	44		Input	No Connection, MFG test only
nWRL	45		Output	Active LOW, write to lower bank of External SRAM
nWRH	46		Output	Active LOW, Write to upper bank of External SRAM
nRD	47		Output	Active LOW, Read from External SRAM or ROM
nRESET	48		Input	Master Reset. SL11R Device active low reset input.
nRAS	49		Output	Active low, DRAM Row Address Select
VDD	50		Power	+3.3 VDC Supply
VDD	51		Power	+3.3 VDC Supply
nCASL	52		Output	Active LOW, DRAM Column Low Address Select
nCASH	53		Output	Active LOW, DRAM Column High Address Select
nDRAMOE	54		Output	Active LOW, DRAM Output Enable
nDRAMWR	55		Output	Active LOW, DRAM Write
nXRAMSEL	56		Output	Active LOW, select external SRAM (16 bit)
nXROMSEL	57		Output	Active LOW, select external ROM
nXMEMSEL	58		Output	Active LOW, select external Memory bus, external SRAM, DRAM, ROM or any memory mapped device
X_PCLK	59		Bidir	See register 0xC006 for more information
SECLK	60	GPIO31	Bidir	SECLK, Serial EEPROM clock, or GPIO31
SEDO	61	GPIO30	Bidir	SEDO, Serial flash EPROM Data, or GPIO30 This pin requires a 5K Ohm pull-up.
GPIO29	62	GPIO29	Bidir	GPIO29
UART_TXD	63	GPIO28	Output	UART Transmit Data (out), or GPIO28
GND	64		GND	Digital ground.
GND	65		GND	Digital ground.
UART_RXD	66	GPIO27	Input	UART Receive Data (in), or GPIO27
nENS	67	GPIO26	Output	Serial EPROM control signal
CLKS	68	GPIO25	Output	Serial EPROM Clock
nDTSRB	69	GPIO24	Output	EPP Data Strobe
nASTRB	70	GPIO23	Output	EPP Address Strobe
nWRITE	71	GPIO22	Output	EPP Write Strobe
P9	72	GPIO21	Output	P Register
P5	73	GPIO20	Output	P Register or PWR_OFF
P4	74	GPIO19	Output	P Register

Fast EPP Pin Assignment and Description (continued)

Pin Name	Pin No.	GPIO pins	Pin Type	GPIO & Fast EPP Pin Chip Revision 1.1
VDD	75		Power	+3.3 VDC Supply
P3	76	GPIO18	Output	P Register or USB_PU (USB DATA+ pull up)
P2	77	GPIO17	Output	P Register
P1	78	GPIO16	Output	P Register
GND	79		GND	Digital ground.
P6	80	GPIO15	Output	P Register
P7	81	GPIO14	Output	P Register
P8	82	GPIO13	Output	P Register
GPIO12	83	GPIO12	Bidir	GPIO12
DATAS	84	GPIO11	Bidir	DATA Strobe for Serial EPROM
VREQ	85	GPIO10	Input	TBD
WAIT	86	GPIO9	Bidir	EPP WAIT signal
VDD1	87		Power	USB +3.3 VDC Supply.
DATA+	88		Bidir	USB Differential DATA Signal High Side.
DATA-	89		Bidir	USB Differential DATA Signal Low Side.
GND1	90		GND	USB Digital Ground.
INTR	91	GPIO8	Input	EPP INTR pin
SD7	92	GPIO7	Bidir	EPP Data bit 7
SD6	93	GPIO6	Bidir	EPP Data bit 6
SD5	94	GPIO5	Bidir	EPP Data bit 5
SD4	95	GPIO4	Bidir	EPP Data bit 4
SD3	96	GPIO3	Bidir	EPP Data bit 3
SD2	97	GPIO2	Bidir	EPP Data bit 2
SD1	98	GPIO1	Bidir	EPP Data bit 1
SD0	99	GPIO0	Bidir	EPP Data bit 0
VDD	100		Power	+3.3 VDC Supply

6.4 DVC 8-Bit DMA Mode Pin Assignment and Description

DVC 8-Bit DMA Mode Pin Assignment and Description

Pin Name	Pin No.	GPIO pins	Pin Type	GPIO & DVC 8-bit DMA modes Pin Chip Revision 1.1
VDD	1		Power	+3.3 VDC Supply
D0	2		Bidir	External Memory Data Bus, Data0
D1	3		Bidir	External Memory Data Bus, Data1
D2	4		Bidir	External Memory Data Bus, Data2
D3	5		Bidir	External Memory Data Bus, Data3
D4	6		Bidir	External Memory Data Bus, Data4
D5	7		Bidir	External Memory Data Bus, Data5
D6	8		Bidir	External Memory Data Bus, Data6
D7	9		Bidir	External Memory Data Bus, Data7
D8	10		Bidir	External Memory Data Bus, Data8

DVC 8-Bit DMA Mode Pin Assignment and Description (continued)

Pin Name	Pin No.	GPIO pins	Pin Type	GPIO & DVC 8-bit DMA modes Pin Chip Revision 1.1
D9	11		Bidir	External Memory Data Bus, Data9
D10	12		Bidir	External Memory Data Bus, Data10
D11	13		Bidir	External Memory Data Bus, Data11
GND	14		GND	Digital ground.
X1	15		Input	External 48-MHz Crystal or Clock Input.
X2	16		Output	External crystal output. No connection when X1 is used for clock input
VDD	17		Power	+3.3 VDC Supply
D12	18		Bidir	External Memory Data Bus, Data12
D13	19		Bidir	External Memory Data Bus, Data13
D14	20		Bidir	External Memory Data Bus, Data14
D15	21		Bidir	External Memory Data Bus, Data15
A20	22		Output	External Memory Address Bus, A20
A19	23		Output	External Memory Address Bus, A19
A18	24		Output	External Memory Address Bus, A18
A17	25		Output	External Memory Address Bus, A17
A16	26		Output	External Memory Address Bus, A16
A15	27		Output	External Memory Address Bus, A15
A14	28		Output	External Memory Address Bus, A14
A13	29		Output	External Memory Address Bus, A13
A12	30		Output	External Memory Address Bus, A12
A11	31		Output	External Memory Address Bus, A11
A10	32		Output	External Memory Address Bus, A10
A9	33		Output	External Memory Address Bus, A9
A8	34		Output	External Memory Address Bus, A8
A7	35		Output	External Memory Address Bus, A7
A6	36		Output	External Memory Address Bus, A6
A5	37		Output	External Memory Address Bus, A5
A4	38		Output	External Memory Address Bus, A4
A3	39		Output	External Memory Address Bus, A3
GND	40		GND	Digital ground.
A2	41		Output	External Memory Address Bus, A2
A1	42		Output	External Memory Address Bus, A1
A0	43		Output	External Memory Address Bus, A0
TEST	44		Input	No Connection, MFG test only
nWRL	45		Output	Active LOW, write to lower bank of External SRAM
nWRH	46		Output	Active LOW, Write to upper bank of External SRAM
nRD	47		Output	Active LOW, Read from External SRAM or ROM
nRESET	48		Input	Master Reset. SL11R Device active low reset input.
nRAS	49		Output	Active LOW, DRAM Row Address Select
VDD	50		Power	+3.3 VDC Supply
VDD	51		Power	+3.3 VDC Supply

DVC 8-Bit DMA Mode Pin Assignment and Description (continued)

Pin Name	Pin No.	GPIO pins	Pin Type	GPIO & DVC 8-bit DMA modes Pin Chip Revision 1.1
nCASL	52		Output	Active LOW, DRAM Column Low Address Select
nCASH	53		Output	Active LOW, DRAM Column High Address Select
nDRAMOE	54		Output	Active LOW, DRAM Output Enable
nDRAMWR	55		Output	Active LOW, DRAM Write
nXRAMSEL	56		Output	Active LOW, select external SRAM (16 bit)
nXROMSEL	57		Output	Active LOW, select external ROM
nXMEMSEL	58		Output	Active LOW, select external Memory bus, external SRAM, DRAM, ROM or any memory mapped device
X_PCLK	59		Bidir	See register 0xC006 for more information
SECLK	60	GPIO31	Bidir	SECLK, Serial EEPROM clock, or GPIO31
SEDO	61	GPIO30	Bidir	SEDO, Serial flash EPROM Data, or GPIO30 This pin requires a 5K Ohm pull-up.
GPIO29	62	GPIO29	Bidir	GPIO29
UART_TXD	63	GPIO28	Output	UART Transmit Data (out), or GPIO28
GND	64		GND	Digital ground.
GND	65		GND	Digital ground.
UART_RXD	66	GPIO27	Input	UART Receive Data (in), or GPIO27
GPIO26	67	GPIO26	Bidir	GPIO26
IRQ1 (in)	68	GPIO25	Bidir	GPIO25, or IRQ1 (in) interrupts the SL11R processor
IRQ0 (in)	69	GPIO24	Bidir	IRQ0 (in) interrupts the SL11R processor or GPIO24
GPIO23	70	GPIO23	Bidir	GPIO23
GPIO22	71	GPIO22	Bidir	GPIO22
nVID_RST	72	GPIO21	Output	VIDEO Reset Pin
P_CONT	73	GPIO20	Output	P_CONT pin
AGC_C	74	GPIO19	Output	AGC_C control pin
VDD	75		Power	+3.3 VDC Supply
AEEP	76	GPIO18	Output	AEPP pin
DSP_CS	77	GPIO17	Output	DSP Chip select pin
EEP2_CS	78	GPIO16	Bidir	Serial EPROM2 chip select
GND	79		GND	Digital ground.
EEP1_CS	80	GPIO15	Output	Serial EPROM1 chip select
DO	81	GPIO14	Output	Serial EPROM Data out pin
SK	82	GPIO13	Input	Serial EPROM Clock pin
PBLK	83	GPIO12	Input	PBLK from CCD
VD	84	GPIO11	Input	VD from CCD
FI	85	GPIO10	Input	FI from CCD
MCK0	86	GPIO9	Bidir	MCK0 from CCD
VDD1	87		Power	USB +3.3 VDC Supply.
DATA+	88		Bidir	USB Differential DATA Signal High Side.
DATA-	89		Bidir	USB Differential DATA Signal Low Side.
GND1	90		GND	USB Digital Ground.

DVC 8-Bit DMA Mode Pin Assignment and Description (continued)

Pin Name	Pin No.	GPIO pins	Pin Type	GPIO & DVC 8-bit DMA modes Pin Chip Revision 1.1
DI	91	GPIO8	Bidir	Serial EPROM Data Input
SD7	92	GPIO7	Input	SDATA port bit 7, or GPIO7
SD6	93	GPIO6	Input	SDATA port bit 6, or GPIO6
SD5	94	GPIO5	Input	SDATA port bit 5, or GPIO5
SD4	95	GPIO4	Input	SDATA port bit 4, or GPIO4
SD3	96	GPIO3	Input	SDATA port bit 3, or GPIO3
SD2	97	GPIO2	Input	SDATA port bit 2, or GPIO2
SD1	98	GPIO1	Input	SDATA port bit 1, or GPIO1
SD0	99	GPIO0	Input	SDATA port bit 0, or GPIO0
VDD	100		Power	+3.3 VDC Supply

7.0 SL11R CPU Programming Guide

This is the preliminary specification for the SL11R Processor Instruction set.

7.1 Instruction Set Overview

This document describes the SL11R CPU Instruction Set, Registers and Addressing modes, Instruction format, etc. The SL11R PROCESSOR uses a unified program and data memory space; although this RAM is also integrated into the SL11R core, provision has been made for external memory as well.

The SL11R PROCESSOR engine incorporates 38 registers: fifteen general-purpose registers, a stack pointer, sixteen registers mapped into RAM, a program counter, and a REGBANK register whose function will be described in a subsequent section.

The SL11R PROCESSOR engine supports byte and word addressing. Subsequent sections of this document will describe:

- The SL11R PROCESSOR Engine (QT Engine) Register Set
- SL11R PROCESSOR Engine Instruction Format
- SL11R PROCESSOR Engine Addressing Modes
- SL11R PROCESSOR Engine Instruction Set

7.2 Reset Vector

On receiving hardware reset, the SL11R Processor jumps to address 0xFFF0, which is an internal ROM address.

7.3 Register Set

The SL11R Processor incorporates 16-bit general-purpose registers called R0..R15, a REGBANK register, and a program counter, along with various other registers. The function of each register is defined as follows:

Name	Function
R0-R14	General Purpose Registers
R15	Stack Pointer
PC	Program Counter
REGBANK	Forms base address for registers R0-R15
FLAGS	Contains flags: defined below
INTERRUPT ENABLE	Bit masks to enable/disable various interrupts

7.4 General-Purpose Registers

The general-purpose registers can be used to store intermediate results, and to pass parameters to and return them from subroutine calls.

7.5 General Purpose/Address Registers

In addition to acting as general-purpose registers, registers R8-R14 can also serve as pointer registers. Instructions can access RAM locations by referring to any of these registers. In normal operation, register R15 is reserved for use as a stack pointer.

7.6 REGBANK Register (0xC002: R/W)

Registers R0..R15 are mapped into RAM via the REGBANK register. The REGBANK register is loaded with a base address, of which the 11 most significant bits are used. A read from or write to one of the registers will generate a RAM address by:

- Shifting the 4 least significant bits of the register number left by 1.
- OR-ing the shifted bits of the register number with the upper 11 bits of the REGBANK register.
- Forcing the Least Significant Bit to 0.

For example, if the REGBANK register is left at its default value of 100 hex, a read of register R14 would read address 11C hex.

Register	Hex Value	Binary Value
REGBANK	0100	0 0 0 0 0 0 0 1 0 0 0 x x x x x
R14	000E << 1 = 001C	x x x x x x x x x x 0 1 1 1 0 0
RAM Location	011C	0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0

Note:

Regardless of the value loaded into the REGBANK register, bits 0..4 will be ignored.

7.7 Flags Register (0xC000: Read Only)

The SL11R Processor uses these flags:

FLAG																
bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	I	S	O	C	Z

Z **Zero**: instruction execution resulted in a result of 0

C **Carry/Borrow**: Arithmetic instruction resulted in a carry (for addition) or a borrow (for subtraction)

O **Overflow**: Arithmetic result was either larger than the destination operand size (for addition) or smaller than the destination operand should allow for subtraction

S **Sign**: Set if MS result bit is “1”

I **Global Interrupts** Enabled if “1”

Note:

Flag behavior for each instruction will be described in the following section

7.8 Instruction Format

To understand addressing modes supported by the SL11R Processor, you must know how the instruction format is defined. In general, the instructions include four bits for the instruction ***opcode***, six bits for the source operand, and six bits for the destination operand.

ADD																
bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	opcode				source						destination					

Some instructions, especially single operand-operator and program control instructions, will not adhere strictly to this format. They will be discussed in detail in turn.

7.9 Addressing Modes

This section describes in detail the six-operand field bits referred to in the previous section as ***source*** and ***destination***. Bear in mind that although the discussion refers to bits 0 through 5, the same bit definitions apply to the “source” operand field, bits 6 through 11. These are the basic addressing modes in the SL11R Processor.

Mode	5	4	3	2	1	0
Register	0	0	r	r	r	r
Immediate	0	1	1	1	1	1
Direct	1	0	b/w	1	1	1
Indirect	0	1	b/w	r	r	r
Indirect with Auto Increment	1	0	b/w	r	r	r
Indirect with Index	1	1	b/w	r	r	r

Notes:

- b/w: ‘1’ for byte-wide access, ‘0’ for word access.
- Indirect with auto increment and byte-wide Indirect addressing is illegal with the stack pointer (R15).

7.10 Register Addressing

In register addressing, any one of registers R0-R15 can be selected using bits 0-3. If register addressing is used, operands are always 16-bit operands, since all registers are 16-bit registers. For example, an instruction using register R7 as an operand would fill the operand field like this:

Bits	5	4	3	2	1	0
Register Operand	0	0	0	1	1	1

7.11 Immediate Addressing

In Immediate Addressing, the instruction word is immediately followed by the source operand. For example, The operand field would be filled with:

Bits	5	4	3	2	1	0
Operand field	0	1	1	1	1	1

Note:

In immediate addressing, the source operand *must* be 16 bits wide, eliminating the need for a b/w bit.

7.12 Direct Addressing

In Direct Addressing, the word following the instruction word is used as an address into RAM. Again, the operand can be either byte or word sized, depending on the state of bit 3 of the operand field. For example, to do a word-wide read from a direct address, the ***source*** operand field would be formed like this:

Bits	5	4	3	2	1	0
I/O operand	1	0	0	1	1	1

Note:

For a memory-to-memory move, the instruction word would be followed by two words, the first being the ***source*** address and the second being the ***destination***.

7.13 Indirect Addressing

Indirect addressing is accomplished using address registers R8-15. In Indirect addressing, the operand is found at the memory address pointed to by the register. Since only eight address registers exist, only three bits are required to select an address register. For example, register R10 (binary 1010) can be selected by ignoring bit 3, leaving the bits 010. Bit 3 of the operand field

is then used as the byte/word bit, set to “0” to select word or “1” to select byte addressing. In this example, a byte-wide operand is selected at the memory location pointed to by register R10:

Bits	5	4	3	2	1	0
Memory operand	0	1	1	0	1	0

Note:

For register R15, byte-wide operands are prohibited. If bit 3 is set high, the instruction is decoded differently, as explained at the top of this section.

7.14 Indirect Addressing with Auto Increment

Indirect Addressing with Auto Increment works identically to Indirect Addressing, except that at the end of the read or write cycle, the register is incremented by 1 or 2 (depending whether it is a byte-wide or word-wide access.)

This mode is prohibited for register R15. If bits 0..2 are all high, the instruction is decoded differently, as explained at the top of this section.

7.15 Indirect Addressing with Offset

In Indirect Addressing with Offset, the instruction word is followed by a 16-bit word that is added to the contents of the address register to form the address for the operand. The offset is an unsigned 16-bit word, and will “wrap” to low memory addresses if the register and offset add up to a value greater than the size of the processor’s address space.

7.16 Stack Pointer (R15) Special Handling

Register R15 is designated as the Stack Pointer, and has these special behaviors:

- If addressed in indirect mode, the register pre-decrements on a write instruction, and post-increments on a read instruction, emulating Push and Pop instructions.
- Byte-wide reads or writes are prohibited in indirect mode.
- If R15 is addressed in Indirect with Index mode, it does not auto-increment or auto-decrement.

SL11R - CPU Instruction Set

The instruction set can be roughly divided into three classes of instructions:

- **Dual Operand Instructions** (Instructions with two operands, a source and a destination)
- **Program Control Instructions** (Jump, Call and Return)
- **Single Operand Instructions** (Instructions with only one operand: a destination)

7.17 Dual Operand Instructions

Instructions with source and destination, for ALL dual operand instructions, byte values are zero extended by default.

MOV																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0000				source						Destination					

destination:= source

Flags Affected: none

ADD																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0001				source						Destination					

destination:= destination + source

Flags Affected: Z, C, O, S

ADDC																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0010				source						Destination					

destination:= destination + source + carry bit

Flags Affected: Z, C, O, S

SUB																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0011				source						Destination					

destination:= destination - source

Flags Affected: Z, C, O, S

SUBB																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0100				source						Destination					

destination:= destination - source - carry bit

Flags Affected: Z, C, O, S

CMP																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0101				source						Destination					

[not saved] = destination - source

Flags Affected: Z, C, O, S

AND																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0110				source						Destination					

destination:= destination & source

Flags Affected: Z, S

TEST																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0111				source						Destination					

[not saved]:= destination & source

Flags Affected: Z, S

OR																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1000				source						Destination					

destination:= destination | source

Flags Affected: Z, S

XOR																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1001				source						Destination					

destination:= destination ^ source

Flags Affected: Z, S

7.18 Program Control Instructions

Jcc JUMP RELATIVE cccc																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1100				cccc				0	Offset						

PC:= PC + (offset*2)(offset is a 7-bit *signed* number from -64..+63)

JccL JUMP ABSOLUTE cccc																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1100				cccc				1	0	Destination					

PC:= [destination](destination is computed in the normal fashion for operand fields)

Rcc RET cccc																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1100				cccc				1	0	010111					

PC:= [R15]

R15++

Ccc CALL cccc																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1010				cccc				1	0	Destination					

R15--

[R15]:= PC

PC = [destination]

INT																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1010				0000				0	int vector						

[R15]:= PC

R15--

PC = [int vector * 2]

This instruction allows the programmer to implement software interrupts. *Int vector* is multiplied by two, and zero extended to 16 bits.

Note: Interrupt vectors 0 through 31 *may* be reserved for hardware interrupts, depending on the application.

The condition (cccc) bits for all of the above instructions are defined as:

Condition	cccc Bits	Description	JUMP mnemonic	CALL mnemonic	RET mnemonic
Z	0000	Z=1	JZ	CZ	RZ
NZ	0001	Z=0	JNZ	CNZ	RNZ
C / B	0010	C=1	JC	CC	RC
NC / AE	0011	C=0	JNC	RNC	RNC
S	0100	S=1	JS	CS	RS
NS	0101	S=0	JNS	CNS	RNS
O	0110	O=1	JO	CO	RO
NO	0111	O=0	JNO	CNO	RNO
A / NBE	1000	(Z=0 AND C=0)	JA	CA	RA
BE / NA	1001	(Z=1 OR C=1)	JBE	CBE	RBE
G / NLE	1010	(O= S AND Z=0)	JG	CG	RG
GE / NL	1011	(O=S)	JGE	CGE	RGE
L / NGE	1100	(O≠S)	JL	CL	RL
LE / NG	1101	(O≠S OR Z=1)	JLE	CLE	RLE
(not used)	1110				
Unconditional	1111	Unconditional	JMP	CALL	RET

Note: For the JUMP mnemonics, adding an “L” to the end indicates a long or absolute jump. Adding an “S” to the end indicates a short or relative jump. If nothing is added, the assembler will choose “S” or “L.”

7.19 Single Operand Operation Instructions

Since Single operand instructions do not require a source field, the format of the Single operand Operation instructions is slightly different.

Instruction																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101***							[param]			destination					

Notice that the ***opcode*** field is expanded to seven bits wide. The four most significant bits for all instructions of this class are “1101.”

In addition, there is space for an optional three bit immediate value, which is used in a manner appropriate to the instruction. The destination field functions exactly as it does in the dual operand operation instructions.

Note:

- For the SHR, SHL, ROR, ROL, ADDI and SUBI instructions, the three-bit *count* or *n* operand is incremented by 1 before it is used.
- The SL11R QT assembler takes this into account.

SHR																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101000							count-1			destination					

destination:= destination >> count

Flags Affected: Z, C, S

Note:

- The SHR instruction shifts in sign bits.
- The C flag is set with last bit shifted out of LSB.

SHL																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101001							count-1			destination					

destination:= destination << count

Flags Affected: Z, C, S

Note: The C flag is set with last bit shifted out of MSB.

ROR																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101010							count-1			destination					

Works identically to the SHR instruction, except that the LSB of *destination* is rotated into the MSB, as opposed to SHR, which discards that bit

Flags Affected: Z, C, S

ROL																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101011							count-1			destination					

Works identically to the SHL instruction, except that the MSB of *destination* is rotated into the LSB, as opposed to SHL, which discards that bit

Flags Affected: Z, C, S

ADDI																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101100							n-1			destination					

destination:= destination + n

Flags Affected: Z, S

SUBI																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101101							n-1			destination					

destination:= destination - n

Flags Affected: Z, S

NOT																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101111							000			destination					

destination:= ~destination (bitwise 1's complement negation)

Flags Affected: Z, S

NEG																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101111							001			destination					

destination:= -destination (2's complement negation)

Flags Affected: Z, O, C, S

CBW																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101111							010			destination					

Sign-extends a byte in the lower eight bits of [destination] to a 16-bit signed word (integer).

Flags Affected: Z, S

7.20 Miscellaneous Instructions

STI																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101111							111			000000					

Sets interrupt enable flag

Flags Affected: I

Note: The STI instruction takes effect 1 cycle after it is executed.

CLI																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101111							111			000001					

Clears interrupt enable flag

Flags Affected: I

STC																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101111							111			000010					

Set Carry bit.

Flags Affected: C

CLC																
bit:	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	1101111							111			000011					

Clear Carry bit.

Flags Affected: C

7.21 Built-in Macros

For the programmer's convenience, the SL11R QT assembler implements several built-in macros. The table below shows the macros, and the mnemonics for the code that the assembler will generate for these macros.

Macro	Assembler will Generate
INC X	ADDI X, 1
DEC X	SUBI X, 1
PUSH X	MOV [R15], X
POP X	MOV X, [R15]

7.22 SL11R Processor Instruction Set Summary

Mnemonic	Operands	Description	Opcode MSb LSb	Flags Affected	Clock Cycles	Notes
MOV	s,d	Move s to d	0000 ssss ssdd dddd	None	5	10,11
ADD	s,d	Add s to d	0001 ssss ssdd dddd	Z,C,O,S	5	10,11
ADDC	s,d	Add s to d with carry	0010 ssss ssdd dddd	Z,C,O,S	5	10,11
SUB	s,d	Subtract s from d	0011 ssss ssdd dddd	Z,C,O,S	5	10,11
SUBB	s,d	Subtract s from d with carry	0100 ssss ssdd dddd	Z,C,O,S	5	10,11
CMP	s,d	Compare d with s	0101 ssss ssdd dddd	Z,C,O,S	5	10,11
AND	s,d	AND d with s	0110 ssss ssdd dddd	Z,S	5	10,11
TEST	s,d	Bit test d with s	0111 ssss ssdd dddd	Z,S	5	10,11
OR	s,d	OR d with s	1000 ssss ssdd dddd	Z,S	5	10,11
XOR	s,d	XOR d with s	1001 ssss ssdd dddd	Z,S	5	10,11
Jcc	c,v	Jump relative on condition ’c’	1100 cccc 0ooo oooo	None	3	11
JccL	c,d	Jump absolute on condition ’c’	1100 cccc 10dd dddd	None	4	11
Rcc	c	Return on condition ’c’	1100 cccc 1001 0111	None	7	11
Ccc	c,d	Call subroutine on condition ’c’	1010 cccc 10dd dddd	None	7	11
Int	v	Software interrupt	1010 0000 0vvv vvvv	None	7	11
SHR	n,d	Shift right out of carry	1101 000n nndd dddd	Z,C,S	4	10,12,11
SHL	n,d	Shift left into carry	1101 001n nndd dddd	Z,C,S	4	10,12,11
ROR	n,d	Rotate right	1101 010n nndd dddd	Z,C,S	4	10,12,11
ROL	n,d	Rotate left	1101 011n nndd dddd	Z,C,S	4	10,12,11
ADDI	n,d	Add immediate	1101 100n nndd dddd	Z,S	4	11
SUBI	n,d	Subtract immediate	1101 101n nndd dddd	Z,S	4	11
NOT	d	1’s complement	1101 1110 00dd dddd	Z,S	4	11
NEG	d	2’s complement	1101 1110 01dd dddd	Z,O,C,S	4	11
CBW	d	Sign-extend d(7:0) to d(15:0)	1101 1110 10dd dddd	Z,S	4	11
STI		Enable interrupts	1101 1111 1100 0000	None	3	11
CLI		Disable interrupts	1101 1111 1100 0001	None	3	11
STC		Set carry	1101 1111 1100 0010	C	3	11
CLC		Clear carry	1101 1111 1100 0011	C	3	11

Notes:

10. The number in the “clock cycles” column reflects the number of clock cycles for register or immediate accesses. For each occurrence of other types of accesses, include the appropriate “clock adder” as listed in the Addressing Modes table below.
11. All clock cycle values assume zero wait-states.
12. A shift of one is done in four clock cycles, each additional shift adds two more clock cycles.

Opcode field descriptions									Clock Adder
Field	Description	Addressing mode	5	4	3	2	1	0	
S	Source	Register	0	0	r	r	r	r	0
D	Destination	Immediate	0	1	1	1	1	1	0
C	Condition code	Direct	1	0	b/w	1	1	1	1
O	Signed offset	Indirect	0	1	b/w	r	r	r	1
V	Interrupt vector	Indirect with Auto Increment	1	0	b/w	r	r	r	2
N	Count value -1	Indirect with Index	1	1	b/w	r	r	r	3

b/w: ’1’ = byte access, ’0’ = word access.

Indirect with auto-increment and byte-wide indirect addressing is illegal with R15.

8.0 SL11R - Electrical Specification

8.1 Absolute Maximum Ratings

This section lists the absolute maximum ratings of the SL11R. Stresses above those listed can cause permanent damage to the device. Exposure to maximum rated conditions for extended periods can affect device operation and reliability.

Parameter	Range
Storage temperature	–40°C to 125°C
Voltage on any pin with respect to ground	–0.3V to 7.3V
Power Supply Voltage (VDD)	3.3V±10%
Power Supply Voltage (VDD1)	3.3V±10%
Lead Temperature (10 seconds)	180°C
Junction Temperature (Tjmax)	125°C

8.2 Recommended Operating Conditions

Parameter	Min.	Typical	Max.
Power Supply Voltage, VDD	3.0V	3.3V	3.6V
Power Supply Voltage, VDD1	3.0V		3.6V
Operating Temperature	0°C		65°C

8.3 Crystal Requirements (XTAL1, XTAL2)

Crystal Requirements, (XTAL1, XTAL2)	Min.	Typical	Max.
Operating Temperature Range	0°C		65°C
Series Resonant Frequency		48 MHz	
Frequency Drift over Temperature			±20 PPM
Accuracy of Adjustment			±30 PPM
Series Resistance			100Ω
Shunt Capacitance	3 pF		6 pF
Load Capacitance		20 pF	
Driver Level	20 µW		5 mW
Mode of Vibration 3rd overtone			

8.4 External Clock Input Characteristics (XTAL1)

Parameter	Min.	Typical	Max.
Clock Input Voltage @ XTAL1 (XTAL2 is Opened)	1.5V		
Clock Frequency		48 MHz	

8.5 SL11R DC Characteristics

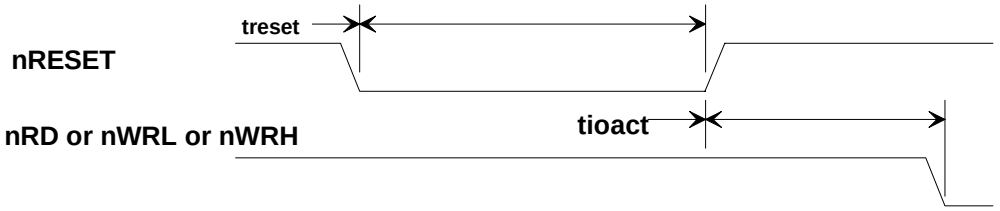
Parameter	Description	Min.	Typical	Max.
V_{IL}	Input Voltage LOW	–0.5V		0.8V
V_{IH}	Input Voltage HIGH	2.0V		V_{DD} + 0.3V
V_{OL}	Output Voltage LOW (I_{OL} = 4 mA)			0.4V
V_{OH}	Output Voltage HIGH (I_{OH} = –4 mA)	2.4V		
I_{OH}	Output Current HIGH	4 mA		
I_{OL}	Output Current LOW	4 mA		
C_{IN}	Input Capacitance			20 pF
I_{CC}	Supply Current (VDD)			< 30 mA
I_{USB}	Supply Current (VDD1)			< 10 mA
Pd	Power dissipation			0.7W
$I_{CC\ +USB\ Susp.}$	Suspend Supply Current			< 220 µA

8.6 SL11R USB Transceiver Characteristics

Parameter	Description	Min.	Typical[13]	Max.
V_{IHYS}	Hysteresis On Input (Data+, Data–)	0.1V		200 mV
V_{USBIH}	USB Input Voltage HIGH		1.5 V	2.0V
V_{USBIL}	USB Input Voltage LOW	0.8V	1.3 V	
V_{USBOH}	USB Output Voltage HIGH	2.2V		
V_{USBOL}	USB Output Voltage LOW			0.7V
Z_{USBH}[14]	Output Impedance HIGH STATE	28Ω		42Ω
Z_{USBL}[14]	Output Impedance LOW STATE	28Ω		42Ω

Notes:
13. All typical values are VDDx = 3.3 V and TAMB= 25°C.
14. Z_{USBX} Impedance Values includes an external resistor of 28-42Ω ± 1%

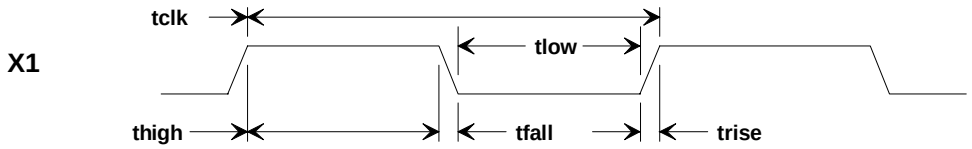
8.7 SL11R Reset Timing



Parameter	Description	Min.	Typical	Max.
treset	nRESET Pulse width	16 clocks		
tioact	nRESET high to nRD or nWRx active	16 clocks		

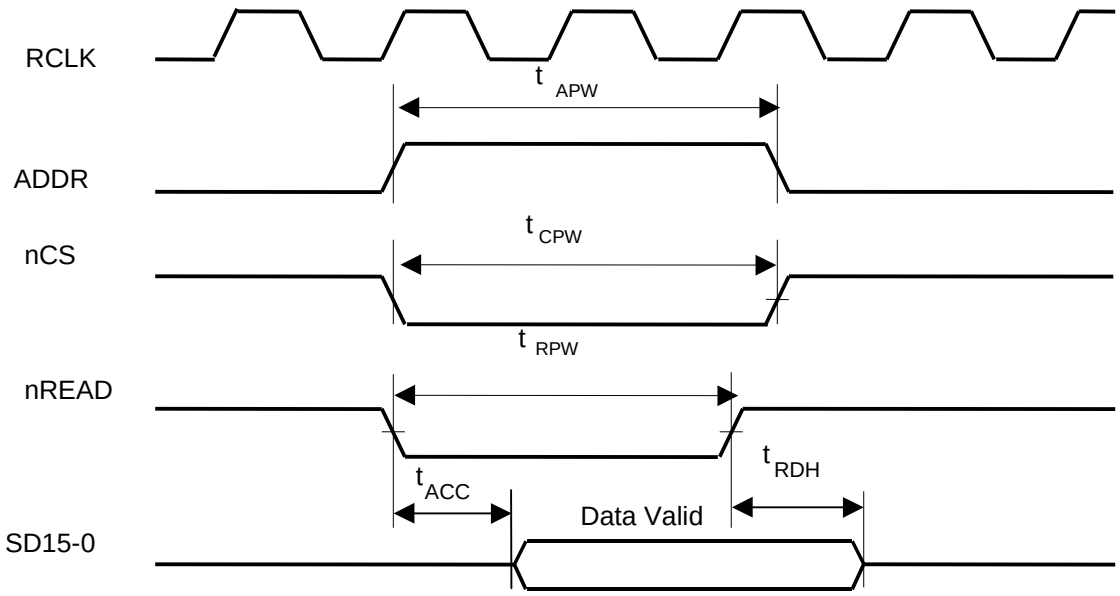
Note: Clock is 48 MHz nominal.

8.8 SL11R Clock Timing Specifications



Parameter	Description	Min.	Typical	Max.
tclk	Clock period (48 MHz)	20.0 ns	20.8 ns	
thigh	Clock high time	9 ns		11 ns
tlow	Clock low time	9 ns		11 ns
trise	Clock rise time			5.0 ns
tfall	Clock fall time			5.0 ns
	Duty Cycle	–5%		+5%

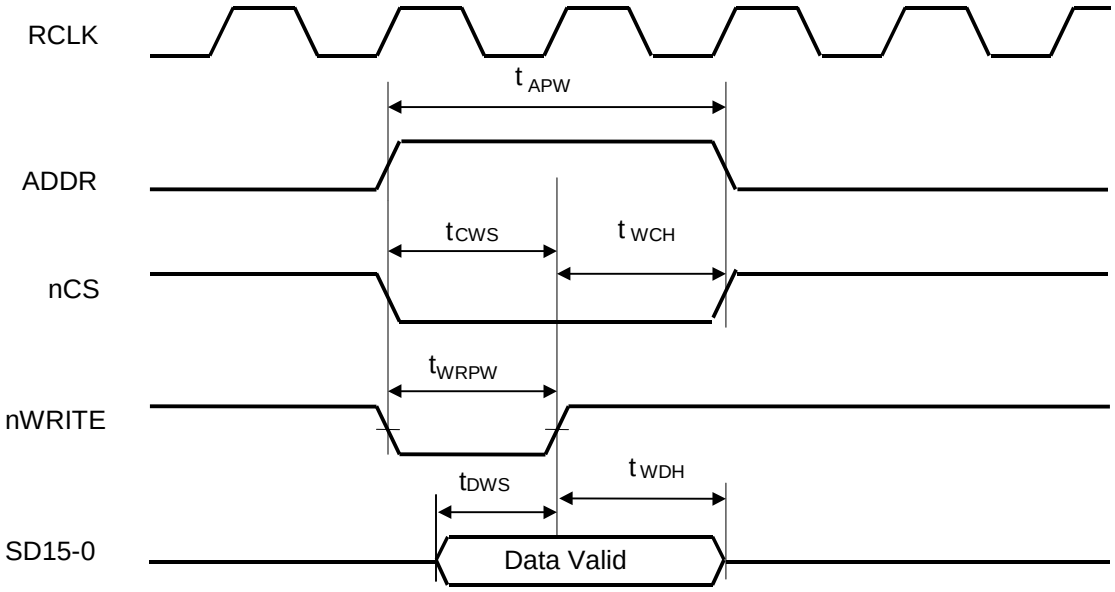
8.9 8/16-bit DMA & DVC 8-bit DMA Mode: SDATA Port I/O Read Cycle (Non-DMA)



Parameter	Description	Min.	Typical	Max.
t_{APW}	ADDR pulse width	30 ns		
t_{CPW}	nCS pulse width	30 ns		
t_{RPW}	Read pulse width	30 ns		
t_{ACC}	Read access time			25 ns
t_{RDH}	Read high to data hold			10 ns

Note: RCLK is the resulting Clock (see Register 0xC006)

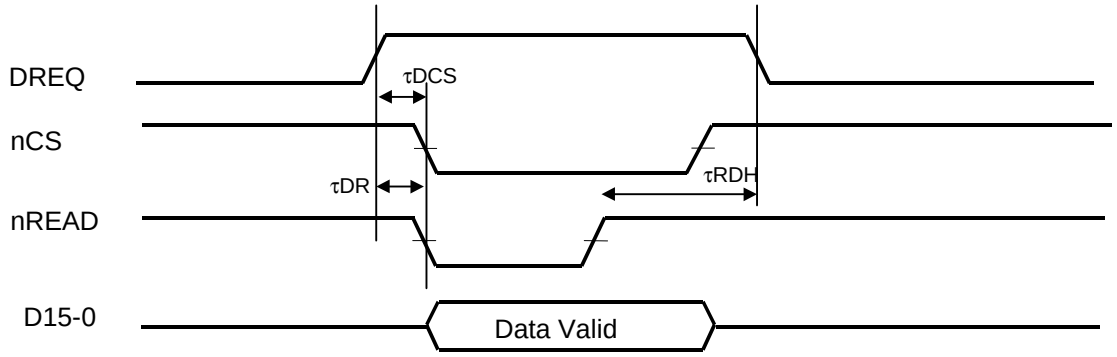
8.10 8/16-bit DMA & DVC 8-bit DMA Mode: SDATA Port I/O Write Cycle (Non-DMA)



Parameter	Description	Min.	Typical	Max.
t_{APW}	ADDR pulse width	20 ns		
t_{CWS}	nCS low to write high set-up	10 ns		
t_{WCH}	Write high to CS high hold	5 ns		
t_{WRPW}	Write pulse width	10 ns		
t_{DWS}	Data setup to write high set-up	10 ns		
t_{WDH}	Write high to data hold	5 ns		

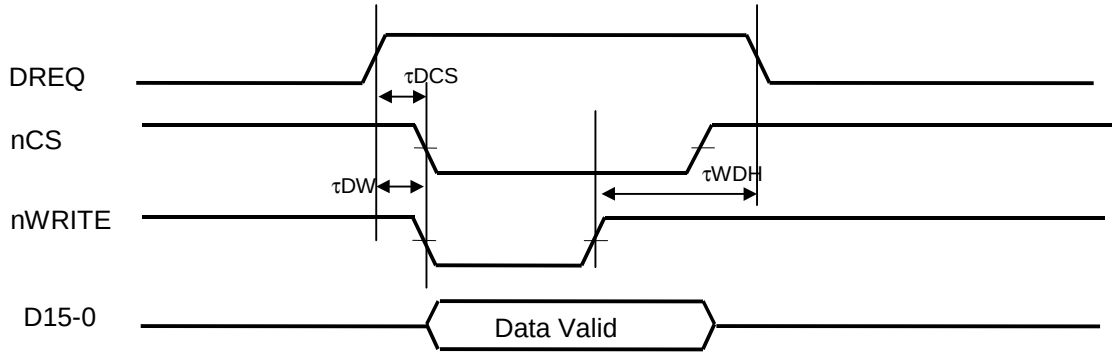
Note: RCLK is the resulting Clock (see Register 0xC006)

8.11 8/16-bit DMA & DVC 8-bit DMA Mode: SDATA, DMA Read Cycle



Parameter	Description	Min.	Typical	Max.
τDCS	DREQ high to CS low	5 ns		
τDR	DREQ high to read low	5 ns		
τRDH	Read high to DREQ low hold			30 ns

8.12 8/16-bit DMA & DVC 8-bit DMA Mode: SDATA, DMA Write Cycle



Parameter	Description	Min.	Typical	Max.
τDCS	DREQ high to CS low	5 ns		
τDWDH	Write high to DREQ low hold			30 ns
τDW	DREQ high to write low	5 ns		

8.13 SL11R Signals Name convention

Doc. Signal Name	SL11R Pin Name	Doc. Signal Name	SL11R Pin Name
/RAS	nRAS	/WE (DRAM)	nDRAMWR
/UCAS	nCASH	/LCAS	nCASL
Dout	Data15-0	/OE	nDRAMOE
Din	Data15-0	Address	A20-0
/CS	nXRAMSEL	/RD	nRD
/WE (SRAM)	nWRL & nWRH		

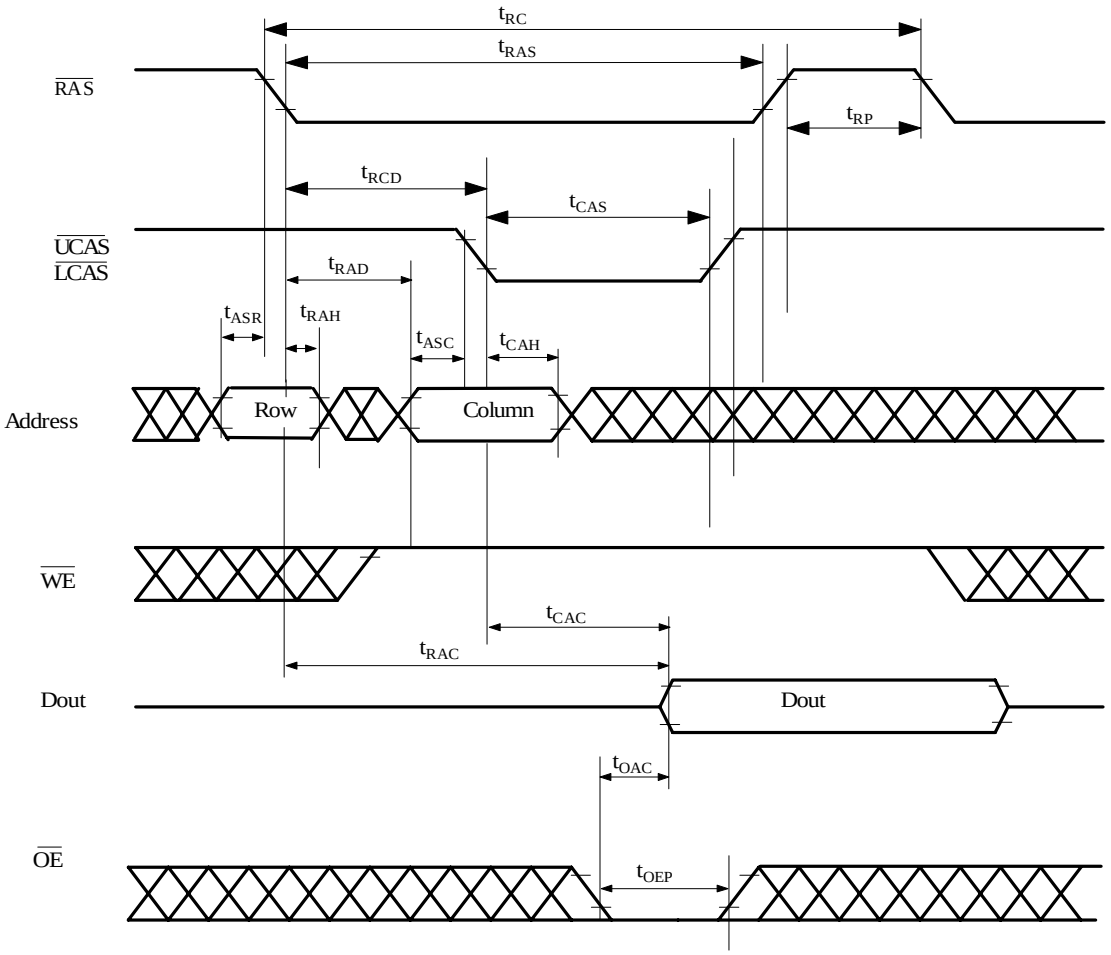
8.14 SL11R DRAM Timing

This timing is based on the SL11R Processor Clock (PCLK) = (2/3) of RCLK = 32MHz (see the register 0xC006 for information about PCLK).

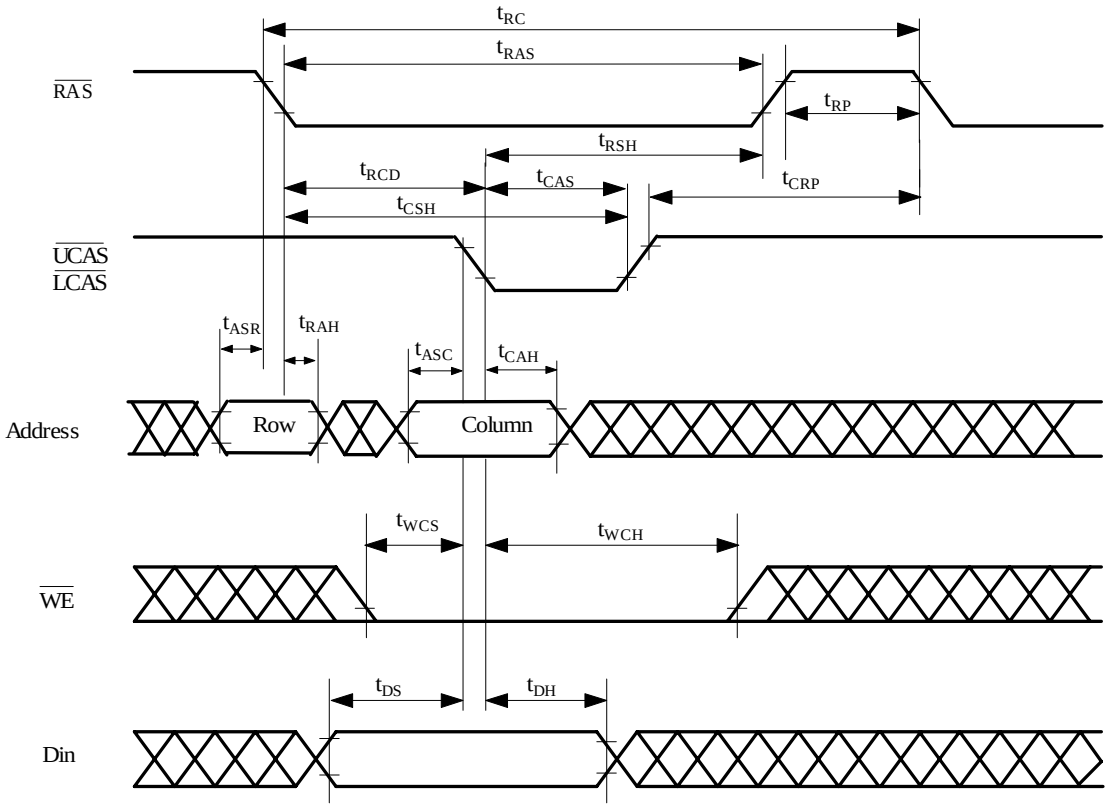
Parameter	Description	Min.	Typical	Max.
t_{RAS}	/RAS pulse width	80 ns		
t_{CAS}	/CAS pulse width	20 ns		
t_{RP}	/RAS precharge time	60 ns		
t_{RCD}	/RAS to /CAS delay time	64 ns		
t_{ASR}	Row address set-up time	20 ns		
t_{RAH}	Row address hold time	36 ns		
t_{ASC}	Column address set-up time	20 ns		
t_{CAH}	Column address hold time	36 ns		
t_{WCS}	Write command set-up time	25 ns		
t_{DS}	Data set-up time	05 ns		
t_{DH}	Data hold time	40 ns		
t_{CRP}	Delay time, /CAS pre-charge to /RAS	05 ns		
t_{T}	Transition time (rise and fall)	03 ns		
t_{RPC}	/RAS precharge to /CAS hold time	00 ns		
t_{CSR}	/CAS set-up time	05 ns		
t_{CPN}	/CAS precharge time	10 ns		
t_{CHR}	/CAS hold time	60 ns		
t_{CAC}	Access time from /CAS			20 ns
t_{RAC}	Access time from /RAS	80 ns		
t_{OAC}	Access time from /OE	20 ns		
t_{RC}	Cycle time read	150 ns		
t_{OFF}	Data out to High Z	05 ns		

Note: This timing is base on EDO DRAM timing 16Mx16 devices. When the SL11R processor is set up for a higher speed (i.e. 48MHz clock), then the faster parts (i.e. 50ns or 60ns) should be used.

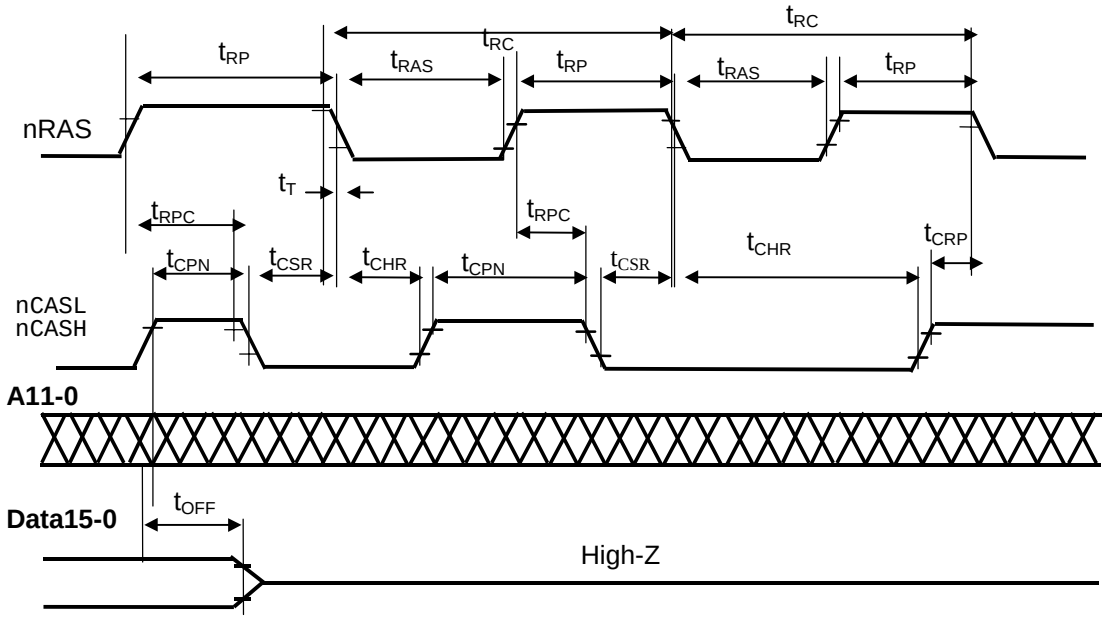
8.15 SL11R DRAM Read Cycle



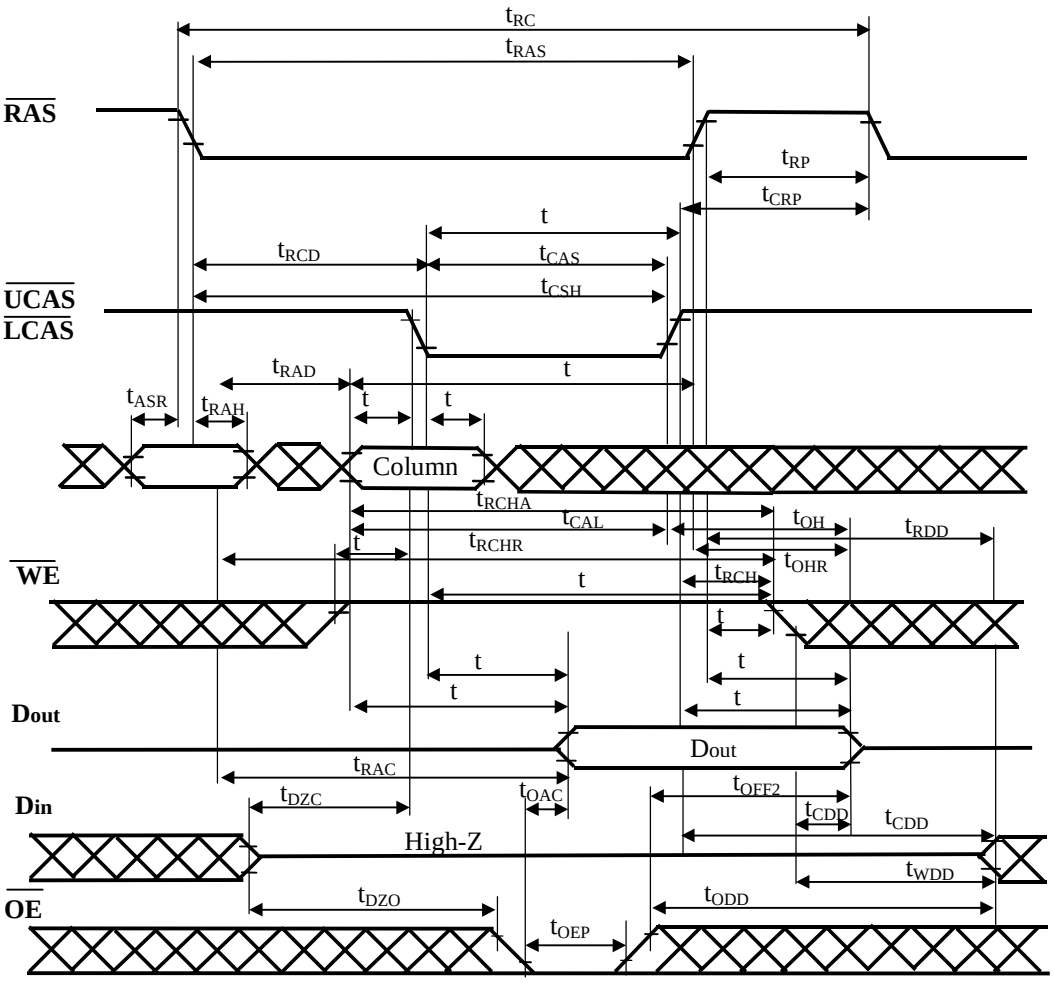
8.16 SL11R DRAM Write Cycle



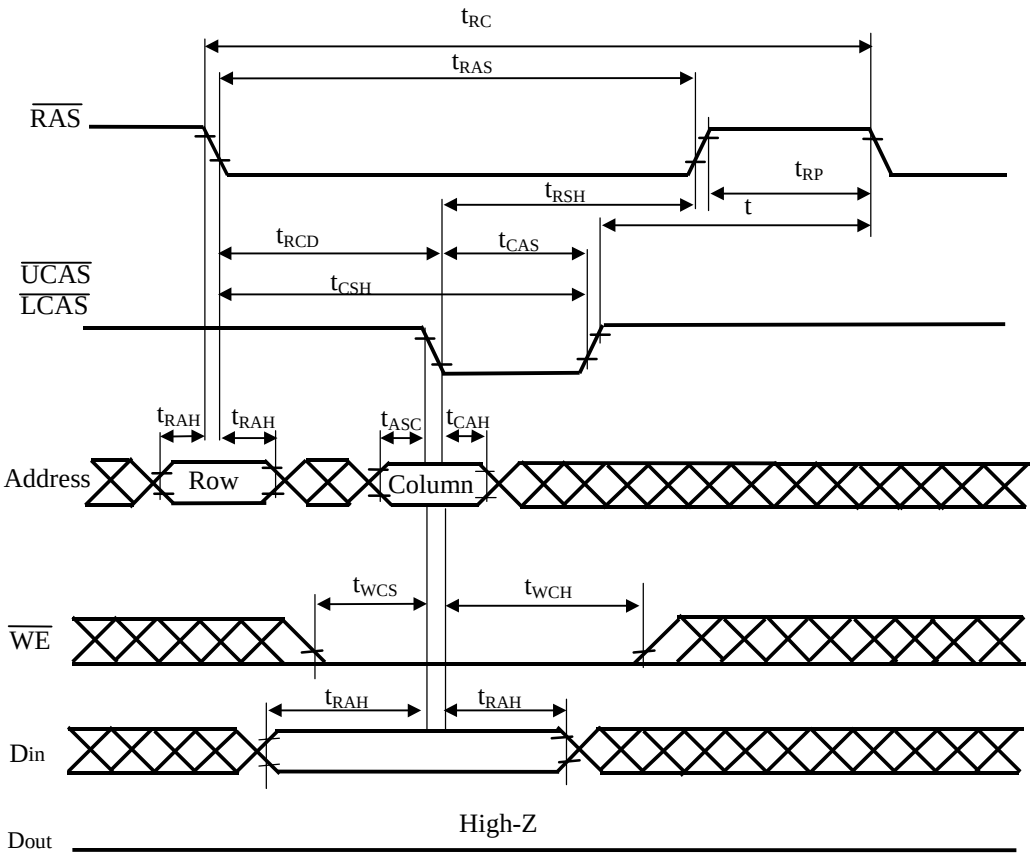
8.17 SL11R CAS-Before-RAS Refresh Cycle



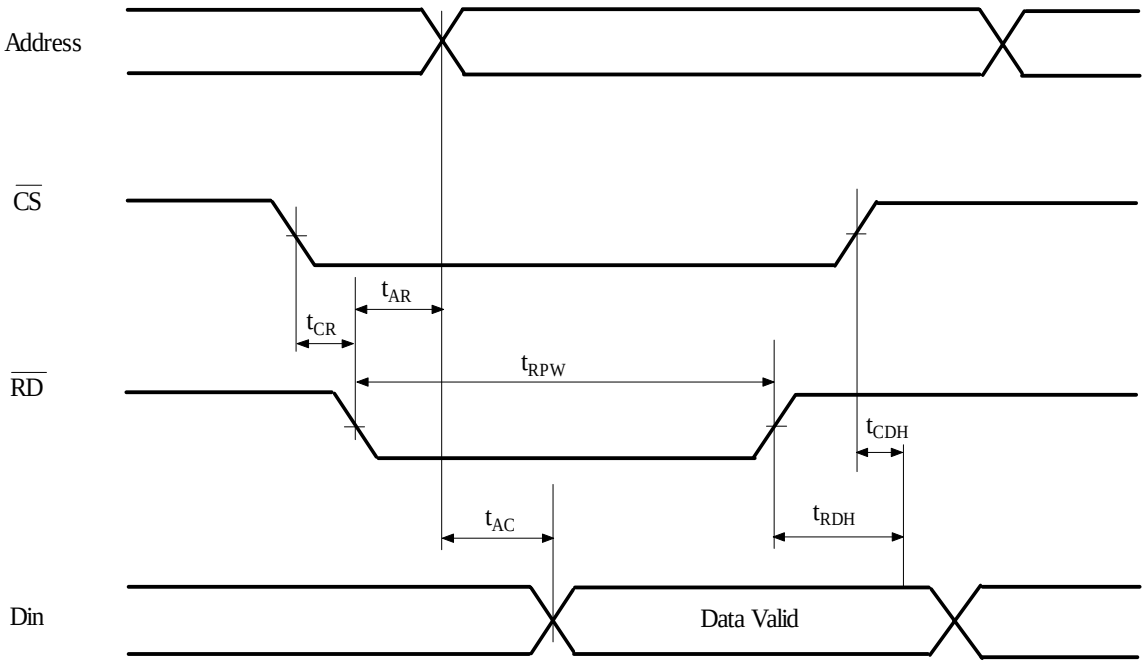
8.18 SL11R DRAM Page Mode Read Cycle



8.19 DRAM Page Mode Write Cycle



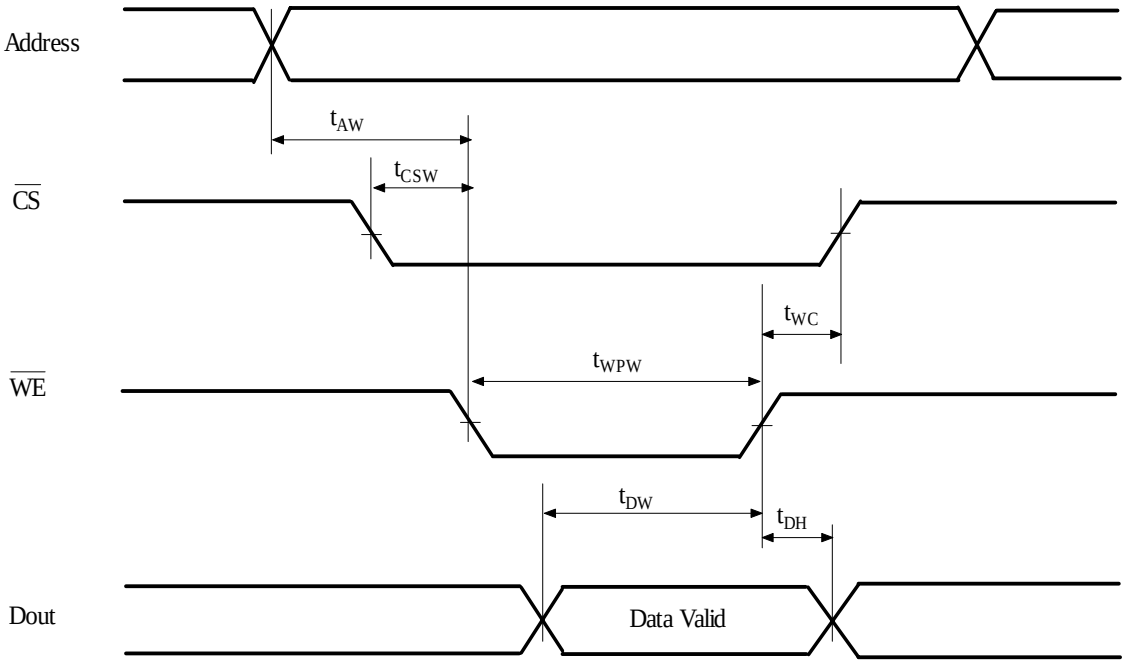
8.20 SL11R SRAM Read Cycle



Parameter	Description	Min.	Typical	Max.
t_{CR}	CS low to RD low	1 ns		
t_{RDH}	RD high to data hold	5 ns		
t_{CDH}	CS high to data hold	3 ns		
t_{RPW}[15]	RD low time	28 ns		31 ns
t_{AR}	RD low to address valid	1 ns		3 ns
t_{AC}[16]	RAM access to data valid			12 ns

Notes:
15. 0 wait state cycle.
16. t_{AC} means at 0 wait states, with PCLK = 2/3 RCLK, the SRAM access time should be 12ns max. For a 1 wait state cycle, with PCLK = 2/3 RCLK, the SRAM access time should be at 12 + 31ns = 43ns max. See register 0xC006 description for PCLK information.

8.21 SL11R SRAM Write Cycle

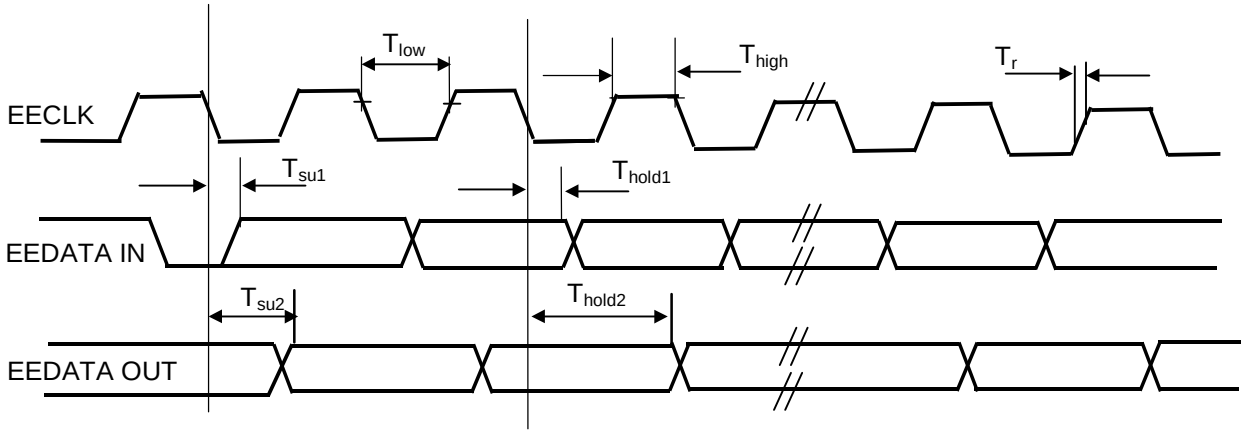


Parameter	Description	Min.	Typical	Max.
t_{AW}	Write address valid to WE low	13 ns		
t_{CSW}	CS low to WE low	13 ns		
t_{DW}	Data valid to WE high	25 ns		
t_{WPW}[17]	WE pulse width	28 ns		
t_{DH}	Data hold from WE high	5 ns		
t_{WC}	WE high to CS high	15 ns		

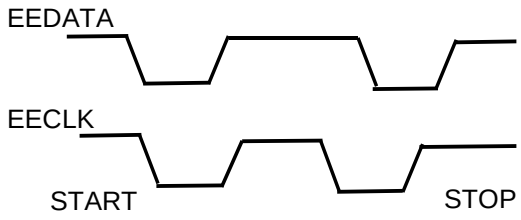
Note:
17. This is at 1 wait state with PCLK = 2/3 RCLK. For 2-wait states, add 31 ns.

8.22 2-Wire Serial Interface EEPROM Timing

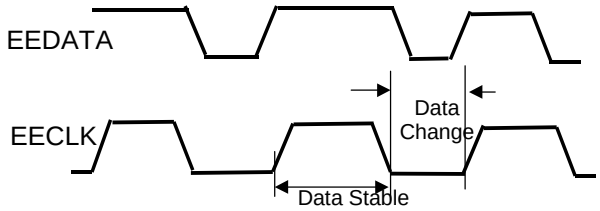
1-EEPROM Bus Timing- Serial I/O



2-Start and Stop Definition



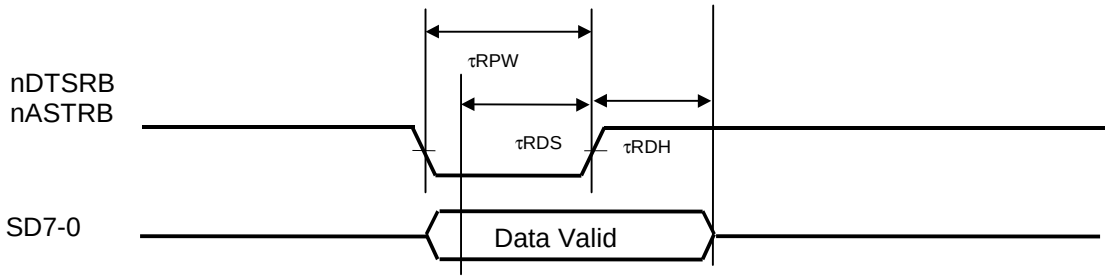
3- Data Validity



Note: Timing will conform to standard as illustrated in ATMEL AT24COX data sheet

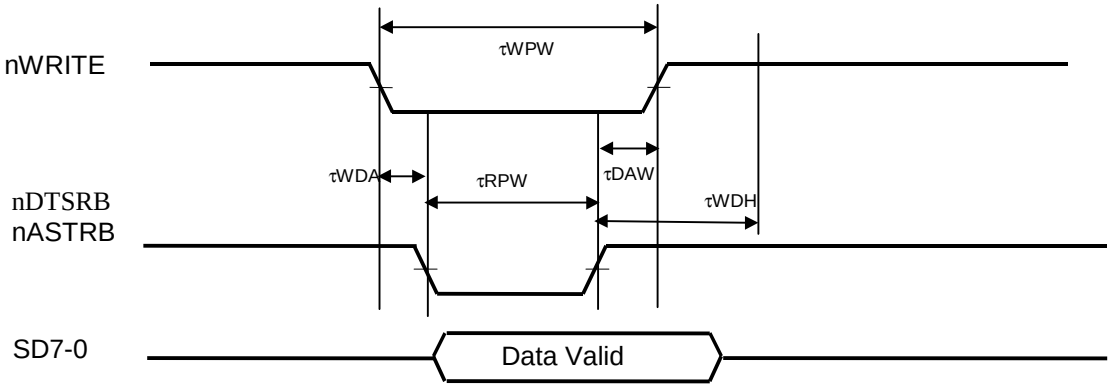
Parameter	Min./Max. Timing	Notes
T_{low}	4.7 µs min.	See ATMEL Data Sheet for
T_{high}	4.0 µs min.	Complete Timing Detail
T_r	1.0 µs max.	
T_{su1}	200 ns max.	
T_{hold1}	0 ns	
T_{su2}	4.5 µs min.	
T_{hold2}	100 ns max.	

8.23 Fast EPP Data/Address Read Cycle



Parameter	Description	Min.	Typical	Max.
τRPW	nDTSRB or nASTRB pulse width		50 ns	
τRDS	Data setup before nDTSRB or nASTRB high	5 ns		
τRDH	nDTSRB or nASTRB high to data hold			30 ns

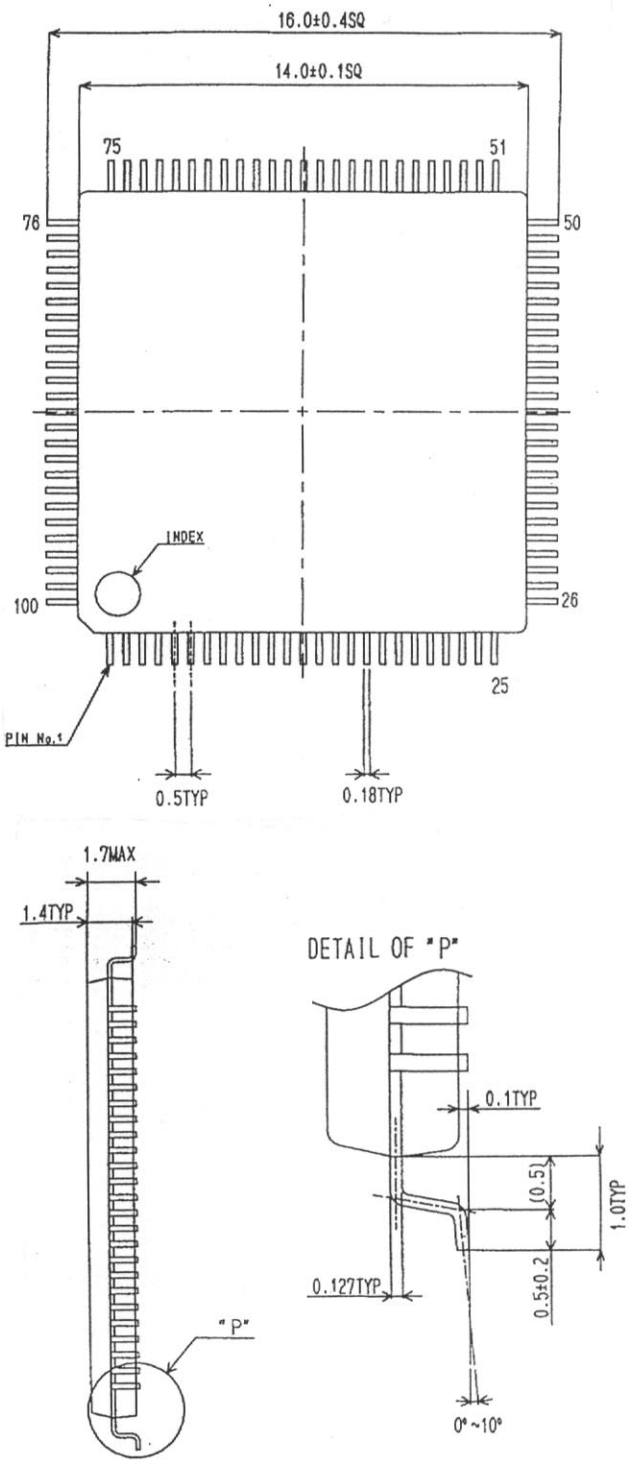
8.24 Fast EPP Data/Address Write Cycle



Parameter	Description	Min.	Typical	Max.
τWPW	nWRITE pulse width		85 ns	
τRPW	nDTSRB or nASTRB pulse width		50 ns	
τWDA	nWRITE low to nDTSRB or nASTRB low		10 ns	
τDAW	nDTSRB or nASTRB high to nWRITE high		25 ns	
τWDH	nDTSRB or nASTRB high to data hold			30 ns

9.0 Package information

9.1 Drawings and Dimensions



9.2 Package Markings

YYWW = Date code

XXXX = Product code

9.3 Thermal Specifications

Parameter	Min.	Typ.	Max.
Junction Temperature (T_{jmax})			125°C
Package thermal impedance (θ_{ja})	65°C/W		75°C/W
Dissipated power @ 65°C ambient (P_{max})			0.8W

10.0 Revision History

Document Title: SL11R USB Controller/16-Bit RISC Processor Data Sheet Document Number: 38-08006				
REV.	**ECN NO.**	**ISSUE DATE**	**ORIG. OF CHANGE**	**DESCRIPTION OF CHANGE**
**	110565	12/14/01	BHA	Converted to Cypress Format from ScanLogic