SL16
Hardware Specification

Technical Reference

ScanLogic Corporation
4 Preston Court,
Bedford, MA 01730
http//:www.scanlogic.com

1. INTRODUCTION **6**
1.1 OVERVIEW 6
1.2 SL16 FEATURES 6
1.3 SL16 BLOCK DIAGRAM 8
1.4 SL16 16-BIT RISC PROCESSOR 8
1.5 3KX16 MASK ROM AND BIOS 8
1.6 INTERNAL RAM 9
1.7 CLOCK GENERATOR 9
1.8 USB INTERFACE 9
1.9 PROCESSOR CONTROL REGISTERS 9
1.10 INTERRUPTS 9
1.11 UART INTERFACE 9
1.12 SERIAL FLASH EEPROM INTERFACE (I2C) 9
1.13 EXTERNAL SRAM/DRAM/EPROM INTERFACE 9
1.14 GENERAL TIMERS AND WATCH DOG TIMER 10
1.15 SPECIAL GPIO FUNCTION FOR SUSPEND, RESUME AND LOW POWER MODES 10
1.16 PROGRAMMABLE PULSE/PWM INTERFACE 10
1.17 SL16 INTERFACES: SL16 MODE AND GPIO MODE 10

2. DEFINITIONS **11**

3. REFERENCES **12**

4. INTERFACE **13**
4.1 INTERNAL MASKED ROM: 0XE800-0XFFFF 13
4.2 EXTERNAL ROM: 0XC100-0XE800 13
4.3 INTERNAL RAM: 0X0000-0X0DFF 14
4.4 CLOCK GENERATOR 15
4.5 USB INTERFACE 15
4.5.1 USB Global Control & Status Register (0xC080: R/W) *16*
4.5.2 USB Frame Number Register (0xC082: Read Only) *16*
4.5.3 USB Address Register (0xC084: R/W) *16*
4.5.4 USB Command Done Register (0xC086: Write Only) *17*
4.5.5 USB Endpoint 0 Control & Status Register (0xC090: R/W) *17*
4.5.6 USB Endpoint 1 Control & Status Register (0xC092: R/W) *17*
4.5.7 USB Endpoint 2 Control & Status Register (0xC094: R/W) *17*
4.5.8 USB Endpoint 3 Control & Status Register (0xC096: R/W) *17*
4.5.9 USB Endpoint 0 Address Register (0x0120: R/W) *18*
4.5.10 USB Endpoint 1 Address Register (0x0124: R/W) *18*
4.5.11 USB Endpoint 2 Address Register (0x0128: R/W) *19*
4.5.12 USB Endpoint 3 Address Register (0x012C: R/W) *19*
4.5.13 USB Endpoint 0 Count Register (0x0122: R/W) *19*
4.5.14 USB Endpoint 1 Count Register (0x0126: R/W) *19*
4.5.15 USB Endpoint 2 Count Register (0x012A: R/W) *19*
4.5.16 USB Endpoint 3 Count Register (0x012E: R/W) *19*
4.6 PROCESSOR CONTROL REGISTERS 19
4.6.1 Version Address Register (0xC004: Read Only) *19*
4.6.2 Configuration Register (0xC006: R/W) *20*
4.6.3 Speed Control Register (0xC008: R/W) *21*
4.6.4 Power Down Control Register (0xC00A: R/W) *22*
4.6.5 Breakpoint Register (0xC014: R/W) *22*
4.7 INTERRUPTS 23
4.7.1 Hardware Interrupts *23*

4.7.2 Interrupt Enable Register (0xC00E: R/W) ... 24
4.7.3 GPIO Interrupt Control Register (0xC01C: R/W) ... 24
4.7.4 Software Interrupts ... 24
4.8 UART INTERFACE. ... 26
4.8.1 UART Control Register (0xC0E0: R/W) ... 26
4.8.2 UART Status Register (0xC0E2: Read Only) ... 27
4.8.3 UART Transmit Data Register (0xC0E4: Write Only) ... 27
4.8.4 UART Receive Data Register (0xC0E4: Read Only) ... 27
4.9 SERIAL FLASH EEPROM INTERFACE (I2C) ... 28
4.10 EXTERNAL SRAM, EPROM, DRAM ... 30
4.10.1 Memory Control Register (0xC03E: R/W) ... 30
4.10.2 Extended Memory Control Register (0xC03A: R/W) ... 31
4.10.3 Extended Page 1 Map Register (0xC018: R/W) ... 31
4.10.4 Extended Page 2 Map Register (0xC01A: R/W) ... 32
4.10.5 DRAM Control Register (0xC038: R/W) ... 32
4.10.6 Memory Map ... 33
4.11 GENERAL TIMERS AND WATCH DOG TIMER ... 35
4.11.1 Timer 0 Count Register (0xC010: R/W) ... 35
4.11.2 Timer 1 Count Register (0xC012: R/W) ... 35
4.11.3 Watchdog Timer Count & Control Register (0xC00C: R/W) ... 35
4.12 SPECIAL GPIO FUNCTION FOR SUSPEND, RESUME AND LOW POWER MODES ... 36
4.13 PROGRAMMABLE PULSE/PWM INTERFACE ... 36
4.13.1 PWM Control Register (0xC0E6: R/W) ... 37
4.13.2 PWM Maximum Count Register (0xC0E8: R/W) ... 38
4.13.3 PWM Channel 0 Start Register (0xC0EA: R/W) ... 38
4.13.4 PWM Channel 0 Stop Register (0xC0EC: R/W) ... 38
4.13.5 PWM Channel 1 Start Register (0xC0EE: R/W) ... 38
4.13.6 PWM Channel 1 Stop Register (0xC0F0: R/W) ... 38
4.13.7 PWM Channel 2 Start Register (0xC0F2: R/W) ... 38
4.13.8 PWM Channel 2 Stop Register (0xC0F4: R/W) ... 39
4.13.9 PWM Channel 3 Start Register (0xC0F6: R/W) ... 39
4.13.10 PWM Channel 3 Stop Register (0xC0F8: R/W) ... 39
4.13.11 PWM Cycle Count Register (0xC0FA: R/W) ... 39

5. SL16 INTERFACES: SL16 MODE AND GPIO MODE ... 40

5.1 SL16 MODE ... 40
5.1.1 DMA Control Register (0xC02A: R/W) ... 41
5.1.2 Low DMA Start Address Register (0xC02C: R/W) ... 41
5.1.3 High DMA Start Address Register (0xC02E: R/W) ... 41
5.1.4 Low DMA Stop Address Register (0xC030: R/W) ... 42
5.1.5 High DMA Stop Address Register (0xC032: R/W) ... 42
5.1.6 Mailbox Protocol ... 42
5.1.7 INBUFF Data Register (0xC0C4: R/W) ... 42
5.1.8 OUTBUFF Data Register (0xC0C4: R/W) ... 42
5.1.9 STATUS Register (0xC0C2: Read Only) ... 43
5.1.10 DMA Protocol ... 43
5.1.11 DMA Control Register (0xC0C0: R/W) ... 44
5.2 GPIO MODE (GPIO) ... 45
5.2.1 I/O Control Register 0 (0xC022: R/W) ... 45
5.2.2 I/O Control Register 1 (0xC028: R/W) ... 45
5.2.3 Output Data Register 0 (0xC01E: R/W) ... 45
5.2.4 Output Data Register 1 (0xC024: R/W) ... 45
5.2.5 Input Data Register 0 (0xC020: R/W) ... 46
5.2.6 Input Data Register 1 (0xC026: R/W) ... 46

6. PHYSICAL CONNECTION 47

6.1 SL16 PACKAGE TYPE 47
6.2 SL16 PIN ASSIGNMENT AND DESCRIPTION 47

7. SL16 CPU PROGRAMMING GUIDE 50

7.1 INSTRUCTION SET OVERVIEW 50
7.2 RESET VECTOR 50
7.3 REGISTER SET 50
7.4 GENERAL PURPOSE REGISTERS 50
7.5 GENERAL PURPOSE/ADDRESS REGISTERS 50
7.6 REGBANK REGISTER (0XC002: R/W) 51
7.7 FLAGS REGISTER (0XC000: READ ONLY) 51
7.8 INSTRUCTION FORMAT 51
7.9 ADDRESSING MODES 52
7.10 REGISTER ADDRESSING 52
7.11 IMMEDIATE ADDRESSING 52
7.12 DIRECT ADDRESSING 52
7.13 INDIRECT ADDRESSING 53
7.14 INDIRECT ADDRESSING WITH AUTO INCREMENT 53
7.15 INDIRECT ADDRESSING WITH OFFSET 53
7.16 STACK POINTER (R15) SPECIAL HANDLING 53
7.17 DUAL OPERAND INSTRUCTIONS 53
7.18 PROGRAM CONTROL INSTRUCTIONS 55
7.19 SINGLE OPERAND OPERATION INSTRUCTIONS 56
7.20 MISCELLANEOUS INSTRUCTIONS 58
7.21 BUILT-IN MACROS 58

8. SL16 - ELECTRICAL SPECIFICATION 59

8.1 ABSOLUTE MAXIMUM RATINGS 59
8.2 RECOMMENDED OPERATING CONDITIONS 59
8.3 CRYSTAL REQUIREMENTS (XTAL1, XTAL2) 59
8.4 EXTERNAL CLOCK INPUT CHARACTERISTICS (XTAL1) 59
8.5 SL16 DC CHARACTERISTICS 60
8.6 SL16 USB TRANSCEIVER CHARACTERISTICS 60
8.7 SL16 RESET TIMING 61
8.8 SL16 CLOCK TIMING SPECIFICATIONS 61
8.9 SL16 PORT I/O READ CYCLE (NON-DMA) 62
8.10 SL16 PORT I/O WRITE CYCLE (NON-DMA) 63
8.11 SL16 DMA READ CYCLE 64
8.12 SL16 DMA WRITE CYCLE 64
8.13 SL16 SIGNALS NAME CONVENTION 65
8.14 SL16 DRAM TIMING 65
8.15 SL16 DRAM READ CYCLE 66
8.16 SL16 DRAM WRITE CYCLE 67
8.17 SL16 CAS-BEFORE-RAS REFRESH CYCLE 68
8.18 SL16 DRAM PAGE MODE READ CYCLE 69
8.19 SL16 DRAM PAGE MODE WRITE CYCLE 70
8.20 SL16 SRAM READ CYCLE 71
8.21 SL16 SRAM WRITE CYCLE 72
8.22 SL16 I2C SERIAL FLASH EEPROM TIMING 73

Example 1 Changing SL16 CPU Speed 15
Example 2 SL16 extended memory setup: 30

Table 1 Internal Masked ROM (SL16 BIOS) *13*
Table 2 Internal RAM memory usage *14*
Table 3 Hardware Interrupt Table *23*
Table 4 Software Interrupt Table *25*
Table 5 Memory Map *33*

Figure 1 SL16 Block Diagram *8*
Figure 2 UART Port Connection *26*
Figure 3 I2C 2K-byte connection *28*
Figure 4 I2C 16K Connection *29*
Figure 5 Special GPIO pull up connection example *36*
Figure 6 PWM Block Diagram *36*
Figure 8 SL16 Mode Block Diagram *40*
Figure 7 GPIO mode Block Diagram *46*

1. INTRODUCTION

1.1 Overview

The SL16 is a low cost, high speed Universal Serial Bus (USB) RISC based Controller. It contains a 16-bit RISC processor with built in SL16 BIOS ROM to greatly reduce firmware development efforts. Its serial flash EEPROM (I2C) interface offers low cost storage for USB device configuration and "customer product specific functions". New functions can be programmed into the I2C by downloading it from a USB Host PC. This unique architecture provides the ability to upgrade products, in the field, without changing the peripheral hardware.

The SL16 Processor can execute code either from internal ROM/RAM or external SRAM and ROM. The SL16 Programmable bi-directional Data Port supports both DMA and I/O modes. A built in USB port supports up to 12 Mbits/sec. the maximum USB transfer rate. All USB protocol modes are supported; Isochronous (up to 1024 bytes/packet) Bulk, Interrupt and Control. The SL16 power source requires only 3.3Volt, and it can be powered via a USB host PC or a Hub. Resume, Suspend and Low power modes are available.

The SL16 offers optimal solution for variety of peripheral products such as: Scanners, Digital Cameras (Video and Still), Color Printers, MFU, Fax's, External Storage devices, Monitors, Connectivity box's, and other peripherals that traditionally interface via EPP or SCSI to host PC.

1.2 SL16 Features

- ScanLogic is offering Development Kit with all of its product line. These Development Kits includes; multiple peripheral Mini-port class drivers for WIN95 OSR2.1 and Windows 98, firmware source code and demo USB source code for variety of applications. Also, available SL16 "C" compiler, debugger, and assembler with reference demo board.
- 48 MHz 16 bit RISC Processor
- Up to 16 bit Programmable Bi-directional Data I/O or DMA port
- Up to 32 bit General Purpose I/O (GPIO) channels.
- 6Kx8 internal Mask ROM with built-in BIOS in support of comprehensive SL16 BIOS interrupt BIOS calls (see [Ref. 1] **SL11R_BIOS** for information), which include USB functions, I2C, UART and Boot-Up options (Boot-up from I2C or External ROM). Executable code can run from 8-bit or 16-bit external SRAM or 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 as data and/or code.
- Two-wire serial EEPROM (I2C) interface port with SL11R BIOS support to allow on board flash EEPROM programming
- Flexible Programmable external memory wait-states and 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 8/16 DMA data path interface.
- Supports 12 MHz/48MHz external crystal or clock.
- Executable code or data can be loaded either from USB port or via UART port. The code/data is moved to RAM buffer for either debugging purposes (utilizing break point register), or to be programmed, as a new value added function, into an I2C.

- USB Port (12 Mbits/sec) including built-in USB transceiver. All USB standard protocol modes are supported; isochronous mode up to 1024 packet size, Bulk, Interrupt and Control modes.
- Four end points are available. Each endpoint utilizing bi-directional DMA port to move data to/from Memory buffer to/from USB. Independently, data can be send/received to/from the Data Port.
- Built-in two Timers, a Watch dog timer (WDT), four programmable channels PWM and four Programmable Timing Generator outputs.
- Four PWM or Programmable Timing Generator outputs channels available. Each channel provides programmable timing generator sequence which can be used to interface to various CCD, CIS, and CMOS image sensors, or can be used for other type of applications.
- Suspend, Resume and Low Power modes are supported.
- UART interface supports from 7,200 Baud to 115.2K Baud.
- USB Generic Min-Port Driver for WIN95 OSR2.1 and Windows 98 are available.
- “C” Compiler, Debugger and QT-Assembler are available
- Package: 100 LPQFP, 0.5 micron.
- Power requirements 3.3v

1.3 SL16 Block Diagram

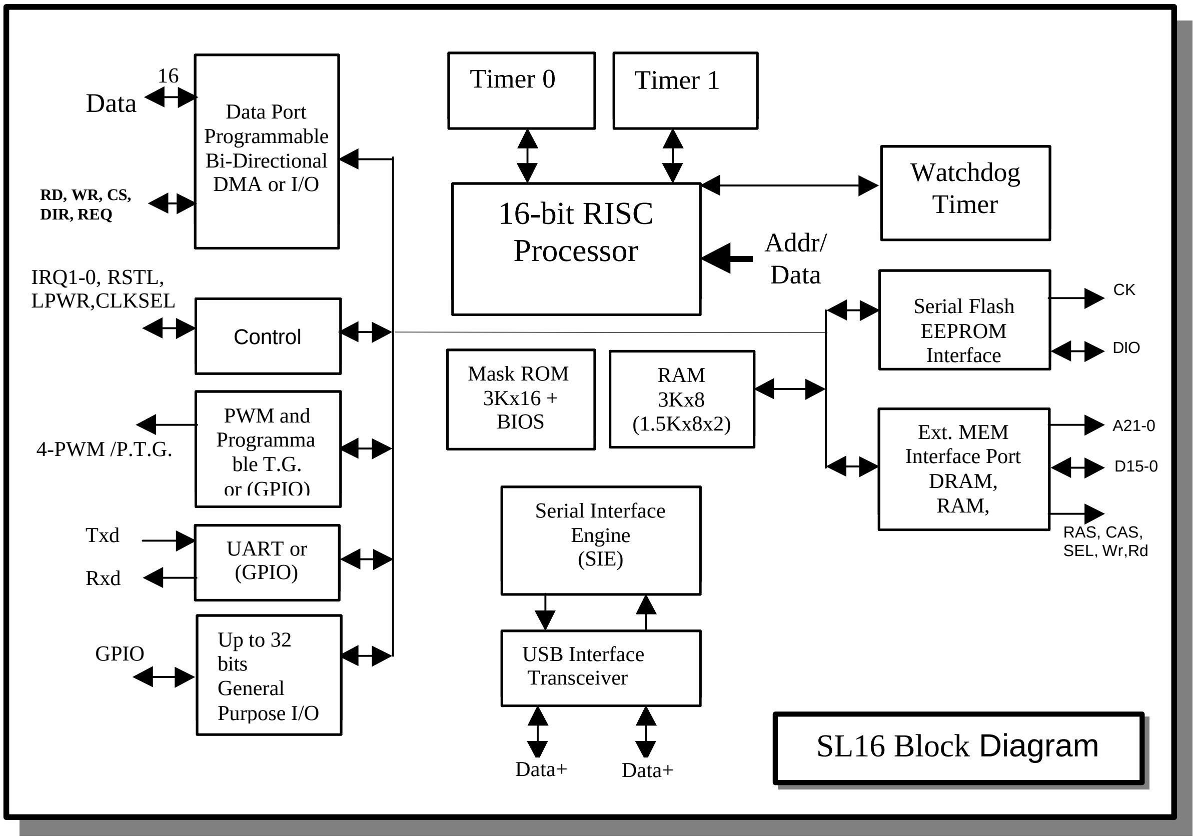

Data
16
RD, WR, CS, DIR, REQ
Data Port Programmable Bi-Directional DMA or I/O
IRQ1-0, RSTL, LPWR,CLKSEL
Control
4-PWM /P.T.G.
PWM and Programmable T.G. or (GPIO)
Txd
Rxd
UART or (GPIO)
GPIO
Up to 32 bits General Purpose I/O
Timer 0
Timer 1
16-bit RISC Processor
Addr/ Data
Watchdog Timer
Serial Flash EEPROM Interface
CK
DIO
Mask ROM 3Kx16 + BIOS
RAM 3Kx8 (1.5Kx8x2)
Ext. MEM Interface Port DRAM, RAM,
A21-0
D15-0
RAS, CAS, SEL, Wr,Rd
Serial Interface Engine (SIE)
USB Interface Transceiver
Data+
Data+
SL16 Block Diagram

Figure 1 SL16 Block Diagram

1.4 SL16 16-Bit RISC Processor

The SL16 can be used as general purpose 16 bit embedded processor. It includes USB interface (Universal Serial Interface Bus), up to 32 bit GPIO in support of variety of functions and modes. The 16-bit main data port can be used either in I/O or DMA bi-directional modes. Also, the SL16 contains 4 PWM channels or four Programmable Time Generator (PTG) signals, a UART, Serial flash EEPROM interface, an additional External DRAM or SRAM interface for extended memory, two Timers, one WDT, internal mask BIOS ROM (3kx16) and SRAM (3Kx8). The SL16 is optimized to offer maximum flexibility in the implementation of variety of applications such as; Embedded Digital Video USB controller, USB scanner Controller, USB cable modems, Printers, external Storage Devices, MFU, and etc.

The SL16 contains a specialized instruction set (RISC) that are highly optimized to provide efficient coding for variety of applications such as video processing algorithms, Network data packets translation and USB transaction processing. The SL16 support simple software interface for all the USB transaction processing, which support bulk mode up to 64 Bytes/packet, Isochronous mode up to 1024 Bytes/packet, Interrupt and control modes.

1.5 3Kx16 Mask ROM and BIOS

The SL16 has a built in 3Kx16 Mask ROM, which contains the SL16 BIOS ROM. This BIOS ROM provides software interface for USB and boot-up option for I2C or external 8/16 EEPROM.

1.6 Internal RAM

The SL16 contains a 3K x 8 internal buffer memory, RAM. The RAM can be used for code/program, variables, buffer I/O, DMA data (i.e. Video data), and USB packets. The memory can be accessed by the 16 Bit processor for data manipulation or by the SIE (Serial Interface Engine), which receive or send USB host data.

1.7 Clock Generator

The 48 MHz external Crystal may be used with the SL16 (or 12MHz). Two pins, X1 and X2, are provided to connect a lower cost crystal circuit to the device. Circuitry is provided to generate the internal clocks requirements of the device. If an external 48 MHz clock is available in the application, it may be used in lieu of the crystal circuit by connecting directly to the X1 input pin.

The 12 MHz external crystal may be used with the SL16 Controller. Two pins, X1 and CLK, are provided to connect a lower cost crystal circuit to the device. The PLL circuitry is provided to generate the internal 48 MHz clock requirements of the device. If an external 12 MHz clock is available, it may be used in lieu of the crystal circuit by connecting directly to the CLK input pin. The selection of the 12MHz is controlled under software setup.

1.8 USB Interface

The SL16 has a built in SIE and USB transceiver, which meets the USB (Universal Serial Bus) specification v1.0. The transceiver is capable or transmitting or receiving serial data at the USB maximum data rate, 12 Mbits/sec. The SL16 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 Endpoint 3 support interrupt transfers, bulk transfers up to 64 Bytes/packet, or Isochronous transfers up to 1024 Bytes/packet size.

1.9 Processor Control Registers

The SL16 provides software control registers that can be used to setup the chip mode, clock control, read software version and software breakpoint control.

1.10 Interrupts

The SL16 provides 127 interrupt vectors for SL11R BIOS software interface (see [Ref. 1] **SL11R_BIOS**).

1.11 UART Interface

The SL16 has a built-in UART interface, which supports 7,200 to 115.2KBaud. It can be utilized as a development port or for other interface requirements. Our development environment for the SL16 chip includes “C” compiler, debugger and assembler. One can download modified code to internal SRAM and debug it utilizing build in Breakpoint register and Breakpoint Interrupt to break on any specified address location.

1.12 Serial flash EEPROM Interface (I2C)

The SL16 provides interface to external serial flash EEPROM. The interface is implemented using General Purpose I/O signals. Variety of serial EEPROM formats can be supported, currently the BIOS ROM supports two-wire serial flash EEPROM type. Serial EEPROM can be used to store specific Peripheral USB configuration and add on value functions. Also, it can be used for field product upgrades.

1.13 External SRAM/DRAM/EPROM Interface

The SL16 provides a multiplexed address port and 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.

1.14 General Timers and Watch Dog Timer

The SL16 has two built in programmable timers that can provide an interrupt to the SL16 Engine. The timers decrement on every microsecond Clock tick. Interrupt occurs on timer reaching zero. A separate Watchdog timer is also provided for monitoring certain activities. The Watchdog timer can also interrupt the SL16 processor.

1.15 Special GPIO function for Suspend, Resume and Low Power modes

The SL16 CPU supports suspend, resume and set the CPU running at low power mode. The SL16 BIOS assigns GPIO18 for the USB DATA+ line pull-up (This pin can simulate the USB cable remove or insert while the USB power is still applied to the board) and the GPIO20 for controlling power off function.

1.16 Programmable Pulse/PWM Interface

The SL16 has four built-in PWM outputs channels available. Each channel provides programmable timing generator sequence which can be used to interface to various line CCD, CIS, CMOS image sensors or can be used for other type of applications.

1.17 SL16 Interfaces: SL16 Mode and GPIO Mode

The SL16 supports fast DMA mode and GPIO mode. On the GPIO mode and SL16 Mode, there are 4 pins, which are used by the UART and the I2C that can not be used as the GPIO. The SL16 Mode includes the **Mailbox Protocol** and **DMA Protocol**. The **Mailbox Protocol** allows asynchronous exchange of data between external Processor (i.e. DSP or Microprocessors) and SL16, via SD15-SD0 (GPIO15-0) bi-directional data port. The DMA Protocol allows the large data can be transferred from or to SL16 memory devices via the 8/16 DMA port.

Note: *The Fast DMA and PWM Interface will not be supported when the SL16 is in the GPIO mode*.

2. DEFINITIONS

USB	**U**niversal **S**erial **B**us
SL16	The SL16 is a Scanlogic **USB** Controller, which provides multiple functions on a single chip. The families of these chips are SL11R, SL11P2USB and SLEPP2USB. In general, the only difference between SL11R and SL16 is the interface mode (see the Configuration Register (0xC006: R/W)).
QT	**Q**uick stream data **T**ransfer engine, which contain a small set of RISC instructions designed for **USB SL16** controller.
QTS	Name Convention that represent utility tools, for example '**QTS**' indicate all tools, which interface with the RS232 serial interface port.
QTU	Name Convention that represent utility tools, for example '**QTU**' indicate all tools, which 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, bi-directional channel controlled by the host device. This mode provides separate address and data cycles over the eight data lines of the interface.
I2C	2-wire Serial flash EEPROM interface.
R0-R15	SL16 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
SL16 BIOS	The SL16 BIOS functional design is equivalent with the SL11R BIOS functional design. See the SL11R BIOS documentation for more in details.

3. REFERENCES

[Ref. 1] SL11R_BIOS

[Ref. 2] SL11R_TOOLS

[Ref. 3] Universal Serial Bus Specification 1.0

4. INTERFACE

4.1 Internal Masked ROM: 0xE800-0xFFFF

The SL16 has a built-in a 3Kx16 internal masked ROM that contained software bootstrap to allow program from I2C or external 8/16-bit ROM can be downloaded or executed. In addition, the internal masked ROM contains SL16 BIOS interrupt calls function (see [Ref. 1] **SL11R_BIOS** for information) that supports all the interface of USB, I2C, UART and Boot-Up options (Boot-up from I2C or External ROM). This SL16 BIOS ROM can help the users to reduce USB software development and other interface supported. The SL16 Chip is ready for all the USB enumeration and download/program code.

The SL16 Internal Masked ROM (i.e. SL16 BIOS) is mapped from address 0xE800 to 0xFFFF. Upon the power-up or hardware reset, the SL16 processor jumps to the address of 0xFFF0, which will contain a long jump to the beginning of the internal ROM of address 0xE800. See table bellows:

Address	Memory Description
0xE800-0xFFEF	SL16 BIOS code/data space
0xFFF0-0xFFF3	Jump to 0xE800
0xFFF4-0xFFF9	Reserved for future use.
0xFFFA-0xFFFB	ROM BIOS Checksum
0xFFFC-0xFFFD	SL16 BIOS Revision
0xFFFE-0xFFFE	Peripheral Revision
0xFFFF-0xFFFF	QT Engine Instruction Revision

Table 1 Internal Masked ROM (SL16 BIOS)

4.2 External ROM: 0xC100-0xE800

The SL16 reserved address from 0xC100 to 0xE800 for external ROM interface. On the default, the SL16 BIOS will scan for the signature ID = 0xCB36 at the location 0xC100 to allow a code can be started to execute at address 0xC102 (see [Ref. 1] **SL11R_BIOS** for more information). The signal nXROMSEL is used for the external ROM mapped from 0xC100 to 0xE800 on the default. However, The Extended Memory Control can be used to configure multiple windows for external ROM setup.

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

4.3 Internal RAM: 0x0000-0x0DFF

The SL16 contains a 1.5Kx16 internal buffer memory. The 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 the video data is read from the camera interface and is sent to the USB port by the internal SL11's USB DMA engine. The SL16 BIOS also use the internal RAM for USB buffers, BIOS's variables and user's data/code. Program executable code or data can reside in multiple locations; in internal masked ROM (6Kx8), Internal RAM (3Kx8), or in external ROM, external SRAM. Program code or data can also be loaded to either internal or external RAM from the USB port, from the RS232 port, and from the I2C.

The SL16 Internal RAM is mapped from 0x0000 to 0x0DFF. See the internal RAM memory usage as shown below:

Address	Memory Description
0x0000 - 0x00FF	Hardware/Software Interrupts
0x0100 - 0x01FF	Register Banks/USB Control/Software Stack
0x0200 - 0x021F	Hardware Interrupts stack
0x0220- 0x0343 [1]	SL16 BIOS internal buffers & variables
0x0334 - 0x0DFF	User's Programming Space

Table 2 Internal RAM memory usage

- The addresses from 0x0000 to 0x00FF are reserved for hardware and software interrupt vectors (see [Ref. 1] **SL11R_BIOS** for more information).
- The addresses from 0x0100 to 0x01FF are reserved for Internal Register Banks (i.e. SL16 register R0-R15 bank0 and R0-R15 bank1) and software stack. Others are reserved for USB Control registers and other read/write control registers.
- The addresses from 0x0200 to 0x021F are reserved for hardware interrupt stack.
- The addresses from 0x0220 to 0x0343 are the available internal RAM that can be used for user's code. The user's code can be download via the USB port or UART interface (see [Ref. 1] **SL11R_BIOS** for more information).

---

[1] This address may be changed due to the new SL16 BIOS revision updated. The new SL16 BIOS may require more internal memory for its variable usage in any new SL16 BIOS.

4.4 Clock Generator

The SL16 as an option to use either a 48 MHz or 12MHz external crystal or oscillator as its clock source. SL16 includes an internal PLL that can be configured by software. At power-up stage, the SL16 BIOS default configuration sets the clock processor at 2/3 of X1 (of the external provided clock)) and the SL16 processor Speed at ½ of the external clock. See the Speed Control Register (0xC008: R/W) and the Configuration Register (0xC006: R/W) for more information.

Example 1 Changing SL16 CPU Speed

Default of the SL16 BIOS assumes to use 48MHz input clock, then the SL16 processor clock is (2/3)*48MHz/2 = 16MHz. See example below:

```
mov     [0xC006],0x10           ;clock = 2/3*X1
mov     [0xC008],1              ;at 24MHz
```

If the X1 input clock is 48MHz, then the maximum speed of the SL16 processor can be set at follows:

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

If the X1 input clock is 12MHz, then the maximum speed of the SL16 processor can be set to:

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

4.5 USB Interface

The SL16 has a built in transceiver that meets the USB (Universal Serial Bus) specification v1.0. The transceivers are connecting directly to the physical layer of the USB engine. The transceiver is capable or transmitting or receiving serial data at the USB maximum data rate, 12 Mbits/sec. The SL16 has four USB-DMA engines for four USB end points. Each of the USB-DMA engines is independently responsible for each USB transaction that is automatically transferred the data from the memory to/from USB port. The 16-bit processor has the ability to set up pointers and block sizes in buffer memory for USB transactions.

The SL16 Controller contains a number of Registers which provide overall control and status functions for USB transactions. The first sets of registers are for overall control and status functions, while the second groups are 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 v1.0. Sec 5.3.1

The SL16 also include the SL16 BIOS that provides a set of subroutines via interrupt calls for all USB interface functions required to communicate to/from an USB host (refer to [Ref. 1] **SL11R_BIOS** for more information). The SL16 BIOS will simplified and reduce the firmware software development.

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

The USB Global Control & Status Register allow to enable/disable and read the current 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-D5	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 after 3mS there is no activity (UA).
- The US and UA bits are cleared after they are read by the SL16 processor.
- D15-D4 are the reserved bits, should be written with zeros.
- The SL16 BIOS will set the UE=1 upon the power-up.

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

The Frame Number Register contains the 11 bit ID Number of 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 SL16 BIOS use this register to detect USB activity for internal BIOS software 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 SL16 BIOS modify this register upon receiving the SET_ADDRESS (see [Ref. 3] **Universal Serial Bus Specification 1.0** on Chapter 9 for more information) from the host.

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

This is the USB command done register that is only used by the control point i.e. end point 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 SL16 BIOS modify this register upon the command completion on end point 0.

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

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

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

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

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

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

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

General description for all endpoint from Endpoint 0 to Endpoint 3:
The SL16 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 Endpoint 3 support interrupt transfers, bulk transfers up to 64 Bytes/packet, or Isochronous transfers up to 1024 Bytes/packet size.

USB Endpoints Control (For Writing)

Each of the endpoint Control Register, when written has the following functions assigned:

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

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

Note:

- 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** is written, and if the direction of the transfer does not match DIR, then the transaction is ignored (as if not enabled).

- 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 (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, it is ignored (so that if the host misses an **ACK** and resends data, we only see the data once).

- The DATA0/DATA1 toggle bit is automatic done by the hardware. To reset this DATA0/DATA1 toggle bit to DATA0, the **Enable** on the **D1** bit should be toggle (i.e. set to ‘0’ and set to ‘1’).

- When the Zero Length on the **D5** is set, the host will receive the zero length USB packet, regardless of the number of bytes in the USB Count register.

- The SL16 BIOS has full control on the USB end point 0. The SL16 BIOS responses all the numeration from host. On other end points, the SL16 BIOS can be used to control under BIOS interrupt calls (see [Ref. 1] **SL11R_BIOS**)

- The SL16 BIOS will set all the USB Control & Status register of end point 1 to end point 3 to zero upon receiving the SET_CONFIG (see [Ref. 3] **Universal Serial Bus Specification 1.0** on Chapter 9 for more information) command from host.

4.5.9 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 USB endpoint is incremented to the number of byte that has been setup to the USB Endpoint Count Register.

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

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

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

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

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

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

4.5.13 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 any transfer, the USB endpoint Count Register is decrement to zero upon the success of the USB transfer.

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

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

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

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

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

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

4.6 Processor Control Registers

The SL16 provides software control registers that can be used to setup the chip mode, clock control, read software version and software breakpoint control.

4.6.1 Version Address Register (0xC004: Read Only)

The Version Address Register stores the current version of the SL16. This register is reserved and not used. The new Version Address Register is located in the SL16 ROM BIOS at address 0xFFFC.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
N7	N6	N5	N4	N3	N2	N1	N0	D7	D6	D5	D4	D3	D2	D1	D0

D15-D8	N7-N0	Version Number left of Decimal Point.
D7-D0	D7-D0	Version Number right of Decimal Point.

Example

Revision 0.4 => N(7-0) = 00000000, D(7-0) = 00000100

4.6.2 Configuration Register (0xC006: R/W)

The Configuration Register is used to configure the SL16 into the appropriate mode, and to select 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

D6-4 C2-0

Clock Configuration bits. These bits select clock source. The clock may come from an outside pin like X1 or X_PCLK or it may come from the PLL multiplier 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.

D2,D1 M1,0: SL16 mode selected that defined as shown:

M1	M0	Mode
0	0	GPIO Mode
0	1	reserved
1	0	reserved
1	1	SL16 Mode

Note:
This M1 and M0 bits should be set one for SL16 mode.

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:
On the default, this bit will be set to zero by the SL16 BIOS. This bit should be set to zero.

D15-D7 Reserved should be set to all zeros.

Where:

PCLK	is connected to the SL16 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 using the PLL, the X1 input pin clock should be 12 MHz and the software should set the **C2=1** to allow the multiply by four.
- **X_PCLK** is a bi-direction 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, which is only on the mode C2=0, C1=1, C0=0.
- After the power-up, the SL16 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).

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

The Speed Control Register allows the operation of the SL16 processor at a number of speed selections. A four bit divider (SPD3-0 + 1) selects the speed as shown below. Note speed will also depend on 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	SL16 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:

On power up, lowest speed is selected for lower power operation. The SL16 BIOS will re-setup this clock to 24MHz after the power-up.

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

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

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

There are two ways to enter to power down mode:

Suspend or **Halt**.

D5	USB	Enable restarts on USB transition. Will result in device power up.
D4	GPIO	Enable restarts on GPIO transition. Will result 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 allows 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	Suspend mode stops all clocks in the SL16 to save power. 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.

4.6.5 Breakpoint Register (0xC014: R/W)

The Breakpoint Register holds the breakpoint address. When an access to this address is done an INT127 occurs.

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.

4.7 Interrupts

The SL16 provides 127 interrupt vectors. The first 64 vectors are the hardware interrupts and the next 64 interrupt vectors are the software interrupts (see the [Ref. 1] **SL11R_BIOS** for more information).

4.7.1 Hardware Interrupts

The SL16 allocates address from 0x0000 to 0x003E for hardware interrupts. The hardware interrupt vectors are shown below:

Interrupt Number	Vector Address	Interrupt Type
0	0x0000	Timer0 ♣
1	0x0002	Timer1 ♦
2	0x0004	GP IRQ0 ♦
3	0x0006	GP IRQ1 ♦
4	0x0008	UART Tx ♣
5	0x000A	UART Rx ♣
6	0x000C	Fast DMA Done ♦
7	0x000E	USB Reset
8	0x0010	USB SOF ♣♣
9	0x0012	USB Endpoint0 No Error ♣
10	0x0014	USB Endpoint0 Error ♣
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	SL16 Mailbox TX Empty (SL16 Mode Only) ♦
18	0x0024	SL16 Mailbox RX Full (SL16 Mode Only) ♦
19-63	0x0026- 0x003E	Reserved ♦

Table 3 Hardware Interrupt Table

Note: ♣ These hardware interrupt vectors are reserved for internal SL16-BIOS usage. User should not attempt to overwrite these functions.

♣♣ The SOF interrupt is generated when there is an incoming SOF on the USB.

♦ These hardware interrupt vectors are initialized to return on the interrupt.

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

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

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

This is a global hardware interrupt enable register that allows controlling these previous hardware interrupt vectors. The SL16 BIOS default setup 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 (SL16 Mode Only)
D5	USB	USB Interrupt enable.
D4	FDMA	Fast DMA Done Interrupt enable (SL16 Mode Only).
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.

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

This register defines the polarity of the GPIO interrupt on the IRQ1 (GPIO26) and IRQ0 (GPIO25). The **GPIO** bit on the Interrupt Enable Register must be set to allows these below setting:

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.

4.7.4 Software Interrupts

The SL16 allocates address from 0x0040 to 0x00FE for software interrupts. The software interrupt vectors are shown below:

Interrupt Number	Vector Address	Interrupt Type
64 (0x40)	0x0080	I2C_INT ♣
65 (0x41)	0x0082	Reserve for future extension of other Serial EEPROM
66 (0x42)	0x0084	UART_INT ♣
67 (0x43)	0x0086	SCAN_INT ♣
68 (0x44)	0x0088	ALLOC_INT ♣
69 (0x45)	0x008A	**Data**: start of free memory. Default=0x200 ♣♣
70 (0x46)	0x008C	IDLE_INT
71 (0x47)	0x008E	IDLER_INT
72 (0x48)	0x0090	INSERT_IDLE_INT
73 (0x49)	0x0092	PUSHALL_INT ♣
74 (0x4a)	0x0094	POPALL_INT ♣
75 (0x4b)	0x0096	FREE_INT ♣
76 (0x4c)	0x0098	REDO_ARENA ♣
77 (0x4d)	0x009A	HW_SWAP_REG ♣
78 (0x4e)	0x009C	HW_REST_REG ♣
79 (0x4f)	0x009E	SCAN_DECODE_INT
80 (0x50)	0x00A0	USB_SEND_INT ♣
81 (0x51)	0x00A2	USB_RECEIVE_INT ♣
82 (0x52)	0x00A4	Reserved
83 (0x53)	0x00A6	USB_STANDARD_INT
84 (0x54)	0x00A8	**Data**: Standard loader vector. Defaut=0 ♣♣
85 (0x55)	0x00AA	USB_VENDOR_INT
86 (0x56)	0x00AC	**Data**: USB_Vendor loader. Default = 0xff ♣♣
87 (0x57)	0x00AE	USB_CLASS_INT
88 (0x58)	0x00B0	**Data**: USB_Class_Loader. Default = 0 ♣♣
89 (0x59)	0x00B2	USB_FINISH_INT
90 (0x5a)	0x00B4	**Data**: Device Descriptor. Default = Scanlogic Device Desc ♣♣
91 (0x5b)	0x00B6	**Data**: Configuration Desc. Default = Scanlogic Configuration ♣♣
92 (0x5c)	0x00B8	**Data**: String Descriptor. Default = Scanlogic String Desc. ♣♣
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	Reserve for future addition secondary USB Port
105 (0x69)	0x00D2	POWER_DOWN_SUBROUTINE
106-109	0xD4-0xDA	Reserve for future secondary USB Port
110-124	0xDE-0xF8	User's ISR or internal peripheral interrupt
125-127	0xFA-0xFE	Reserve for the Debugger

Table 4 Software Interrupt Table

Note: ♣ These software vectors are reserved for the internal SL16-BIOS. User should not overwrite these functions.

♣♣ These vectors are used as the data pointers. 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.

4.8 UART Interface.

The SL16 Controller UART port supports a range of baud rates from 7200 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 on receiver buffer transitioning to FULL or transmit transitioning to buffer EMPTY.

The SL16 BIOS uses the UART port for all software debugging process. It is recommended that user should include this interface into their hardware design. For example user can add UART interface via only 4-pin header, which require off board external RS-232 transceiver (i.e. MAX202 RS-232 transceiver can resided on the external serial cable, which use 5V and Ground via the 4 pin header).

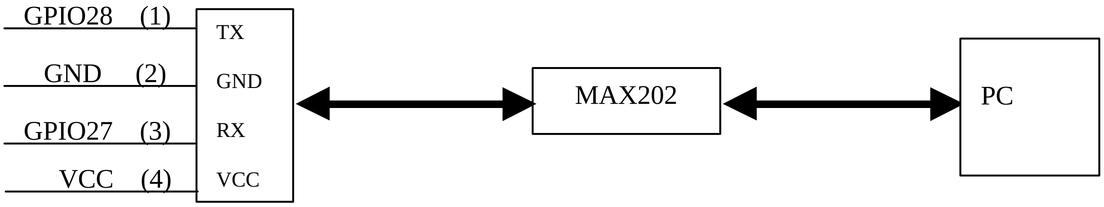


Figure 2 UART Port Connection

The SL16 BIOS uses GPIO28 for data transmit (TX) and GPIO27 for data receive (RX). These two pins can not be used for any other purpose.

Note: The SL16 BIOS will setup the default Baud rate for the UART as 14400 baud.

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

The SL16 allocates two General Purpose I/O signals for the UART function, they are GPIO28 (UART_TXD) and GPIO27 (UART_RXD). At the power, the SL16 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.

4.8.2 UART Status Register (0xC0E2: Read Only)

This register is used by the SL16 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. Sets when data moves from buffer to output shift register.

Note:

No error detection for receiving data is supported.

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

This register is used by the SL16 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

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

This register is used by the SL16 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.

4.9 Serial flash EEPROM Interface (I2C)

The SL16 provides interface to external serial flash EEPROM. The interface is implemented using General Purpose I/O signals. Variety of serial EEPROM formats can be supported, currently the BIOS ROM supports two-wire serial flash EEPROM type. Serial EEPROM can be used to store specific Peripheral USB configuration and add on value functions. Also, it can be used for field product upgrades.

The SL16 BIOS uses this interrupt to read and write from/to an external serial flash EEPROM. The recommended serial EEPROM device is a 2-Wire Serial CMOS EEPROM (AT24CXX Device Family). Currently, the SL16 BIOS Revision 1.1 allows reading/writing to/from EEPROM, up to 2K Bytes, which is 16K bits I2C 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 for either to execute this code or use it as external look up table data source. The advantage of the I2C/EEPROM interface is saving space and cost to compare it with using an external 8-bit PROM/EPROM.

The SL16 BIOS uses two GPIO pins, GPIO31 and GPIO30 to interface to external serial EEPROM (see below figure):

- GPIO31 is connected to the Serial Clock Input (SCL).
- GPIO30 is connected to the Serial Data (SDA).
- It is recommended to add 5K to 15K pull-up resistors on the Data line (i.e. GPIO30).
- Pin 1 (A0), Pin 2 (A1), Pin 3 (A2), Pin 4 (GND) and Pin 7 Write Protect) are connected to Ground.

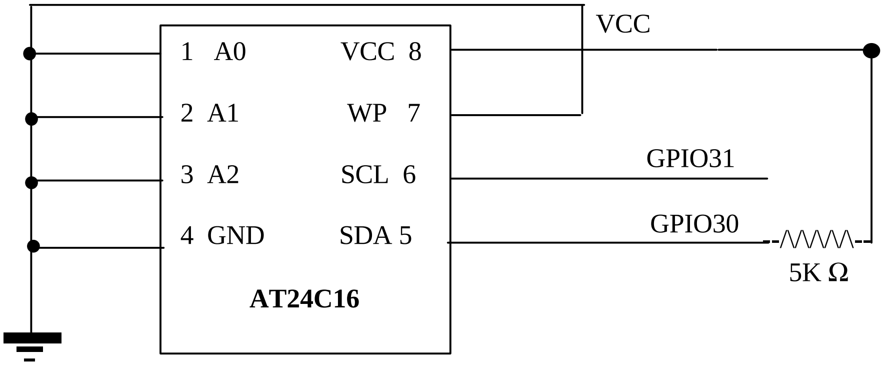


Figure 3 I2C 2K-byte connection

The current SL16 BIOS only support up to 2Kbyte serial EEPROM. To read and write to a device that is larger than 2Kbytes, greater than 16K bits, the SL16-BIOS requires additional serial EEPROM to be connected as shown below:

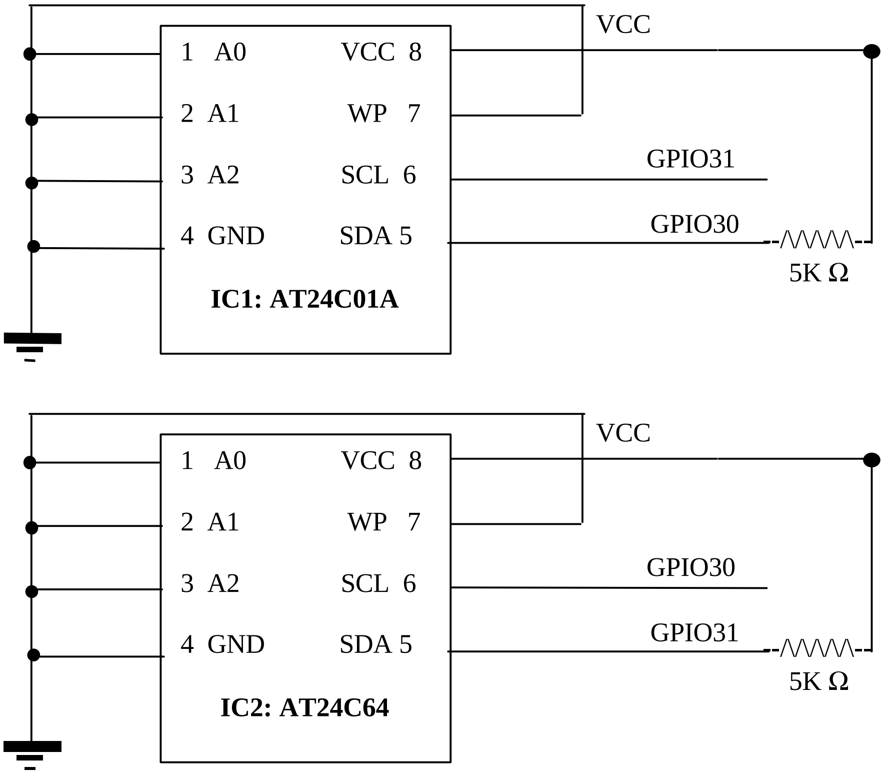


Figure 4 I2C 16K Connection

In this example, the SL16 BIOS will first access the (small) program residing on **IC1** serial flash EEPROM, and then it will access the second **IC2 EEPROM** (see [Ref. 1] **SL11R_BIOS** for more information).

4.10 External SRAM, EPROM, DRAM

The SL16 has a multiplexed address ports, 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 SL16 BIOS is setup for external SRAM and serial flash EEPROM. Also, the external memory interface is set up as 16-bit and 7 wait states for both external SRAM and EEPROM. DRAM controller needs to be set up by the user.

Example 2 SL16 extended memory setup:

```
internal_rom_start:
        mov     [0xC03A],0x0077         ;set 16-bit ROM & 7 wait
        cmp     [0xC100],0xCB36         ;check for special vector ROM
        je      0xC102
        mov     [0xC006],0x10           ;2/3 clock
        mov     [0xC008],1              ;at 24mhz
        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.

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

This register provides the control Wait State for the internal RAM and internal 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	one-wait state for internal RAM is added
D1	RO	one-wait state for internal ROM is added
D0	DB	DEBUG mode is enabled. Internal address bus is echoed to external address pins.

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

This register provides the control Wait State for the external SRAM/DRAM/EPROM interfaces.

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, ‘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.

4.10.3 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 set to 0000h.

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) and 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 SL16 access the address 0x8000-0x9FFF.

4.10.4 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 set to 0000h.

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 SL16 access the address 0xA000-0xBFFF.

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

A multiplexed address port and 16 bit data port has been 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. So this register is designed to control the DRAM function.

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 CAS signal is issued prior to 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.

4.10.6 Memory Map

The total memory space allocated by the SL16 is 64K-byte. Program, data, and I/O space are contained within 64K-byte address space. The program code or data can be stored in either external RAM, or external ROM.

The SL16 allows extended data (video) to be stored on an external EDO DRAM. The entire (video image) data can be DMA directly to DRAM without software intervention. The total of DMA size can be up-to 2M-byte addressing space. The SL16 processor can access DRAM data via address space, from **0x8000** to **0xBFFF**.

The SL16 Controller provides a 16 bit Memory interface that can support a wide variety of external DRAM, RAM and ROM devices. The SL16 Controller memory space is byte addressable and is divided as follows:

FUNCTION	ADDRESS
Internal RAM	0x0000 – 0x0BFF
External RAM	0x0C00 – 0x7FFF♣
Extended Page 1/DRAM	0x8000 – 0x9FFF
Extended Page 2/DRAM	0xA000 – 0xBFFF
Memory Mapped Registers	0xC000 – 0xC0FF
External ROM	0xC100 – 0xE7FF♣♣
Internal ROM	0xE800 – 0xFFFF

Table 5 Memory Map

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

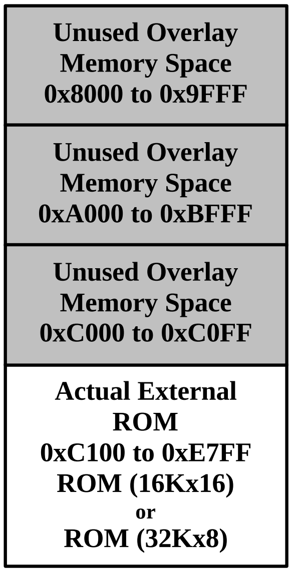
Note:

♣ The External RAM address from 0x0000 to 0x0C00 will not be accessible from the SL16 processor. This is an overlay memory space between internal RAM and external RAM. The actual total size of the external RAM will be (0x8000-0x0C00), which is 29K-byte. The signal name nXRAMSEL on SL16-pin56 will be active when the CPU access address from 0x0C00 to 0x7FFF.

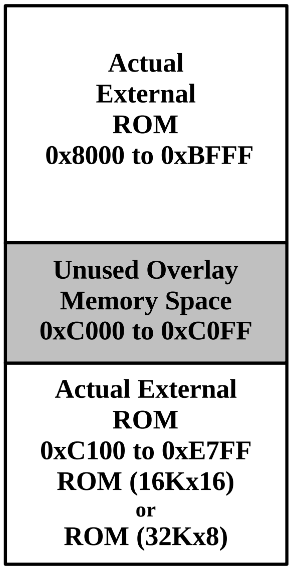
Unused Overlay Memory Space 0x0000 to 0x0C00
Actual External RAM 0x0C00 to 0x7FFF SRAM (16Kx16) or SRAM (32Kx8)

♣♣When the bit 12 (ROM Merge Bit) of the Extended Memory Controller Register at address 0xC03A is equal to zero, 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 name nXROMSEL on SL16-pin57 will be active when the CPU accesses the address from 0xC100 to 0xE7FF. The signal name nXMEMSEL on SL16-pin58 will be active when the CPU accesses the address from 0x8000 to 0xBFFF. When the bit 12 (ROM Merge Bit) of the Extended Memory Controller Register at address 0xC03A is equal one, then the External ROM address space will be mapped into the multiple windows, which are: 0x 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-byte equal to 25.75K.

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


Unused Overlay Memory Space 0x8000 to 0x9FFF
Unused Overlay Memory Space 0xA000 to 0xBFFF
Unused Overlay Memory Space 0xC000 to 0xC0FF
Actual External ROM 0xC100 to 0xE7FF ROM (16Kx16) or ROM (32Kx8)

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


Actual External ROM 0x8000 to 0xBFFF
Unused Overlay Memory Space 0xC000 to 0xC0FF
Actual External ROM 0xC100 to 0xE7FF ROM (16Kx16) or ROM (32Kx8)

4.11 General Timers and Watch Dog Timer

The SL16 Controller has two built in programmable timers that can provide an interrupt to the SL16 Engine. The timers decrement on every microsecond clock tick. Interrupt occurs on timer reaching zero.

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

The SL16 BIOS uses the timer 0 for time-outs function and the power down mode. At the end of the power up, the SL16 BIOS disable the timer 0 interrupt. If the user wishes to use the 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.

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

The SL16 provides timer 1 for user application. The SL16 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

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

The SL16 provides Watchdog timer to monitor certain activities. The Watchdog timer can also interrupt the SL16 processor. The default value of this register is set to all zero.

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:

- Must send a Reset Count (RC), before time-out occurs to avoid Watchdog going off.
- 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 watch dog timeout occurred.
- The WT value will be cleared on the next external reset.

4.12 Special GPIO function for Suspend, Resume and Low Power modes

The SL16 CPU supports suspend, resume and set the CPU running at low power mode. The SL16 BIOS assigns GPIO18 for the USB DATA+ line pull-up (This pin can simulate the USB cable remove or insert while the USB power is still applied to the board) and the GPIO20 for controlling power off function. The GPIO20 can be used for device low power mode, it will turn of or disable external powers to the peripheral in suspend mode. Once USB power is resumed, the external power can be enabled again to the peripheral. The SL16 BIOS will execute the pull up interrupt upon the power-up. To use this feature, the GPIO18 pin must be connected to the DATA+ line of the USB connector (see Figure below). For more information about these function, see the [Ref. 1] **SL11R_BIOS**.

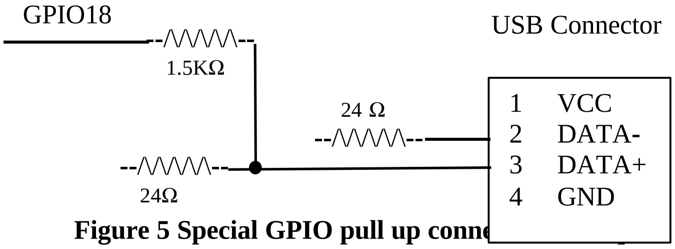


Figure 5 Special GPIO pull up conne

4.13 Programmable Pulse/PWM Interface

The SL16 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 individual Start and Stop Counter Registers which are provided for each of the four output channels.

To set up for PWM operation, the Maximum Count Register is set to the desired maximum count value. Then the start and stop value for each channel is also written with the required values. The start and stop values are chosen to achieve the desired pulse widths each cycle between the counter start and the 10-bit Maximum Count Register value. When the channels are not enabled in the Control Register, the associated I/O pins revert to GPIO use.

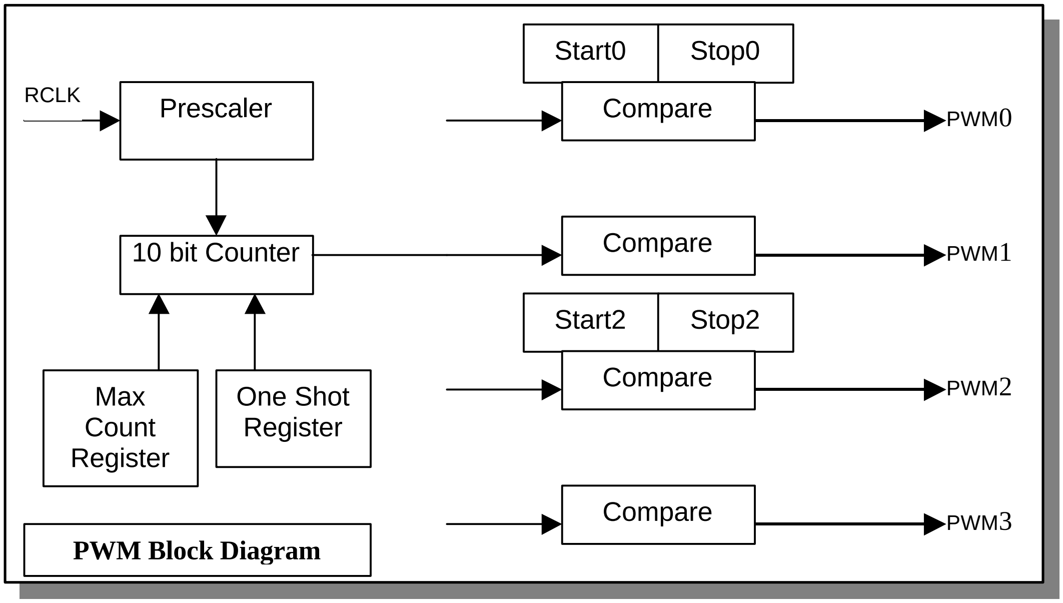


Figure 6 PWM Block Diagram

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

4.13.1 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	48.00M Hz
001	24.00M Hz
010	06.00M Hz
011	01.50M Hz
100	375K Hz
101	93.80K Hz
110	23.40K Hz
111	05.90K Hz

D8	OS	Enable One Shot Mode for PWM channels. One Shot mode will allow the number of counter cycles set in the PWM cycle count register to run and then it will stop. 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.

Note:

- If not enabled, i.e., if set = '0', the pins are GPIO. To force the outputs to “0” or “1”.
- If start register = stop register, then output stays “0”.
- If stop register > max count register, then output stays “1”.

4.13.2 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.

4.13.3 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.

4.13.4 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.

4.13.5 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.

4.13.6 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.

4.13.7 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.

4.13.8 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.

4.13.9 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.

4.13.10 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.

4.13.11 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

5. SL16 INTERFACES: SL16 MODE AND GPIO MODE

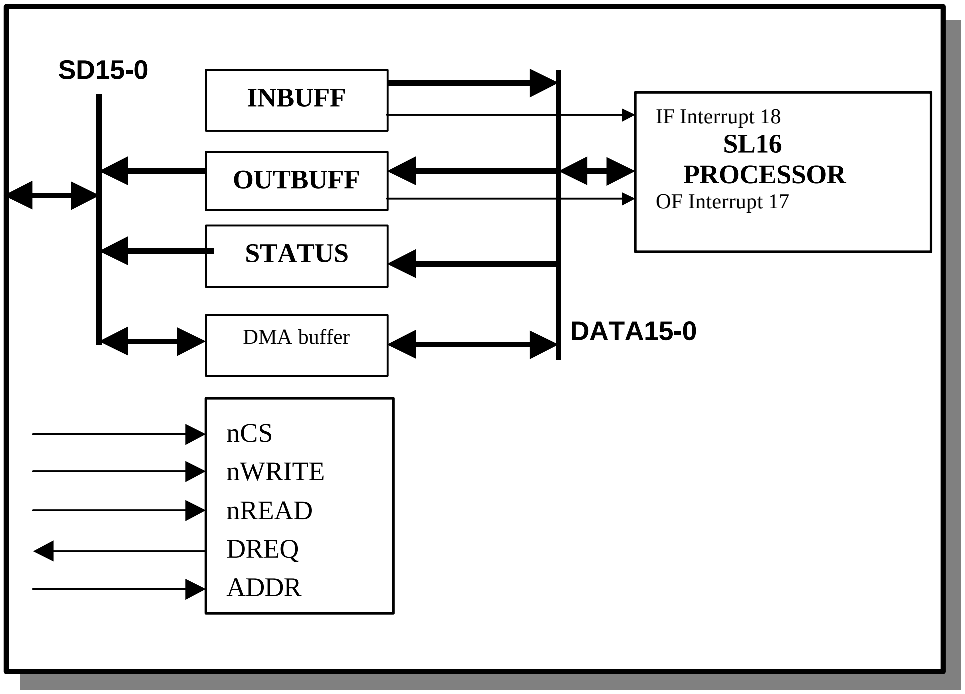
5.1 SL16 Mode

This SL16 Mode includes the **Mailbox Protocol** and **DMA Protocol**. The **Mailbox Protocol** allows asynchronous exchange of data between external Processor (i.e. DSP or Microprocessors) and SL16, via SD15-SD0 (GPIO15-0) bi-directional data port. The DMA Protocol allows the large data can be transferred from or to SL16 memory devices via the 8/16 DMA port.

The SL16 has four built-in PWM outputs channels available under SL16 mode. Each channel provides programmable timing generator sequence which can be used to interface to various line CCD, CIS, CMOS image sensors or can be used for other type 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 GPIO Mode.*



5.1.1 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 of S15-SD0/SD7-SD0 is automatically written into the RAM of the SL16, under fast DMA control. The DMA must be enabled in the DMA Control and Address register.

D2	TSZ	Transfer Size. 8 bit when set = '1', 16-bit when set = '0'
D1	DIR	DMA Direction. When set = '0' ***Peripheral to Memory***. When set = '1' Memory to Peripheral.
D0	DMA	DMA Enabled when set = '1'. Bit clears to ‘0’ when DMA is done.

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

This register contains the starting SL16 memory address of the low order word.

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.

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

This register contains the starting SL16 memory address of the high order word.

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 must be matched on the High DMA Stop Address register, when select the DMA to or from DRAM.

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

This register contains the stopping SL16 memory address of the low order word. This is the last DMA address in memory. DMA will stop when this address is reached. If the **FDMA** bit in the Interrupt Enable Register (0xC00E: R/W) is enabled, an interrupt will be generated when this address is reached.

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 stopping DMA address

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

This register contains the stopping SL16 memory address of the high order word. 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, when select the DMA to or from DRAM.

5.1.6 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), the ADDR pin should be driven high. The ADDR pin should be driven low when accesses the Mailbox (STATUS). The external processor and SL16 can both access to the INBUFF, OUTBUFF & STATUS Mailbox. The SL16 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 can not access the Mailbox during the DMA in progress.*

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

The external processor will write to this register with the ADDR signal set to one and the SL16 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

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

The SL16 will write to this register and the external processor will read from this register with the ADDR signal set to one. The SL16 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

5.1.9 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 SL16 also can access this register.

5.1.10 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, the SL16 is ready to DMA to or from the external device (scanner, printer, camera, modem or etc.) by asserting the DREQ signal when data is requested or ready to send. When the external device is ready to send Data it inserts **nWRITE** signal, Data must be available at this point, or if external device is ready to accept data it inserts the **nREAD** signal. If the **FDMA** bit in the Register 0xC00E is enabled, the interrupt will be generated after the DMA transfer done.

Note:

*To enable the Fast DMA done interrupt, the bit **FDMA** in the Register 0xC00E must be enabled.*

This mode can be used to move large amount of data to from to variety of Peripherals such as, Scanner, Printer, Cable Modem, External Storage device, and others. For example for a DVC, video data from the camera can be moved via DMA to internal memory buffer for it to be transferred to the USB host. At the same time, this data can be setup to transfer to the host via the USB DMA engine (i.e. no SL16 Processor is involved, since USB has its own DMA engine).

User also can set either 8 bits or 16 bits transfers and the direction, Peripheral to internal memory or SL16 to Peripheral. The DMA Start Address Registers contain the starting of DMA data location to be written into the SL16 memory. The DMA Stop Address Registers contain the stopping DMA address in the SL16 memory. DMA operation will stop when it reaches to the DMA Stop Address registers and if the **FDMA** bit in the Register 0xC00E is enabled, an interrupt will be issued to indicate DMA operation is done. For more information of controlling the DMA operation, see SL16 section for more in details.

5.1.11 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 setup.

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	EH	EL

D2	DREQ	External DREQ DMA Enable, if set to '1', the SL16 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	EH	High Byte Enable
D0	EL	Low Byte Enable

5.2 GPIO Mode (GPIO)

On the GPIO mode, the SL16 has up to 32 general purpose IO signals are available. However there are 4 pins, which are used by the UART and the I2C that can not be used as the GPIO pins.

When the SL16 interface is in the GPIO mode, A number of General Purpose Digital I/O (GPIO) pins are supported by the SL16 Controller. Some of these pins can be assigned to special functions. However, when not configured as special functions, the pins can be used as GPIO. The special functions for example PWM Output will override the GPIO function.

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 in the Output Data Register even though the outputs are not enabled.

Note: *The Fast DMA and PWM Interface will not be supported in the GPIO mode*.

5.2.1 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, that corresponding GPIO data pin becomes an output pin. When any bit of this register set to zero, that 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.

5.2.2 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, that corresponding GPIO data pin becomes an output pin. When any bit of this register set to zero, that 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.

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

This register controls the output 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
O15	O14	O13	O12	O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1	O0

D15-0	O15-0	Output Pin Data

5.2.4 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
O15	O14	O13	O12	O11	O10	O9	O8	O7	O6	O5	O4	O3	O2	O1	O0

D15-0	O31-16	Output Pin Data

5.2.5 Input Data Register 0 (0xC020: R/W)

This register controls 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

5.2.6 Input Data Register 1 (0xC026: R/W)

This register controls 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

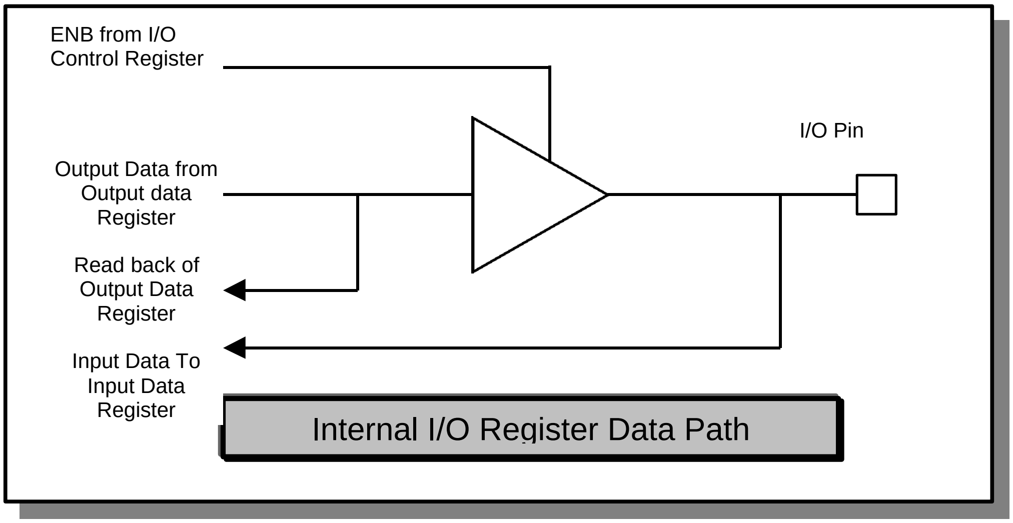

ENB from I/O Control Register
I/O Pin
Output Data from Output data Register
Read back of Output Data Register
Input Data To Input Data Register
Internal I/O Register Data Path

Figure 8 GPIO mode Block Diagram

6. PHYSICAL CONNECTION

6.1 SL16 Package Type

The SL16 is 100 PQFP.

6.2 SL16 Pin Assignment and Description

Pin Name	Pin No	GPIO pins	Pin Type	**SL16 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
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. SL16 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 Flash EEPROM clock, or GPIO31
SEDO	61	GPIO30	Bidir	SEDO, Serial flash EPROM Data, or GPIO30 This pin requires a 5K Ohm 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
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 SL16 processor
PWM3, or IRQ0 (in)	69	GPIO24	Bidir	IRQ0 (in) interrupts the SL16 processor or PWM. See the PWM Control register setup 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	Output	DMA Request Enable. DREQ indicates that SL16 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 bi-directional SDATA Port, or GPIO18
nWRITE	77	GPIO17	Bidir	Active input low signal used to indicate write data transfers to the general bi-directional SD15-0 Data Port. Signal is driven high for read transfers to the SL16, or GPIO17
nREAD	78	GPIO16	Bidir	Active low input signal used to indicate read data transfers from the general bi-directional SD15-0 Data Port. Or GPIO16
GND	79		GND	Digital ground.
SD15	80	GPIO15	Bidir	Main bi-directional SDATA port bit 15, or GPIO15
SD14	81	GPIO14	Bidir	Main bi-directional SDATA port bit 14, or GPIO14
SD13	82	GPIO13	Bidir	Main bi-directional SDATA port bit 13, or GPIO13

SD12	83	GPIO12	Bidir	Main bi-directional SDATA port bit 12, or GPIO12
SD11	84	GPIO11	Bidir	Main bi-directional SDATA port bit 11, or GPIO11
SD10	85	GPIO10	Bidir	Main bi-directional SDATA port bit 10, or GPIO10
SD9	86	GPIO9	Bidir	Main bi-directional 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 bi-directional SDATA port bit 8, or GPIO8
SD7	92	GPIO7	Bidir	Main bi-directional SDATA port bit 7, or GPIO7
SD6	93	GPIO6	Bidir	Main bi-directional SDATA port bit 6, or GPIO6
SD5	94	GPIO5	Bidir	Main bi-directional SDATA port bit 5, or GPIO5
SD4	95	GPIO4	Bidir	Main bi-directional SDATA port bit 4, or GPIO4
SD3	96	GPIO3	Bidir	Main bi-directional SDATA port bit 3, or GPIO3
SD2	97	GPIO2	Bidir	Main bi-directional SDATA port bit 2, or GPIO2
SD1	98	GPIO1	Bidir	Main bi-directional SDATA port bit 1, or GPIO1
SD0	99	GPIO0	Bidir	Main bi-directional SDATA port bit 0, or GPIO0
VDD	100		Power	+3.3 VDC Supply

7. SL16 CPU PROGRAMMING GUIDE

This is the preliminary specification for the SL16 Processor Instruction set.

7.1 Instruction Set Overview

This document describes the SL16 CPU Instruction Set, Registers and Addressing modes Instruction format etc. The SL16 PROCESSOR uses a unified program and data memory space; although this RAM is also integrated into the SL16 core, provision has been made for external memory as well.

The SL16 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 SL16 PROCESSOR engine supports byte and word addressing. Subsequent sections of this document will describe:

- The SL16 PROCESSOR Engine (QT Engine) Register Set
- SL16 PROCESSOR Engine Instruction Format
- SL16 PROCESSOR Engine Addressing Modes
- SL16 PROCESSOR Engine Instruction Set

7.2 Reset Vector

On receiving hardware reset, the SL16 Processor jumps to address 0xFFF0, which is an internal ROM address.

7.3 Register Set

The SL16 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
FLAGS	Contains flags: defined below
INTERRUPT ENABLE	Bit masks to enable/disable various interrupts

7.4 General Purpose Registers

The general-purpose registers are exactly what their name implies. They 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 set aside to be used 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 SL16 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

Before discussing addressing modes supported by the SL16 Processor, it is necessary to define the instruction format. 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 SL16 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:

- The "b/w" bit defined for some addressing modes is set to 1 for byte-wide access, and 0 for word access.
- The definitions for bits 5 and 4 for immediate and direct addressing appear to conflict with the bits defined for Indirect and Indirect with Auto Increment. This conflict is eliminated by disallowing indirect with auto increment and byte-wide Indirect addressing 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 the 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 with 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.
- SL16 - 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)

J*cc*L 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)

R*cc* 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 softare 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: 1) For the JUMP mnemonics, adding an “L” to the end indicates an 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.”

Also, 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. This is possible because an instruction such as SHR [destination],0 is semantically meaningless.
- The SL16 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 SL16 QT assembler implements several built-in macros. The table below shows the macros, and the memnonics 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]

8. SL16 - ELECTRICAL SPECIFICATION

8.1 Absolute Maximum Ratings

This section lists the absolute maximum ratings of the SL16. 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.

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

8.2 Recommended Operating Conditions

Parameter	Min.	Typical	Max
Power Supply Voltage, VDD	3.0 V	3.3v	3.6 V
Power Supply Voltage, VDD1	3.0 V		3.6 V
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
Parallel Resonant Frequency		48MHz	
Frequency Drift over Temperature			+/- 20 ppm
Accuracy of Adjustment			+/- 30 ppm
Series Resistance			50 ohms
Shunt Capacitance	3 pf		6 pf

8.4 External Clock Input Characteristics (XTAL1)

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

8.5 SL16 DC Characteristics

Symbol	Parameter	Min.	Typical	Max
IL	Input Voltage LOW	-0.5 V		0.8 V
V_{IH}	Input Voltage HIGH	2.0V		VDD+ 0.3V
V_{OL}	Output Voltage LOW(IoL=4ma)			0.4 V
V_{OH}	Output Voltage HIGH(IoH=-4ma)	2.4 V		
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)			< 100mA
I_{USB}	Supply Current (VDD1)			< 50mA

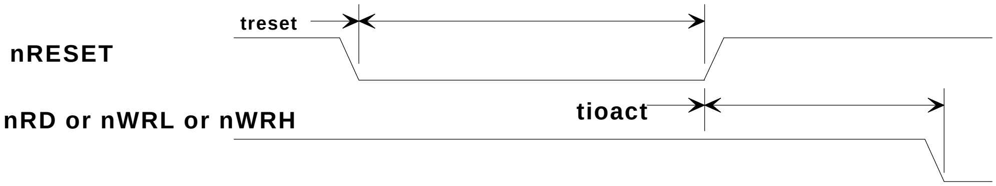
8.6 SL16 USB Transceiver Characteristics

Symbol	Parameter	Min.	Typical	Max
V_{IHYS}	Hysteresis On Input (Data+, Data-)	0.1 V		200 mV
V_{USBIH}	USB Input Voltage HIGH		1.5 V	2.0 V
V_{USBIL}	USB Input Voltage LOW	0.8 V	1.3 V	
V_{USBOH}	USB Output Voltage HIGH	2.2 V		
V_{USBOL}	USB Output Voltage LOW			0.7 V
Z_{USBH}	Output Impedance HIGH STATE	28 Ohms		43 Ohms
Z_{USBL}	Output Impedance LOW STATE	28 Ohms		43 Ohms
I_{USB}	Transceiver Supply p-p Current (3.3V)			< 150mA

Notes:

- All typical values are VDD2 = 3.3 V and TAMB= 25°C.
- Z_{USBX} Impedance Values includes an external resistor of 24 Ohms ± 1%

8.7 SL16 RESET TIMING

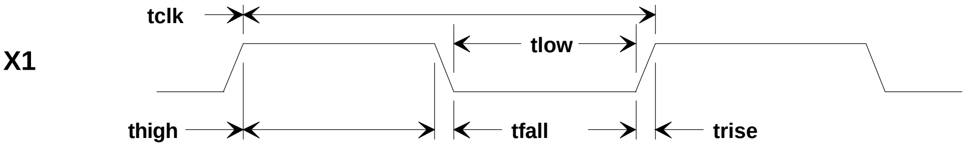


SL11R RESET TIMING

Symbol	Parameter	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.

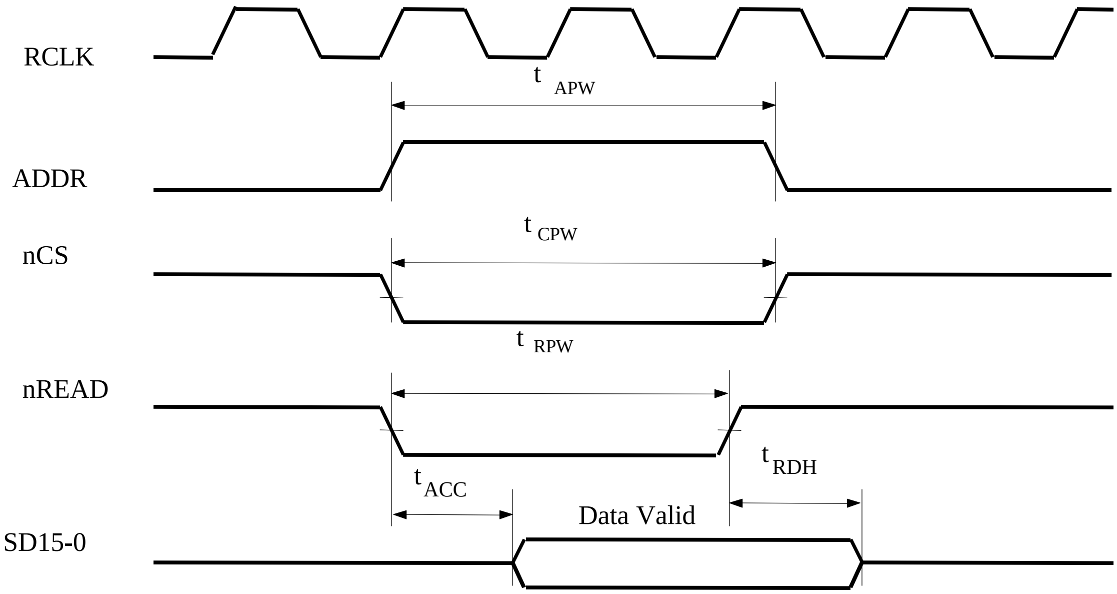
8.8 SL16 Clock Timing Specifications



SL11R CLOCK TIMING

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

8.9 SL16 Port I/O Read Cycle (Non-DMA)

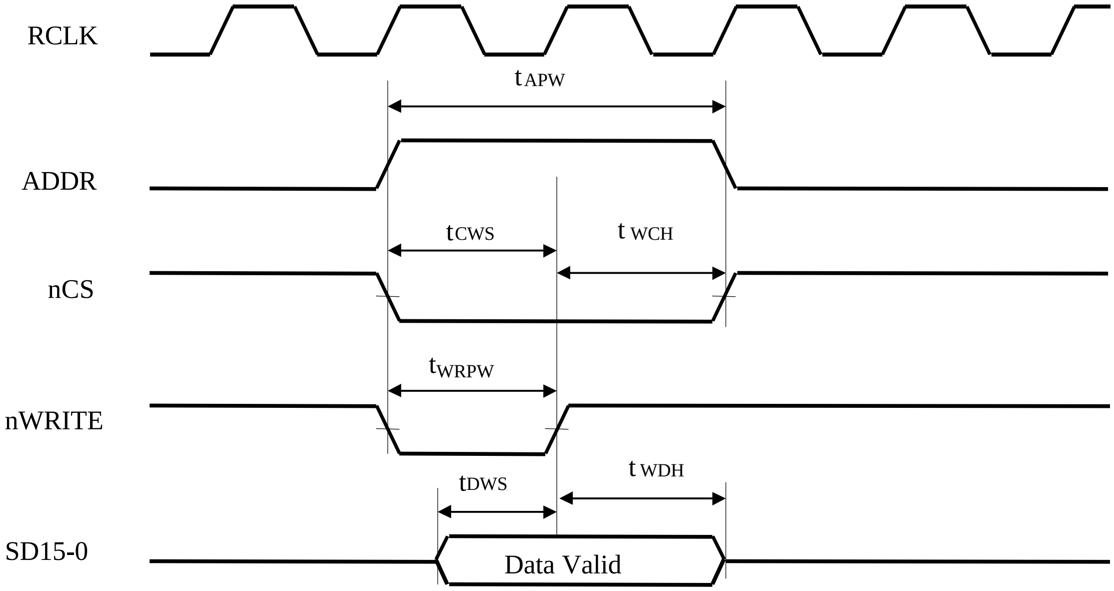


Symbol	**Parameter**	**Min**	**Typical**	**Max**
t_{APW}	ADDR pulse width	30 ns		
t_{CPW}	nCS pulse width	30 ns		
t_{RPW}	Read pulse width	30 ns		
t_{ACC}	Read access time			25 ns
t_{RDH}	Read high to data hold			10 ns

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

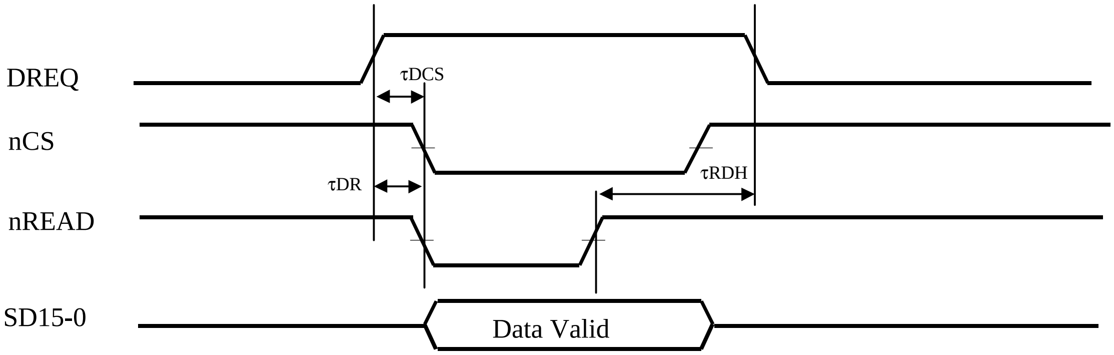
8.10 SL16 Port I/O Write Cycle (Non-DMA)

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



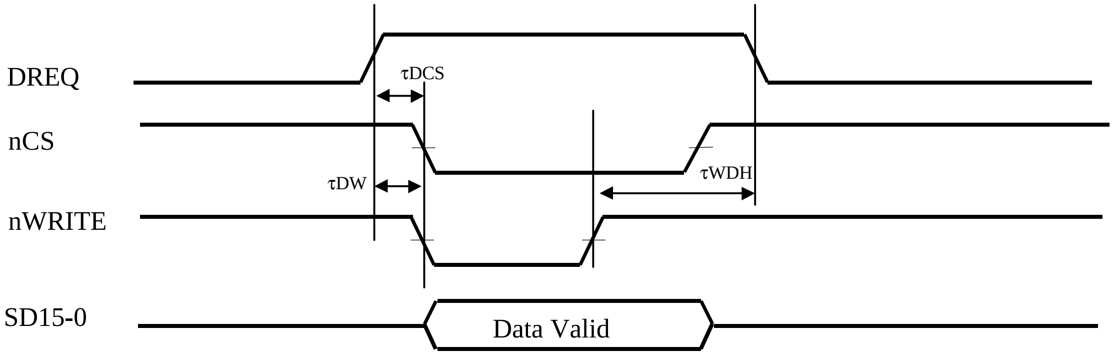
Symbol	Parameter	Min	Typical	Max
t_{APW}	ADDR pulse width	20 ns		
t_{CWS}	nCS low to write high setup	10 ns		
t_{WCH}	Write high to CS high hold	5 ns		
t_{WRPW}	Write pulse width	10 ns		
t_{DWS}	Data setup to write high setup	10 ns		
t_{WDH}	Write high to data hold	5 ns		

8.11 SL16 DMA Read Cycle



Symbol	Parameter	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 SL16 DMA Write Cycle



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

8.13 SL16 Signals Name convention

Doc. Signal Name	SL16 Pin Name	Doc. Signal Name	SL16 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		

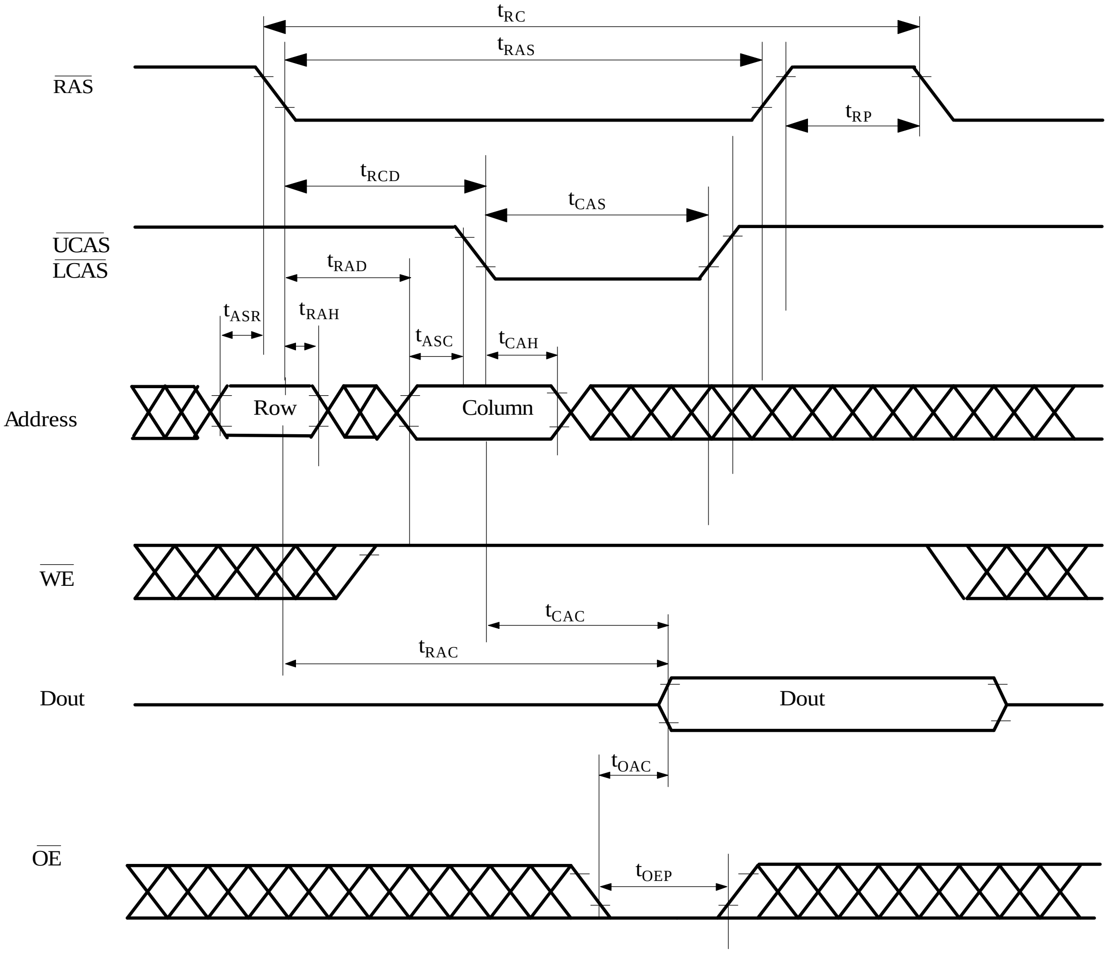
8.14 SL16 DRAM Timing

This timing is based on the SL16 Processor Clock (PCLK) = (2/3) of RCLK = 32MHz (see the register 0xC006 for information about PCLK).

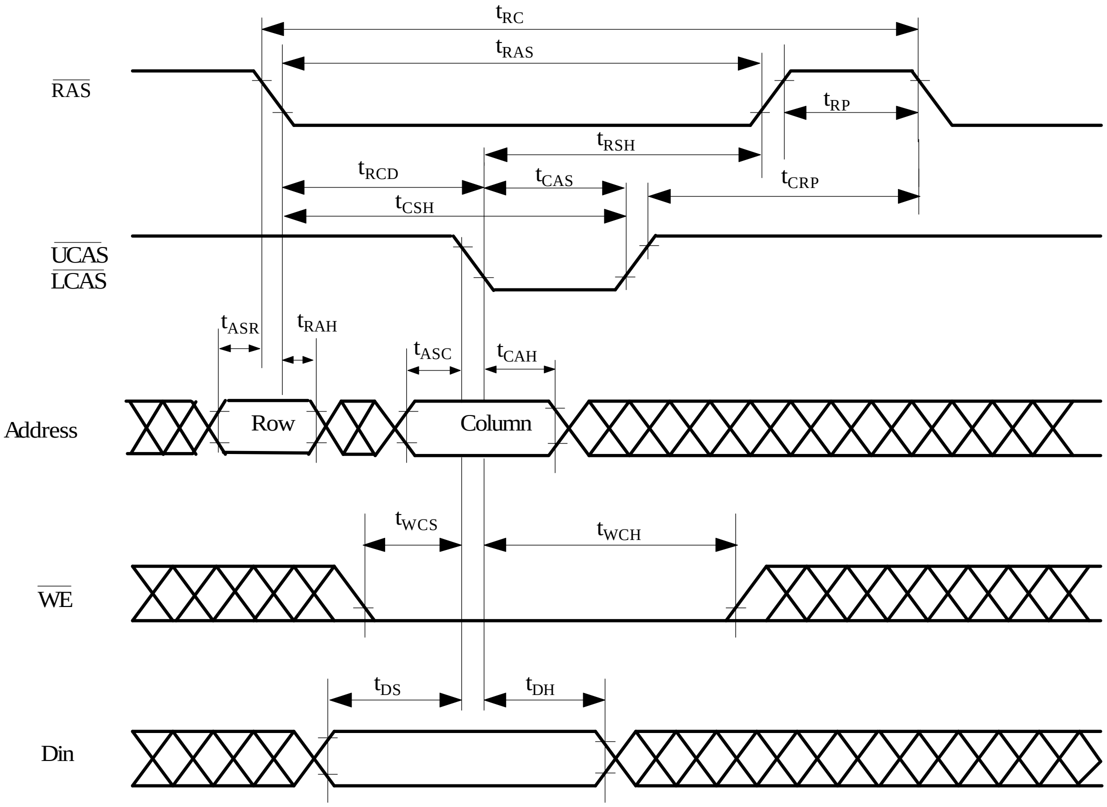
Symbol	Parameter	Min	Typical	Max
t_{RAS}	/RAS pulse width	80ns		
t_{CAS}	/CAS pulse width	20ns		
t_{RP}	/RAS precharge time	60ns		
t_{RCD}	/RAS to /CAS delay time	64ns		
t_{ASR}	Row address setup time	20ns		
t_{RAH}	Row address hold time	36ns		
t_{ASC}	Column address setup time	20ns		
t_{CAH}	Column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

Note: This timing is base on the EDO DRAM timing 16Mx16 devices. When setup the SL16 processor for the higher speed (i.e. 48MHz clock), then the faster parts (i.e. 50ns or 60ns) should be used.

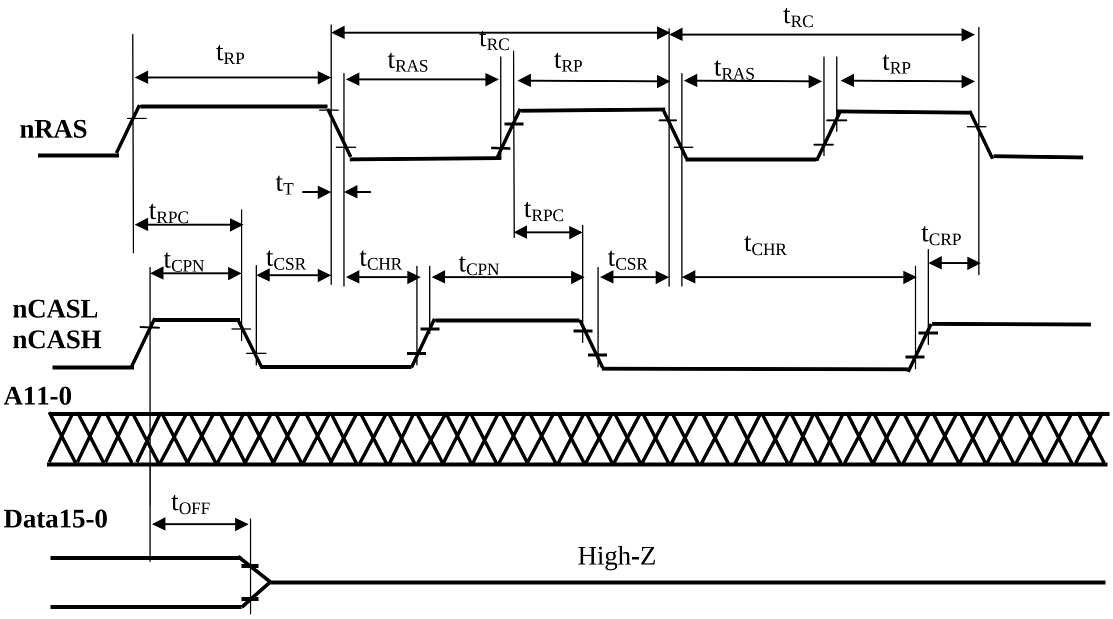
8.15 SL16 DRAM Read Cycle



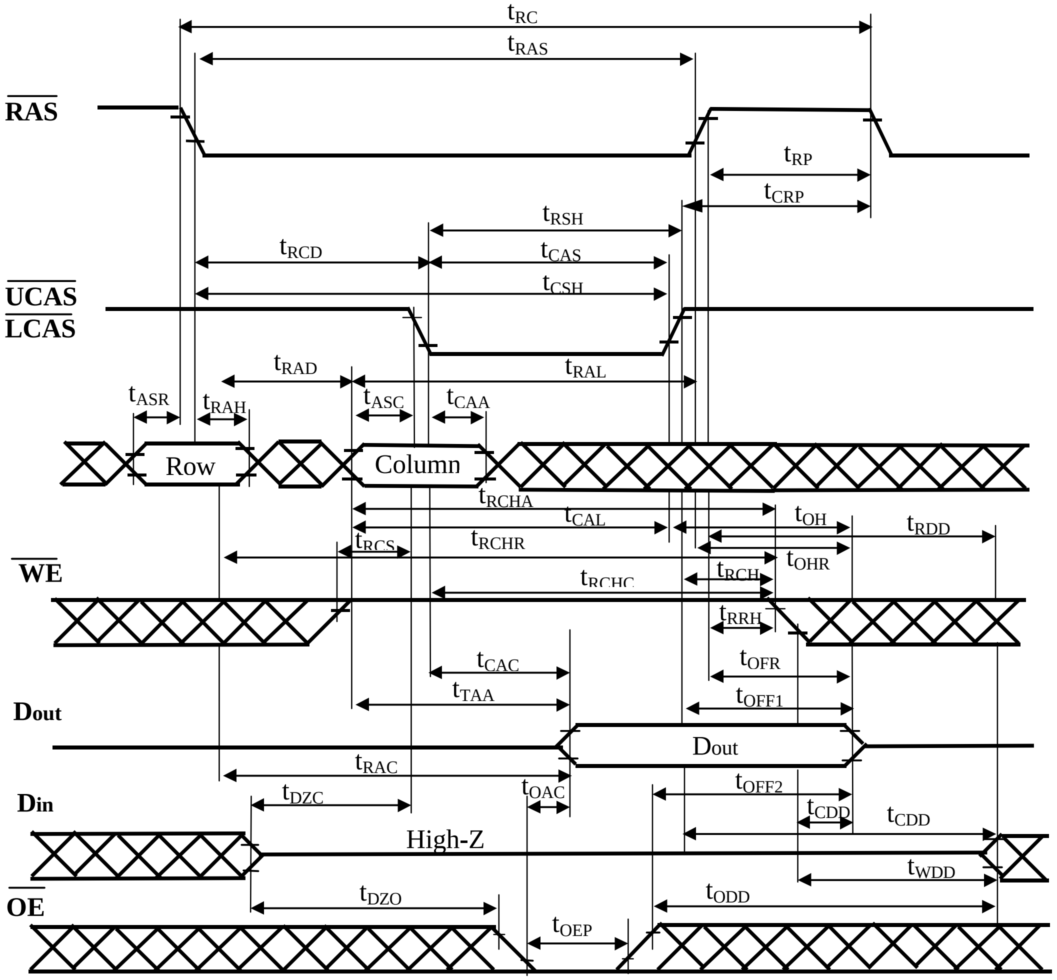
8.16 SL16 DRAM Write Cycle



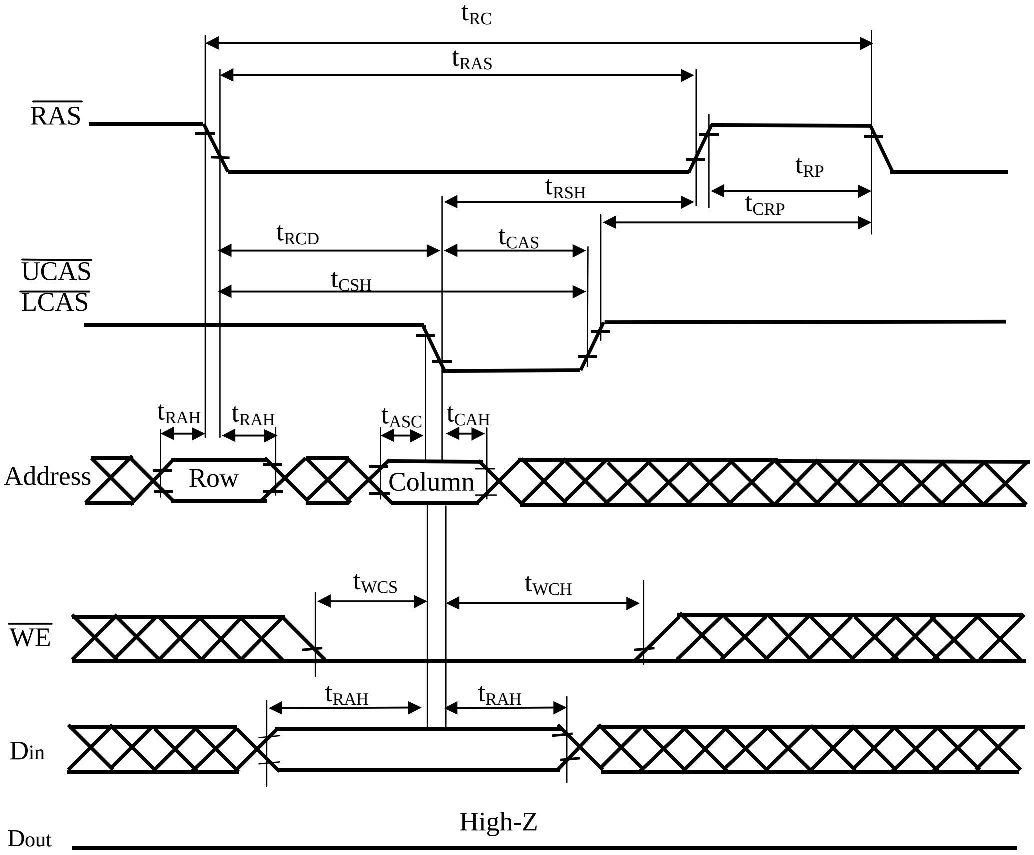
8.17 SL16 CAS-Before-RAS Refresh Cycle



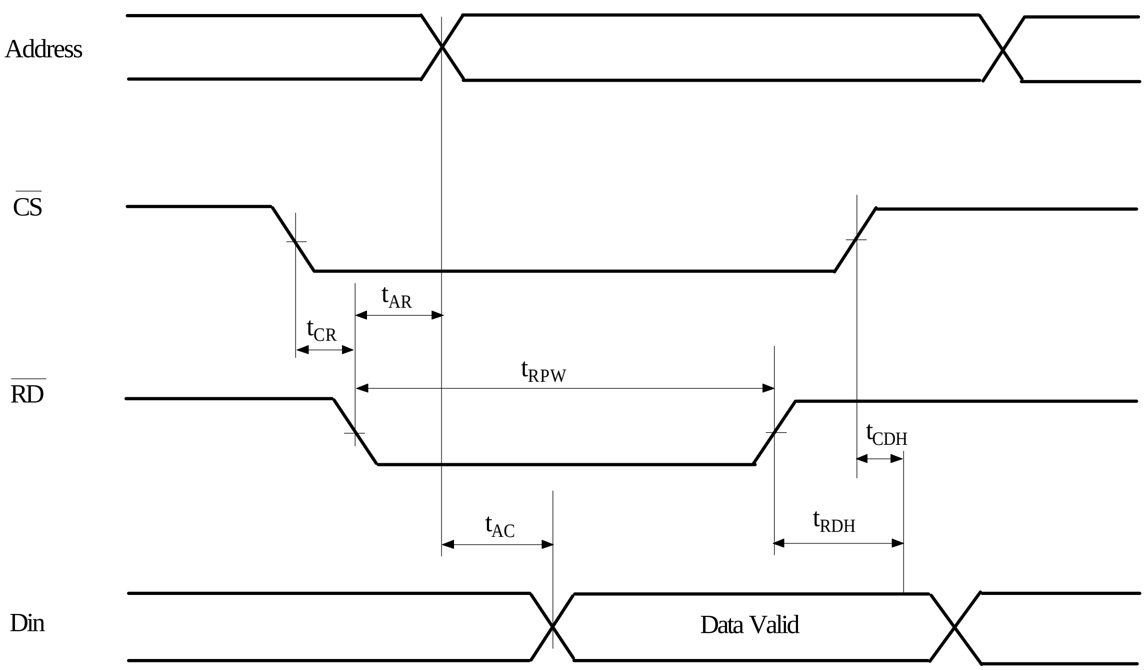
8.18 SL16 DRAM Page Mode Read Cycle



8.19 SL16 DRAM Page Mode Write Cycle



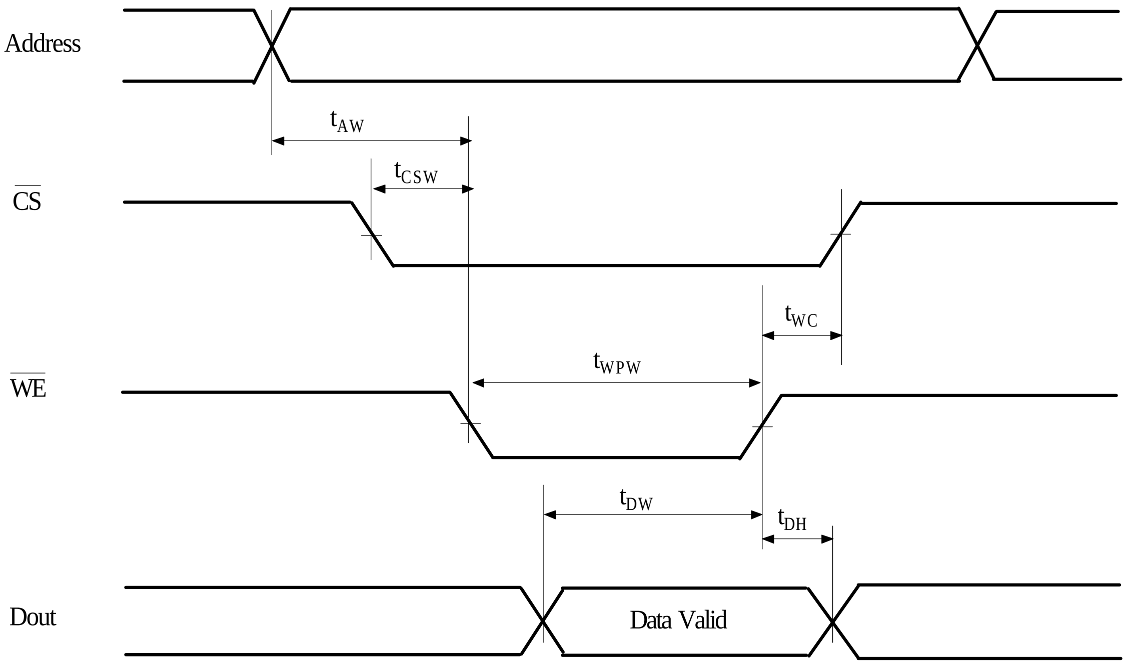
8.20 SL16 SRAM Read Cycle



Symbol	Parameter	Min	Typical	Max
t_{CR}	CS low to RD low	1ns		
t_{RDH}	RD high to data hold			0ns
t_{CDH}	CS high to data hold			0ns
t_{AC} *	Ram access to data valid			12ns

- t_{AC} means at 1 wait state, with PCLK = 2/3 RCLK, the SRAM access time should be at least 12ns. For a 2 wait state, with PCLK = 2/3 RCLK, the SRAM access time should be at least 12 + 31ns = 43ns(see the register 0xC006 for information about PCLK).

8.21 SL16 SRAM Write Cycle

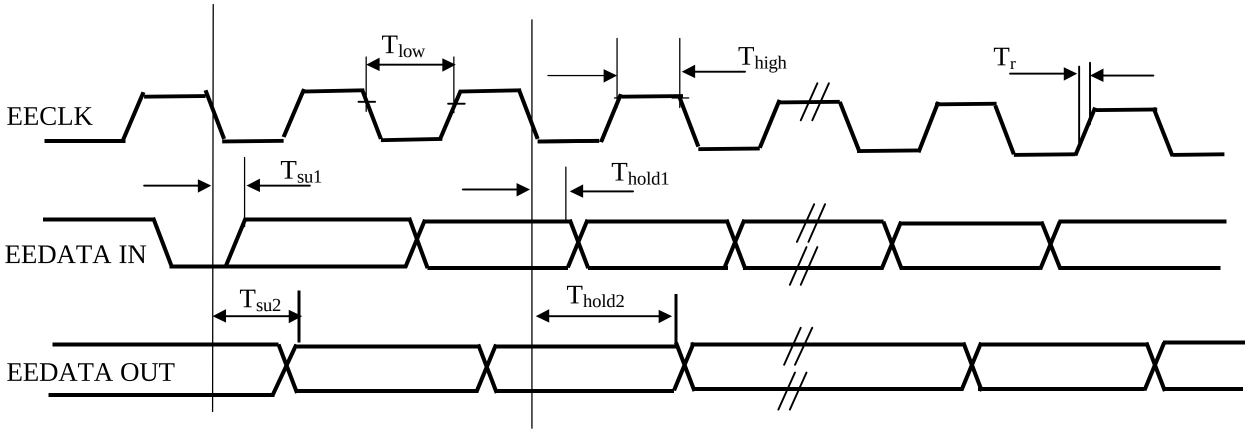


Symbol	Parameter	Min	Typical	Max
t_{AW}	Write address valid to WE low	13ns		
t_{CSW}	CS low to WE low	13ns		
t_{DW}	Data valid to WE high	25ns		
t_{WPW} *	WE pulse width	28ns		
t_{DH}	Data hold from WE high	03ns		
t_{WC}	WE high to CS high	15ns		

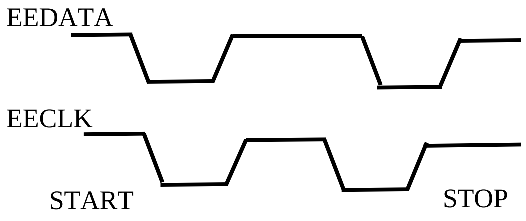
- This is at 1 wait state with PCLK = 2/3 RCLK. For 2-wait states, add 31ns.

8.22 SL16 I2C Serial flash EEPROM timing

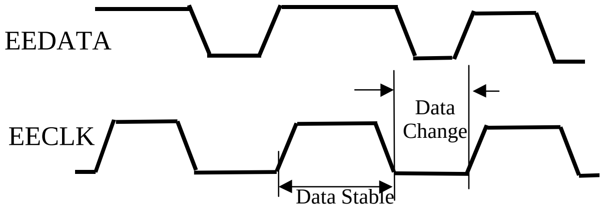
1-EEPROM Bus Timing- Serial I/O



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}	200ns max	
T_{hold1}	0ns	
T_{su2}	4.5 µs min	
T_{hold2}	100ns max	