STANDARD
MICROSYSTEMS
CORPORATION

USB97C102

Multi-Endpoint USB Peripheral Controller with Integrated 5 Port HUB

FEATURES

- High Performance USB Peripheral Controller Engine
  - Integrated USB Transceiver
  - Serial Interface Engine (SIE)
  - 8051 Microcontroller (MCU)
  - Patented Memory Management Unit (MMU)
  - 4 Channel 8237 DMA Controller (ISADMA)
  - 4K Byte On Board USB Packet Buffer
  - Quasi-ISA Peripheral Interface
  - USB Bus Snooping Capabilities
  - GPIOs
- Pin Compatible with SMSC USB97C100
- Complete USB Specification 1.1 Compatibility
  - Isochronous, Bulk, Interrupt, and Control Data Independently Configurable per Endpoint
  - Dynamic Hardware Allocation of - Packet Buffer for Virtual Endpoints
  - Multiple Virtual Endpoints (up to 16 TX, 16 RX Simultaneously)
  - Multiple Alternate Address Filters
  - Dynamic Endpoint Buffer Length Allocation (0-1280 Byte Packets)
- USB Full (12Mbps) and Low Speed Capability
- MMU and SRAM Buffer Allow Buffer Optimization and Maximum Utilization of USB Bandwidth
  - 128 Byte Page Size
  - 10 Pages Maximum per Packet
  - 32 Deep Receive Packet Queue
  - Up to 5 Deep Transmit Packet Queue, per Endpoint
  - Hardware Generated Packet Header Records Each Packet Status Automatically
  - Simultaneous Arbitration Between MCU, SIE, and ISA DMA Accesses
- Extended Power Management
  - Standard 8051 "Stop Clock" Modes
  - Additional USB and ISA Suspend Resume Events
  - Internal 8MHz Ring Oscillator for Immediate Low Power Code Execution
  - 24, 16, 12, 8, 4, and 2 MHz PLL Taps For on the Fly MCU and DMA Clock Switching
  - Independent Clock/Power Management for SIE, MMU, DMA and MCU
- DMA Capability with ISA Memory
  - Four Independent Channels
  - Transfer Between Internal and External Memory
  - Transfer Between I/O and Internal Memory
  - External Bus Master Capable
- Scatter Gather DMA
  - Four Independent Channels
  - Up to 16 Transfers can be Programmed to Occur Consecutively Without MCU Intervention

- External MCU Memory Interface
  - 1M Byte Code and Data Storage via 16K Windows
  - Flash, SRAM, or EPROM
  - Downloadable via USB, Serial Port, or ISA Peripheral
- Quasi-ISA Interface Allows Interface to New and "Legacy" Peripheral Devices
  - 1M ISA Memory Space via 4K MCU Window
  - 64K ISA I/O Space via 256 Byte MCU Window
  - 4 External Interrupt Inputs
  - 4 DMA Channels
  - Variable Cycle Timing
  - 8 Bit Data Path
- 3.3 Volt, Low Power Operation
- 5 Volt Tolerant Operation on I/O Signal Pins
- On Board Crystal Driver Circuit
- 128 Pin QFP Package

GENERAL DESCRIPTION

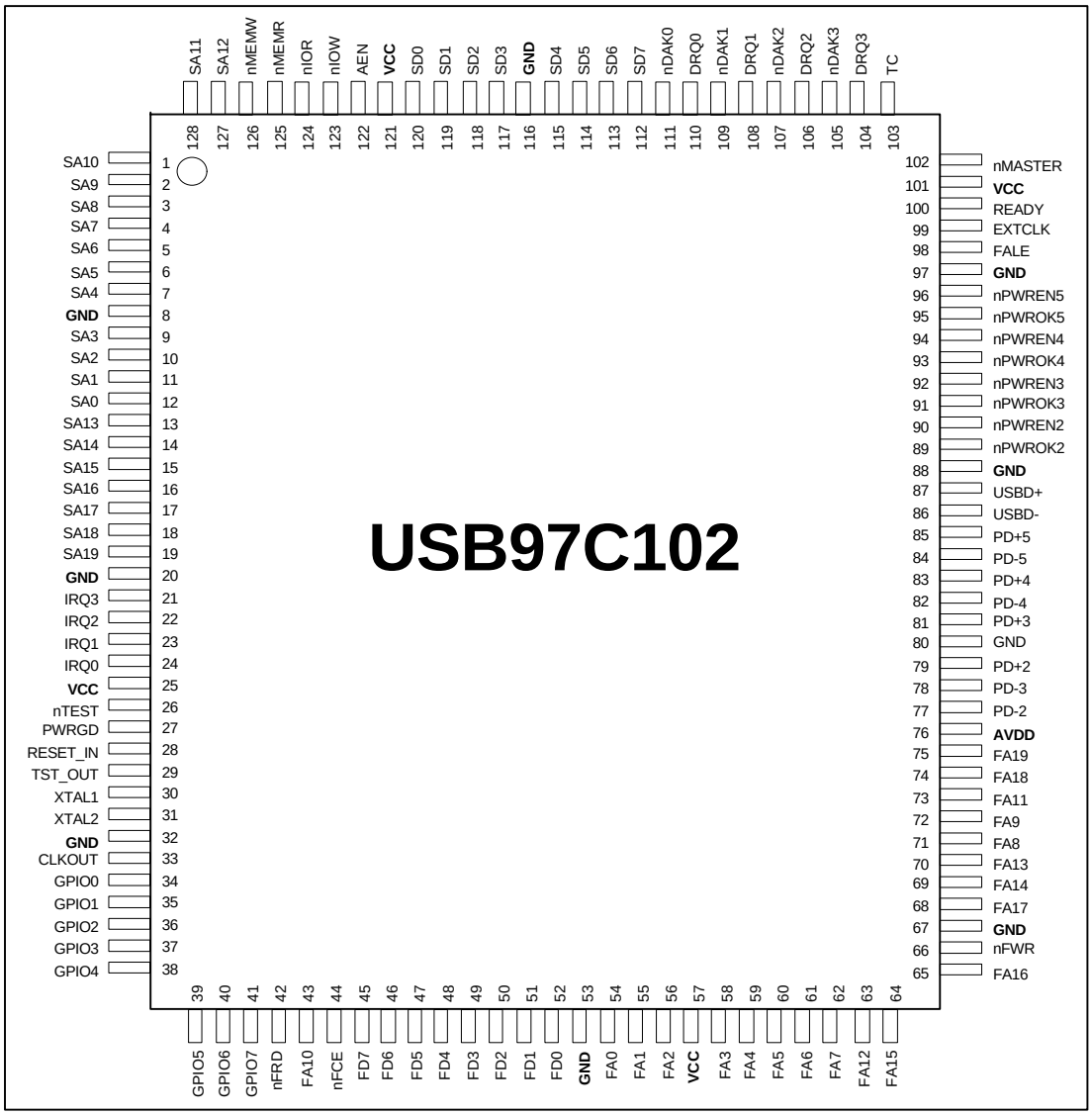
The USB97C102 is a flexible, general purpose USB peripheral interface and controller ideally suited for multiple endpoint applications. The USB97C102 provides an ISA-like bus interface, which will allow virtually any PC peripheral to be placed at the end of a USB connection. Its unique dynamic buffer architecture overcomes the throughput disadvantages of existing fixed FIFO buffer schemes allowing maximum utilization of the USB connection's overall bandwidth. This architecture minimizes the integrated microcontroller's participation in the USB data flow, allowing back-to-back packet transfers to block oriented devices. The efficiency of this architecture allows floppy drives to coexist with other peripherals such as serial and parallel ports on a single USB link.

The USB97C102 allows external program code to be downloaded over the USB to allow easy implementation of varied peripheral USB Device Classes and combinations. This also provides a method for convenient field upgrades and modifications.

TABLE OF CONTENTS

80 Arkay Drive
Hauppauge, NY 11788
(516) 435-6000
FAX (516) 273-3123

PIN CONFIGURATION



DESCRIPTION OF PIN FUNCTIONS

Table 1 - USB97C102 Pin Configuration

QFP PIN NUMBER	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
		ISA INTERFACE	
100	READY	Channel is ready when high. ISA memory or slave devices use this signal to lengthen a bus cycle from the default time. Extending the length of the bus cycle can only be done when the bus cycles are derived from the Internal DMA controller core. 8051 MCU generated Memory or I/O accesses cannot and will not be extended even if READY is asserted low by an external ISA slave device. The external slave device negates this signal after decoding a valid address and sampling the command signals (nIOW, nIOR, nMEMW, and nMEMR). When the slave's access has completed, this signal should be allowed to float high.	IP
104, 106, 108, 110	DRQ[3:0]	DMA Request channels 3-0; active high. These signals are used to request DMA service from the DMA controller. The requesting device must hold the request signal until the DMA controller drives the appropriate DMA acknowledge signal (nDACK[3:0]).	I
105, 107, 109, 111	nDACK [3:0]	DMA Acknowledge channels 3-0; active low. These signals are used to indicate to the DMA requesting device that it has been granted the ISA bus.	O8
103	TC	DMA Terminal Count; active high. This signal is used to indicate that a DMA transfer has completed.	O8
19-13, 127-7, 9-12	SA[19:0]	System Address Bus These signals address memory or I/O devices on the ISA bus.	O8
112-115, 117-120	SD[7:0]	System Data Bus These signals are used to transfer data between system devices.	I/O8
122	AEN	Address Enable This signal indicates address validation to I/O devices. When low this signal indicates that an I/O slave may respond to addresses and I/O commands on the bus. This signal is high during DMA cycles to prevent I/O slaves from interpreting DMA cycles as valid I/O cycles.	O8
123	nIOW	I/O Write; active low. This signal indicates to the addressed ISA I/O slave to latch data from the ISA bus.	O8

QFP PIN NUMBER	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
124	nIOR	I/O Read; active low. This signal indicates to the addressed ISA I/O slave to drive data on the ISA bus.	O8
125	nMEMR	Memory read; active low This signal indicates to the addressed ISA memory slave to drive data on the ISA bus.	O8
126	nMEMW	Memory write; active low This signal indicates to the addressed ISA memory slave to latch data from the ISA bus.	O8
102	nMASTER	External Bus master, active low This signal forces the USB97C102 to immediately tri-state its external bus, even if internal transactions are not complete. All shared ISA signals are tri-stated, except 8237 nDACKs, which can be used in gang mode to provide external bus-master handshaking. This pin must be used with some handshake mechanism to avoid data corruption.	IP
21-24	IRQ[3:0]	Interrupt Request 3-0; active high These signals are driven by ISA devices on the ISA bus to interrupt the 8051.	I
30	XTAL1/ Clock In	24MHz Crystal or clock input. This pin can be connected to one terminal of the crystal or can be connected to an external clock when a crystal is not used.	ICLKx
31	XTAL2	24MHz Crystal This is the other terminal of the crystal.	OCLKx
99	EXTCLK	Alternate clock to 8237 An external clock can be used for the internal 8237. This clock can be used to synchronize the 8237 to other devices.	ICLK
33	CLKOUT	Clock output. This clock frequency is the same as the 8051 running clock. This clock is stopped when the 8051 is stopped. Peripherals should not use this clock when they are expected to run when the 8051 is stopped. This clock can be used to synchronize other devices to the 8051.	O8
USB INTERFACE			
87, 86	USBD- USBD+	USB Upstream Connection signals These are two point-to-point signals and driven differentially.	IOUSB
96, 94, 92, 90.	nPWREN[5:2]	USB Power Enable A low signal on this pin applies power to the associated USB port (port #5 through #2). This output signal is active low.	O24

QFP PIN NUMBER	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
95, 93, 91, 89.	nPWROK[5:2]	USB Over-Current Sense Input to indicate an over-current condition for a bus powered USB device on an external downstream port (port #5 through #2).	I
85, 84, 83, 82, 81, 78, 79, 77.	nPD+[5:2], nPD -[5:2]	USB Downstream Connection Signals These are two point-to-point signals and driven differentially. They are used as standard “Walk Up” USB Port Connections	IOUSB
FLASH INTERFACE			
45-52	FD[7:0]	Flash ROM Data Bus These signals are used to transfer data between 8051 and the external FLASH.	IO8
75, 74, 68, 65, 64, 69, 70, 63, 73, 43, 72, 71, 62-58, 56-54	FA[19:0]	Flash ROM Address Bus These signals address memory locations within the FLASH.	O8
42	NFRD	Flash ROM Read; active low	O8
66	NFWR	Flash ROM Write; active low	O8
44	nFCE	Flash ROM Chip Select; active low	O8
98	FALE	Flash ROM address latch enable	O8
POWER SIGNALS			
25,57,76, 101,121	VCC	+3.3 Volt Power	
8, 20, 32, 53, 67, 80, 88, 97, 116	GND	Ground Reference	
MISCELLANEOUS			
41-34	GPIO[7:0]	General Purpose I/O. These pins can be configured as inputs or outputs under software control.	I/O24
27	PWRGD	Active high input. This signal is used to indicate to that chip that a good power level has been reached. When inactive/low, all pins are Tri-stated except TST_OUT and a POR is generated.	I
28	RESET_IN	Power on reset; active high This signal is used by the system to reset the chip. It also generates an internal POR.	I

QFP PIN NUMBER	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
29	TST_OUT	XNDR Chain output This signal is used for testing the chip via an internal XNDR Chain.	O8
26	nTEST	Test input This signal is a manufacturing test pin. User can pull it high or leave it unconnected.	IP

BUFFER TYPE DESCRIPTIONS

Table 2 - USB97C102 Buffer Type Description

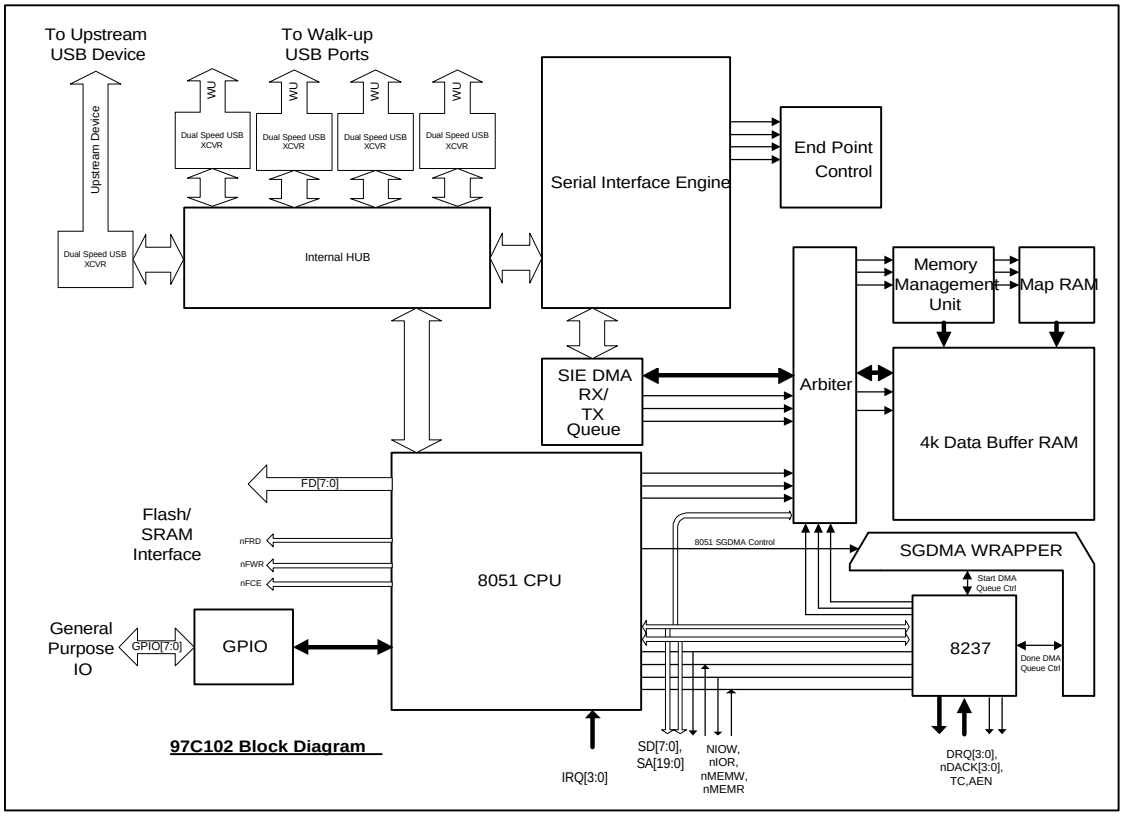
BUFFER	DESCRIPTION
I	Input (no pull-up)
IP	Input 90µA with internal pull-up
O8	Output with 8mA drive
I/O8	Input/output with 8mA drive
I/O16	Input/output with 16mA drive
O24	Output, 24mA sink, 12mA source.
I/ODP24	Input/Output drain , 24mA sink, 12mA source with 90µA pull-up
ICLKx	XTAL clock input
OCLKx	XTAL clock output
ICLK	Clock input (TTL levels)
IOUSB	Defined in USB specification; V1.1

CODE DEBUGGER INTERFACE

This interface is made available by driving NTEST=0 and TSTOUT=0 (In this mode, TSTOUT is an input of the chip). In this mode, the pin functions are defined as follows:

QFP PIN NUMBER	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
64, 69, 70, 63, 73, 43, 72, 71, 62-58, 56-54	FA[15:0]	8051 Address Bus	I
42	NFRD	Data Read Strobe; active low	I
66	NFWR	Data Write; active low	I
112-115, 117-120	FD[7:0]	Data Bus	I/O8
44	nFCE	8051 T1IN timer signal	O8
75	FA19	8051 T0IN timer signal	O8
74	FA18	8051 WAKE interrupt signal	O8
68	FA17	8051 INT1 interrupt signal	O8
65	FA16	8051 INT0 interrupt signal	O8

USB97C102 BLOCK DIAGRAM



FUNCTIONAL DESCRIPTION

The USB97C102 incorporates a USB Serial Interface Engine (SIE), 8051 Microcontroller Unit (MCU), Serial Interface Engine DMA (SIEDMA), a programmable 8237 ISA bus DMA controller (ISADMA), 4K bytes of SRAM for data stream buffering, and a patented MMU (Memory Management Unit) to dynamically manage buffer allocation. The semi-automatic nature of the SIEDMA, ISADMA, and MMU blocks frees the MCU to provide enumeration, protocol and power management. A bus arbiter integrated into the MMU assures that transparent access between the SIEDMA, ISADMA, and MCU to the SRAM occurs.

Serial Interface Engine (SIE)

The SIE is a USB low-level protocol interpreter. The SIE controls the USB bus protocol, packet generation/extraction, parallel-to-serial/serial-to-parallel conversion, CRC coding/decoding, bit stuffing, and NRZI coding/decoding.

The SIE can be dynamically configured as having any combination of 0-16 transmit, and 0-16 receive endpoints, for up to 4 independent addresses. There are 3 alternate and one local address. The alternate addresses, for example, can be used for Hub addresses. The SIE can also "Receive All Addresses" for bus snooping.

Micro Controller Unit (MCU)

The 8051 embedded controller is a static CMOS MCU which is fully software compatible with the industry standard Intel 80C51 micro-controller. All internal registers of the USB97C102 blocks are mapped into the external memory space of the MCU.

A detailed description of the microcontroller's internal registers and instruction set can be found in the "USB97C102 Programmer's Reference Guide".

SIEDMA

This is a simplified DMA controller, which automatically transfers data between SIE and SRAM via MMU control. The SIEDMA appends a status header containing frame number, endpoint, and byte count to each incoming packet before notifying the MCU of its arrival. This block's operation is transparent to the firmware.

Memory Management Unit (MMU) Register Description

This patented MMU consists of a 4k buffer RAM which is allocated in 32 pages of 128 bytes. Packets can be allocated with up to 10 pages each (1280 bytes). The buffer can therefore concurrently hold up to 32 packets with a 64 byte payload. For isochronous pipes, it can hold 3 packets with a 1023 byte payload each, and still have room for two more 64 byte packets.

This block supports 16 independent transmit FIFO queues (one for each endpoint), and a single receive queue. Each endpoint can have up to five transmit packets queued. The receive queue can accept 32 packets of any size combination before forcing the host to back off.

The arbiter makes the single-ported buffer RAM appear to be simultaneously available to the MCU, the four channels of the ISADMA, and the SIEDMA for receiving and transmitting packets.

ISADMA

This is an industry standard 8237 DMA controller to transfer data between the ISA bus and the SRAM under MMU control. This DMA contains status and control registers which can be accessed and programmed by the 8051 controller. The 8237 can run at 2, 4, or 8 MHz internally, or via an external clock to synchronize it with another source.

SGDMA

A four channel Scatter-Gather DMA will run the 8237 DMA controller once the MCU indicates which packets to transfer. The SGDMA performs scattering/gathering operations from the MMU to/from the external ISA memory. It also allows ISA device to/from MMU transfer.

Applications

The USB97C102 enables entirely new I/O applications, as well as new form factors for existing Legacy I/O applications. PC98 compliance encourages the elimination of DMA, IRQ and addressing conflicts via total on-board ISA elimination. With the USB97C102, the ISA bus can be eliminated from motherboards without sacrificing the huge infrastructure of Legacy I/O ports. By moving these devices to the flexible USB bus, new form factors such as monitor peripheral clusters are also possible (mouse, keyboard, serial, parallel ports in a USB connected monitor). PC system designers are no longer constrained by the physical borders of the motherboard. The USB97C102 is ideal for USB peripherals which require considerable bandwidth, such as floppy drives, audio, IR, etc. The following block diagrams illustrate these applications.

TYPICAL PC MOTHERBOARD APPLICATION

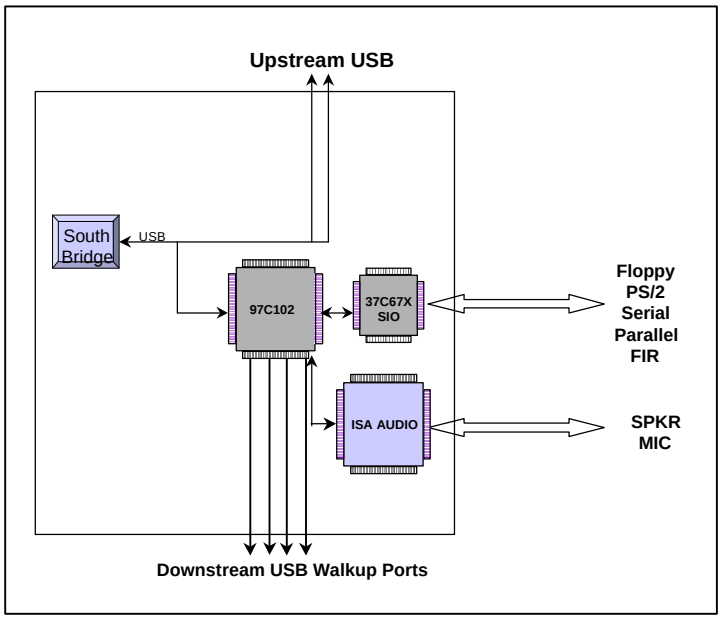


FIGURE 2 – USB97C102 CONFIGURED IN A PC MOTHERBOARD

TYPICAL MONITOR APPLICATION

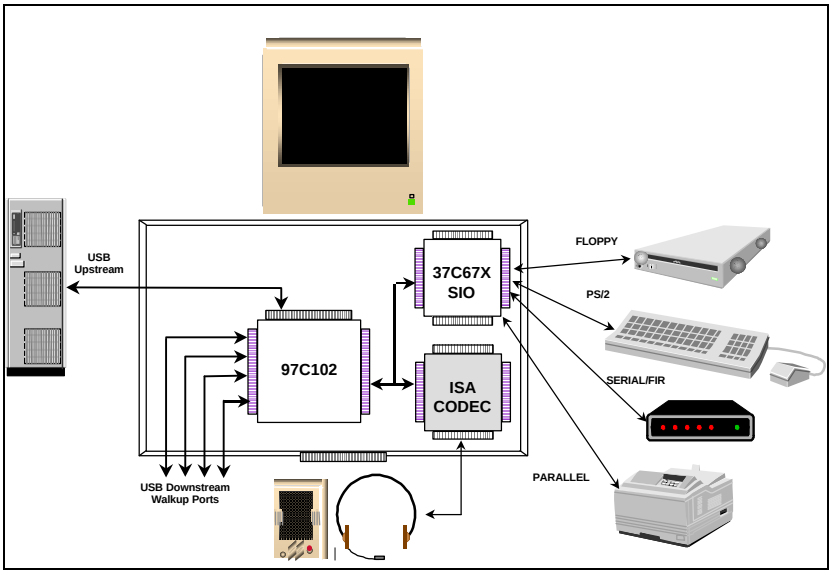


FIGURE 3 – USB97C102 CONFIGURED FOR MONITOR, HUB, AND PERIPHAL CONSTELLATION

TYPICAL FLOPPY DRIVE APPLICATION

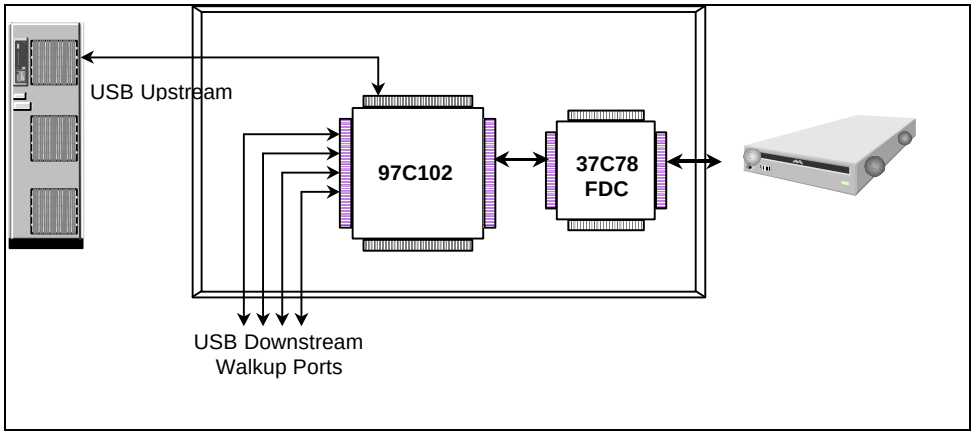


FIGURE 4 – USB97C102 CONFIGURED FOR FLOPPY DRIVE APPLICATION AND WALKUP PORTS

TYPICAL SIGNAL CONNECTIONS

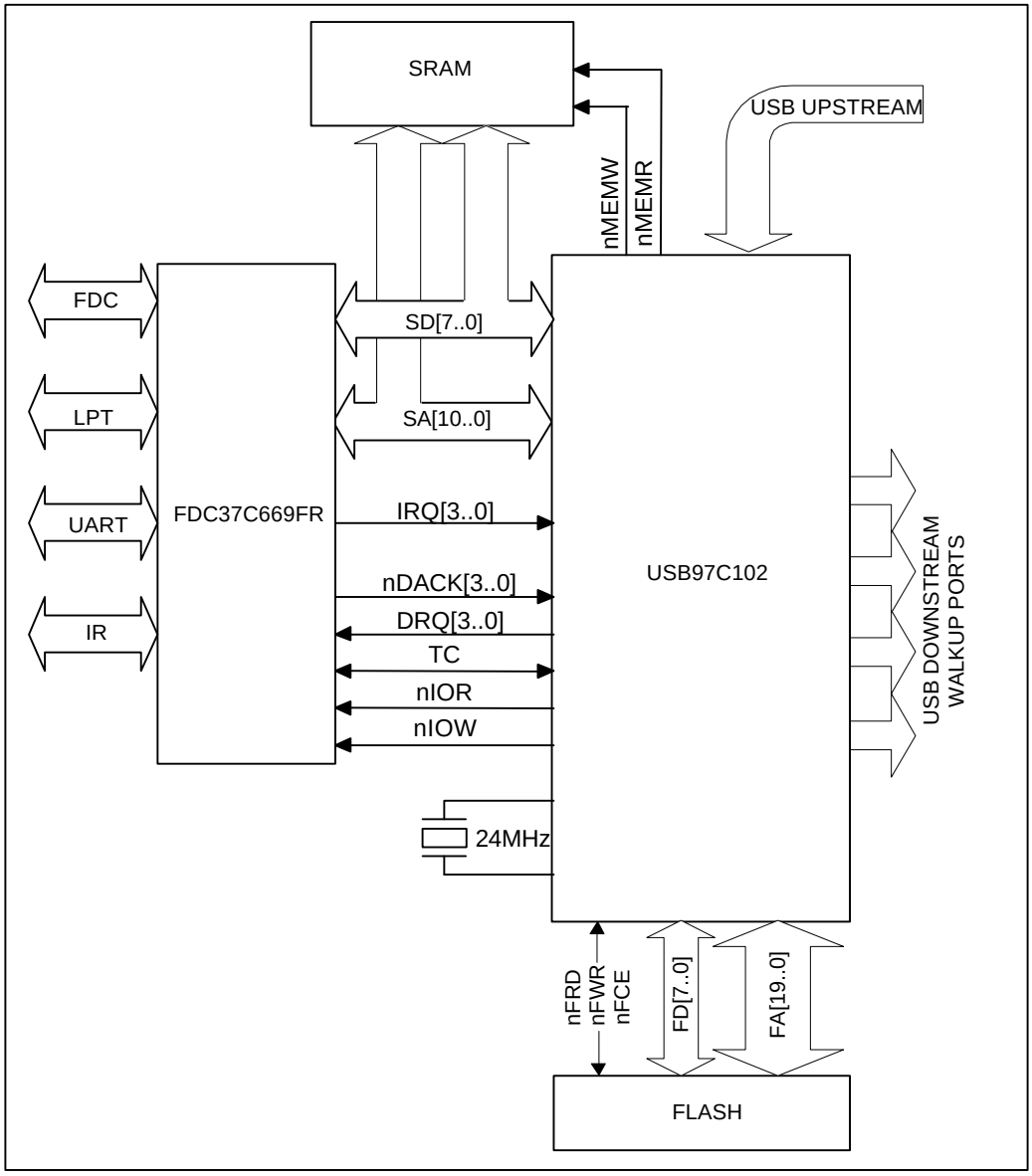


FIGURE 5 – USB97C102 CONFIGURED WITH FDC CONTROLLER AND WALKUP PORTS

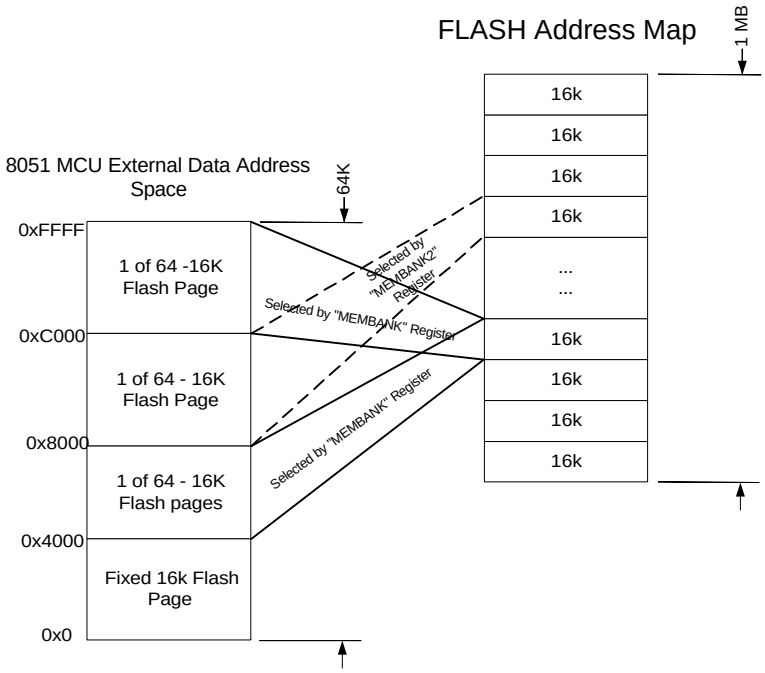
MCU Memory Map

The 64K memory map is as follows from the 8051's viewpoint:

Code Space

Table 3 - MCU Code Memory Map

8051 ADDRESS	CODE SPACE	ACCESS
0xC000-0xFFFF	Movable 16k FLASH page 1 of 64 16k pages in External FLASH (0x00000-0xFFFFF) selected by MEM_BANK Register Default: 0x04000-0x07FFF FLASH	External FLASH
0x8000-0xBFFF	Movable 16k FLASH page 1 of 64 16k pages in External FLASH (0x00000-0xFFFFF) selected by MEM_BANK2 Register Default: 0x0660-0x03FFF FLASH	External FLASH
0x4000-0x7FFF	Movable 16k FLASH page 1 of 64 16k pages in External FLASH (0x00000-0xFFFFF) selected by MEM_BANK Register Default: 0x04000-0x07FFFFLASH	External FLASH
0x0000-0x3FFF	Fixed 16k FLASH Page 0x00000-0x03FFF FLASH	External FLASH



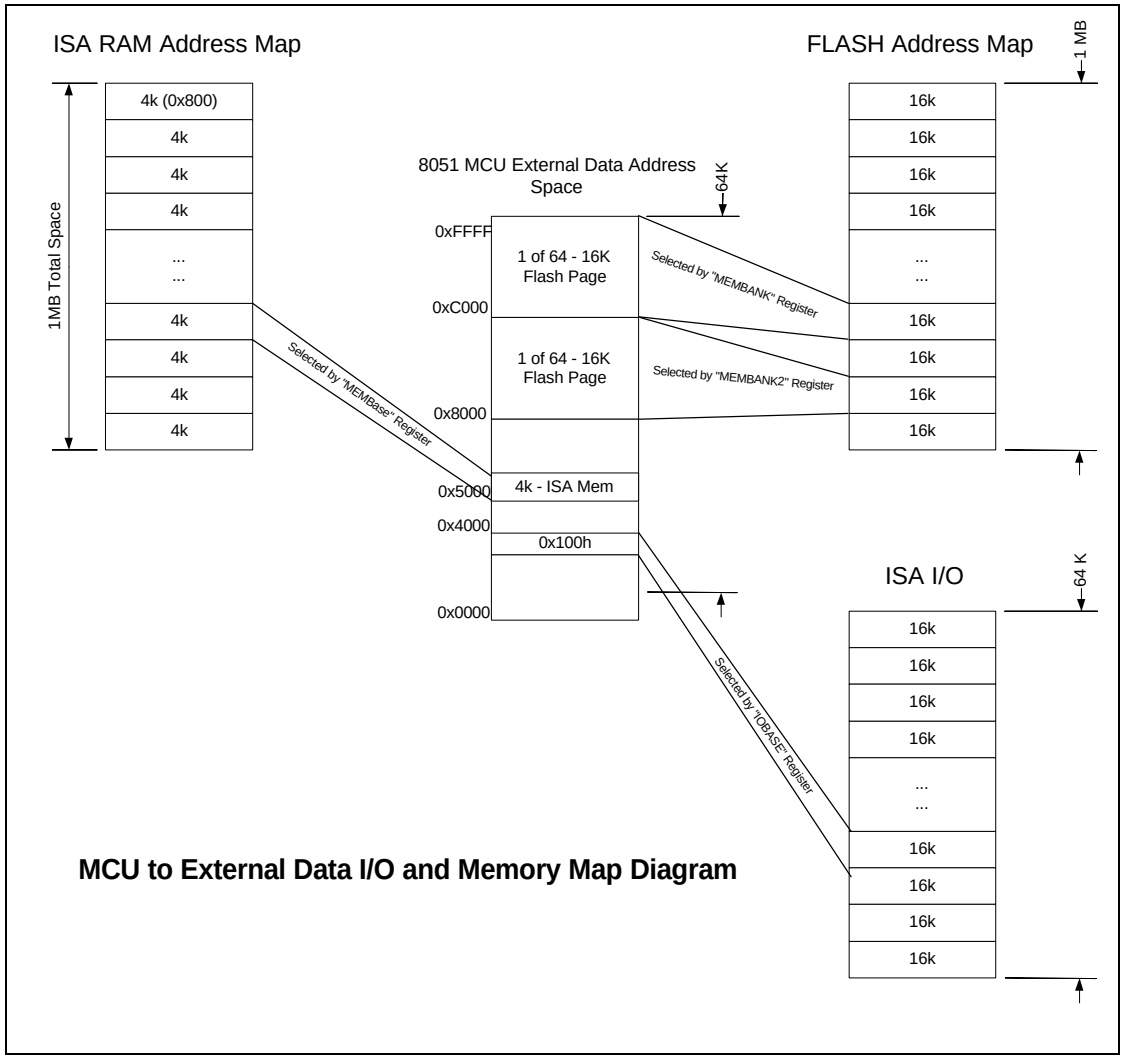
MCU to External Code Space Map Diagram

Data Space

Table 4 - MCU Data Memory Map

8051 ADDRESS	DATA SPACE	ACCESS
0xC000-0xFFFF	Movable 16k FLASH page 1 of 64 16k pages in External FLASH (0x00000-0xFFFFF) selected by MEM_BANK Register Default: 0x04000-0x07FFF FLASH	External FLASH
0x8000-0xBFFF	Movable 16k FLASH page 1 of 64 16k pages in External FLASH (0x00000-0xFFFFF) selected by MEM_BANK2 Register Default: 0x004000-0x037FFF FLASH	External FLASH
0x7000-0x7FFF	0x7F80-0x7F9F SIE Reg 0x7F70-0x7F7F ISA Reg 0x7F50-0x7F6F MMU Reg 0x7F20-0x7F2F Power Reg 0x7F10-0x7F1F Configuration Reg 0x7F00-0x7F0F Runtime Reg Note 1.	Internal
0x6000-0x6FFF	0x6000 MMU Data Register	Internal
0x5000-0x5FFF	0x5000-0x5FFF ISA MEMORY Window	ISA
0x4000-0x4FFF	0x4000-0x40FF ISA I/O Window	ISA
0x3000-0x3FFF		Not used
0x2000-0x2FFF		Not used
0x1000-0x1FFF		Not used
0x0000-0x00FF	Registers and SFR's	Internal

Note 1: The MCU, MMU, and SIE block registers are external to the 8051, but internal to the USB97C102. These addresses will appear on the FLASH bus, but the read and write strobes will be inhibited.



MCU to External Data I/O and Memory Map Diagram

ISADMA Memory Map

The Internal Memory buffer is virtualized into the 8237's 64K address map as 32 independent 1k blocks. After the MMU has allocated a given packet size for a specific PNR, the MMU will make that packet appear to the 8237 as a contiguous block of data in the address ranges depicted in table 5.

Table 5 - ISADMA Memory Map

8237 MEMORY ADDRESS	DESCRIPTION
0x8000-0xFFFF	32 blocks of 1K Window to Packet
0x0000-0x7FFF	32K Window to External ISA RAM

MCU Block Register Summary

Table 6 - MCU Block Register Summary

ADDRESS	NAME	R/W	DESCRIPTION	PAGE
RUNTIME REGISTERS				
7F00	ISR_0	R/W	INT0 Source Register	28
7F01	IMR_0	R/W	INT0 Mask Register	29
7F02	ISR_1	R/W	INT1 Source Register	30
7F03	IMR_1	R/W	INT1 Mask Register	31
7F06	DEV_REV	R	Device Revision Register	31
7F07	DEV_ID	R	Device ID Register	31
UTILITY REGISTERS				
7F18	GPIOA_DIR	R/W	GPIO Configuration Register	34
7F19	GPIOA_OUT	R/W	GPIO Data Output Register	35
7F1A	GPIOA_IN	R	GPIO Data Input Register	35
7F1B	UTIL_CONFIG	R/W	Miscellaneous Configuration Register	36
POWER MANAGEMENT REGISTERS				
7F27	CLOCK_SEL	R/W	8051 and 8237 Clock Select Register	38
7F28	MEM_BANK2	R/W	Flash Bank Select 2	39
7F29	MEM_BANK	R/W	Flash Bank Select	39
7F2A	WU_SRC_1	R/W	Wakeup Source	39
7F2B	WU_MSK_1	R/W	Wakeup Mask	40
7F2C	WU_SRC_2	R/W	Wakeup Source	40
7F2D	WU_MSK_2	R/W	Wakeup Mask	41
ISA BUS CONTROL REGISTERS				
7F10	GP1Data	R/W	GP FIFO Data Port #1	32
7F11	GP1Status	R	GP FIFO status Port #1	33
7F12	GP2Data	R/W	GP FIFO Data Port #2	32
7F13	GP2Status	R	GP FIFO status Port #2	33
7F14	GP3Data	R/W	GP FIFO Data Port #3	32
7F15	GP3Status	R	GP FIFO status Port #3	33
7F16	GP4Data	R/W	GP FIFO Data Port #4	32
7F17	GP4Status	R	GP FIFO status Port #4	34
7F70	BUS_REQ	R/W	ISA Bus Request Register	42
7F71	IOBASE	R/W	8051 ISA I/O Window Base Register	44
7F72	MEMBASE	R/W	8051 ISA Memory Window Base Register	45
7F73	BUS_STAT	R	ISADMA Request Status	43
7F74	BUS_MASK	R/W	ISADMA Request Interrupt Mask	44
7F7E	MCU_TEST2	N/A	Reserved for Test	
7F7F	MCU_TEST1	N/A	Reserved for Test	

SGDMA Block Register Summary

Table 7 – SGDMA Block Register Summary

ADDRESS	NAME	R/W	DESCRIPTION	PAGE
CHANNEL 0 REGISTERS				
7FB0	SGDMA_START_FIFO0	R/W	Packet Number Start FIFO Register	56
7FB1	SGDMA_DONE_FIFO0	R/W	Packet Number Done FIFO Register	56
7FB2	SGDMA_ADHI0	R/W	ISA Address High Byte Register	56
7FB3	SGDMA_ADLO0	R/W	ISA Address Low Byte Register	57
7FB4	SGDMA_SIZEHI0	R/W	Transfer Size High Register	57
7FB5	SGDMA_SIZELO0	R/W	Transfer Size Low Register	57
7FB6	SGDMA_TOTAL_PKTS0	R	Total Packets in Channel Register	58
7FB7	SGDMA_DONE_PKTS0	R	Total Packets in Done FIFO Register	58
7FB8	SGDMA_STS0	R	Status Register	58
7FB9	SGDMA_CMD0	R/W	Command Register	60
CHANNEL 1 REGISTERS				
7FBA	SGDMA_START_FIFO1	R/W	Packet Number Start FIFO Register	56
7FBB	SGDMA_DONE_FIFO1	R/W	Packet Number Done FIFO Register	56
7FBC	SGDMA_ADHI1	R/W	ISA Address High Byte Register	56
7FBD	SGDMA_ADLO1	R/W	ISA Address Low Byte Register	57
7FBE	SGDMA_SIZEHI1	R/W	Transfer Size High Register	57
7FBF	SGDMA_SIZELO1	R/W	Transfer Size Low Register	57
7FC0	SGDMA_TOTAL_PKTS1	R	Total Packets in Channel Register	58
7FC1	SGDMA_DONE_PKTS1	R	Total Packets in Done FIFO Register	58
7FC2	SGDMA_STS1	R	Status Register	58
7FC3	SGDMA_CMD1	R/W	Command Register	60
CHANNEL 2 REGISTERS				
7FC4	SGDMA_START_FIFO2	R/W	Packet Number Start FIFO Register	56
7FC5	SGDMA_DONE_FIFO2	R/W	Packet Number Done FIFO Register	56
7FC6	SGDMA_ADHI2	R/W	ISA Address High Byte Register	56
7FC7	SGDMA_ADLO2	R/W	ISA Address Low Byte Register	57
7FC8	SGDMA_SIZEHI2	R/W	Transfer Size High Register	57
7FC9	SGDMA_SIZELO2	R/W	Transfer Size Low Register	57
7FCA	SGDMA_TOTAL_PKTS2	R	Total Packets in Channel Register	58
7FCB	SGDMA_DONE_PKTS2	R	Total Packets in Done FIFO Register	58
7FCC	SGDMA_STS2	R	Status Register	58
7FCD	SGDMA_CMD2	R/W	Command Register	60
CHANNEL 3 REGISTERS				
7FCE	SGDMA_START_FIFO3	R/W	Packet Number Start FIFO Register	56
7FCF	SGDMA_DONE_FIFO3	R/W	Packet Number Done FIFO Register	56
7FD0	SGDMA_ADHI3	R/W	ISA Address High Byte Register	56
7FD1	SGDMA_ADLO3	R/W	ISA Address Low Byte Register	57

ADDRESS	NAME	R/W	DESCRIPTION	PAGE
7FD2	SGDMA_SIZEHI3	R/W	Transfer Size High Register	57
7FD3	SGDMA_SIZELO3	R/W	Transfer Size Low Register	57
7FD4	SGDMA_TOTAL_PKTS3	R	Total Packets in Channel Register	58
7FD5	SGDMA_DONE_PKTS3	R	Total Packets in Done FIFO Register	58
7FD6	SGDMA_STS3	R	Status Register	58
7FD7	SGDMA_CMD3	R/W	Command Register	60
PIO REGISTERS				
7FD8	PIO_ADHI	R/W	Upper Byte of the ISA Address	63
7FD9	PIO_ADMID	R/W	Middle Byte of the ISA Address	63
7FDA	PIO_ADLO	R/W	Lower Byte of the ISA Address	63
7FDB	PIO_DATA	R/W	PIO Data Register	63
7FDC	PIO_CSR	R/W	PIO Command Register	64

MMU Block Register Summary

Table 8 - MMU Block Register Summary

ADDRESS	NAME	R/W	DESCRIPTION	PAGE
MMU REGISTERS				
6000	MMU_DATA	R/W	8051-MMU Data Window Register FIFO	65
7F40	TX_FIFO0	R	TX_FIFO Counter 0	66
7F41	TX_FIFO1	R	TX_FIFO Counter 1	66
7F42	TX_FIFO2	R	TX_FIFO Counter 2	66
7F43	TX_FIFO3	R	TX_FIFO Counter 3	66
7F44	TX_FIFO4	R	TX_FIFO Counter 4	66
7F45	TX_FIFO5	R	TX_FIFO Counter 5	66
7F46	TX_FIFO6	R	TX_FIFO Counter 6	66
7F47	TX_FIFO7	R	TX_FIFO Counter 7	66
7F48	TX_FIFO8	R	TX_FIFO Counter 8	66
7F49	TX_FIFO9	R	TX_FIFO Counter 9	66
7F4A	TX_FIFOA	R	TX_FIFO Counter A	66
7F4B	TX_FIFOB	R	TX_FIFO Counter B	66
7F4C	TX_FIFOC	R	TX_FIFO Counter C	66
7F4D	TX_FIFOD	R	TX_FIFO Counter D	66
7F4E	TX_FIFOE	R	TX_FIFO Counter E	66
7F4F	TX_FIFOF	R	TX_FIFO Counter F	66
7F50	PRL	R/W	8051-MMU Pointer Register (Low)	66
7F51	PRH	R/W	8051-MMU Pointer Register (High) & R/W	66
7F52	MMUTX_SEL	R/W	8051-MMU TX FIFO Select for Commands	67
7F53	MMUCR	W	8051-MMU Command Register	67
7F54	ARR	R	8051-MMU Allocation Result Register	68
7F55	PNR	R/W	8051-MMU Packet Number Register	68
7F56	PAGS_FREE	R/W	Pages Free In the MMU	69
7F57	TX_MGMT	R	TX Management Register 2	70
7F58	RXFIFO	R	RX Packet FIFO Register (All EPs)	70
7F59	POP_TX	R	POP TX FIFO	71
7F60	TXSTAT_A	R	TX Packet FIFO Status Register (EP0-3)	72
7F61	TXSTAT_B	R	TX Packet FIFO Status Register (EP4-7)	73
7F62	TXSTAT_C	R	TX Packet FIFO Status Register (EP8-11)	74
7F63	TXSTAT_D	R	TX Packet FIFO Status Register (EP12-15)	75
7F64	MMU_TESTx	N/A	Reserved for Test	
7F65	MMU_TESTx	N/A	Reserved for Test	
7F66	MMU_TESTx	N/A	Reserved for Test	
7F67	TX_MGMT	R/W	TX Management Register 1	76
7F6E	MMU_TESTx	N/A	Reserved for Test	
7F6F	MMU_TESTx	N/A	Reserved for Test	

SIE Block Register Summary

Table 9 - SIE Block Register Summary

ADDRESS	NAME	R/W	DESCRIPTION	PAGE
SIE Control Registers				
7F80	EP_CTRL0	R/W	Endpoint 0 Control Register	78
7F81	EP_CTRL1	R/W	Endpoint 1 Control Register	78
7F82	EP_CTRL2	R/W	Endpoint 2 Control Register	78
7F83	EP_CTRL3	R/W	Endpoint 3 Control Register	78
7F84	EP_CTRL4	R/W	Endpoint 4 Control Register	78
7F85	EP_CTRL5	R/W	Endpoint 5 Control Register	78
7F86	EP_CTRL6	R/W	Endpoint 6 Control Register	78
7F87	EP_CTRL7	R/W	Endpoint 7 Control Register	78
7F88	EP_CTRL8	R/W	Endpoint 8 Control Register	78
7F89	EP_CTRL9	R/W	Endpoint 9 Control Register	78
7F8A	EP_CTRL10	R/W	Endpoint 10 Control Register	78
7F8B	EP_CTRL11	R/W	Endpoint 11 Control Register	78
7F8C	EP_CTRL12	R/W	Endpoint 12 Control Register	78
7F8D	EP_CTRL13	R/W	Endpoint 13 Control Register	78
7F8E	EP_CTRL14	R/W	Endpoint 14 Control Register	78
7F8F	EP_CTRL15	R/W	Endpoint 15 Control Register	78
7F90	FRAMEL	R	USB Frame Count Low	79
7F91	FRAMEH	R	USB Frame Count High	80
7F92	SIE_ADDR	R/W	USB Local Address Register	80
7F93	SIE_STAT	R	SIE Status Register	83
7F94	SIE_CTRL1	R/W	SIE Control Register 1	84
7F95	SIE_TST1	R/W	Reserved Test Register	
7F96	SIE_TST2	R/W	Reserved Test Register	
7F97	SIE_EP_TEST	R/W	Reserved Test Register	
7F98	SIE_CONFIG	R/W	SIE Configuration Register	85
7F99	ALT_ADDR1	R/W	Secondary Local Address Register #1	81
7F9A	SIE_TST3	R/W	Reserved Test Register	
7F9B	SIE_TST4	R/W	Reserved Test Register	
7F9C	SIE_TST5	R/W	Reserved Test Register	
7F9D	SIE_TST6	R/W	Reserved Test Register	
7F9E	ALT_ADDR2	R/W	Secondary Local Address Register #2	81
7F9F	ALT_ADDR3	R/W	Secondary Local Address Register #3	82
7FA9	SIE_CTRL2	R/W	SIE Control Register 2	86
7FAA	EPCMD	R/W	Endpoint Command Register	87
7FAB	NONCTRL_EP1	R/W	Non-Control Endpoint Register 1 (High)	89
7FAC	NONCTRL_EP2	R/W	Non-Control Endpoint Register 2 (Low)	90
7FAD	PCKTHEAD	R	Packet Header FIFO	90

ADDRESS	NAME	R/W	DESCRIPTION	PAGE
7FAE	MMPCMD	R/W	Mem-Management Command Register	91
7FAF	MMPSTATE	R/W	Mem-Management State Register	92
7FEC	IN_NAKLO	R/W	IN NAK Register Low	93
7FED	IN_NAKHI	R/W	IN NAK Register High	94
7FEE	OUT_NAKLO	R/W	OUT NAK Register Low	95
7FEF	OUT_NAKHI	R/W	OUT NAK Register High	96

Table 10 - HUB Block Register Summary

ADDRESS	NAME	R/W	DESCRIPTION	PAGE
HUB REGISTERS				
7FA0	IdVendor-Low Byte	R/W	Low byte Vendor ID in little endian format (Bit 0 is the LSB)	
7FA1	IdVendor-High Byte	R/W	High byte Vendor ID in little endian format (Bit 0 is the LSB)	
7FA2	IdProduct-Low Byte	R/W	Low byte Product ID value in little endian format (Bit 0 is the LSB). This value is initialized by firmware upon initialization/power up. This value must be initialized prior to the Hub device participating in and USB enumeration transactions.	
7FA3	IdProduct-High Byte	R/W	High byte Product ID value in little endian format (Bit 0 is the LSB). This value is initialized by firmware upon initialization/power up. This value must be initialized prior to the Hub device participating in and USB enumeration transactions.	
7FA4	BcdDevice - Low Byte	R/W	This 8-bit value defines the USB device release number, which is assigned by the system manufacture.	
7FA5	BcdDevice - High Byte	R/W	This 8-bit value defines the USB device release number, which is assigned by the system manufacture.	
7FA6	HubControl1	R/W	Hub Control register 1	100

MCU REGISTER DESCRIPTION

MCU Runtime Registers

Table 11 - Interrupt 0 Source Register

ISR_0 (0x7F00 - RESET=0x00)			INTERRUPT 0 SOURCE REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	IRQ3	R/W	External interrupt input. 0 = Inactive 1 = Active
6	IRQ2	R/W	External interrupt input. 0 = Inactive 1 = Active
5	IRQ1	R/W	External interrupt input. 0 = Inactive 1 = Active
4	IRQ0	R/W	External interrupt input. 0 = Inactive 1 = Active
3	RX_PKT	R/W	1 = A Packet Number (PNR) has been successfully queued on the RXFIFO.
2	TX_EMPTY	R/W	1 = Whenever an enabled TX Endpoint's FIFO becomes empty. This will occur when the last queued packet in one of the 16 TX queues is successfully transferred to the Host.
1	TX_PKT	R/W	1 = A Packet was successfully transmitted.
0	ISADMA	R/W	1 = When a selected 8237 channels in BUS_STAT/BUS_MASK register pair either reached Terminal Count or have a new DMA Request Pending.

The bits in this register are cleared by writing a '1' to the corresponding bit.

Note 1: TX_EMPTY is useful for warning of USB performance degradation. This interrupt indicates that the next time the Host polls the affected endpoint, it will receive a NAK for that endpoint, thus reducing effective overall bandwidth due to retries. Firmware must use TX_STAT A, B, and C to determine which endpoint queue is empty.

Note 2: When ISADMA causes an interrupt, the 8237 CH_STAT register should also be read and serviced when the bit causing the interrupt is to be rearmed. When ISR_0 is read and the ISADMA bit is cleared, any other low-to-high transitions in the BUS_STAT register bits that are not masked will still cause an interrupt.

Note 3: If an expected IN token at EP1 (when in isochronous mode) does not arrive, the packet number will be removed from the TX_FIFO and saved in the TX_COMPLETION_FIFO and a TX_PKT interrupt will be sent to the MCU. It is the responsibility of the MCU to remove pages assigned to that packet. (Note if the MCU masks the TX_PKT interrupt, it will not be notified).

Table 12 - Interrupt 0 Mask

IMR_0 (0x7F01- RESET=0xFF)			INTERRUPT 0 MASK REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	IRQ3	R/W	External interrupt input mask 0 = Enable Interrupt 1 = Mask Interrupt
6	IRQ2	R/W	External interrupt input mask 0 = Enable Interrupt 1 = Mask Interrupt
5	IRQ1	R/W	External interrupt input mask 0 = Enable Interrupt 1 = Mask Interrupt
4	IRQ0	R/W	External interrupt input mask 0 = Enable Interrupt 1 = Mask Interrupt
3	RX_PKT	R/W	Received Packet MMU Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
2	TX_EMPTY	R/W	Transmit Queue Empty MMU Interrupt 0 = Enable Interrupt 1 = Mask Interrupt
1	TX_PKT	R/W	Transmit Packet MMU Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
0	ISADMA	R/W	ISADMA Status Change Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt

Table 13 - Interrupt 1 Source Register

ISR_1 (0x7F02- RESET=0x00)			INTERRUPT 1 SOURCE REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:5]	Reserved	R/W	Reserved
4	EOT	R/W	1 = The SIE returned to Idle State. Marks the end of each transaction.
3	SOF	R/W	1 = When a Start of Frame token is correctly decoded. Generated by the write strobe to the Frame Count register.
2	ALLOC	R/W	1 = MCU Software Allocation Request complete interrupt. This interrupt is not generated for hardware (SIEDMA) allocation requests.
1	RX_OVRN	R/W	1 = A receive condition has occurred that will stop the current receive buffer to not be processed. The SIE automatically recovers from this condition after its cause has been alleviated (e.g. any partially allocated packets will be released. See Note 2).
0	PWR_MNG	R/W	1 = A wakeup or power management event in the WU_SRC_1 or WU_SRC_2 registers has gone active.

Note 1: The bits in this register are cleared by writing a '1' to the corresponding bit.

Note 2: The RX_OVRN interrupt should be considered by firmware as a general Receive Overrun of the SIE, meaning that a packet destined for the RAM buffer could not be received and was not acknowledged back to the Host. The firmware should check to see if the RX Packet Number FIFO Register (RXFIFO) is full. If it is empty, then there may be too many transmit packets queued for the device to receive anything, or the last packet may have been corrupted on the wire. If it is not empty, then one or more receive packets must be dequeued before the device can continue to receive packets. In the normal course of operation, the MCU should respond to a RX_PKT interrupt as often as possible and let the buffering logic do its job.

Note 3: The RX_OVRN Interrupt can also be triggered if a non-isochronous packet exceeds 64 bytes or if an isochronous packet exceeds the programmed limit (SIE_CTRL2 – Page 86). When a packet that is too long is detected, the packet will be discarded from memory and the RX Overrun Interrupt will be triggered. For non-isochronous packets, the hardware will stall the Rx Endpoint at the handshake after reception.

Note 4: The RX_OVRN Interrupt can also be triggered if an Enpoint receives a packet while it is stalled. The packet will be discarded from memory.

Table 14 - Interrupt 1 Mask

IMR_1 (0x7F03- RESET=0xFF)			INTERRUPT 1 MASK REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:5]	Reserved		Reserved
4	EOT	R/W	EOT interrupt mask 0 = Enable Interrupt 1 = Mask Interrupt
3	SOF	R/W	Start of Frame Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
2	ALLOC	R/W	MCU Software Allocation Complete Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
1	RX_OVRN	R/W	Receive Overrun Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
0	PWR_MNG	R/W	Power Management Wakeup Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt

Table 15 - Device Revision Register

DEV_REV (0x7F06- RESET=0x00)			DEVICE REVISION REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	BCD ‘00’ HEX 0x00	R	This register defines additional revision information used internally by SMSC. The value is silicon revision dependent.

Table 16 - Device Identification Register

DEV_ID (0x7F07- RESET=0x26)			DEVICE IDENTIFICATION REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	BCD '26' HEX 0x26	R	This register defines additional revision information used internally by SMSC

Table 17 – 8051 GP FIFO1

GP_FIFO1 (0x7F10- RESET=0xXX)			8051 GP FIFO1
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	GP_FIFO1	R/W	8 byte deep GP FIFO. This data FIFOs must not be read unless the associated status bit indicates that FIFO is not empty.

Table 18 – 8051 GP FIFO2

GP_FIFO2 (0x7F12 - RESET=0xXX)			8051 GP FIFO2
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	GP_FIFO2	R/W	8 byte deep GP FIFO. This data FIFOs must not be read unless the associated status bit indicates that FIFO is not empty.

Table 19 – 8051 GP FIFO3

GP_FIFO3 (0x7F14 - RESET=0xXX)			8051 GP FIFO3
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	GP_FIFO3	R/W	8 byte deep GP FIFO. This data FIFOs must not be read unless the associated status bit indicates that FIFO is not empty.

Table 20 – 8051 GP FIFO4

GP_FIFO4 (0x7F16 - RESET=0xXX)			8051 GP FIFO4
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	GP_FIFO4	R/W	8 byte deep GP FIFO. This data FIFOs must not be read unless the associated status bit indicates that FIFO is not empty.

FIFO Status Registers

Table 21 – 8051 GP FIFO 1 STATUS

GPFIFO1_STS (0x7F11 – RESET=0x01)			8051 GP FIFO status
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:2]	Reserved	R	Reserved
1	GPFIFO1_FULL	R	GP FIFO 1 full status 0 = Not FULL 1 = FULL
0	GPFIFO1_EMPTY	R	GP FIFO 1 empty status 0 = Has one or more TX packet 1 = Empty

Table 22 – 8051 GP FIFO 2 STATUS

GPFIFO2_STS (0x7F13 – RESET=0x01)			8051 GP FIFO 2 status
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:2]	Reserved	R	Reserved
1	GPFIFO2_FULL	R	GP FIFO 2 full status 0 = Not FULL 1 = FULL
0	GPFIFO2_EMPTY	R	GP FIFO 2 empty status 0 = Has one or more TX packet 1 = Empty

Table 23 – 8051 GP FIFO 3 STATUS

GPFIFO3_STS (0x7F15 – RESET=0x01)			8051 GP FIFO 3 status
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:2]	Reserved	R	Reserved
1	GPFIFO3_FULL	R	GP FIFO 3 full status 0 = Not FULL 1 = FULL
0	GPFIFO3_EMPTY	R	GP FIFO 3 empty status 0 = Has one or more TX packet 1 = Empty

Table 24 – 8051 GP FIFO 4 STATUS

GPFIFO4_STS (0x7F17 – RESET=0x01)			8051 GP FIFO status
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:2]	Reserved	R	Reserved
1	GPFIFO4_FULL	R	GP FIFO 4 full status 0 = Not FULL 1 = FULL
0	GPFIFO4_EMPTY	R	GP FIFO 4 empty status 0 = Has one or more TX packet 1 = Empty

Table 25 - GPIO Direction Register

GPIOA_DIR (0x7F18- RESET=0x00)			MCU UTILITY REGISTERS
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	GPIO7	R/W	GPIO7 Direction 0 = In 1 = Out
6	GPIO6	R/W	GPIO6 Direction 0 = In 1 = Out
5	GPIO5	R/W	GPIO5 Direction 0 = In 1 = Out
4	GPIO4/SOF	R/W	GPIO4 Direction 0 = In 1 = Out
3	GPIO3/T1	R/W	GPIO3 Direction 0 = In 1 = Out
2	GPIO2/T0	R/W	GPIO2 Direction 0 = In 1 = Out
1	GPIO1/TXD	R/W	GPIO1 Direction 0 = In 1 = Out
0	GPIO0/RXD	R/W	GPIO0 Direction 0 = In 1 = Out

Note: The Timer inputs T[1:0] can be configured as outputs and left unconnected so that software can write to the bits to trigger the timer. Otherwise, the Timer inputs can be used to count external events or internal SOF receptions.

Table 26 - GPIO Output Register

GPIOA_OUT (0x7F19- RESET=0x00)			GPIO DATA OUTPUT REGISTER A
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	GPIO7	R/W	GPIO7 Output Buffer Data
6	GPIO6	R/W	GPIO6 Output Buffer Data
5	GPIO5	R/W	GPIO5 Output Buffer Data
4	GPIO4/SOF	R/W	GPIO4 Output Buffer Data
3	GPIO3/T1	R/W	GPIO3 Output Buffer Data
2	GPIO2/T0	R/W	GPIO2 Output Buffer Data
1	GPIO1/TXD	R/W	GPIO1 Output Buffer Data
0	GPIO0/RXD	R/W	GPIO0 Output Buffer Data

Table 27 - GPIO Input Register

GPIOA_IN (0x7F1A- RESET=0xXX)			GPIO INPUT REGISTER A
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	GPIO7	R	GPIO7 Input Buffer Data
6	GPIO6	R	GPIO6 Input Buffer Data
5	GPIO5	R	GPIO5 Input Buffer Data
4	GPIO4/SOF	R	GPIO4 Input Buffer Data
3	GPIO3/T1	R	GPIO3 Input Buffer Data
2	GPIO2/T0	R	GPIO2 Input Buffer Data
1	GPIO1/TXD	R	GPIO1 Input Buffer Data
0	GPIO0/RXD	R	GPIO0 Input Buffer Data

Table 28 - Utility Configuration Register

UTIL_CONFIG (0x7F1B- RESET=0x00)			UTILITY CONFIGURATION REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:5]	Reserved	R	Reserved
4	GPIO4/SOF	R/W	GPIO4/SOF Output Select Mux 0 = GPIO4 1 = SOF port
3	GPIO3/T1	R/W	P3.5 Timer 1 input trigger source 0 = GPIO3 1 = SOF FRAME write strobe
2	GPIO2/T0	R/W	P3.4 Timer 0 input trigger source 0 = GPIO2 1 = SOF FRAME write strobe
1	GPIO1/TXD	R/W	GPIO1/TXD Output Select Mux 0 = GPIO1 1 = P3.1
0	GPIO0/RXD	R/W	P3.0 RXD/GPIO0 Input Select Mux 0 = RXD<=GPIO0 1 = RXD<='0'

Note1: In Counter mode, the 8051 must sample T[1:0] as a '1' in one instruction cycle, and then '0' in the next. So for 12MHz, the SOF Pulse must be active for at least 1us.

Note 2: Missing SOF packets can be reconstructed by using the Timer mode to count the number of 8051 instruction cycles since the last valid Frame was received.

Note 3: A GPIO can be used to output nSOF pulses. This can be done by configuring a GPIO as an output and writing to the GPIO out register to generate low pulses each time a SOF packet is received.

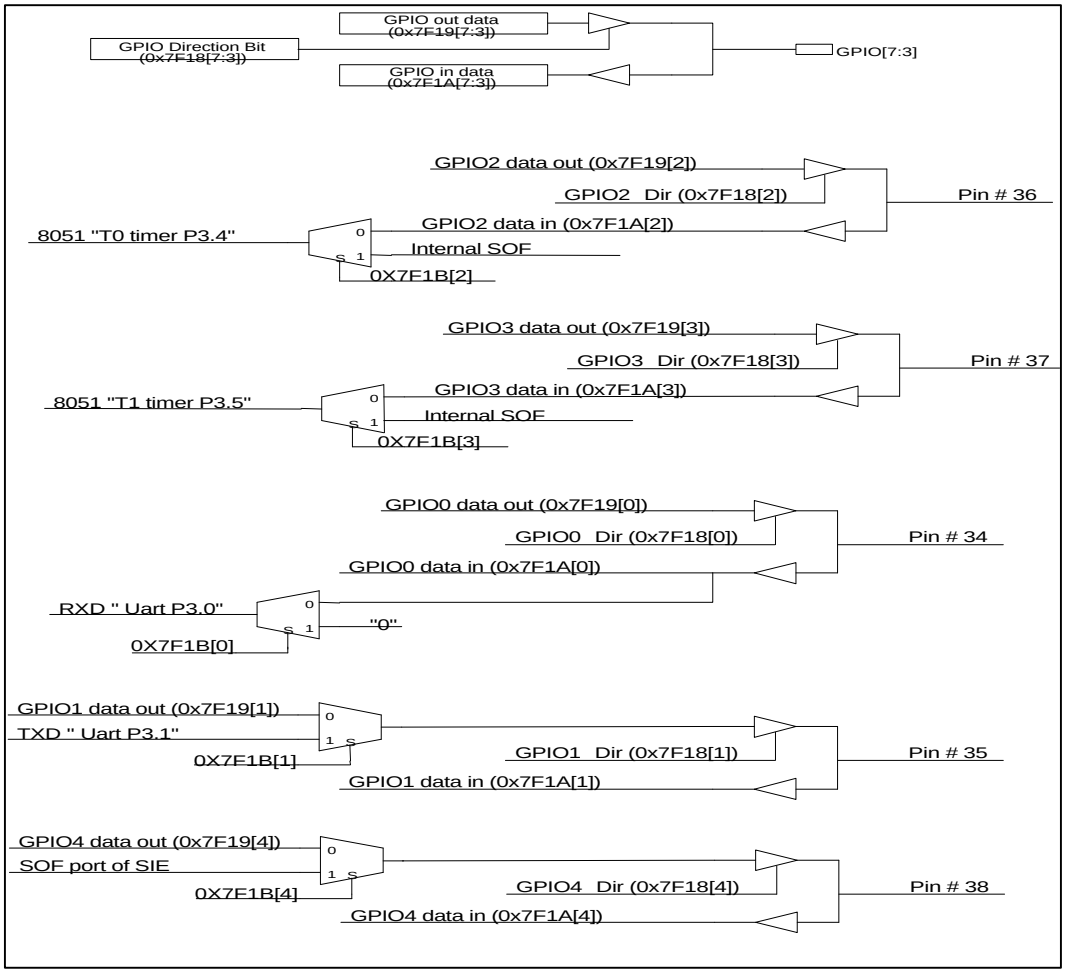


FIGURE 6 - GPIO MUXING BLOCK DIAGRAM

MCU POWER MANAGEMENT REGISTERS

Table 29 - MCU/ISADMA Clock Source Select

CLOCK_SEL (0x7F27 - RESET=0x40)			MCU/ISADMA CLOCK SOURCE SELECT
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	SLEEP	R/W	When PCON. 0 = 1 and SLEEP has been set to 1, the ring oscillator will be gated off, then all oscillators will be turned off for maximum power savings. (These two signals can be used to generate nFCE)
6	ROSC_EN	R/W	0 = Ring Oscillator Disable. 1 = Ring Oscillator Enable. ROSC_EN must be set to 1 before the MCU can be switched to the internal Ring Oscillator Clock source.
5	MCUCLK_SRC	R/W	MCUCLK_SRC overrides MCUCLK_x clock select and switches the MCU to the Ring Oscillator. 0 = Use Ring Oscillator. ROSC_EN must be enabled by the MCU first. 1 = Use clock specified in MCU_CLK_[1:0]
[4:3]	MCU_CLK[1:0]	R/W	[4:3] = 00: 8MHz [4:3] = 01: 12MHz [4:3] = 10: 16MHz [4:3] = 11: 24MHz
2	ISADMACLK_EXT	R/W	Selects an external clock source for the 8237 ISADMA controller for synchronizing the DMA with another block. NOTE: This will initially be an external input, but may eventually be used within the block to optimize performance, or as some other internal clock source. 0 = Use ISADMACLK[1..0] select 1 = Use EXT_IN clock source for 8237
[1:0]	ISADMACLK[1:0]	R/W	[1:0] = 00: Stopped [1:0] = 01: 2MHz [1:0] = 10: 4MHz [1:0] = 11: 8MHz

Note 1: The 8051 may program itself to run off of an internal Ring Oscillator having a frequency range between 4 and 12MHz. This is not a precise clock, but is meant to provide the 8051 with a clock source, without running the 24MHz crystal oscillator or the PLL

Note 2: Switching between fast and slow clocks is recommended to save power.

Note 3: Clock switching can be done on the fly as long as both clocks are running. When switching, it takes a total of six clocks (3 clocks of the original clock plus 3 clocks of the switching clock) to guarantee the switching.

Note 4: Time TBD is required from ROSC_EN=1 to MCUCLK_SRC=0.

Table 30 - FLASH Bank Select Register 2

MEM_BANK2 (0x7F29 - RESET=0x00)			FLASH BANK SELECT REGISTER 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:6]	Reserved	R	Reserved
[5:0]	A[19:14]	R/W	This register selects which 16k page resides at 0x8000-0xBFFF in Code Space and 0x8000-0xBFFF in Data Space.

Table 31 - FLASH Bank Select Register

MEM_BANK (0x7F29 - RESET=0x01)			FLASH BANK SELECT REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:6]	Reserved	R	Reserved
[5:0]	A[19:14]	R/W	This register selects which 16k page resides at 0x4000-0x7FFF in Code Space and 0xC000-0xFFFF in Data Space. The 0x0000-0x3FFF page will always reflect the 16K FLASH page 0 (0x00000-0x03FFF).

Table 32 - Wakeup Source 1 Register

WU_SRC_1 (0x7F2A – RESET=0x00)			WAKEUP SOURCE 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	R/W	Reserved
2	USB_Reset	R/W	This bit is set when the SIE detects simultaneous logic lows on D+ and D- (Single-Ended 0) for 32 to 64 full speed bit times, or 4 to 8 low speed bit times (or 2.5<t<5.5us). The USB_Reset signal may be as long as 10ms. SETUP tokens can be NAK'd for up to 10ms after the Reset signal is released.
1	Resume	R/W	This bit is set on detection of Global Resume state (when there is a transition from the "J" state while in Global Suspend).
0	Suspend	R/W	Suspend – If 3ms of IDLE state are detected by the hardware, then this bit will be set.

Note 1: Only low to high transitions for the associated inputs sets these bits.
Note 2: The bits in this register are cleared by writing a ‘1’ to the corresponding bit.
Note 3: Unmasked Wakeup Source bits generate an INT1 PWR_MNG interrupt, and restart the 8051 when its clock is stopped. This restarts the Ring Oscillator and crystal oscillator for the MCU to resume from <500µA operation.
Note 4: To initiate USB Remote Wakeup, the SIE_Resume bit should be used in the SIE_CONFIG register.

Table 33 - Wakeup Mask 1 Register

WU_MSK_1 (Note 1) (0x7F2B - RESET=0x07)			WAKEUP MASK 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	R	Reserved
2	USB_Reset	R/W	External wakeup event. 0 = Enabled 1 = Masked
1	Resume	R/W	External wakeup event. 0 = Enabled 1 = Masked
0	Reserved	R	Reserved

Note 1: Interrupt events enabled by these bits are routed to the PWR_MNG Bit 0 in the ISR_1 register.

Table 34 - Wakeup Source 2 Register

WU_SRC_2 (0x7F2C - RESET=0x00)			WAKEUP SOURCE 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:4]	'0'	R/W	Reserved
3	IRQ3	R/W	External Interrupt state since WU_SRC_2 was last read. 0 = Unchanged 1 = Changed
2	IRQ2	R/W	External Interrupt state since WU_SRC_2 was last read. 0 = Unchanged 1 = Changed
1	IRQ1	R/W	External Interrupt state since WU_SRC_2 was last read. 0 = Unchanged 1 = Changed
0	IRQ0	R/W	External Interrupt state since WU_SRC_2 was last read. 0 = Unchanged 1 = Changed

Note 1: Any transition from high to low, or low to high on the associated input sets these bits.
Note 2: The bits in this register are cleared by writing a '1' to the corresponding bit.
Note 3: Since this register will report any status change, when devices are to be powered down while monitored, the appropriate bits must be masked until the device is armed correctly.

Table 35 - Wakeup Mask 2 Register

WU_MSK_2 (0x7F2D - RESET=0x0F)			WAKEUP MASK 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7 :4]	'0'	R	Reserved
3	IRQ3	R/W	External wakeup event enable. 0 = Enabled 1 = Masked
2	IRQ2	R/W	External wakeup event enable. 0 = Enabled 1 = Masked
1	IRQ1	R/W	External wakeup event enable. 0 = Enabled 1 = Masked
0	IRQ0	R/W	External wakeup event enable. 0 = Enabled 1 = Masked

Note: Interrupt events enabled by these bits are be routed to the PWR_MNG Bit 0 in the ISR_1 register.

MCU ISA Interface Registers

Table 36 – ISA Bus Request Register

BUS_REQ (0x7F70 – RESET=0x00)			ISA BUS REQUEST REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	INH_TC3	R/W	This bit inhibits DMA channel 3 TC.**See Note Below 0 = TC is driven onto the ISA bus via EOP as before. 1 = TC is forced inactive.
6	INH_TC2	R/W	This bit inhibits DMA channel 2 TC.** See Note Below 0 = TC is driven onto the ISA bus via EOP as before. 1 = TC is forced inactive.
5	INH_TC1	R/W	This bit inhibits DMA channel 1 TC.** See Note Below 0 = TC is driven onto the ISA bus via EOP as before. 1 = TC is forced inactive.
4	INH_TC0	R/W	This bit inhibits DMA channel 0 TC.** See Note Below 0 = TC is driven onto the ISA bus via EOP as before. 1 = TC is forced inactive.
3	RESET_8237	R/W	Writing a '1' holds the 8237 hardware reset input active. Writing '0' releases it for normal operation. May be used for clock switching or power management functions.
2	AEN	R	This bit reflects the status of the 8237's AEN pin. This bit does not generate an interrupt
1	HLDA	R/W	The 8051 can grant the bus when it is ready via HLDA. This should tri-state any common signals between the 8051 and the 8237 on the ISA bus.
0	HREQ	R	This bit reflects the status of the 8237's HREQ bus request pin. This bit does not generate an interrupt.

Note: HLDA Example: When the 8051 is running at 24MHz, and the 8237 is running at 2MHz, the 8237 may take up to 1.5us to complete a transfer after deasserting HLDA When running the 8051 at 24MHz, wait states must be added when the 8237 is running at 2 or 4 MHz. When running the 8051 at 12MHz, wait states must be added when the 8237 is running at 2 MHz.

Note**: The “Inhibit” function is not valid for Memory-to-Memory DMA cycles

Table 37 - ISA Bus Status Register

BUS_STAT (0x7F73 - RESET=0xXX)			ISA BUS STATUS REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	CH3RQ	R	Channel 3 DMA Request 0 = No Request Pending 1 = Request Pending
6	CH2RQ	R	Channel 2 DMA Request 0 = No Request Pending 1 = Request Pending
5	CH1RQ	R	Channel 1 DMA Request 0 = No Request Pending 1 = Request Pending
4	CH0RQ	R	Channel 0 DMA Request 0 = No Request Pending 1 = Request Pending
3	CH3TC	R	Channel 3 Terminal Count Reached 0 = No 1 = Yes
2	CH2TC	R	Channel 2 Terminal Count Reached 0 = No 1 = Yes
1	CH1TC	R	Channel 1 Terminal Count Reached 0 = No 1 = Yes
0	CH0TC	R	Channel 0 Terminal Count Reached 0 = No 1 = Yes

Note 1: Each bit in this register reflects the current value of the corresponding bit in the 8237 CH_STAT status register.

Note 2: The 8237 clears bits 3..0 in the CH_STAT status register when the 8051 reads it through the ISA Bus I/O Window.

Note 3: Reading the BUS_STAT register does not clear or otherwise affect the BUS_STAT register.

Note 4: The ISADMA bit in ISR_0 is latched high whenever any bit in BUS_STAT that is enabled in BUS_MASK transitions from low to high.

Note 5: This register is intended (1) to provide a view into the status of the 8237 without having to assume control of the ISA bus during DMA transfers, and (2) to provide a means for generating the ISADMA interrupt in ISR_0 which indicates that a DMA transfer has completed and that the 8051 should take control of the bus and setup the 8237 for its next transfer. Bits 7-4 can be used to generate additional interrupt requests from the DREQ pins, or simply to monitor channel request status by masking them.

Table 38 - ISA Bus Status Mask Register

BUS_MASK (0x7F74 - RESET=0xFF)			ISA BUS STATUS MASK REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	CH3RQ_MASK	R/W	Channel 3 DMA Request ISADMA Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
6	CH2RQ_MASK	R/W	Channel 2 DMA Request ISADMA Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
5	CH1RQ_MASK	R/W	Channel 1 DMA Request ISADMA Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
4	CH0RQ_MASK	R/W	Channel 0 DMA Request ISADMA Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
3	CH3TC_MASK	R/W	Channel 3 Terminal Count ISADMA Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
2	CH2TC_MASK	R/W	Channel 2 Terminal Count ISADMA Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
1	CH1TC_MASK	R/W	Channel 1 Terminal Count ISADMA Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
0	CH0TC_MASK	R/W	Channel 0 Terminal Count ISADMA Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt

Table 39 - ISA I/O Window Base Register

IOBASE (0x7F71 - RESET=0x00)			ISA I/O WINDOW BASE REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	SA[15:8]	R/W	When the 8051 reads or writes to the ISA I/O Window, this register is combined with the 8 bit offset in the 256 byte window and presented as the 64k I/O Space address during an 8051-ISA IOR or IOW cycle

Table 40 - ISA Memory Window Base Register

MEMBASE (0x7F72 - RESET=0x00)			ISA MEMORY WINDOW BASE REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	SA[19:12]	R/W	When the 8051 reads or writes to the ISA Memory Window, this register is combined with the 12 bit offset in the 4k byte window and presented as the 1Mbyte Memory address during an 8051-ISA MEMR or MEMW cycle.

8237 (ISADMA) REGISTER DESCRIPTION

Memory Map

Table 41 - ISADMA Memory Map

8237 MEMORY ADDRESS	DESCRIPTION
0xFC00-0xFFFF	1k Window to Packet with PNR=0x1F
0xF800-0xFBFF	1k Window to Packet with PNR=0x1E
0xF400-0xF7FF	1k Window to Packet with PNR=0x1D
0xF000-0xF3FF	1k Window to Packet with PNR=0x1C
0xEC00-0xEFFF	1k Window to Packet with PNR=0x1B
0xE800-0xEBFF	1k Window to Packet with PNR=0x1A
0xE400-0xE7FF	1k Window to Packet with PNR=0x19
0xE000-0xE3FF	1k Window to Packet with PNR=0x18
0xDC00-0xDFFF	1k Window to Packet with PNR=0x17
0xD800-0xDBFF	1k Window to Packet with PNR=0x16
0xD400-0xD7FF	1k Window to Packet with PNR=0x15
0xD000-0xD3FF	1k Window to Packet with PNR=0x14
0xCC00-0xCFFF	1k Window to Packet with PNR=0x13
0xC800-0xCBFF	1k Window to Packet with PNR=0x12
0xC400-0xC7FF	1k Window to Packet with PNR=0x11
0xC000-0xC3FF	1k Window to Packet with PNR=0x10
0xBC00-0xBFFF	1k Window to Packet with PNR=0x0F
0xB800-0xBBFF	1k Window to Packet with PNR=0x0E
0xB400-0xB7FF	1k Window to Packet with PNR=0x0D
0xB000-0xB3FF	1k Window to Packet with PNR=0x0C
0xAC00-0xAFFF	1k Window to Packet with PNR=0x0B
0xA800-0xABFF	1k Window to Packet with PNR=0x0A
0xA400-0xA7FF	1k Window to Packet with PNR=0x09
0xA000-0xA3FF	1k Window to Packet with PNR=0x08
0x9C00-0x9FFF	1k Window to Packet with PNR=0x07
0x9800-0x9BFF	1k Window to Packet with PNR=0x06
0x9400-0x97FF	1k Window to Packet with PNR=0x05
0x9000-0x93FF	1k Window to Packet with PNR=0x04
0x8C00-0x8FFF	1k Window to Packet with PNR=0x03

8237 MEMORY ADDRESS	DESCRIPTION
0x8800-0x8BFF	1k Window to Packet with PNR=0x02
0x8400-0x87FF	1k Window to Packet with PNR=0x01
0x8000-0x83FF	1k Window to Packet with PNR=0x00
0x0000-0x7FFF	32K Window to External ISA RAM

The actual packet may be composed of up to 10 different 128 byte non-contiguous packets, but the MMU re-maps the internal addresses automatically such that the 8237 and 8051 only need to reference the packet number and offset within the packet. For example, suppose a 312 (0x138) byte packet is received by the SIEDMA from the host. The patented MMU allocates 384 bytes for the packet (including an 8 byte status header) and returns a PNR tag of 0x0A. The SIEDMA engine will place 0x0A in the receive packet queue and notify the 8051. The 8051 will take that PNR, examine the packet through its own PNR/Pointer registers, and determine the offset for the payload data it wants to transfer from the packet, say 0x027. The address it must calculate for the 8237 base address register would therefore be 0xA827 (0xA800+0x027). Each channel can be programmed with a different (or same) Packet Number and offset and the data will appear to it as ordinary contiguous RAM (see table 32 for more information).

Software written to this model will work for virtually any Endpoint number and Buffer size combination.

Runtime Registers

The DMA controller has a block of 16 R/W registers which normally occupy I/O locations 0x00-0x0F on the ISA bus. When they are located at 0x0000-0x000F on the ISA bus, the 8051 can access them by programming the IOBASE Register to 0x00, and reading or writing from 0x4000-0x400F.

Table 42 - 8237 Registers in ISA I/O Space

0x0000	Channel 0: Current Address H/L
0x0001	Channel 0: Byte Count H/L
0x0002	Channel 1: Current Address H/L
0x0003	Channel 1: Byte Count H/L
0x0004	Channel 2: Current Address H/L
0x0005	Channel 2: Byte Count H/L
0x0006	Channel 3: Current Address H/L
0x0007	Channel 3: Byte Count H/L
0x0008	Status/Command Register
0x0009	Write Request Register
0x000A	Write Single Mask Register
0x000B	Write Mode Register
0x000C	Clear Byte Ptr F/F - Read Temp Register
0x000D	Master Clear
0x000E	Clear Mask
0x000F	Write All Mask Bits

Note: To write to these registers, HLDA must be logic low.

Table 43 - 8237 Address Programming Guide

8237 INTERNAL ADDRESS PROGRAMMING GUIDE		
BIT	**NAME**	**DESCRIPTION**
15	INT_EXT	Indicates whether this address refers to Internal Buffer RAM or External ISA Memory Space 0 = External 1 = Internal When this bit is set to zero (0), I/O capability is added to External Memory DMA. This capability can only be used for DMA channels 2 or 3.
[14:10]	PN[4:0]/SA[14:10]	External Address -or- Internal Packet Number SA[14..10] when INT_EXT=0 PN[4..0] when INT_EXT=1
[9:0]	PTR[9:0]/SA[9:0]	External Address -or- Internal Packet Offset Pointer SA[9..0] when INT_EXT=0 PTR[9..0] when INT_EXT=1

Note: SA[19..15] are driven low when the 8237 is accessing external ISA memory. PTR10 is driven low when the 8237 is accessing internal buffer RAM.
Note that the actual transfer size for the ISADMA is limited to 1024 bytes, which limits the payload data to 1016 bytes per transfer when the 8 byte header is skipped. Also note that the 8051 still has access to 1Meg of external RAM through the MEMBASE register and it is independent of the 8237's 32k external limit.

Table 44 - Channel 0 Current Address Register

CH0_ADDR (ISA 0x0000)			**CHANNEL 0 CURRENT ADDRESS**
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	CH0_ADDRL	R/W	Lower 8 bits of Base and Current Address when Byte F/F = 0
[7:0]	CH0_ADDRH	R/W	Upper 8 bits of Base and Current Address when Byte F/F = 1

Note: Byte F/F is an internal Flip Flop which reflects which byte (high or low) is being written. The CLEAR_FF register should be written to before writing this register to guarantee which byte (high or low) is being written. See the Address Programming Table for 16 bit Address definitions.

Table 45 - Channel 0 Byte Count Register

CH0_CNT (ISA 0x0001)			CHANNEL 0 BYTE COUNT
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	CH0_CNTL	R/W	Lower 8 bits of Byte Count when Byte F/F = 0
[7:0]	CH0_CNTH	R/W	Upper 8 bits of Byte Count when Byte F/F = 1

Note: The CLEAR_FF register should be written to before writing this register to guarantee which byte (high or low) is being written. See Address Programming Table for 16 bit Address definitions.

Table 46 - Channel 1 Current Address Register

CH1_ADDR (ISA 0x0002)			CHANNEL 1 CURRENT ADDRESS
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	CH1_ADDRL	R/W	Lower 8 bits of Base and Current Address when Byte F/F = 0
[7:0]	CH1_ADDRH	R/W	Upper 8 bits of Base and Current Address when Byte F/F = 1

Note: The CLEAR_FF register should be written to before writing this register to guarantee which byte (high or low) is being written. See the Address Programming Table for 16 bit Address definitions.

Table 47 - Channel 1 Byte Count Register

CH1_CNT (ISA 0x0003)			CHANNEL 1 BYTE COUNT
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	CH1_CNTL	R/W	Lower 8 bits of Byte Count when Byte F/F = 0
[7:0]	CH1_CNTH	R/W	Upper 8 bits of Byte Count when Byte F/F = 1

Note: The CLEAR_FF register should be written to before writing this register to guarantee which byte (high or low) is being written. See Address Programming Table for 16 bit Address definitions.

Table 48 - Channel 2 Current Address Register

CH2_ADDR (ISA 0x0004)			CHANNEL 2 CURRENT ADDRESS
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	CH2_ADDRL	R/W	Lower 8 bits of Base and Current Address when Byte F/F = 0
[7:0]	CH2_ADDRH	R/W	Upper 8 bits of Base and Current Address when Byte F/F = 1

Note: The CLEAR_FF register should be written to before writing this register to guarantee which byte (high or low) is being written. See the Address Programming Table for 16 bit Address definitions.

Table 49 - Channel 2 Byte Count Register

CH2_CNT (ISA 0x0005)			CHANNEL 2 BYTE COUNT
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	CH2_CNTL	R/W	Lower 8 bits of Byte Count when Byte F/F = 0
[7:0]	CH2_CNTH	R/W	Upper 8 bits of Byte Count when Byte F/F = 1

Note: The CLEAR_FF register should be written to before writing this register to guarantee which byte (high or low) is being written. See Address Programming Table for 16 bit Address definitions.

Table 50 - Channel 3 Current Address Register

CH3_ADDR (ISA 0x0006)			CHANNEL 3 CURRENT ADDRESS
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	CH3_ADDRL	R/W	Lower 8 bits of Base and Current Address when Byte F/F = 0
[7:0]	CH3_ADDRH	R/W	Upper 8 bits of Base and Current Address when Byte F/F = 1

Note: The CLEAR_FF register should be written to before writing this register to guarantee which byte (high or low) is being written. See the Address Programming Table for 16 bit Address definitions.

Table 51 - Channel 3 Byte Count Register

CH3_CNT (ISA 0x0007)			CHANNEL 3 BYTE COUNT
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	CH3_CNTL	R/W	Lower 8 bits of Byte Count when Byte F/F = 0
[7:0]	CH3_CNTH	R/W	Upper 8 bits of Byte Count when Byte F/F = 1

Note: The CLEAR_FF register should be written to before writing this register to guarantee which byte (high or low) is being written. See Address Programming Table for 16 bit Address definitions.

Table 52 - Channel Status Register

CH_STAT (ISA 0x0008)			CHANNEL STATUS REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	CH3RQ	R	Channel 3 DMA Request 0 = No Request Pending 1 = Yes Request Pending
6	CH2RQ	R	Channel 2 DMA Request 0 = No Request Pending 1 = Yes Request Pending
5	CH1RQ	R	Channel 1 DMA Request 0 = No Request Pending 1 = Yes Request Pending
4	CH0RQ	R	Channel 0 DMA Request 0 = No Request Pending 1 = Yes Request Pending
3	CH3TC	R	Channel 3 Terminal Count Reached 0 = No 1 = Yes
2	CH2TC	R	Channel 2 Terminal Count Reached 0 = No 1 = Yes
1	CH1TC	R	Channel 1 Terminal Count Reached 0 = No 1 = Yes
0	CH0TC	R	Channel 0 Terminal Count Reached 0 = No 1 = Yes

Note 1: These bits are also visible outside of I/O space in the BUS_STAT register.
Note 2: These bits are cleared when this register is read through the ISA I/O Window.

Table 53 - 8237 Command Register

CH_CMD (ISA 0x0008)			COMMAND REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	DACK_SENS	W	DACK Sense 0 = Active High 1 = Active Low
6	DREQ_SENS	W	DREQ Sense (1 = Active Low, 0 = Active High)
5	WRITE_TIME	W	Write Timing Select 0 = Late Timing 1 = Extended
4	PRIORITY	W	Priority 0 = Fixed 1 = Rotating
3	COMP_TIME	W	Timing 0 = Normal 1 = Compressed
2	CTRL_EN	W	Controller Enable 0 = Enable 1 = Disable
1	ADDR_HOLD	W	Channel 0 Address Hold 0 = Disable 1 = Hold Enable
0	MEM2MEM	W	Memory-to-Memory 0 = Disable 1 = Enable

Table 54 - 8237 Write Single Request Register

CH_REQ (ISA 0x0009)			WRITE REQUEST REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	W	Reserved
2	SET_CLR	W	Force Internal DMA Request Bit 0 = Clear 1 = Set
[1:0]	SEL[1:0]	W	'00' = Select Channel 0 DREQ '01' = Select Channel 1 DREQ '10' = Select Channel 2 DREQ '11' = Select Channel 3 DREQ

Table 55 - 8237 Write Single Mask Register

CH_MASK (ISA 0x000A)			WRITE SINGLE MASK REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	R	Reserved
2	SET_CLR	W	Set Channel Mask Bit 0 = Clear 1 = Set
[1:0]	SEL[1:0]	W	'00' = Select Channel 0 Mask Bit '01' = Select Channel 1 Mask Bit '10' = Select Channel 2 Mask Bit '11' = Select Channel 3 Mask Bit

Table 56 - Write Mode Register

DMA_MODE (ISA 0x000B)			WRITE MODE REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:6]	MODE[1:0]	W	'00' = Demand Mode Select '01' = Single Mode Select '10' = Block Mode Select '11' = Cascade Mode Select
5	INC_DEC	W	Auto-increment/Decrement 0 = Increment 1 = Decrement
4	AUTO_INIT	W	Auto-initialization 0 = Disable 1 = Enable
[3:2]	R/WV[1:0]	W	'00' = Verify Transfer '01' = Write Transfer '10' = Read Transfer '11' = Illegal 'XX' if bits 6 and 7 = '11' Or if CH_CMD register bit 0 = 1 (memory-to-memory transfer)
[1:0]	SEL[1:0]	W	'00' = Select Channel 0 '01' = Select Channel 1 '10' = Select Channel 2 '11' = Select Channel 3

Table 57 - Clear Byte Pointer Flip Flop Register

CLEAR_FF (ISA 0x000C)			CLEAR BYTE POINTER FLIP FLOP
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	BPFF	W	This register must be written to clear the high/low byte pointer flip flop prior to reading or writing new address or word count information to the 8237.

Table 58 - Read Temporary Register

RD_TEMP (ISA 0x000D)			READ TEMPORARY REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	TEMP_BYTE	R	This location holds the value of the last byte transferred in a memory-to-memory operation.

Table 59 - Master Clear Register

MSTR_CLR: (ISA 0x000D)			MASTER CLEAR REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	SW_RESET	W	Writing to this register has the same effect on the registers as a hardware reset. The 8237 will enter the idle state.

Table 60 - Clear Mask Register

CLR_MASK: (ISA 0x000E)			CLEAR MASK REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	CLR_ALL	W	Writing to this register clears the mask bits of all four channels and allows them to receive DMA requests.

Table 61 - Clear All Mask Bits Register

ALL_MASK (ISA 0x000F)			WRITE ALL MASK BITS REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:4]	Reserved	W	Reserved
3	CH3_MASK	W	Channel 3 Mask Bit (1 = Set Mask, 0 = Clear Mask)
2	CH2_MASK	W	Channel 2 Mask Bit (1 = Set Mask, 0 = Clear Mask)
1	CH1_MASK	W	Channel 1 Mask Bit (1 = Set Mask, 0 = Clear Mask)
0	CH0_MASK	W	Channel 0 Mask Bit (1 = Set Mask, 0 = Clear Mask)

SCATTER-GATHER DMA (SGDMA) REGISTER DESCRIPTION

The SGDMA has four DMA channels with each channel having its own set of registers (FIGURE 7). Each register set starts at a different starting BASE address: Channels 0, 1,2, and 3 have starting BASE addresses 7FB0, 7FBA, 7FC4, and 7FCE, respectively. The tables on the following pages describe each register of one register set, using 'x' to denote the DMA channel (x ranges from 0 to 3). The address of each register is denoted as BASE+n, where n is the offset from BASE.

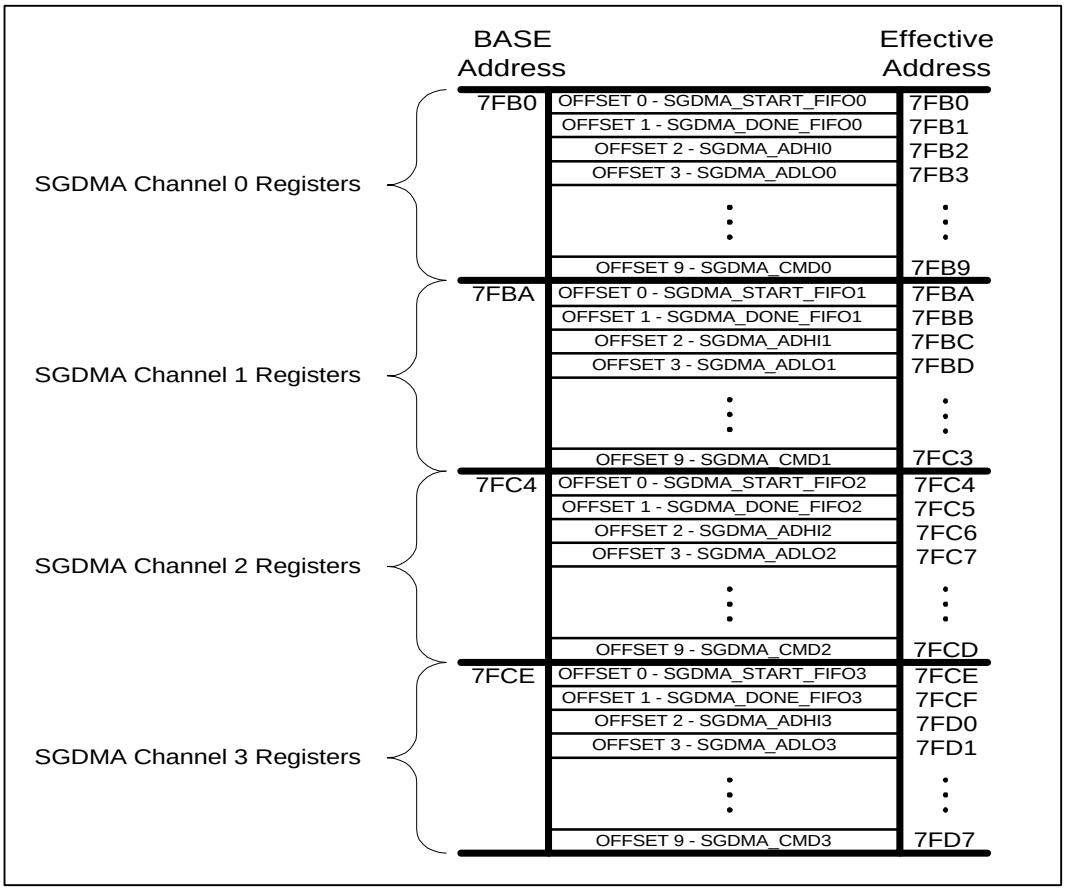


FIGURE 7 – SGDMA REGISTER SPACE

The SGDMA also contains a PIO engine that permits the MCU to access the ISA bus on a cycle stealing basis with the DMA transfers (there is only one PIO engine).

Table 62 – SGDMA Packet Number Start FIFO Register

SGDMA_START_FIFOx [x=0..3] (BASE+0 – RESET = 0x00)			SGDMA_START_FIFO
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:5]	Reserved	R/W	Reserved – Read back as 0
[4:0]	PCKTNUM	R/W	Packet Number to queue on SGDMA_START_FIFOx [x=0..3].

Note 1: This register is used when the channel is configured for an MMU memory operation.

Note 2: When the channel is enabled (CHANNEL_ENABLE=1 in register SGDMA_CMDx [x=0..3]), only the SGDMA can read from these registers. The MCU can only read from these registers when the channel is disabled (CHANNEL_ENABLE=0 in register SGDMA_CMDx [x=0..3]), and not busy (DMA_BUSY=0 in register SGDMA_STSx [x=0..3]). It is the MCU's responsibility to ensure that this FIFO does not overflow.

Table 63 – SGDMA Packet Number Done FIFO Register

SGDMA_DONE_FIFOx [x=0..3] (BASE+1 – RESET = 0x00)			SGDMA_DONE_FIFO
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:5]	Reserved	R/W	Reserved – Read back as 0.
[4:0]	PCKTNO	R/W	Packet Number on top of the SGDMA_DONE_FIFOx [x=0..3].

Note 1: This register is used when the channel is configured for an MMU memory operation.

Note 2: The SGDMA puts the packet number on this FIFO when the DMA transfer is complete, or when the MCU disables a channel that was enabled and busy (meaning a DMA transfer was in progress). When this register is read, the FIFO is popped. It is the MCU's responsibility to ensure that this FIFO does not overflow.

Table 64 – SGDMA ISA Address High Byte Register

SGDMA_ADHIx [x=0..3] (BASE+2 – RESET = 0x00)			SGDMA_ADHI
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	ADHIx [x=0..3]	R/W	Contains the high byte of the ISA Address to use.

Table 65 – SGDMA ISA Address Low Byte Register

SGDMA_ADLOx [x=0..3] (BASE+3 – RESET = 0x00)			SGDMA_ADLO
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	ADLOx [x=0..3]	R/W	Contains the low byte of the ISA Address to use.

Note 1: The SGDMA_ADHIx [x=0..3] and SGDMA_ADLOx [x=0..3] registers are used when the channel [0..3] is configured for an ISA memory operation.

Note 2: The MCU must not write to these registers unless the channel is disabled (CHANNEL_ENABLED=0 in register SGDMA_CMDx [x=0..3]) and not busy (DMA_BUSY=0 in register SGDMA_STSx [x=0..3]).

Table 66 – SGDMA Transfer Size High Register

SGDMA_SIZEHIx [x=0..3] (BASE+4 – RESET = 0x00)			SGDMA_SIZEHI
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	SIZEHIx [x=0..3]	R/W	Contains the high byte of the payload data size (in bytes)

Table 67 – SGDMA Transfer Size Low Register

SGDMA_SIZELOx [x=0..3] (BASE+5 – RESET = 0x00)			SGDMA_SIZELO
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	SIZEHIx [x=0..3]	R/W	Contains the low byte of the payload data size (in bytes).

Note 1: The SGDMA_SIZEHIx [x=0..3] and SGDMA_SIZELOx [x=0..3] registers are used when the channel [0..3] is configured for an ISA memory operation or for a MMU operation without a Packet Header.

Note 2: The MCU must not write to these registers unless the channel is disabled (CHANNEL_ENABLED = 0 in register SGDMA_CMDx [x=0..3]) and not busy (DMA_BUSY=0 in register SGDMA_STSx [x=0..3]).

Table 68 – SGDMA Total Packets in Channel Register

SGDMA_TOTAL_PKTSx [x=0..3] (BASE+6 – RESET = 0x00)			SGDMA_TOTAL_PKTS
BIT	**NAME**	**R**	**DESCRIPTION**
[7:5]	Reserved	R	Reserved – Read back as 0
[4:0]	NUMPCKTS	R	Number of packets currently in the channel [0..3]

Note 1: This register contains a count of the total number of packets in the corresponding SGDMA channel [0..3]. It is incremented when the MCU puts a packet number into the SGDMA_START_FIFOx [x=0..3] and is decremented when the MCU pops a packet from the SGDMA_DONE_FIFOx [x=0..3].

Table 69 – SGDMA Total Packets in the Done FIFO Register

SGDMA_DONE_PCKTSx [x=0..3] (BASE+7 – RESET = 0x00)			SGDMA_DONE_PCKTS
BIT	**NAME**	**R**	**DESCRIPTION**
[7:5]	Reserved	R	Reserved – Read back as 0
[4:0]	NUMPCKTS	R	Number of packets in the SGDMA_DONE_FIFOx [x=0..3]

Note 1: This register contains a count of the total number of packets in the SGDMA_DONE_FIFOx [x=0..3]. It is incremented when the SGDMA puts a packet number into the SGDMA_DONE_FIFOx [x=0..3] and is decremented when the MCU pops a packet from the SGDMA_DONE_FIFOx [x=0..3].

Table 70 – SGDMA Status Register

SGDMA_STSx [x=0..3] (BASE+8 – RESET = 0x00)			SGDMA_STS
BIT	**NAME**	**R**	**DESCRIPTION**
7	DMA_BUSY	R	Will be Set (1) while an SGDMA transfer is in progress
6	ISA_DONE	R	Will be Set (1) after an ISA memory – ISA device transfer is completed. Resets to 0 when the channel is disabled (CHANNEL_ENABLE=0 in register SGDMA_CMDx [x=0..3])
5	M2M_INCOMPLETE	R	(Channel 0 only) Set(1) only when all of the following are true: • Memory-to-Memory transfer • MMU memory is the target • SGDMA_START_FIFO1 is not empty • SGDMA_SIZEHI0 = 0 • SGDMA_SIZELO0 = 0 • SGDMA Channel 0 is enabled • SGDMA Channel 1 is enabled

SGDMA_STSx [x=0..3] (BASE+8 – RESET = 0x00)			SGDMA_STS
BIT	**NAME**	**R**	**DESCRIPTION**
			Clear(0) when they are not all true
5	Reserved	R	Reserved – Read back as 0. (for Channels 1, 2, and 3)
4	Reserved	R	Reserved – Read back as 0.
3	DONE_FIFO_FULLx [x=0..3]	R	Set(1) when SGDMA_DONE_FIFOx [x=0..3] is full Clear(0) when SGDMA_DONE_FIFOx [x=0..3] is not full
2	DONE_FIFO_EMPTYx [x=0..3]	R	Set(1) when SGDMA_DONE_FIFOx[x=0..3] is empty Clear(0) when SGDMA_DONE_FIFOx[x=0..3] is not empty
1	START_FIFO_FULLx [x=0..3]	R	Set(1) when SGDMA_START_FIFOx [x=0..3] is full Clear(0) when SGDMA_START_FIFOx [x=0..3] is not full
0	START_FIFO_EMPTYx [x=0..3]	R	Set(1) when SGDMA_START_FIFOx [x=0..3] is empty Clear(0) when SGDMA_START_FIFOx [x=0..3] is not not empty

Table 71 – SGDMA Command Register

SGDMA_CMDx [x=0..3] (BASE+9 – RESET = 0x00)			SGDMA_CMD
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	R/W	Reserved – Read back as 0
2	PCKT_HDR	R/W	Set(1) to indicate that a packet header is present. Clear(0) to indicate that there is no packet header. Applies only if MEM_OP is set(1).
1	MEM_OP	R/W	Set(1) to indicate a MMU memory operation. Clear(0) to indicate an ISA memory operation. • When clear(0), the packet size comes from the SGDMA_SIZEHIx/LOx [x=0..3] registers, whose value is decremented before being written into the 8237 count register. The memory address comes from the SGDMA_ADHIx/LOx [x=0..3] registers. • When set(1), and PCKT_HDR=0, the packet size comes from the SGDMA_SIZEHIx/LOx [x=0..3] registers, whose value is decremented before being written into the 8237 count register. The High byte of the address is based on the packet number, and the low byte of the address is 0. • When set(1), and PCKT_HDR=1, the high byte of the address is based on the packet number and the low byte of the address is 8. For a MemRd transfer, the packet size is read from the packet header – this value represents the number of bytes of payload data plus header, so this value minus 9 is written into the 8237 counter register. If the packet size in the header is 8 (indicating a zero-byte payload), no DMA transfer will occur but the SGDMA will but the packet number into the SGDMA_DONE_FIFOx [x=0..3]. For a MemWr transfer, the SGDMA_SIZEHIx/LOx [x=0..3] registers contain the number of bytes of payload data, so this value is decremented before being written into the 8237 count register, and this value plus 8 is written into the packet memory at an offset of 6.

SGDMA_CMDx [x=0..3] (BASE+9 – RESET = 0x00)			SGDMA_CMD
BIT	**NAME**	**R/W**	**DESCRIPTION**
0	CHANNEL_ ENABLE	R/W	Set(1) to enabled the channel. Clear(0) to disable the channel. If this bit is cleared during a SGDMA cycle (before TC), then: • If MEM_OP=1, the packet number is put into the SGDMA_DONE_FIFOx [x=0..3] and the SGDMA returns to idle, ready for another cycle. • If MEM_OP=0, the SGDMA returns to idle and is ready for another cycle. • If a memory-to-memory transfer, the packet number for the MMU channel is put into the SGDMA_DONE_FIFO0x [0 or 1], but none of the updates to the address or size register occur, and the SGDMA returns to idle and is ready for another cycle.

Note 1: The PKT_HDR and MEM_OP bits must not be changed unless the channel is disabled (CHANNEL_ENABLRE=0 in register SGDMA_CMDx [x=0..3]) and not busy (DMA_BUSY=0 in register SGDMA_STSx [x=0..3]).

Note 2: Memory-to-memory transfer is a special case and is handled as follows. A memory-to-memory transfer supports transfers only between MMU memory and ISA memory, in either direction. The source must be Channel 0, the destination must be Channel 1.

- MMU memory to ISA memory transfers: Initially, the MCU performs the following:
  1. Clears the SGDMA_SIZEHI0/LO0 registers.
  2. Sets the SGDMA_ADHI1/LO1 registers to the ISA Address.

  After each TC, the SGDMA will add the size of the completed transfer to the SGDMA_ADHI1/LO1 registers.

  If PKT_HDR=1 in the SGDMA_CMD0 register, the transfer size comes from the MMU packet header. If PKT_HDR=0 in the SGDMA_CMD0 register, the transfer size is the value in SGDMA_SIZEHI1/LO1 registers.

- ISA memory to MMU memory transfers: Initially, the MCU performs the following:
  1. Sets the SGDMA_ADHI0/LO0 to the ISA Address.
  2. Sets the SGDMA_SIZEHI0/LO0 to the ISA buffer size.
  3. Sets the SGDMA_SIZEHI1/LO1 to the session transfer size.

  After each TC, the SGDMA will add the session transfer size to the SGDMA_ADHI0/LO0 registers and subtract the session transfer size from the SGDMA_SIZEHI0/LO0 registers.

The actual transfer size is the lesser of the values in the SGDMA_SIZEHI1/ LO1 and the SGDMA_SIZEHI0/LO0 registers.
If PKT_HDR=1 in register SGDMA_CMD1, then the actual transfer size plus 8 will be written to the packet memory at an offset of 6.
When the SGDMA_SIZEHI0/LO0 registers reach a value of 0, the ISA buffer has been completely transferred. If the SGDMA_START_FIFO1 is not yet empty when the ISA buffer has been completely transferred, the M2M_INCOMPLETE bit in the SGDMA_STS0 register will be set.

- During a memory-to-memory transfer, if either Channel 0 or Channel 1 is Disabled during the SGDMA cycle, then the packet number for the MMU channel is put into the SGDMA_DONE_FIFOxx [0 or 1]. However, none of the updates to the address or size registers occur, and the SGDMA returns to idle and is ready for another cycle.

- For memory-to-memory transfers, the SGDMA_ADHI0, SGDMA_ADLO0, SGDMA_ADHI1, SGDMA_ADLO1, SGDMA_SIZEHI0, and SGDMA_SIZELO0 must not be read by the MCU until both channels 0 and 1 are not Busy (DMA_BUSY=0 in SGDMA_STS0 and SGDMA_STS1).

PIO REGISTER DESCRIPTION

Table 72 – Upper Byte of the PIO ISA Address

PIO_ADHI (0x7FD8 – RESET = 0x00)			PIO_ADHI
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:4]	Reserved	R/W	Reserved – Read back as 0
[3:0]	PIOADDRHI	R/W	The upper 4 bits of the ISA address

Table 73 –Middle Byte of the PIO ISA Address

PIO_ADMID (0x7FD9 – RESET = 0x00)			PIO_ADMID
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	PIOADDRMID	R/W	The middle byte of the ISA address

Table 74 –Low Byte of the PIO ISA Address

PIO_ADLO (0x7FDA – RESET = 0x00)			PIO_ADLO
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	PIOADDRLO	R/W	The low byte of the ISA address

Note 1: The MCU must not write these registers until PIO_BUSY=0 in the PIO_CSR register
Note 2: PIO Read transfers and the first transfer of PIO Read Multiple transfers are initiated by writing to the PIO_ADLO register.

Table 75 – PIO Data Register

PIO_DATA (0x7FDB – RESET = 0x00)			PIO_DATA
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	PIODATA	R/W	The Data for the PIO transfer

Note 1: The MCU must not read this register until PIO_BUSY=0 in the PIO_CSR register.
Note 2: PIO Write transfers are initiated by writing to the PIO_DATA register.
PIO Read Multiple transfers (except for the first transfer) are initiated by reading the PIO_DATA register

Table 76 – PIO Command/Status Register

PIO_CSR (0x7FDC – RESET = 0x00)			PIO_CSR			
BIT	**NAME**	**R/W**	**DESCRIPTION**			
7	PIO_BUSY	R	Will be set(1) when a PIO transfer is in progress			
6	Reserved	R/W	Reserved – Read back as 0			
[5:3]	STROBEWIDTH	R/W	Programs the width of the RD/WR strobes (nIOR, nIOW, nMEMR, nMEMW)			
			5	**4**	**3**	**Strobe Width**
			0	0	0	Invalid
			0	0	1	Invalid
			0	1	0	Invalid
			0	1	1	3 DMACLKs
			1	0	0	4 DMACLKs
			1	0	1	5 DMACLKs
			1	1	0	6 DMACLKs
			1	1	1	Invalid
			For ISA Read accesses, the data must be valid one DMACLK cycle before the trailing edge of the read strobe.			
2	MEMORY	R/W	Set(1) to indicate an ISA memory transfer. Clear(0)to indicate an ISA I/O transfer.			
[1:0]	TRANSFERTYPE	R/W	**1**	**0**	**Transfer Type**	
			0	0	Disable	
			0	1	Write	
			1	0	Read	
			1	1	Read Multiple	

Note 1: The STROBEWIDTH and MEMORY bits of the PIO_CSR register must not be changed unless TRANSFERTYPE=00 (disabled) and PIO_BUSY=0 in the PIO_CSR register.

Note 2: The Ready input can be used to stretch the width of the RD/WR strobes generated by the PIO. When the READY input is high, the strobes will be as programmed in the PIO_CSR register.

- PIO Writes: In order to affect the nIOW or nMEMW strobes, the READY signal must go low at least 2 DMACLK periods before the scheduled rising edge of the strobe, and then the strobe will remain low until 3 DMACLK periods after READY goes high. PIO Reads: In order to affect the nIOR or nMEMW strobes, the READY signal must go low at least 3 DMACLK periods before the scheduled rising edge of the strobe, and then the strobe will remain low for 5 DMACLK periods after READY goes high. The read Data must be valid within 4 DMACLKS after READY goes high so that the Data is available at least 1 DMACLK period before the rising edge of the read strobe.

MEMORY MANAGEMENT UNIT (MMU) REGISTER DESCRIPTION

MMU Interface Registers

Table 77 - MMU Data Window Register

MMU_DATA (0x6000)			MMU DATA WINDOW REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	[D7:D0]	R/W	Data Packet Window. When RCV in the PRH register = '1', this is the byte pointed to by the packet number on the top of the RXFIFO, and the packet offset of PRH:PRL. When RCV in the PRH register = '0', this is the byte pointed to by the packet number in the PNR register, and the packet offset of PRH:PRL.

Note:

1. The Read FIFO may take at most 1.218µs after the PNH is written to present valid data.
2. The Write FIFO may take at most 2.520µs after writing the last byte of data to the FIFO to finish writing that data to the buffer.
3. The worst case sequential access times to the FIFOs while the 8237 is simultaneously arbitrating for the MMU, and a USB packet is currently being transferred, is 588ns.
    a) (READ) Therefore, after changing the PRH register, the 8051 should wait at least 2 instruction cycles (at 12MHz) before reading from this register. After waiting, the 8051, in auto-increment mode (PRH bit 6=1), can read a byte every cycle (at up to 16MHz).
    b) (WRITE) The data register mode can be switched to write at any time, and data can be written immediately on every instruction cycle. After writing data, the 8051 should wait at least 3 instruction cycles (at 12MHz) before changing the PNR or PRH :PRL registers for a Read . Again, after waiting 1.218µs, the 8051 can read a byte every instruction cycle.

Each endpoint will have a three bit up/dn counter which will maintain the number of packets queued for transmit at that endpoint. These counters are readable through these registers (there are sixteen – from 7F40 to 7F4F).

Table 78 – Tx FIFO Counter

TX_FIFOx (0x7F40-0x7F4F)			Tx FIFO Counter
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	R	Reserved
[2:0]	TxFIFO Count	R	This field will contain the number of packets that are queued for transmit at each endpoint. It is incremented when there is a push on the Tx FIFO of the corresponding endpoint, and it is decremented when there is a pop on the Tx FIFO of the corresponding endpoints.

Table 79 - Pointer Register (Low)

PRL (0x7F50)			POINTER REGISTER (LOW)
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	A[7:0]	R/W	LSB of the (0-1277 Max) offset of the allocated Packet Pointed to by PNR. The byte(s) pointed to by this register can be read and written to by the MCU at 0x6000.

Note 1: This register must be written before PRH .
Note 2: The value read from this register is not necessarily what was last written to it, but actually the last address used to access the buffer RAM.

Table 80 - Pointer Register (High)

PRH (0x7F51)			POINTER REGISTER (HIGH)
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	RCV	R/W	0 = The packet at 0x6000 is the packet pointed to by the PNR register. 1 = The packet available at 0x6000 is the packet pointed to by the packet on the top of the RX Packet Number FIFO.
6	AUTO_INCR	R/W	0 = Auto-increment is disabled 1 = Causes the PRH:PRL register to be automatically incremented each time the 0x6000 data window is accessed.
5	READ	R/W	Data register direction. This bit is required for the MMU/Arbiter to provide a transparent interface to the buffer RAM for the MCU. When first set, the MMU immediately fills the read FIFO. The MCU must wait 2.5us (60 Arbiter clocks) after writing to the MMU_DATA register before changing this bit from '0' to '1'. 0 = WRITE 1 = READ
[4:3]	Reserved	R	Reserved

PRH (0x7F51)			POINTER REGISTER (HIGH)
BIT	**NAME**	**R/W**	**DESCRIPTION**
[2:0]	A[10:8]	R/W	MSB of the (0-1277 Max) offset of the allocated Packet Pointed to by PNR. The byte(s) pointed to by this register can be read and written to by the MCU at 0x6000.

Note: This register must be written after PRL for its value to take effect.

Table 81 - Transmit FIFO Select Register

MMUTX_SEL (0x7F52)			TRANSMIT FIFO SELECT REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:4]	Reserved	R	Reserved
[3:0]	EP[3:0]	R/W	This register selects which Endpoint Commands "110" and "111" will affect when issued to the MMU

Note: This register must be written before writing the “Enqueue Packet into Endpoint x” or the “Reset TX Endpoint x” command to the MMUCR .

Table 82 - MMU Command Register

MMUCR (0x7F53)			MMU COMMAND REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:5]	MMU_CMD	W	MMUCR COMMAND SET
4	Reserved	W	Reserved, writes are ignored and read return “0”
[3:0]	N[3:0]	W	Number of 128 byte Pages. N[3..0]=0000 indicates 1 page, and N[3..0]=1001 indicates 10 pages, or 1280 bytes.

MMU COMMAND Bits 7, 6, and 5 Description:

000 NOOP, No operation

001 Allocate Memory : N3-0 specify how many 128 byte pages to allocate for that packet (up to 10 pages allowed (1280 bytes) per packet.) Immediately generates a "FAILED" code at the ARR and the code is cleared when complete. Can generate an ALLOC interrupt to MCU upon completion. When an allocation request cannot be completed due to insufficient memory, the FAILED bit in the ARR will remain set. Any subsequent release of memory pages (by either the MMUCR or the SIEDMA) will cause the MMUCR to automatically continue the allocate command until all requested pages have been successfully allocated. Software should never issue another allocate command until the previous allocate command has been successfully completed.

010 RESET MMU : Frees all buffer RAM, clears interrupts, and resets queue pointers.

011 Remove Packet from top of RX Queue : To be issued after MCU has completed processing the packet number at the RXFIFO.

100 Remove and Release Top of RXFIFO : Same as (011), but also frees all memory used by the packet. This command is especially useful as a quick way to "ignore" bad packets.

101 Release specific Packet : Frees all pages allocated to the packet specified in the PNR.

110 Enqueue Packet into Endpoint x : Places the Packet number indicated by the PNR register in the transmit queue of the endpoint pointed to by the MMUTX_SEL register. The MMUTX_SEL register must be written before this command is issued.

111 Reset TX Endpoint x : Resets the TX FIFO holding the packet numbers awaiting transmission and the TXFIFO_STAT bits of the endpoint pointed to by the MMUTX_SEL register. The MMUTX_SEL register must be written before this command is issued. This command does not release any memory allocated to packets that are dequeued.

Table 83 - Allocation Result Register

ARR (0x7F54)			**ALLOCATION RESULT REGISTER**
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	FAILED	R	
[6:5]	Reserved	R	Reserved
[4:0]	P[4:0]	R	Returns Packet Number (0-31, 0x00-0x1F) from an allocation command. This can be written directly into the PNR register

Table 84 - Packet Number Register

PNR (0x7F55)			**PACKET NUMBER REGISTER**
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:5	Reserved	R	Reserved
[4:0]	P[4:0]	R/W	Packet selector to access packet at 0x6000 buffer window

MMU Free Pages Register

MMU Free Pages bits, and a global NAK_ALLRX (this can only NACK OUT and Bulk packets) control bit for the firmware to view the real time status of the 32 page allocation bits. This allows the MCU to set NAK_ALLRX which would inhibit the SIE from asking the SIEDMA to allocate packets, MCU checks how many pages are left, issue an allocate if enough are free, and then release the SIE/SIEDMA. For the current design, the number of free pages would range from 0x00 to 0x1F (32) pages left unallocated.

The indication of pages free may be invalid during an allocation or deallocation.

Table 85 - PAGES FREE IN THE MMU

PAGS_FREE (0x7F56 - RESET=0x20)			**PAGES FREE IN THE MMU**
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	NAK_ALLRX	R/W	NACK All received packets 0 = Normal Operation (Default) 1 = NACK all RX packets
6	Reserved	0	Reserved
[5:0]	PAGS_FREE	R	These bits indicate the number of free pages in the MMU.

Note 1: Firmware can set a NAK_ALLRX bit to inhibit the SIE from asking the SIEDMA to allocate any pages while the MCU is observing the page free bits.

Note 2: This register is used to indicate how many pages are left in many situations, including after an RX_OVRN, before a multi-packet allocation, etc. This eliminates the possibility of a failed allocation, simplifying software without adding additional hardware to abort an allocation.

32 BYTE DEEP TX COMPLETION FIFO REGISTER

Table 86 - TX Management Register 2

TX_MGMT (0x7F57 - RESET=0x80)			TX Management Register
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	CTX_EMTY	R	Completed TX FIFO empty status 0 = Has one or more TX packet 1 = Empty
6	CTX_FULL	R	Completed TX FIFO full status 0 = Not FULL 1 = FULL
5	Reserved	R	Reserved
[4:0]	CTX_FIFO	R	This is the data port for the 32 deep TX completion FIFO. This FIFO is automatically updated by hardware with the last successfully completed transmit packet. It is the responsibility of software to ensure that this FIFO never overflows and/or becomes full.

Table 87 - Receive Packet Number FIFO Register

RXFIFO (0x7F58)			NEXT RX PACKET NUMBER FIFO REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	RXFIFO_EMPTY	R	1 = No pending packets from the host to be processed
6	RXFIFO_FULL	R	1 = The SIEDMA will not accept packets from the host (via RX Overflow)
5	Reserved	R	Reserved
[4:0]	P[4:0]	R	Packet Number When a packet has been received, and the 8-byte header has been written by the SIEDMA, the associated Packet Number is placed in this FIFO.

A "complete" reception requires that the 8 byte status header is correctly written into the packet buffer, with the correct data, and moved into the RX Packet Number FIFO. A "successful" reception requires that the CRC and PID check bits of a "complete" reception are good. The hardware queues only "complete" packets. Firmware must determine if "complete" packets were "successful". Corrupted token packets causes the complete data payload to be ignored.

Tx FIFO POP Register

This register is used to help software manage TX Queues. This will provide a method to handle a CLEAR_FEATURE:ENDPOINT_STALL condition gracefully. When read, this register will return the Packet Number of the next packet waiting on the TX queue pointed to by MMUTX_SEL register, AND it will pop that Packet Number off of the selected TX FIFO.

Table 88 - POP TX FIFO

POP_TX (0x7F59 – RESET=0x80)			POP TX FIFO
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	POPTX_STAT	R	POP TX FIFO empty status 0 = Has one or more TX packet 1 = Empty
[6:5]	Reserved	R	Reserved
[4:0]	POP_TX	R	This 5 bit value is the packet number or handle that is at the top of the TX FIFO pointer to by MMUTX_SEL. The TX FIFO is popped when this register is read.

Note: It is the software's responsibility to ensure that the appropriate TX EP is disabled during this operation, and to issue a deallocate command if desired.

Table 89 - Transmit FIFO Status Register A

TXSTAT_A (0x7F60 - RESET=0x55)			TRANSMIT FIFO STATUS REGISTER A
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EP3TX_EMPTY	R	Endpoint 3 Transmit Packet FIFO Status Bits [7:6]='11' Invalid Bits [7:6]='10' Empty (No Packets queued) Bits [7:6]='01' Full (5 Packets queued) Bits [7:6]='00' Partially Full (1, 2, 3, or 4 Packets queued)
6	EP3TX_FULL	R	
5	EP2TX_EMPTY	R	Endpoint 2 Transmit Packet FIFO Status Bits [5:4]='11' Invalid Bits [5:4]='10' Empty (No Packets queued) Bits [5:4]='01' Full (5 Packets queued) Bits [5:4]='00' Partially Full (1, 2, 3, or 4 Packets queued)
4	EP2TX_FULL	R	
3	EP1TX_EMPTY	R	Endpoint 1 Transmit Packet FIFO Status Bits [3:2]='11' Invalid Bits [3:2]='10' Empty (No Packets queued) Bits [3:2]='01' Full (5 Packets queued) Bits [3:2]='00' Partially Full (1, 2, 3, or 4 Packets queued)
2	EP1TX_FULL	R	
1	EP0TX_EMPTY	R	Endpoint 0 Transmit Packet FIFO Status Bits [1:0]='11' Invalid Bits [1:0]='10' Empty (No Packets queued) Bits [1:0]='01' Full (5 Packets queued) Bits [1:0]='00' Partially Full (1, 2, 3, or 4 Packets queued)
0	EP0TX_FULL	R	

Table 90 - Transmit FIFO Status Register B

STAT_B (0x7F61 - RESET=0x55)			TRANSMIT FIFO STATUS REGISTER B
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EP7TX_EMPTY	R	Endpoint 7 Transmit Packet FIFO Status Bits [7:6]='11' Invalid Bits [7:6]='10' Empty (No Packets queued) Bits [7:6]='01' Full (5 Packets queued) Bits [7:6]='00' Partially Full (1, 2, 3, or 4 Packets queued)
6	EP7TX_FULL	R	
5	EP6TX_EMPTY	R	Endpoint 6 Transmit Packet FIFO Status Bits [5:4]='11' Invalid Bits [5:4]='10' Empty (No Packets queued) Bits [5:4]='01' Full (5 Packets queued) Bits [5:4]='00' Partially Full (1, 2, 3, or 4 Packets queued)
4	EP6TX_FULL	R	
3	EP5TX_EMPTY	R	Endpoint 5 Transmit Packet FIFO Status Bits [3:2]='11' Invalid Bits [3:2]='10' Empty (No Packets queued) Bits [3:2]='01' Full (5 Packets queued) Bits [3:2]='00' Partially Full (1, 2, 3, or 4 Packets queued)
2	EP5TX_FULL	R	
1	EP4TX_EMPTY	R	Endpoint 4 Transmit Packet FIFO Status Bits [1:0]='11' Invalid Bits [1:0]='10' Empty (No Packets queued) Bits [1:0]='01' Full (5 Packets queued) Bits [1:0]='00' Partially Full (1, 2, 3, or 4 Packets queued)
0	EP4TX_FULL	R	

Table 91 - Transmit FIFO Status Register C

TXSTAT_C (0x7F62 - RESET=0x55)			TRANSMIT FIFO STATUS REGISTER C
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EP11TX_EMPTY	R	Endpoint 11 Transmit Packet FIFO Status Bits [7:6]='11' Invalid Bits [7:6]='10' Empty (No Packets queued) Bits [7:6]='01' Full (5 Packets queued) Bits [7:6]='00' Partially Full (1, 2, 3, or 4 Packets queued)
6	EP11TX_FULL	R	
5	EP10TX_EMPTY	R	Endpoint 10 Transmit Packet FIFO Status Bits [5:4]='11' Invalid Bits [5:4]='10' Empty (No Packets queued) Bits [5:4]='01' Full (5 Packets queued) Bits [5:4]='00' Partially Full (1, 2, 3, or 4 Packets queued)
4	EP10TX_FULL	R	
3	EP9TX_EMPTY	R	Endpoint 9 Transmit Packet FIFO Status Bits [3:2]='11' Invalid Bits [3:2]='10' Empty (No Packets queued) Bits [3:2]='01' Full (5 Packets queued) Bits [3:2]='00' Partially Full (1, 2, 3, or 4 Packets queued)
2	EP9TX_FULL	R	
1	EP8TX_EMPTY	R	Endpoint 8 Transmit Packet FIFO Status Bits [1:0]='11' Invalid Bits [1:0]='10' Empty (No Packets queued) Bits [1:0]='01' Full (5 Packets queued) Bits [1:0]='00' Partially Full (1, 2, 3, or 4 Packets queued)
0	EP8TX_FULL	R	

Table 92 - Transmit FIFO Status Register D

TXSTAT_D (0x7F63 - RESET=0x55)			TRANSMIT FIFO STATUS REGISTER D
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EP15TX_EMPTY	R	Endpoint 15 Transmit Packet FIFO Status Bits [7:6]='11' Invalid Bits [7:6]='10' Empty (No Packets queued) Bits [7:6]='01' Full (5 Packets queued) Bits [7:6]='00' Partially Full (1, 2, 3, or 4 Packets queued)
6	EP15TX_FULL	R	
5	EP14TX_EMPTY	R	Endpoint 14 Transmit Packet FIFO Status Bits [5:4]='11' Invalid Bits [5:4]='10' Empty (No Packets queued) Bits [5:4]='01' Full (5 Packets queued) Bits [5:4]='00' Partially Full (1, 2, 3, or 4 Packets queued)
4	EP14TX_FULL	R	
3	EP13TX_EMPTY	R	Endpoint 13 Transmit Packet FIFO Status Bits [3:2]='11' Invalid Bits [3:2]='10' Empty (No Packets queued) Bits [3:2]='01' Full (5 Packets queued) Bits [3:2]='00' Partially Full (1, 2, 3, or 4 Packets queued)
2	EP13TX_FULL	R	
1	EP12TX_EMPTY	R	Endpoint 12 Transmit Packet FIFO Status Bits [1:0]='11' Invalid Bits [1:0]='10' Empty (No Packets queued) Bits [1:0]='01' Full (5 Packets queued) Bits [1:0]='00' Partially Full (1, 2, 3, or 4 Packets queued)
0	EP12TX_FULL	R	

Table 93 - TX Management Register 1

TX_MGMT (0x7F67 - RESET=0x00)			TX Management Register 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:1]	Reserved	R	Reserved
0	MEM_DALL	R/W	Memory deallocate Mode 0 = Auto 1 = Manual deallocation, but the TX FIFO Pop is still automatic. This control bit selects between Auto and Manual memory pages deallocation. This bit should be statically set at the start of operation, and can not be changed during or if about to transmit. This bit defaults to "0" for normal operation. When set, the MCU handles freeing up the memory pages.

SERIAL INTERFACE ENGINE (SIE) REGISTER DESCRIPTION

Packet Header Definition

The following header contains information to determine endpoint, status, length of the received packet, and the payload "received data".

Table 94 - Packet Header Definition

OFFSET	MSB 7	6	5	4	3	2	1	LSB 0
<n + 7	Payload Data Byte n-1 (n is the payload data size, which is Byte Count -8) - For 0 Length Packet, Byte Count = 0x008 - For 1 byte Packet, Byte Count = 0x009 - For any Packet, (Byte Count-1) points to last byte of payload data							
0x008	Payload Data Byte 0							
0x007	0	0	0	0	0	BYTE COUNT[10..8]		
0x006	BYTE COUNT[7..0]							
0x005	EXTENDED FRAME COUNT[15..11]					FRAME COUNT[10..8]		
0x004	FRAME COUNT[7..0]							
0x003	RESERVED							
0x002	0	TMP_ADDRESS[6..0]						
0x001	0	0	0	0	PACKET ID[3..0]			
0x000	Bad_CRC	Last_TOG	Bad_TOG	0	ENDPOINT[3..0]			

Packet Description:

1. Offset 0 to 7 is the packet header.
   a) Offset 0x000 to 0x005 is generated by the SIE.
      i) Offset 0x000 bit bit 5 - Bad_TOG- This bit is set when the SIE receives an unexpected toggle. This is not necessarily an error condition, This bit could indicate a condition when the return handshake packet is lost. Note that this bit is not set for isochronous transfers.

LAST PACKET TOGGLE VALUE	CURRENT PACKET TOGGLE VALUE	"BAD TOG" BIT
0	0	1
0	1	0
1	0	0
1	1	1

      ii) Offset 0x000 bit Last_TOG is the last toggle bit received.
      iii) Offset 0x000 bit Bad_CRC, is set when the SIE detects a bad CRC.
   b) Offset 0x006 to 0x007 is generated by the SIEDMA.
3. Offset 8 to n+7 is the actual data received from the USB bus and stored in memory.

SIE Interface Registers

The architecture of the USB97C102 is such that there are no data FIFO's associated with individual endpoints. The MMU does not differentiate packets by endpoint number. The firmware must read the endpoint number from the packet header to pass the packet on to the appropriate endpoint handler. This makes the chip dynamic and flexible in allocating buffers to store any payload size from 0 to 1280 bytes. Each endpoint can be configured separately via the following register.

Table 95 - Endpoint Control Registers

EP_CTRL[15..0] (0x7F8F-0x7F80 - RESET=0x00)			ENDPOINT CONTROL REGISTERS
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	TX_ISO	R/W	Bit 7 instructs the SIE how to handle handshakes for transmit endpoints during "IN" transactions, and how the SIEDMA engine should handle packet queue status after packet transmission. When a TX endpoint is configured for isochronous operation (Bit 7 = '1'), all packet transmissions are considered successful and the SIEDMA must move the packet number into the TX Completion FIFO. When the TX endpoint is non-isochronous (Bit 7 = '0'), then the SIE must receive a valid ACK handshake from the host before the packet is released. This guarantees data integrity for non-isochronous transactions. Successfully transmitted packets are automatically de-queued and the packet is released. 0 = Non-Isochronous 1 = Isochronous
6	RX_ISO	R/W	Bit 6 instructs the SIE how to handle handshakes for receive endpoints during "OUT" and "SETUP" transactions. Once a packet matches the 7-bit Function Address, the SIE must begin page allocation and generate a new packet in buffer RAM. The MCU must check PID_Valid and CRC_Valid bits and dequeue "bad" packets. The SIE will use bit 6 to inhibit handshakes when enabled. 0 = Non-isochronous 1 = Isochronous
5,3	TX_CONT[1:0]	R/W	0,0= Endpoint is disabled, and does not send handshakes. 0,1= Send a STALL handshake for an IN transaction directed at this EP. 1,0= Normal Operation. ACK or NAK is sent depending on whether data is in the EPXs TX_QUEUE. 1,1= Send a NAK handshake for an IN transaction directed at this EP, regardless of TX_QUEUE status. (Note 3)
4,2	RX_CONT[1:0]	R/W	0,0= Endpoint is disabled, and does not send handshakes. 0,1= Send a STALL handshake for an OUT transaction directed at this EP. 1,0= Normal Operation. ACK or NAK is sent depending on RX_OK status 1,1= Send a NAK handshake for an OUT transaction directed at this EP (Note 1)

EP_CTRL[15..0] (0x7F8F-0x7F80 - RESET=0x00)			ENDPOINT CONTROL REGISTERS
BIT	**NAME**	**R/W**	**DESCRIPTION**
1	TX_TOGGLE	R/W	This bit is toggled after each successful transmission. TX_TOGGLE can be reset or cleared by the MCU but the MCU must insure that the endpoint is disabled before modifying them. For isochronous transmits, this bit won't be toggled by the hardware.
0	RX_TOGGLE	R	This bit reflects the last DATA0/DATA1 toggle. For isochronous receives, this bit will still hold the received toggle, but it won't be checked for Toggle error.

Note 1: There is one Endpoint Control Register per virtual endpoint. When the SIE decodes a token, the endpoint number is used to index which EP_CTRL register bits should be used to respond to the SIE and SIEDMA.

Note 2: This register allows firmware to throttle back RX packets to any specific endpoint(s) until the firmware decides congestion has subsided.

Note 3: If the firmware needs to STALL an endpoint, it should first be taken off-line by setting RX_CONT1=0, and then RX_CON0=1.

Note 4: This allows firmware to manage TX endpoint(s) and hold queued data until the firmware is ready, even if the host is asking. This is not as critical as the RX version, but it may be required for Isochronous synchronization, as well as STALL recovery.

Note 5: These registers can be written to at any time but the SIE won't be affected until after the current transaction (on the particular endpoint) is completed. If a particular register is written several times during an SIE transaction, only the last value written will take effect after the SIE transaction is complete.

Table 96 - LSB FRAME Count Register

FRAMEL 0x7F90 Reset 0x00			FRAME COUNT REGISTER (LOW)
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	FRAME[7:0]	R	The 11 bit Frame Number from each SOF packet is loaded with the RISING edge of EOT when SOF_TOKEN = '1' and ACK = '1'.

Note: This register is always the last correctly received valid SOF Frame number. Garbled and invalid SOF tokens do not alter this register. However, the LSB FRAME/MSB FRAME Count registers will be incremented by the hardware when a missing SOF is detected, or when there is an error in frame number. A SOF is defined as missing if it does not occur within a range of +/- 3 USBCLKs from the expected frame length. The expected frame length is the previous interval between SOFs.

Table 97 - MSB FRAME Count Register

FRAMEH 0x7F91 Reset 0x00			FRAME COUNT REGISTER (HIGH)
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	EXT_FR[15:11]	R	Extended Frame Count. The extended count bits are loaded with the RISING edge of EOT when SOF_TOKEN = '1' and ACK = '1'. The extended Frame count bit must also be enabled (EN_EXTFRAME = '1' in SIE_CONFIG).
[2:0]	FRAME[10:8]	R	Frame Number from each SOF packet is loaded with the RISING edge of EOT when SOF_TOKEN = '1' and ACK = '1'.

Note: This register is always the last correctly received valid SOF Frame number. Garbled and invalid SOF tokens do not alter this register. However, the LSB FRAME/MSB FRAME Count registers will be incremented by the hardware when a missing SOF is detected, or when there is an error in frame number. A SOF is defined as missing if it does not occur within a range of +/- 3 USBCLKs from the expected frame length. The expected frame length is the previous interval between SOFs.

Table 98 - Local Address Register

SIE_ADDR (0x7F92 RESET=0x00)			LOCAL ADDRESS REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	RX_ALL	R/W	1 = Overrides the token address decoding of the SIE such that no compare is done. Token CRC is also ignored when RX_ALL=1. This bit forces all packets transmitted on the wire to be received in the RX Packet Queue
[6:0]	ADDR[6:0]	R/W	This register is only written by the 8051. It is the SIE's local address assigned during enumeration. This SIE address allows Endpoints 0 through 3 to be available. This address can be used for the HUB address.

Note: When RX_ALL is enabled, software should not enable any TX endpoints as they will respond to any Address with the same endpoint and possibly cause contention on the line. Software should also set each RX endpoint RX_ISO bit to prevent handshakes from being sent.

Table 99 - Alternate Address 1 Register

ALT_ADDR1 (0x7F99 - RESET=0x00)			ALTERNATE SIE ADDRESS 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EN_ALTADDR1	R/W	Alternate address. 1 = Enabled, this bit allows Endpoints 15 through 0 to be available as Endpoint ALT0 through Endpoint ALT15. In other words, the SIE can respond to two addresses with up to 32 endpoints distributed between them. 0 = Disabled, this register does not affect EP_OK generation.
6	ALT6	R/W	Alternate address bit 6
5	ALT5	R/W	Alternate address bit 5
4	ALT4	R/W	Alternate address bit 4
3	ALT3	R/W	Alternate address bit 3
2	ALT2	R/W	Alternate address bit 2
1	ALT1	R/W	Alternate address bit 1
0	ALT0	R/W	Alternate address bit 0

Note 1: The Firmware (8051) must make sure that endpoint configurations do not overlap.

Table 100 - Alternate Address 2 Register

ALT_ADDR2 (0x7F9E – RESET=0x00)			ALTERNATE SIE ADDRESS 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EN_ALTADDR2	R/W	Alternate address 2. 1 = Enabled, this bit allows Endpoints 8 through 11 to be available to this address. 0 = Disabled, this register does not affect EP_OK generation.
6	ALT6	R/W	Alternate address bit 6
5	ALT5	R/W	Alternate address bit 5
4	ALT4	R/W	Alternate address bit 4
3	ALT3	R/W	Alternate address bit 3
2	ALT2	R/W	Alternate address bit 2
1	ALT1	R/W	Alternate address bit 1
0	ALT0	R/W	Alternate address bit 0

Table 101 - Alternate Address 3 Register

ALT_ADDR3 (0x7F9F – RESET=0x00)			ALTERNATE SIE ADDRESS 3
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EN_ALTADDR 3	R/W	Alternate address 3. 1 = Enabled, this bit allows Endpoints 12through 15 to be available to this address. 0 = Disabled, this register does not affect EP_OK generation.
6	ALT6	R/W	Alternate address bit 6
5	ALT5	R/W	Alternate address bit 5
4	ALT4	R/W	Alternate address bit 4
3	ALT3	R/W	Alternate address bit 3
2	ALT2	R/W	Alternate address bit 2
1	ALT1	R/W	Alternate address bit 1
0	ALT0	R/W	Alternate address bit 0

Table 102 - SIE Status Register

SIE_STAT (0x7F93 - RESET=0x03)			SIE STATUS REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	ERR	R	Indicates that an error occurred during the last USB transaction. Considered valid on the rising edge of EOT
6	TIMEOUT	R	Indicate that the last USB transaction ended because of an inter-packet time out condition (i.e.:>16 bit times). Considered valid on the rising edge of EOT.
5	SETUP_TOKEN	R	Indicates that the token received was a SETUP token.
4	SOF_TOKEN	R	Indicates that the SOF PID has been received. Considered valid when EOT is '0'.
3	PRE_TOKEN	R	Indicates that the SIE detected a PRE (preamble) packet on the USB bus. The signal is asserted when the SIE has seen a valid SYNC followed by a valid PRE PID.
2	ACK	R	Indicates that the last USB transaction was completed without error or time-out. Considered valid on the rising edge of EOT.
1	USB_RESET	R	When active '1', it indicates that the USB line is being reset. This signal is asserted when the SIE detects a string of single - ended 0's on the bus for a long time. [During USB_RESET, this bit is set(1). When the firmware sends a system reset, this bit is cleared(0)]
0	EOT	R	End - of - Transaction. On transition to a '1', it indicates the end of transaction. On transition to a '0' it indicates the beginning of a new transaction.

Note: This read only register reflects the status signals from the SIE state machine. This register can be polled for test purposes, or by error handling routines for recovery.

Table 103 - SIE Control Register 1

SIE_CTRL1 (0x7F94 - RESET=0x00)			SIE CONTROL REGISTER 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	SIEDMA_DISABLE	R/W	0 = Normal operation 1 = Inhibits SIEDMA operation to facilitate MCU override
6	FORCE_RXOK	R/W	Forces SIE to send Acknowledge during receive. Must be '0' for normal operation.
5	FORCE_TTAG	R/W	0 = Normal operation 1 = Signals that the next byte written to the SIE TX_FIFO is the last payload byte.
4	FORCE_RXOVFLO	R/W	0 = Normal operation. 1 = Forces the SIE to generate RXOVFLO and clear the SIE RX FIFO.
3	FORCE_TXABORT	R/W	0 = Normal operation 1 = Forces a bit-stuff error at the host
2	FORCE_EOT	R/W	0 = Normal operation 1 = Forces an End-of-Transaction for the SIE
1	RTAG_IN	R	Status of RTAG signal from SIE RX FIFO
0	TXOK_IN	R	Status of TXOK from SIE

Note: Bits 7:2 must be set to “0” for normal operation. Altering these bits will cause an abnormal USB behavior.

Table 104 - SIE Configuration Register.

SIE_CONFIG (0x7F98 - RESET=0x40)			SIE CONFIGURATION REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	FSEN	R/W	This bit indicates that the USB97c102 supports 12Mbps USB data rates. This bit **must** be set to a one '1' for normal operation.
6	RST_SIE	R/W	1 = Resets the SIE
5	RST_FRAME	R/W	1 = Clears FRAMEL and Bit 0 through 2 of FRAMEH
4	EN_EXTFRAME	R/W	Extended Frame Count Enable. Expands the Frame count from 11 bits to 16 bits for 8051 use. 0 = Bits 7-3 of FRAMEH are driven to 0. 1 = Bits 7-3 of FRAMEH count 1-0 transitions of bit 2 in FRAMEH.
3	SIE_SUSPEND	R/W	1 = Forces the SIE into USB Suspend Mode. The MCU must determine that Suspend must be entered.
2	SIE_RESUME	R/W	1 = Forces the SIE to transmit Resume signaling on the line.
1	USB_RESUME	R	1 = Indicates Resume signaling has been detected on the line while in the Suspend State. This signal causes a Resume Power Management interrupt).
0	USB_RESET	R	1 = Indicates that the USB line is being reset. Asserted when SE0 is present on the bus for 32 or more 12 Mbps bit times. This causes a USB_RESET Power management interrupt.

Table 105 - SIE Control Register 2

SIE_CTRL 2 (0x7FA9 – RESET=0x00)			SIE CONTROL REGISTER 2		
BIT	**NAME**	**R/W**	**DESCRIPTION**		
7	Reserved	R/W	Reserved – This bit should always be cleared (0)		
6	Reserved	R/W	Reserved – This bit should always be cleared (0)		
5	Reserved	R/W	Reserved – This bit should always be cleared (0)		
4	Reserved	R/W	Reserved – This bit should always be cleared (0)		
3	Reserved	R/W	Reserved – This bit should always be cleared (0)		
2	SET_BUSY_ON_SETUP	R/W	When set (1), a setup pckt rcvd on a control endpoint will set that endpoint to busy in any direction it was not disabled. When clear (0), a setup pckt will have no effect on the endpoint's control condition.		
[1:0]	ISO_LIMIT	R/W	ISO_Limit – Defines the maximum size of an isochronous packet		
			1	**0**	**PAYLOAD SIZE**
			0	0	1023 byte payload
			0	1	512 byte payload total less 8 byte header = 504 byte payload
			1	0	256 byte payload total less 8 byte header = 248 byte payload
			1	1	256 byte payload total less 8 byte header = 248 byte payload

In conjunction with the Endpoint Control Registers defined above, the Endpoint Command Register allows the dynamic modification and configuration of specific endpoints. This register which is new to the SMSC Family of USB devices, allows the MCU to write each individual bit field within the existing register Endpoint set without having to do read / modify / write operations. The Firmware can jam this register with a full constant, or could OR-in an EP number.

This register allows the individual setting and clearing of the bits in the EP_CTRL registers.

Table 106 - Endpoint Command Register

EP_COMM (0x7FAA- RESET=0x00)			**ENDPOINT Command Register**			
BIT	**NAME**	**R/W**	**DESCRIPTION**			
7	TX/RX	R/W	This bit, when set (1) will allow the command specified in bits 6-4 to control the TX endpoint. When this bit is cleared (0), the command control the RX endpoint. In other words, if set (1), the command will affect TX_ISO, TX_ENABLE, STALL_TXEP, and TX_TOGGLE (defined in EPCTRL). If clear (0), the command will affect RX_ISO, RX_ENABLE, STALL_RXEP, and RX_TOGGLE (also defined in EPCTRL).			
[6:4]	COMMAND	R/W	**6**	**5**	**4**	**COMMAND BITS**
			0	0	0	Endpoint is disabled, and does not send handshakes.
			0	0	1	Send a STALL handshake for an IN transaction directed at this EP.
			0	1	0	Normal Operation. ACK or NAK is sent depending on whether data is in the EPXs TX_QUEUE.
			0	1	1	Send a NAK handshake for an IN transaction directed at this EP, regardless of TX_QUEUE status. (Note 3)
			1	0	0	Clear Tx / Rx Toggle bit
			1	0	1	Set Tx / Rx Toggle bit
			1	1	0	Clear Tx / Rx ISO bit
			1	1	1	Set Tx / Rx ISO bit

EP_COMM (0x7FAA- RESET=0x00)			ENDPOINT Command Register				
BIT	**NAME**	**R/W**	**DESCRIPTION**				
[3:0]	EP_Select	R/W	**3**	**2**	**1**	**0**	**ENDPOINT SELECT**
			0	0	0	0	Endpoint 0
			0	0	0	1	Endpoint 1
			0	0	1	0	Endpoint 2
			0	0	1	1	Endpoint 3
			0	1	0	0	Endpoint 4
			0	1	0	1	Endpoint 5
			0	1	1	0	Endpoint 6
			0	1	1	1	Endpoint 7
			1	0	0	0	Endpoint 8
			1	0	0	1	Endpoint 9
			1	0	1	0	Endpoint 10
			1	0	1	1	Endpoint 11
			1	1	0	0	Endpoint 12
			1	1	0	1	Endpoint 13
			1	1	1	0	Endpoint 14
			1	1	1	1	Endpoint 15

Section 8.4.5.4 of the USB Spec V1.1 states that “If a non-control endpoint receives a SETUP PID, it must ignore the transaction and return no response.” In order for the hardware to do this correctly, it needs to know which endpoints are non-control endpoints.

Each bit of the NonControl Endpoint registers will correspond to the associated Endpoint. Bits 0-7 of the NonControl Endpoint 1 register will correspond to Endpoints 7-15. Bits 0-7 of the NonControl Endpoint 2 will correspond to Endpoints 0-7. The MCU will write these registers, and set the corresponding bit=1 for each endpoint that is a non-control endpoint. The hardware will not respond to a Setup PID for any endpoint whose corresponding bit is set (1).

Table 107 - NonControl Endpoint Register 1 (high endpoints)

NONCTRL_EP1 (0x7FAB – RESET=0x00)			NONCONTROL ENDPOINT REGISTER 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EP15	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID
6	EP14	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID
5	EP13	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID
4	EP12	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID
3	EP11	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID
2	EP10	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID
1	EP9	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID
0	EP8	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID

Table 108 - NonControl Endpoint Register 2 (low endpoints)

NONCTRL_EP2 (0x7FAC – RESET=0x00)			NONCONTROL ENDPOINT REGISTER 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EP7	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID.
6	EP6	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID.
5	EP5	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID.
4	EP4	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID.
3	EP3	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID.
2	EP2	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID.
1	EP1	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID.
0	EP0	R/W	When this bit is set (1), the Endpoint will not respond to a Setup PID. When this bit is cleared (0), the Endpoint will respond to a setup PID.

Table 109 – Packet Header FIFO

PCKTHEAD (0x7FAD – RESET=0x00)			PACKET HEADER FIFO
BIT	**NAME**	**R**	**DESCRIPTION**
[7:0]	PCKTHEAD	R	This eight bit register will contain the first byte of the packet at the top of the Rx Packet Number FIFO.

By reading this register, the MCU can directly access byte 0 of the packet at the top of the RX Packet Number FIFO.

The Memory Management Policy (MMP) feature permits limiting the number of received packets in memory per endpoint. A five bit up/down counter will be implemented for each endpoint. Each counter will be incremented by the MCU to initialize the limit, then decremented by the hardware as packets arrive at its corresponding endpoint, and incremented by the MCU after it releases the packet. If the count reaches 0, and the MMP feature is enabled, then the hardware will not receive the packet and will NAK non-isochronous OUT tokens. If the count is zero, it will not decrement further; if the count is 31, it will not increment further. The MCU can enable or disable this feature independently for each endpoint. The default condition is disabled.

The following register allows the MCU to access and control the up/down counters for each endpoint:

Table 110 – Memory Management Policy Command Register

MMPCMD (0x7FAE)			**ENDPOINT Command Register**			
BIT	**NAME**	**R/W**	**DESCRIPTION**			
[7:5]	COMMAND	R/W	**7**	**6**	**5**	**Command Bits**
			0	0	0	Disable the Memory Management Policy feature. If disabled, the endpoint counters will still count, but there will be no MMP action taken when the counter reaches zero.
			0	0	1	Enable the Memory Management Policy feature.
			0	1	0	Decrement the count – the MCU must have previously made the endpoint busy before executing this command.
			0	1	1	Increment the count – the MCU must have previously made the endpoint busy before executing this command.
			1	0	0	Get State – this will cause the count and the enable/disable state to be latched into the MMPSTAT register.
			1	0	1	Reserved
			1	1	0	Reserved
			1	1	1	Reset the counter to zero and disable the MMP feature.
4	Reserved	R/W	Reserved – Reads back as 0.			

MMPCMD (0x7FAE)			ENDPOINT Command Register				
BIT	**NAME**	**R/W**	**DESCRIPTION**				
[3:0]	EP_Select	R/W	**3**	**2**	**1**	**0**	**Endpoint Select**
			0	0	0	0	Endpoint 0
			0	0	0	1	Endpoint 1
			0	0	1	0	Endpoint 2
			0	0	1	1	Endpoint 3
			0	1	0	0	Endpoint 4
			0	1	0	1	Endpoint 5
			0	1	1	0	Endpoint 6
			0	1	1	1	Endpoint 7
			1	0	0	0	Endpoint 8
			1	0	0	1	Endpoint 9
			1	0	1	0	Endpoint 10
			1	0	1	1	Endpoint 11
			1	1	0	0	Endpoint 12
			1	1	0	1	Endpoint 13
			1	1	1	0	Endpoint 14
			1	1	1	1	Endpoint 15

Table 111 – Memory Management Policy State Register

MMPSTATE 0x7FAF Reset 0x00			Memory Management Policy State Register
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	Enabled	R/W	MMP State latched by the most recent Get State Command in the MMPCMD register. When set(1), the MMP feature will be enabled; when clear(0), the MMP feature will be disabled.
[6:5]	Reserved	R/W	Reserved – Read back as 0.
[4:0]	Count	R/W	Count latched by the most recent Get State Command in the MMPCMD register.

Note 1: The MMPSTATE register is read/write but it is only read in normal operation.

Table 112 – IN_NAKLO REGISTER

IN_NAKLO (0x7FEC – RESET=0x00)			IN_NAKLO REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	IN_NAK_7	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 7 and reset when the MCU writes a '1' to it.
6	IN_NAK_6	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 6 and reset when the MCU writes a '1' to it.
5	IN_NAK_5	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 5 and reset when the MCU writes a '1' to it.
4	IN_NAK_4	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 4 and reset when the MCU writes a '1' to it.
3	IN_NAK_3	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 3 and reset when the MCU writes a '1' to it.
2	IN_NAK_2	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 2 and reset when the MCU writes a '1' to it.
1	IN_NAK_1	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 1 and reset when the MCU writes a '1' to it.
0	IN_NAK_0	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 0 and reset when the MCU writes a '1' to it.

Table 113 – IN_NAKHI REGISTER

IN_NAKHI (0x7FED – RESET=0x00)			IN_NAKHI REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	IN_NAK_15	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 15 and reset when the MCU writes a '1' to it.
6	IN_NAK_14	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 14 and reset when the MCU writes a '1' to it.
5	IN_NAK_13	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 13 and reset when the MCU writes a '1' to it.
4	IN_NAK_12	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 12 and reset when the MCU writes a '1' to it.
3	IN_NAK_11	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 11 and reset when the MCU writes a '1' to it.
2	IN_NAK_10	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 10 and reset when the MCU writes a '1' to it.
1	IN_NAK_9	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 9 and reset when the MCU writes a '1' to it.
0	IN_NAK_8	R/W	This bit is set when the SIE responds with a NAK to IN tokens on EP 8 and reset when the MCU writes a '1' to it.

Table 114 – OUT_NAKLO REGISTER

OUT_NAKLO (0x7FEE – RESET=0x00)			OUT_NAKLO REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	OUT_NAK_7	R/W	This bit is This bit is set(1) when after the SIE responds with a NAK to OUT tokens on EP 7 and reset(0) after when the MCU writes a '1' to it.
6	OUT_NAK_6	R/W	This bit is set(1) when after the SIE responds with a NAK to OUT tokens on EP 6 and reset(0) when after the MCU writes a '1' to it.
5	OUT_NAK_5	R/W	This bit is set(1) when the SIE responds with a NAK to OUT tokens on EP 5 and reset(0) when after the MCU writes a '1' to it.
4	OUT_NAK_4	R/W	This bit is set(1) when the SIE responds with a NAK to OUT tokens on EP 4 and reset(0) when after the MCU writes a '1' to it.
3	OUT_NAK_3	R/W	This bit is set(1) when the SIE responds with a NAK to OUT tokens on EP 3 and reset(0) when after the MCU writes a '1' to it.
2	OUT_NAK_2	R/W	This bit is set(1) when the SIE responds with a NAK to OUT tokens on EP 2 and reset(0) when after the MCU writes a '1' to it.
1	OUT_NAK_1	R/W	This bit is set(1) when the SIE responds with a NAK to OUT tokens on EP 1 and reset(0) when after the MCU writes a '1' to it.
0	OUT_NAK_0	R/W	This bit is set(1) when the SIE responds with a NAK to OUT tokens on EP 0 and reset(0) when after the MCU writes a '1' to it.

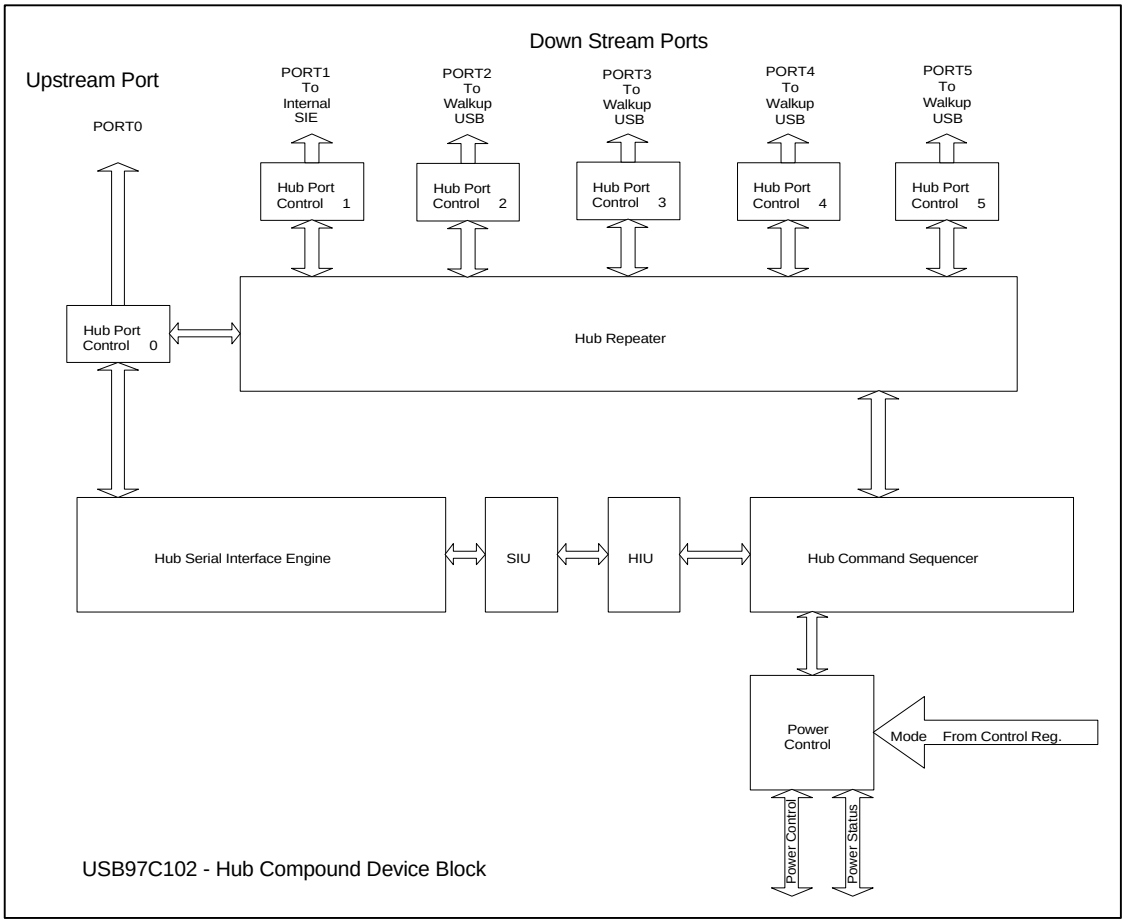
Table 115 – OUT_NAKHI REGISTER

OUT_NAKHI (0x7FEF – RESET=0x00)			OUT_NAKHI REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	OUT_NAK_15	R/W	This bit is set(1) when after the SIE responds with a NAK to OUT tokens on EP 15 and reset(0) when after the MCU writes a '1' to it.
6	OUT_NAK_14	R/W	This bit is set(1) after t when the SIE responds with a NAK to OUT tokens on EP 14 and reset(0) when after the MCU writes a '1' to it.
5	OUT_NAK_13	R/W	This bit is set(1) when after the SIE responds with a NAK to OUT tokens on EP 13 and reset(0) when after the MCU writes a '1' to it.
4	OUT_NAK_12	R/W	This bit is set(1) when after the SIE responds with a NAK to OUT tokens on EP 12 and reset(0) when after the MCU writes a '1' to it.
3	OUT_NAK_11	R/W	This bit is set(1) when after the SIE responds with a NAK to OUT tokens on EP 11 and reset(0) when after the MCU writes a '1' to it.
2	OUT_NAK_10	R/W	This bit is set(1) when after the SIE responds with a NAK to OUT tokens on EP 10 and reset(0) when after the MCU writes a '1' to it.
1	OUT_NAK_9	R/W	This bit is set(1) when after the SIE responds with a NAK to OUT tokens on EP 9 and reset(0) when after the MCU writes a '1' to it.
0	OUT_NAK_8	R/W	This bit is set(1) when after the SIE responds with a NAK to OUT tokens on EP 8 and reset(0) when after the MCU writes a '1' to it.

USB HUB BLOCK

The registers shown below interface the Internal 8051 MCU with the SMC DBC98C99 internal Hub Block. The MCU, subsequent to reset and initialization, must initialize the HUB register block as its first task. Initialization of the registers below, must be accomplished within two (2) ms after the de-assertion of reset. The MCU must initialize the registers before the up stream host controller relinquishes its reset pulse to the internal HUB block so that the Host Controller can enumerate the device.

Below is a block diagram of the HUB block. As indicated in the diagram, the HUB block consists of the Hub Repeater, Control and Command sequencer.



SIU System Interface Unit

This module consists of address decoding and multiplex logic. The address decoder logic is used to compare the address received from the host during a SETUP, IN or OUT token transfer with the address of the HUB. There are two address decodes, one for the HUB endpoint and one for the Remote Device Bay Control endpoint which is not part of the HUB Block.

HIU Hub Interface Unit

The Hub Interface Unit (HIU) provides the hub controller function of this compound device. The hub controller provides the functionality for Host to HUB communication. The HUB specific control and status commands defined in the USB and HUB device class specification permit the host controller to configure the HUB and control and monitor each down stream port. The HUB control block, for the most part, is like a full speed device on USB and hence it consists of all the function blocks needed to implement a device. Included in the functionality required is endpoint 0 control, enumeration, control packet decoding, status maintenance and reporting. Additional functions that will be performed by the HUB block include:

- Provide Hub descriptors defined the HUB USB Device Class Specification V 1.1
- Hub Configuration
- Hub and Port Status
- Interrupt endpoint for status change reporting
- Port Power control
- Frame Timer logic
- Fault Recovery
- Selective Suspend and Resume on a port by port basis
- Selective Reset on a port by port basis
- The ability to decode the preamble PID and allowing Low Speed port enabling.
- Full-speed/Low-speed USB transceivers implemented internally; One placed on the upstream port and four placed on the downstream ports
- Reflecting Remote Resume to Upstream and enabled down stream USB ports

HUB Block Register Summary

The Register definitions defined below are defined in Table 116 and. These registers are memory mapped into the 8051 MCU memory space defined in Figure 4.

Table 116 - HUB Block Register Summary

ADDRESS	NAME	R/W	DESCRIPTION	PAGE
7FA0	IdVendor-Low Byte	R/W	Low byte Vendor ID in little endian format (Bit 0 is the LSB)	
7FA1	IdVendor-High Byte	R/W	High byte Vendor ID in little endian format (Bit 0 is the LSB)	
7FA2	IdProduct-Low Byte	R/W	Low byte Product ID value in little endian format (Bit 0 is the LSB). This value is initialized by firmware upon initialization/power up. This value must be initialized prior to the Hub device participating in and USB enumeration transactions.	
7FA3	IdProduct-High Byte	R/W	High byte Product ID value in little endian format (Bit 0 is the LSB). This value is initialized by firmware upon initialization/power up. This value must be initialized prior to the Hub device participating in and USB enumeration transactions.	
7FA4	BcdDevice - Low Byte	R/W	This 8-bit value defines the USB device release number, which is assigned by the system manufacture.	
7FA5	BcdDevice - High Byte	R/W	This 8-bit value defines the USB device release number, which is assigned by the system manufacture.	
7FA6	HubControl1	R/W	Hub Control register 1	100
7FA7	HubControl2	R/W	Hub Control register 2	102

Table 117 – Hub Control Register1

HubControl1 (0x7FA6- RESET=0x00)			HUB CONTROL REGISTER1
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	NhubReset	R/W	NHubReset – When this bit is asserted (0), the hub controller is in a reset state. The hub will not respond to any enumeration or device requests. When this bit is de-asserted (1), the hub controller is ready to receive packets from the Root Host Controller. Each Port will then be enabled via a control packet from the Host
6	Reserved	R/W	Reserved – This bit should always be cleared (0)
5	HubBypass5	R/W	When this bit is set(1), Ports 1 and 5 are no longer connected to the hub. Port 1 (which is connected to the rest of the 97C102) is connected to port 5. Port 5 becomes the upstream of Port 1. See figure on next page.
4	HubBypass4	R/W	When this bit is set(1), Ports 1 and 4 are no longer connected to the hub. Port 1 (which is connected to the rest of the 97C102) is connected to port 4. Port 4 becomes the upstream of Port 1. See figure on next page.
3	HubBypass3	R/W	When this bit is set(1), Ports 1 and 3 are no longer connected to the hub. Port 1 (which is connected to the rest of the 97C102) is connected to port 3. Port 3 becomes the upstream of Port 1. See figure on next page.
2	HubBypass2	R/W	When this bit is set(1), Ports 1 and 2 are no longer connected to the hub. Port 1 (which is connected to the rest of the 97C102) is connected to port 2. Port 2 becomes the upstream of Port 1. See figure on next page.
1	ForceSE0	R/W	Force Single Ended zero (SE0). – This bit will force a SE0 condition on the upstream port of Port 1 (as selected by the Host_EmuX bits). It is the responsibility of the 8051 MCU to make sure the duty cycle of the SE0 assertion is within the USB specified range for the intended operation (EOP = exactly 2 low speed periods; Disconnect > 2.5µs).
0	GangedPWR	R/W	Ganged Power Sense enable – When this bit is set (1), the Power Control block of the HUB Compound device will internally OR the power OK sense pins (nPWROK[5:2]) and Power Enable (nPWREN[5:2]) pins. This will allow the system designer the ability to reduce implementation costs by reducing the external current hardware. In this mode, since only one Sense and Enable PIN is required, the unused input pins must be tied to VDD (1) and the unused output pins may be left unconnected.

The diagram shown in **FIGURE 8** shows the Hub configured for hub bypass mode 2.

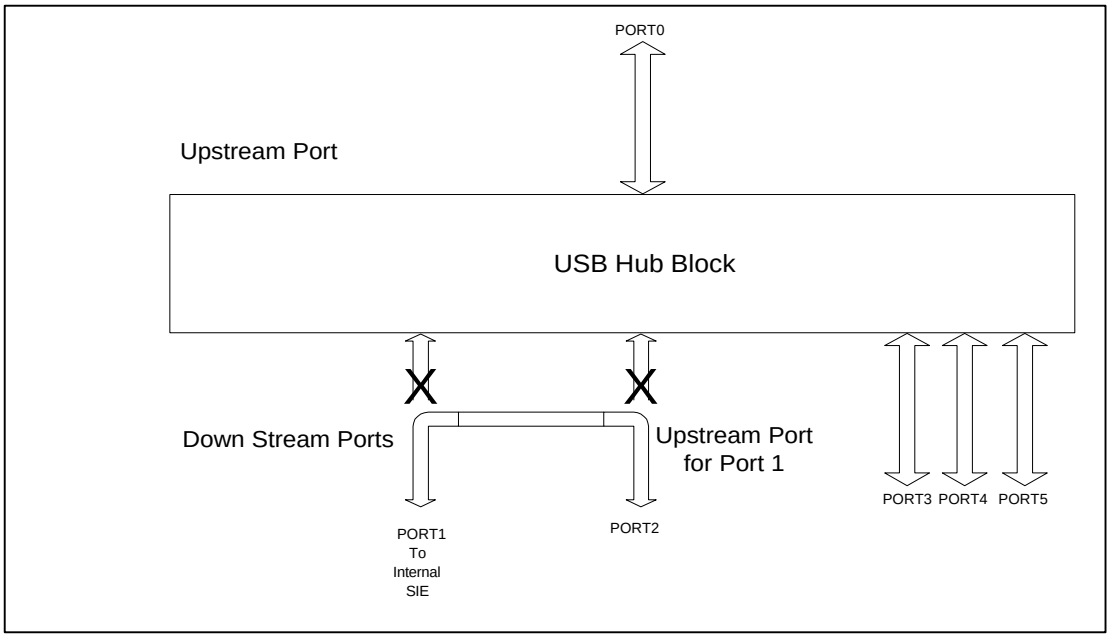


FIGURE 8 – HUBBYPASS2

Table 118 – Hub Control Register2

HubControl2 (0x7FA7- RESET=0x00)			HUB CONTROL REGISTER2
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	Reserved	R/W	Reserved – This bit should always be set to zero (0)
6	Reserved	R/W	Reserved – This bit should always be set to zero (0)
5	EXTXCVR5	R/W	EXTXCVREN – This bit when set (1) will disable the internal Dual Speed USB transceiver and enable the associated control lines on port #5 (internal transceiver bypass). When this bit is clear, the internal transceiver is used. Note 1
4	EXTXCVR4	R/W	EXTXCVREN – This bit when set (1) will disable the internal Dual Speed USB transceiver and enable the associated control lines on port #4 (internal transceiver bypass). When this bit is clear, the internal transceiver is used. Note 1
3	EXTXCVR3	R/W	EXTXCVREN – This bit when set (1) will disable the internal Dual Speed USB transceiver and enable the associated control lines on port #3 (internal transceiver bypass). When this bit is clear, the internal transceiver is used. Note 1
2	EXTXCVR2	R/W	EXTXCVREN – This bit when set (1) will disable the internal Dual Speed USB transceiver and enable the associated control lines on port #2 (internal transceiver bypass). When this bit is clear, the internal transceiver is used. Note 1
1	Reserved	R/W	Reserved – This bit should always be set to zero (0)
0	Reserved	R/W	Reserved – This bit should always be set to zero (0)

Note1: When the port is in external transceiver mode, the associated port PD+ and PD- pins may be left floating. When the port is not in external transceiver mode, the associated transceiver interface input pins should be grounded and the output pins may be left floating.

DC PARAMETERS

MAXIMUM GUARANTEED RATINGS

Operating Temperature Range ..... 0^{o}C to $+70^{o}$C
Storage Temperature Range ..... -55^{o} to $+150^{o}$C
Lead Temperature Range (soldering, 10 seconds) ..... $+325^{o}$C
Positive Voltage on any pin, with respect to Ground ..... V_{cc}+0.3V
Negative Voltage on any pin, with respect to Ground ..... -0.3V
Maximum V_{cc} ..... +5V

*Stresses above the specified parameters could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

Note: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. When this possibility exists, it is suggested that a clamp circuit be used.

DC ELECTRICAL CHARACTERISTICS (T_A = 0°C - 70°C, V_{cc} = *+3.3 V* ± 10%)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	COMMENTS
I Type Input Buffer						
Low Input Level	V_{ILI}			0.8	V	TTL Levels
High Input Level	V_{IHI}	2.0			V	
ICLK Input Buffer						
Low Input Level	V_{ILCK}			0.4	V	
High Input Level	V_{IHCK}	2.2			V	
Input Leakage (All I and IS buffers)						
Low Input Leakage	I_{IL}	-10		+10	uA	V_{IN} = 0
High Input Leakage	I_{IH}	-10		+10	mA	V_{IN} = V_{CC}

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	COMMENTS
O8 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA @ V_{CC} = 5V
						I_{OL} = 4 mA @ V_{CC} = 3.3V
High Output Level	V_{OH}	2.4			V	I_{OH} = -4 mA @ V_{CC} = 5V
						I_{OH} = -2 mA @ V_{CC} = 3.3V
Output Leakage	I_{OL}	-10		+10	uA	V_{IN} = 0 to V_{CC} (Note 1)
I/O8 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA @ V_{CC} = 5V
						I_{OL} = 4 mA @ V_{CC} = 3.3V
High Output Level	V_{OH}	2.4			V	I_{OH} = -4 mA @ V_{CC} = 5V
						I_{OH} = -2 mA @ V_{CC} = 3.3V
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to V_{CC} (Note 1)
I/O16 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 16 mA @ V_{CC} = 5V
						I_{OL} = 8 mA @ V_{CC} = 3.3V
High Output Level	V_{OH}	2.4			V	I_{OH} = -8 mA @ V_{CC} = 5V
						I_{OH} = -4 mA @ V_{CC} = 3.3V
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to V_{CC} (Note 1)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	COMMENTS
I/O24 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 24 mA @ V_{CC} = 5V I_{OL} = 12 mA @ V_{CC} = 3.3V
High Output Level	V_{OH}	2.4			V	I_{OH} = -12 mA @ V_{CC} = 5V I_{OH} = -6 mA @ V_{CC} = 3.3V
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to V_{CC} (Note 1)
IO-U **Note 2**						
Supply Current Unconfigured	I_{CCINIT}	TBD	TBD	TBD	mA	@ V_{CC} = 3.3V
Supply Current Active	I_{CC}		60	100	mA	@ V_{CC} = 3.3V
Supply Current Standby	I_{CSBY}				µA	@ V_{CC} = 3.3V

Note 1: Output leakage is measured with the current pins in high impedance.
Note 2: See Appendix A for USB DC electrical characteristics.

CAPACITANCE T_A = 25°C; fc = 1MHz; V_{CC} = *3.3V*

PARAMETER	SYMBOL	LIMITS			UNIT	TEST CONDITION
		MIN	TYP	MAX		
Clock Input Capacitance	C_{IN}			20	pF	All pins except USB pins (and pins under test tied to AC ground)
Input Capacitance	C_{IN}			10	pF	
Output Capacitance	C_{OUT}			20	pF	

USB PARAMETERS

The following tables and diagrams were obtained from the USB specification

USB DC PARAMETERS

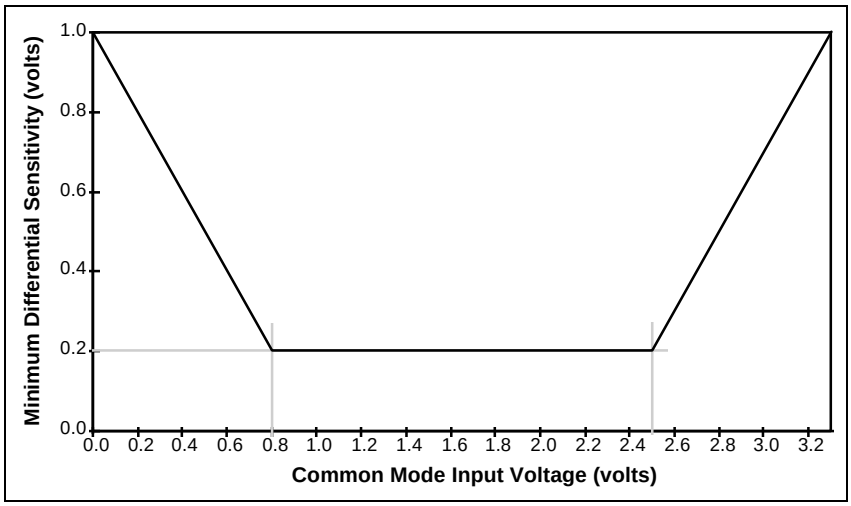


FIGURE 9 - DIFFERENTIAL INPUT SENSITIVITY OVER ENTIRE COMMON MODE RANGE

Table 119 - DC Electrical Characteristics

PARAMETER	SYMBOL	CONDITIONS (NOTE 1, 2)	MIN	TYP	MAX	UNIT
Supply Voltage:						
Powered (Host or Hub) Port	VBUS		2.97		3.63	V
Supply Current:						
Function	ICC	Note 4			100	mA
Un-configured Function (in)	ICCINIT	Note 5			100	uA
Suspend Device	ICCS				200	uA
Leakage Current:						
Hi-Z State Data Line Leakage	ILO	0 V < VIN < 3.3 V	-10		10	uA
Input Levels:						
Differential Input Sensitivity	VDI	|(D+) - (D-)|, and FIGURE 9	0.2			V
Differential Common Mode Range	VCM	Includes VDI range	0.8		2.5	V

PARAMETER	SYMBOL	CONDITIONS (NOTE 1, 2)	MIN	TYP	MAX	UNIT
Single Ended Receiver Threshold	VSE		0.8		2.0	V
Output Levels:						
Static Output Low	VOL	RL of 1.5 KΩ to 3.6 V			0.3 (3)	V
Static Output High	VOH	RL of 15 KΩ to GND	2.8		3.6 (3)	V
Capacitance						
Transceiver Capacitance	CIN	Pin to GND			20	pF
Terminals						
Bus Pull-up Resistor on Root Port	RPU	(1.5 KΩ +/- 5%)	1.425		1.575	kΩ
Bus Pull-down Resistor on Downstream Port	RPD	(15 KΩ +/- 5%)	14.25		15.75	kΩ

Note 1: All voltages are measured from the local ground potential, unless otherwise specified.
Note 2: All timing use a capacitive load (CL) to ground of 50pF, unless otherwise specified.
Note 3: This is relative to VUSBIN.
Note 4: This is dependent on block configuration set by software.
Note 5: When the internal ring oscillator and waiting for first setup packet.

USB AC PARAMETERS

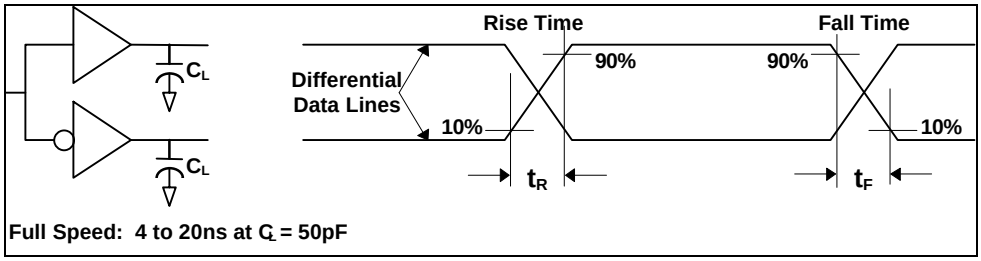


FIGURE 10 - DATA SIGNAL RISE AND FALL TIME

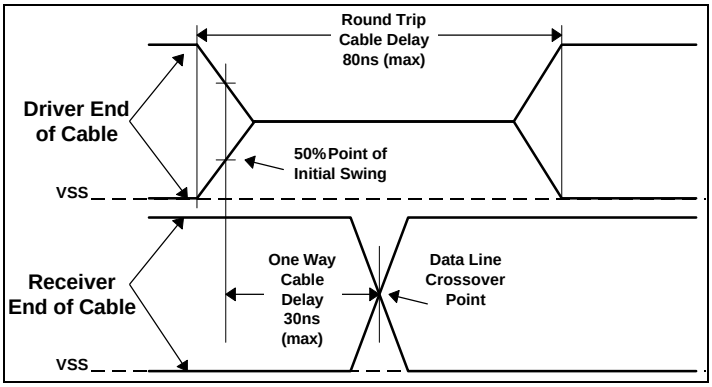


FIGURE 11 - CABLE DELAY

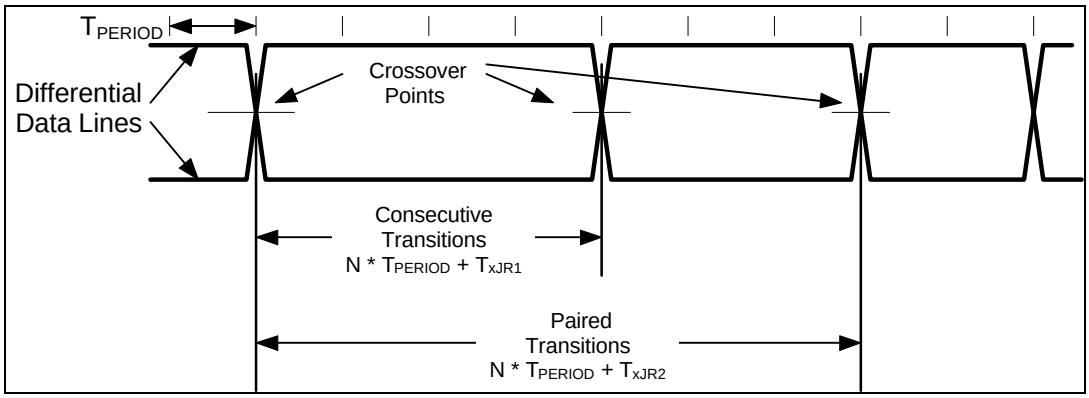


FIGURE 12 - DIFFERENTIAL DATA JITTER

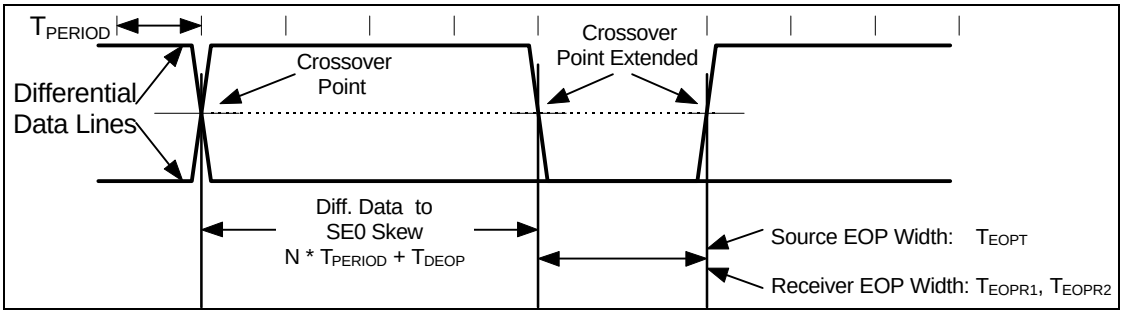


FIGURE 13 - DIFFERENTIAL TO EOP TRANSITION SKEW AND EOP WIDTH

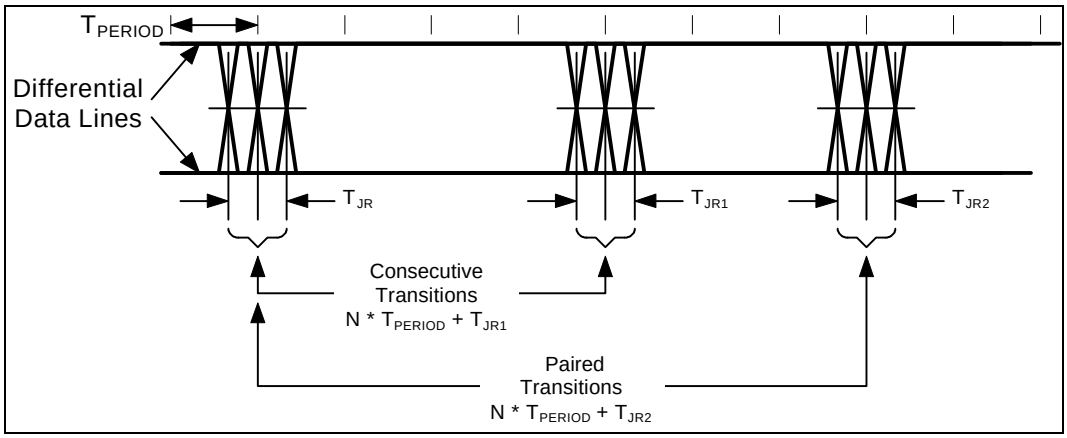


FIGURE 14 - RECEIVER JITTER TOLERANCE

Table 120 - Full Speed (12Mbps) Source Electrical Characteristics

PARAMETER	SYM	CONDITIONS (NOTE 1, 2, 3)	MIN	TYP	MAX	UNIT
DRIVER CHARACTERISTICS:						
Transition Time:		Note 4,5 and FIGURE 10				
Rise Time	TR	CL = 50 pF	4		20	ns
Fall Time	TF	CL = 50 pF	4		20	ns
Rise/Fall Time Matching	TRFM	(TR/TF)	90		110	%
Output Signal Crossover Voltage	VCRS		1.3		2.0	V
Drive Output Resistance	ZDRV	Steady State Drive	28		43	Ω
DATA SOURCE TIMING:						

PARAMETER	SYM	CONDITIONS (NOTE 1, 2, 3)	MIN	TYP	MAX	UNIT
Full Speed Data Rate	TDRATE	Ave. Bit Rate (12 Mb/s +/- 0.25%) Note 8	11.95		12.03	Mbs
Frame Interval	TFRAME	1.0 ms +/- 0.05%	0.9995		1.0005	ms
Clock Period	TPERIOD		80		86	ns
Source Differential Driver Jitter		Note 6, 7 and FIGURE 12				
To next Transition	TDJ1		-3.5		3.5	ns
For Paired Transitions	TDJ2		-4.0		4.0	ns
Source EOP Width	TEOPT	Note 7 and FIGURE 13	160		175	ns
Differential to EOP transition Skew	TDEOP	Note 7 and FIGURE 13	-2		5	ns
Receiver Data Jitter Tolerance		Note 7 and FIGURE 14				
To next Transition	TJR1		-18.5		18.5	ns
For Paired Transitions	TJR2		-9		9.0	ns
Differential Data Jitter		Note 7 and FIGURE 12				
To next Transition	T_XJR1		-18.5		18.5	ns
For Paired Transitions	T_XJR2		-9		9.0	ns
EOP Width at receiver		Note 7 and FIGURE 13				
Must reject as EOP	TEOPR1		40			ns
Must Accept	TEOPR2		82			ns
CABLE IMPEDANCE AND TIMING:						
Cable Impedance (Full Speed)	ZO	(45 Ω +/- 15%)	38.75		51.75	Ω
Cable Delay (One Way)	TCBL	FIGURE 11			30	ns

Note 1: All voltages are measured from the local ground potential, unless otherwise specified.
Note 2: All timing use a capacitive load (CL) to ground of 50pF, unless otherwise specified.
Note 3: Full speed timings have a 1.5KΩ pull-up to 2.8 V on the D+ data line.
Note 4: Measured from 10% to 90% of the data signals.
Note 5: The rising and falling edges should be smoothly transiting (monotonic).
Note 6: Timing differences between the differential data signals.
Note 7: Measured at crossover point of differential data signals.
Note 8: These are relative to the 24 MHz crystal.

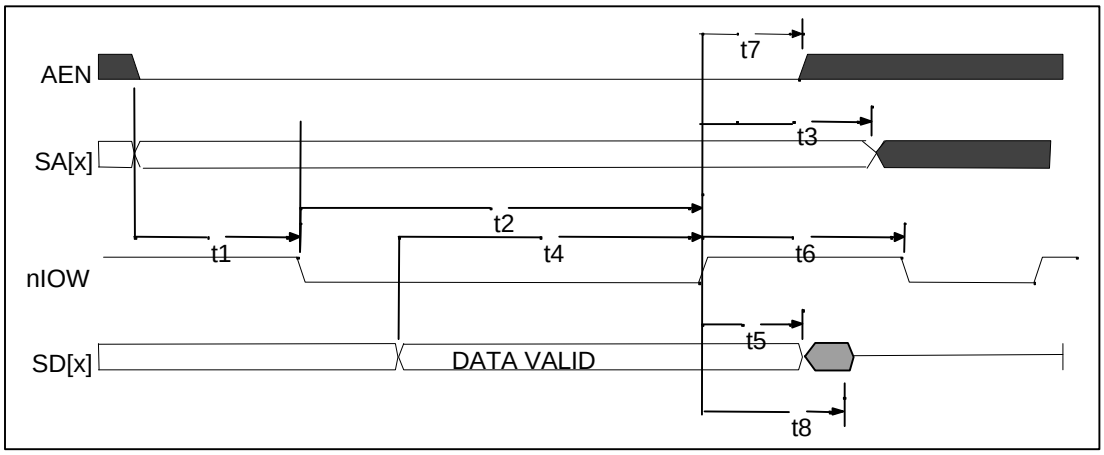


FIGURE 15 - 8051 IO WRITE CYCLE

TABLE 121 – 8051 IO WRITE CYCLE

NAME	DESCRIPTION	MIN	MAX	EQUIATION	UNITS
t1	SA[x] and AEN Valid to nIOW Asserted	106		4t-60	ns
t2	nIOW Asserted to nIOW Deasserted	150		6t-100	ns
t3	nIOW Deasserted to SA[x] Invalid	22		t-20	ns
t4	SD[x] Valid to nIOW Deasserted	150		6t-100	ns
t5	SD[x] Hold from nIOW Deasserted	22		t-20	ns
t6	nIOW Deasserted to nIOW Asserted	25			ns
t7	nIOW Deasserted to AEN Deasserted	22		t-20	ns
t8	nIOW Deasserted to SD[x] tri-state		83	2t	ns

Note: Min and Max delays shown for 8051 clock of 24 MHz, to calculate typical timing delays for other clock frequencies use Oscillator Equations, where $t=1/f_{CLK}$.

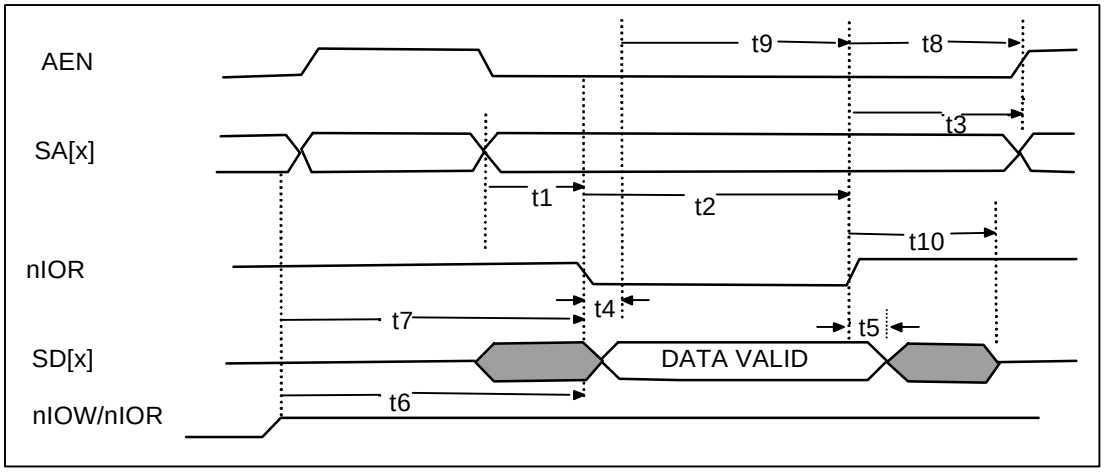


FIGURE 16 – 8051 IO READ CYCLE

TABLE 122 – 8051 IO READ TIMING PARAMETERS

NAME	DESCRIPTION	MIN	MAX	EQUIATION	UNITS
t1	SA[x] and AEN Valid to nIOR Asserted	107		4t-60	ns
t2	nIOR Asserted to nIOR Deasserted	150		6t-100	ns
t3	nIOR Asserted to SA[x] Invalid	32		t-10	ns
t4	nIOR Asserted to Data Valid	0			ns
t5	Data Hold/Float from nIOR Deasserted	0			ns
t6	nIOR Asserted after nIOR Deasserted	32		t-10	ns
t7	nIOR Asserted after nIOW Deasserted	32		t-10	ns
t8	nIOR Asserted to AEN Valid	10			ns
t9	Data Valid to nIOR Deassereted	30			ns
t10	nIOR Deasserted to SD[x] tri-state	32		t-10	ns

Note: Min and Max delays shown for 8051 clk of 24 MHz, to calculate typical timing delays for other clock frequencies use Oscillator Equations, where $t=1/f_{CLK}$.

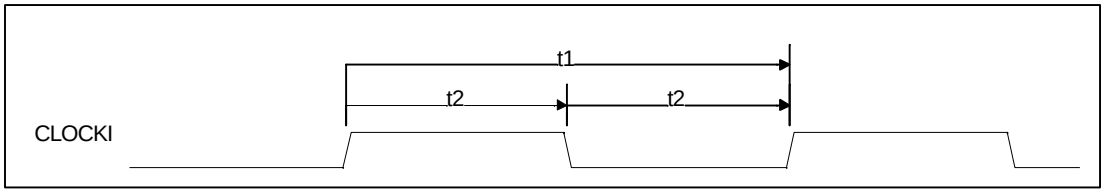


FIGURE 17 - INPUT CLOCK TIMING

TABLE 123 - INPUT CLOCK TIMING PARAMETERS

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t1	CLOCK CYCLE TIME FOR 24 MHZ		41.67		ns
t2	Clock High Time/Low Time for 14.318 MHz	25/16.7		16.7/25	ns
t_r, t_f	Clock Rise Time/Fall Time (not shown)			5	ns

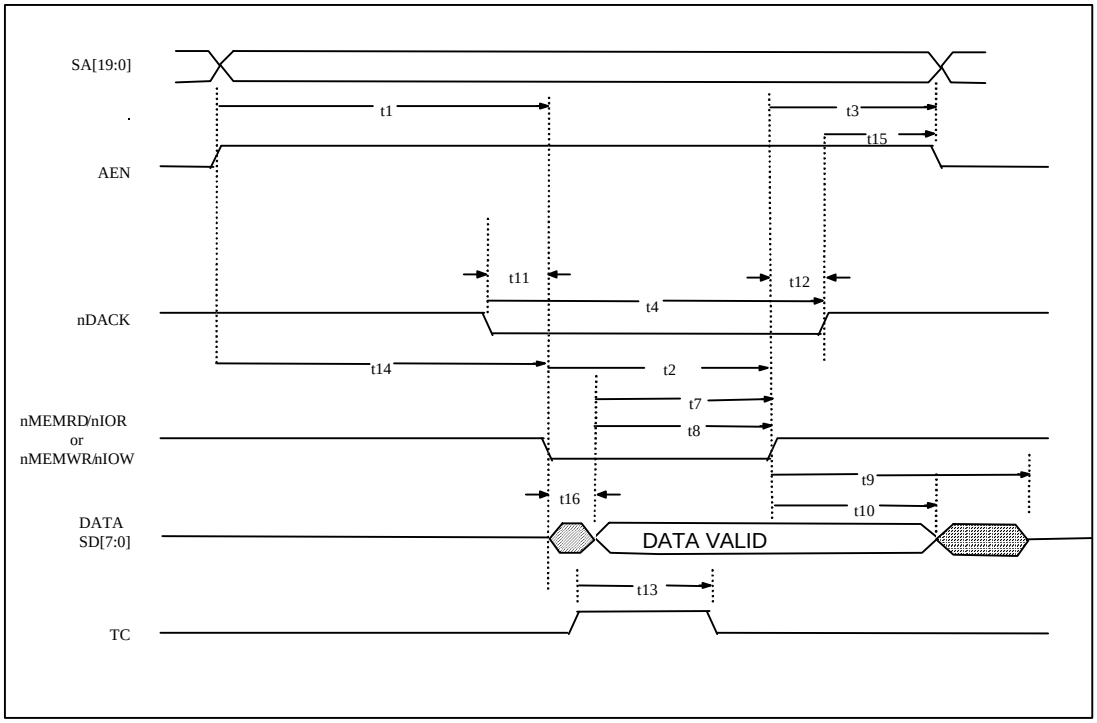


FIGURE 18 - DMA TIMING (SINGLE TRANSFER MODE)

See Table on the following page.

TABLE 124 - DMA TIMING (SINGLE TRANSFER MODE) PARAMETERS

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t1	SA[19:0] Address Setup time to nMEMRD/nIOR or nMEMWR/nIOW Asserted	65			ns
t2	nMEMRD/nIOR or nMEMWR/nIOW Pulsewidth	100			ns
t3	nMEMRD/nIOR or nMEMWR/nIOW deasserted to SA[19:0] Address valid Hold time	30			ns
t4	nDACK Width	150			ns
t7	Data Setup Time to nIOR High	50			ns
t8	Data Set Up Time to nIOW High	40			ns
t9	Data to Float Delay from nIOR High	25		50	ns
t10	Data Hold Time from nIOW High	10			ns
t11	nDACK Set Up to nIOW/nIOR Low	22.5			ns
t12	nDACK Hold after nIOW/nIOR High	22.5			ns
t13	TC Pulse Width	60			ns
t14	AEN Set Up to nIOR/nIOW	40			ns
t15	AEN Hold from nDACK	10			ns
t16	nMEMRD/nIOR or nMEMWR/nIOW asserted to Data valid	0			ns

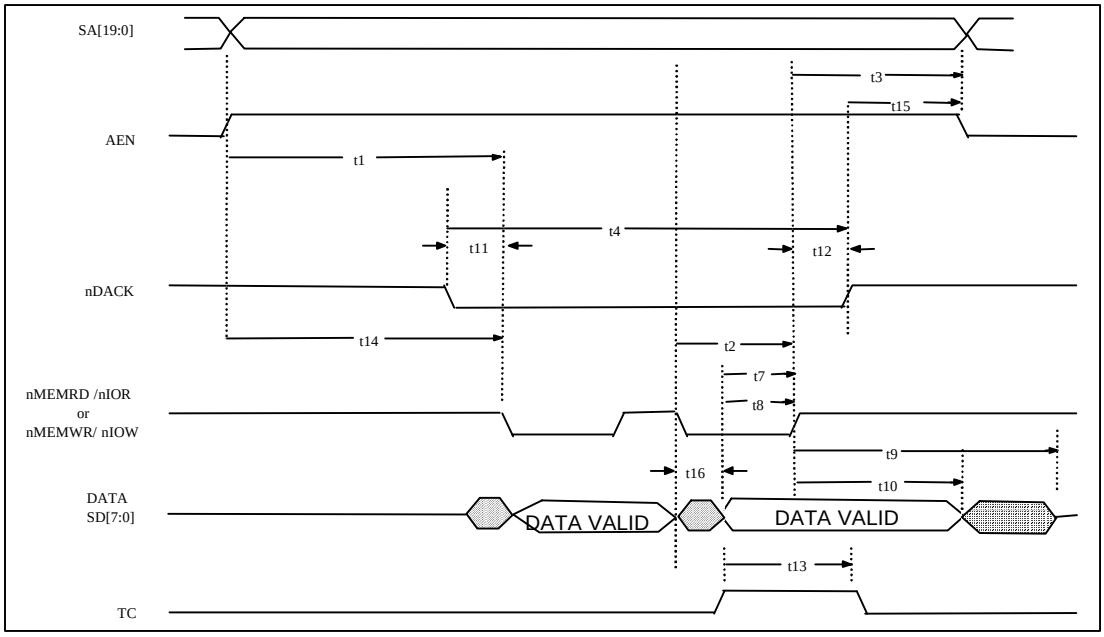


FIGURE 19 - DMA TIMING (BURST TRANSFER MODE)

See Table on the following page.

TABLE 125 - DMA TIMING (BURST TRANSFER MODE) PARAMETERS

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t1	SA[19:0] Address Setup time to nMEMRD/nIOR or nMEMWR/nIOW Asserted	65			ns
t2	nMEMRD/nIOR or nMEMWR/nIOW Pulsewidth	100			ns
t3	nMEMRD/nIOR or nMEMWR/nIOW deasserted to SA[19:0] Address valid Hold time	30			ns
t4	nDACK Width	150			ns
t7	Data Setup Time to nIOR High	50			ns
t8	Data Set Up Time to nIOW High	40			ns
t9	Data to Float Delay from nIOR High	25		50	ns
t10	Data Hold Time from nIOW High	25			ns
t11	nDACK Set Up to nIOW/nIOR Low	22.5			ns
t12	nDACK Hold after nIOW/nIOR High	22.5			ns
t13	TC Pulse Width	60			ns
t14	AEN Set Up to nIOR/nIOW	40			ns
t15	AEN Hold from nDACK	10			ns
t16	nMEMRD/nIOR or nMEMWR/nIOW asserted to Data valid	0			ns

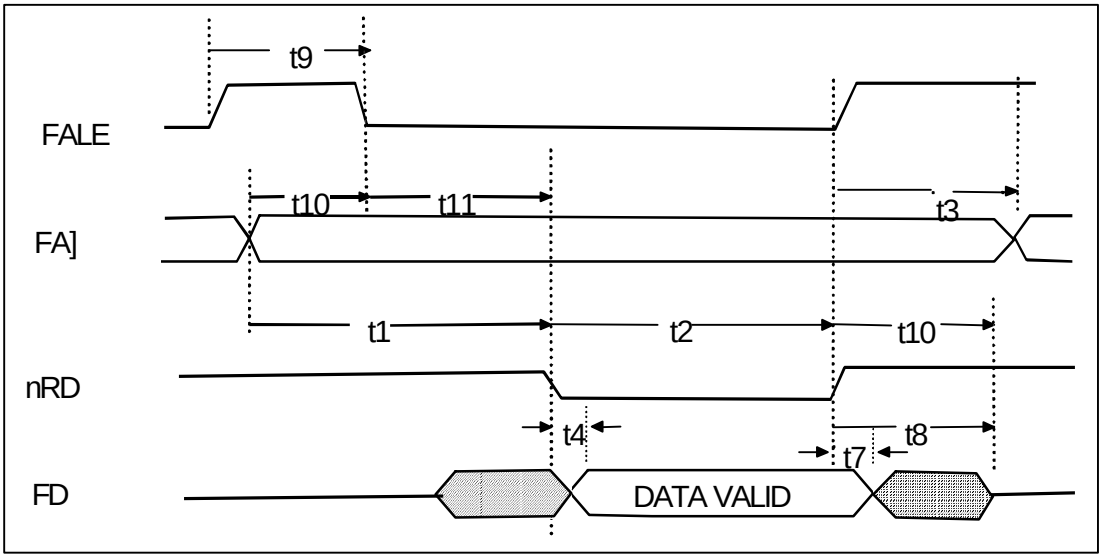


FIGURE 20 - 8051 FLASH PROGRAM FETCH TIMING

Table 126 - 8051 FLASH PROGRAM FETCH TIMING PARAMETERS

	PARAMETER	MIN	TYP	MAX	OSCILLATOR EQUATION	UNITS
t1	FA Valid to nRD asserted	64			2t-20	ns
t2	nRD active pulse width	105			3t-20	ns
t3	nRD deasserted to FA Invalid	32			t-10	ns
t4	nRD asserted to Data Valid	0				ns
t7	FD data Hold from nRD deasserted	0				ns
t8	nRD deasserted to FD data tri-state			32	t-10	ns
t9	FALE active pulse width	53			2t-30	ns
t10	FA address Valid to FALE deasserted	21.66			t-20	ns
t11	FALE deasserted to nRD asserted	21.66			t-20	ns

Note: Min and Max delays shown for an 8051 clock of 24MHz, to calculate timing delays for other clock frequencies use the Oscillator Equations, where T=1/Fclk.

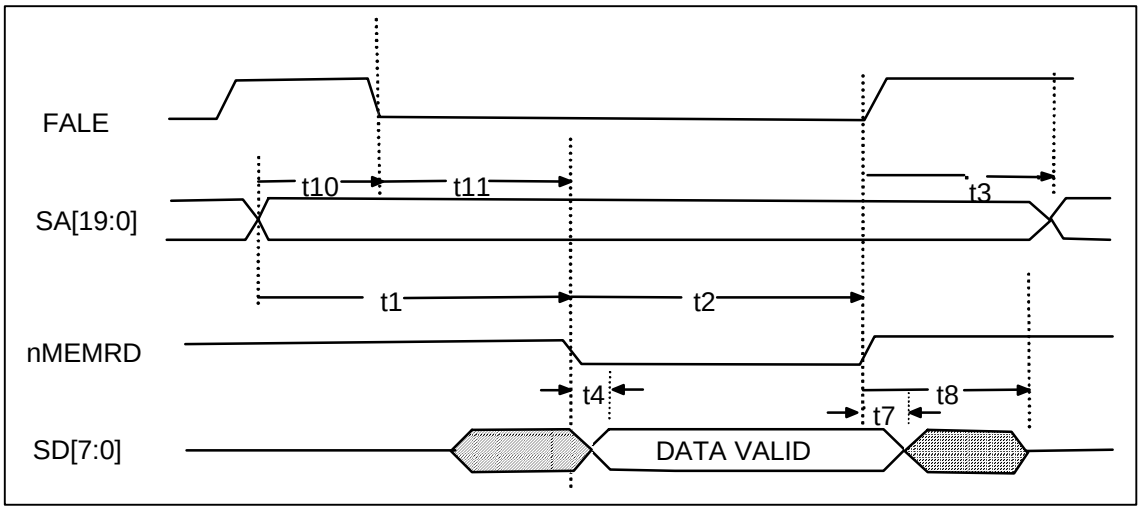


FIGURE 21 - 8051 FLASH MEMORY READ TIMING

Table 127 - 8051 FLASH MEMORY READ TIMING PARAMETERS

	PARAMETER	MIN	TYP	MAX	OSCILLATOR EQUATION	UNITS
t1	SA[19:0] Valid to nMEMRD asserted	107			4t-60	ns
t2	nMEMRD active pulse width	150			6t-100	ns
t3	nMEMRD deasserted to SA[19:0] Invalid	21.66			t-20	ns
t4	nMEMRD asserted to Data Valid	0				ns
t7	SD[7:0] data Hold from nMEMRD deasserted	0				ns
t8	nMEMRD deasserted to SD[7:0] data tri-state			64	2t-20	ns
t11	FALE deasserted to nMEMRD asserted	84		165	3t +/- 40	ns

Note: Min and Max delays shown for an 8051 clock of 24MHz, to calculate timing delays for other clock frequencies use the Oscillator Equations, where T=1/Fclk.

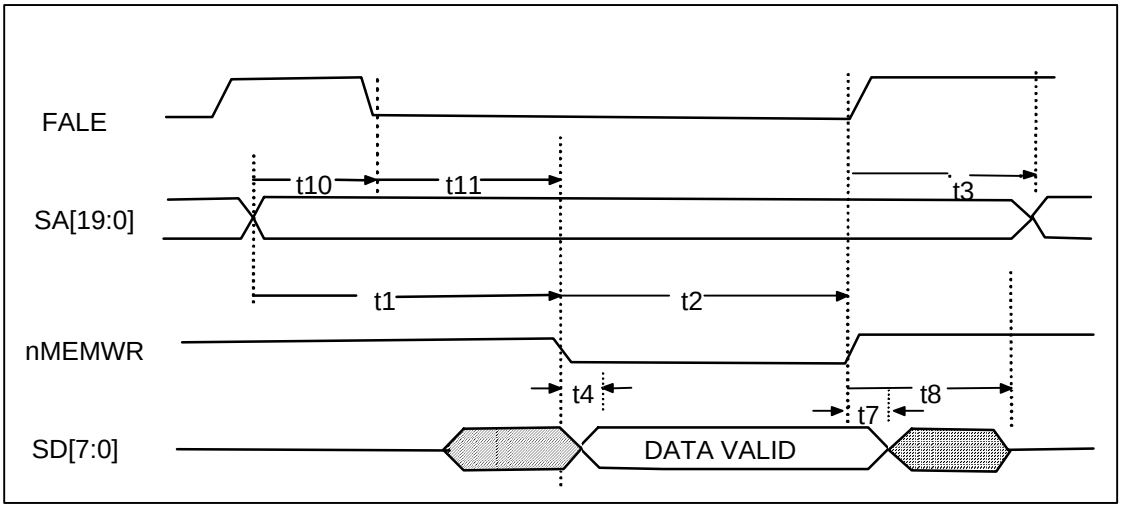


FIGURE 22 - 8051 FLASH MEMORY WRITE TIMING

Table 128 - 8051 FLASH MEMORY READ TIMING PARAMETERS

	PARAMETER	MIN	TYP	MAX	OSCILLATOR EQUATION	UNITS
t1	SA[19:0] Valid to nMEMWR asserted	107			4t-60	ns
t2	nMEMWR active pulse width	150			6t-100	ns
t3	nMEMWR deasserted to SA[19:0] Invalid	21.66			t-20	ns
t4	nMEMWR asserted to Data Valid	32			t-10	ns
t7	SD[7:0] data Hold from nMEMWR deasserted	5				ns
t8	nMEMWR deasserted to SD[7:0] data tri-state			64	2t	ns
t11	FALE deasserted to nMEMWR asserted	85		165	3t +/- 40	ns

Note: Min and Max delays shown for an 8051 clock of 24MHz, to calculate timing delays for other clock frequencies use the Oscillator Equations, where T=1/Fclk.

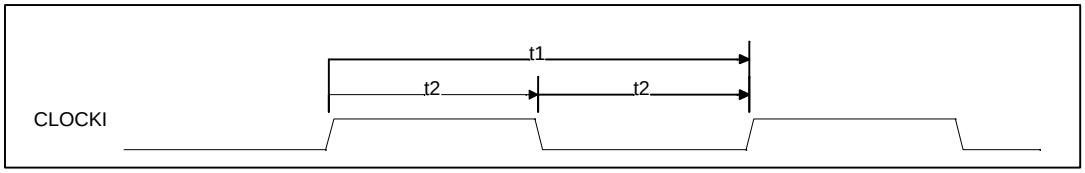
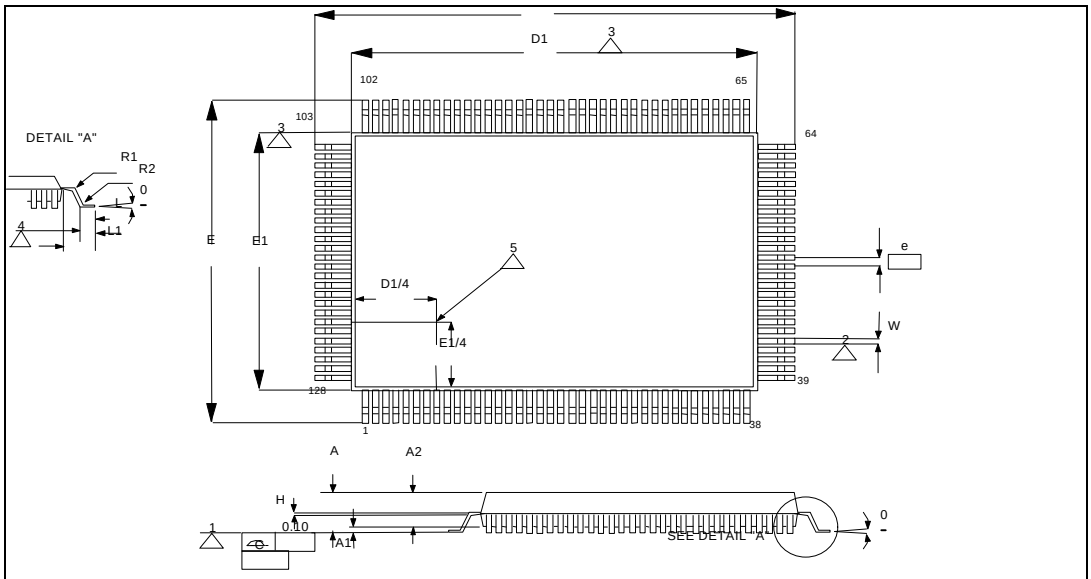


FIGURE 23 - RESET_IN TIMING

Table 129 - RESET_IN TIMING PARAMETERS

	PARAMETER	MIN	TYP	MAX	UNITS
t1	RESET_IN active pulse width	50			ns

MECHANICAL OUTLINE



	MIN	NOM	MAX		MIN	NOM	MAX
A			3.4	L	0.65	0.8	0.95
A1	0.05		0.5	L1		1.95	
A2	2.55		3.05	e	0.5BSC		
D	23.65	23.9	24.15	0	0°		7°
D1	19.9	20	20.1	W	0.1		0.3
E	17.65	17.9	18.15	R1	0.13		
E1	13.9	14	14.1	R2	0.13		0.3
H							

Notes:

1) Coplanarity is 0.08 mm or 3.2 mils
2) Tolerance on the position of the leads is 0.080 mm
3) Package body dimensions D1 and E1 do not include the mold protrusion. mold protrusion is 0.25
4) Dimensions for foot length L measured at the gauge plane 0.25 mm above the
5) Details of pin 1 identifier are optional but must be located within the zone
6) Controlling dimension:

FIGURE 24 - 128 PIN QFP PACKAGE OUTLINE

Circuit diagrams utilizing SMSC products are included as a means of illustrating typical applications; consequently complete information sufficient for construction purposes is not necessarily given. The information has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any licenses under the patent rights of SMSC or others. SMSC reserves the right to make changes at any time in order to improve design and supply the best product possible. SMSC products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of SMSC and further testing and/or modification will be fully at the risk of the customer.

USB97C102 Rev. 5/25/99