STANDARD
MICROSYSTEMS
CORPORATION

USB97C196

USB to Ethernet Bridge Controller with Integrated USB Hub

FEATURES

USB Interface

- USB to Ethernet Bridge Device
- Integrated USB Hub Provides 1 Downstream USB Port
- Supports Bus- or Self-Powered Operation
- High Performance USB Peripheral Controller Engine
  - Integrated USB Transceiver
  - Serial Interface Engine (SIE)
  - Compliant to USB Rev 1.1 Core Specification
  - Supports dedicated Isochronous, Bulk, Interrupt, and Control Data Endpoints
  - Dynamic Hardware Allocation of Packet Buffer for Virtual Endpoints
  - Dynamic Endpoint Buffer Length Allocation (0-1280 Byte Packets)
- Device Supports Self Powered or Bus Powered Designs

System Level Interface

- Embedded 8051 Micro controller (MCU)
  - Standard 8051 "Stop Clock" Modes
  - Additional USB and Ethernet 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 Clock
- Shared USB/Ethernet Memory Management Unit (MMU)
  - 4096 Byte RAM On Board USB/Ethernet Packet Buffer
  - USB Bus Snooping Capabilities
  - 128 Byte Min. Page Size
  - 12 Pages Maximum per Ethernet Packet or USB Bulk Assembled Buffer
  - Up to 32 Deep Receive Packet Queue for USB and Ethernet
  - Up to 5 Deep Transmit Packet Queue, per Endpoint
  - Hardware Generated Packet Descriptor FIFO, Records Each Logical Packet Status Automatically
  - Simultaneous Buffer Arbitration Between MCU, SIE, and CSMA/CD Accesses
  - Extended Power Management
  - Independent Clock/Power Management for SIE, MMU, and CSMA/CD Blocks
- External MCU Memory Interface
  - Up to 128k Byte Code and Data Storage
  - Flash, SRAM, or EPROM
  - Downloadable Code via USB, Serial Port

Network Interface

- Integrates 10BASE-T Transceiver Functions:
  - Driver and Receiver
  - Link Integrity Test
  - Receive Polarity Detection and Correction
- Integrates AUI Interface
  - Supports Standard 10Mbps and 1Mbps Data Rates
- Implements 10 Mbps Manchester Encoding/Decoding and Clock Recovery
- Device Able to Transmit and Receive Data Down to 1 Mbps for Home Networking Applications
  - 7-Wire Serial ENDEC Interface Allows Connection to Home Networking Phys
- Automatic Retransmission, Bad Packet Rejection, and Transmit Padding
- External and Internal Loopback Modes
- Three (3) Direct Driven LEDs for Status/ Diagnostics
- Supports OnNow Technology via Packet Filtering Power Management Technology
- High Performance Chained ("Back-to-Back") Transmit and Receive Operations
- 3.3 Volt, Low Power Operation
- 100 Pin TQFP Package

GENERAL DESCRIPTION

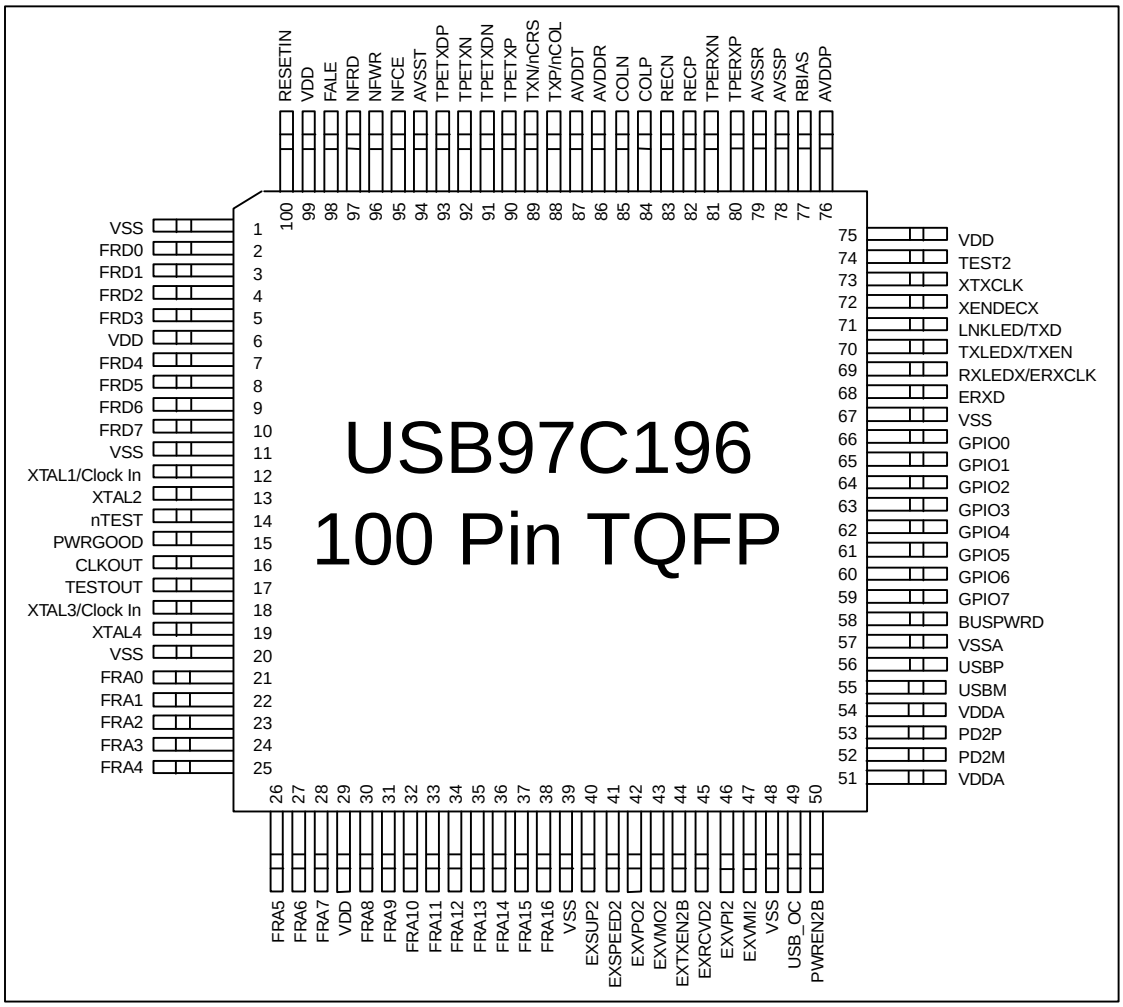
The SMSC USB97C196 is a USB to Ethernet bridge device. The device allows a simple and full-featured link allowing connectivity to an Ethernet LAN via a USB connection. Its unique dynamic buffer architecture overcomes the throughput disadvantages of existing fixed FIFO buffer schemes allowing maximum utilization of the USB connection's overall bandwidth. This architecture minimizes the integrated micro-controller's participation in the USB data flow, allowing back-to-back packet transfers to LAN oriented devices. The efficiency of this architecture allows a high data throughput via a "store and forward" architecture.

The SMSC USB97C196 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 - SMSC USB97C196 Pin Configuration

TQFP PINS	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
		Crystal INTERFACE	
12	XTAL1/ Clock In	**USB 24 MHz 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
13	XTAL2	**USB 24 MHz Crystal:** This is the other terminal of the crystal.	OCLKx
18	XTAL3/ Clock In	**Ethernet 20 MHz 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
19	XTAL4	**Ethernet 20 MHz Crystal:** This is the other terminal of the crystal.	OCLKx
		USB INTERFACE	
52,53	PD2M, PD2P	**USB Downstream Connection signals :** These are two point-to-point signals and are driven differentially PD2P, PD2M are used as a standard “Walk Up” USB Port.	IOUSB
50	PWREN2B	**USB Power Enable –** A low signal on this pin applies power to the downstream USB port. This output signal is active Low.	O24
49	USB_OC	**USB Over-Current Sense.** Input to indicate an over-current condition for a bus powered USB device on the downstream port. This input signal is active Low.	I
55,56	USBM,USBP	**USB Upstream Connection Signals** These are two point-to-point signals and driven differentially.	IOUSB
44	EXTXEN2B	**USB transceiver Output Enable** – This Pin which is active low, enable the external transceiver to transmit data on the USB bus. When this signal is not active (1), the transceiver is in receive mode.	O8
41	EXSPEED2	**USB suspend**- Enables a low power state in the external transceiver while the USB is suspended.	O8

TQFP PINS	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
42,43	EXVPO2, EXVMO2	**Outputs to External Differential Driver** – This is the output from the SIE to the associated USB port external transceiver VPO VMO Result 0 0 SE0 0 1 Logic “0” on USB BUS 1 0 Logic “1” on USB BUS 1 1 Undefined	O24
45	EXRCVD2	**Receive Data from USB Transceiver differential input**	I
46,47	EXVPI2, EXVMI2	**Inputs from External Differential Driver** – These signals are used to detect a single ended zero (SE0, error conditions and interconnect speed from the associated USB port external transceiver. (Inputs to internal USB SIE block) VP VM Result 0 0 SE0 0 1 Low Speed on USB BUS 1 0 High (Full) Speed on USB BUS 1 1 ERROR	I
40	EXSUP2	**Suspend** – This input signal when active high (1) will force a low power state on the associated USB port logic.	O8
		10BASE-T interface	
88,89	TXP/nCOLL TXN/nCRS	INTERNAL ENDEC - (nXENDEC pin open). In this mode TXP and TXN are the AUI transmit differential outputs. They must be externally pulled up using 150 ohm resistors. EXTERNAL ENDEC - (nXENDEC pin tied low). In this mode the pins are inputs used for collision and carrier sense functions.	Differ-ential Output / I
80,81	TPERXP TPERXN	10BASE-T receive differential inputs.	Differ-ential Input
90,92	TPETXP TPETXN	INTERNAL ENDEC - 10BASE-T transmit differential outputs.	Differ-ential Output
93,91	TPETXDP TPETXDN	10BASE-T delayed transmit differential outputs. Used in combination with TPETXP and TPETXN to generate the 10BASE-T transmit pre-distortion.	Differ-ential Output

TQFP PINS	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
71	nLNKLED/ TXD	INTERNAL ENDEC - Link LED output. EXTERNAL ENDEC - Transmit Data output.	OD16/ O162
82,83	RECP,RECN	AUI receive differential inputs.	Differ-ential Input
84,85	COLP,COLN	AUI collision differential inputs. A collision is indicated by a 10MHz signal at this input pair.	Differ-ential Input
70	nTXLED/ nTXEN	INTERNAL ENDEC - Transmit LED output. EXTERNAL ENDEC - Active low Transmit Enable output.	OD16/ O162
68	ERXD	INTERNAL ENDEC – This pin is not used when the Internal ENDEC is enabled. EXTERNAL ENDEC - NRZ receive data input.	I with pullup
73	XTXCLK	Transmit clock input from external ENDEC. Used, only when External ENDEC is used.	IP
69	nRXLED/ RXCLK	INTERNAL ENDEC - Receive LED output. EXTERNAL ENDEC - Receive clock input.	OD16/ I with pullup
Ethernet Misc. and LEDs			
77	RBIAS	A 22kohm 1% resistor should be connected between this pin and analog ground.	N/A
72	NXENDEC	When tied low the SMSC USB97C196 is configured for EXTERNAL ENDEC. When tied high or left open the SMSC USB97C196 will use its Internal ENDEC	IP
FLASH INTERFACE			
10,9,8, 7,5,4,3 ,2	FRD[7:0]	Flash ROM Data Bus These signals are used to transfer data between 8051 and the external FLASH.	IO8
38,37, 36,35, 34,33, 32,31, 30,28, 27,26, 25,24, 23,22, 21	FRA[16:0]	128K Flash ROM Address Bus These signals address memory locations within the FLASH.	O8
97	NFRD	Flash ROM Read; active low	O8
96	NFWR	Flash ROM Write; active low	O8

TQFP PINS	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
95	NFCE	Flash ROM Chip Select; active low	O8
98	FALE	Flash ROM address latch enable	O8
		POWER SIGNALS	
6,29, 75,99	VDD	+3.3 V Power	NA
51,54	VDDA	Digital 3.3V Power Reference for USB Ports	NA
87	AVDDT	Analog power for Ethernet Transmitter	NA
86	AVDDR	Analog power for Ethernet Receiver	NA
76	AVDDP	Analog power for PLL Circuit of Ethernet	NA
94	AVSST	Analog Ground Reference for Ethernet Transmitter	NA
79	AVSSR	Analog Ground Reference for Ethernet Receiver	NA
78	AVSSP	Analog Ground Reference for PLL Circuit	NA
1, 11, 20, 39,48, 67	VSS	Digital Ground Reference	NA
57	VSSA	Digital Ground Reference for USB Ports	NA
		MISCELLANEOUS	
59,60,, 61,62, 63,64, 65,66	GPIO[7:0]	General Purpose I/O. These pins can be configured as inputs or outputs under software control.	I/O16
58	USB_Bus_ Pwrd	USB Power Selection This pin is used to select the device to implement a USB Bus Power or Self Powered mode.	I
100	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
17	TST_OUT	XNOR chain output This signal is used for testing the chip via an internal XNOR Chain. Output only when PWRGOOD is low, otherwise it is Tri-state.	O8
74	TEST2	Reserved for TEST	O
14	NTEST	Test input This signal a manufacturing test pin. User can pull it high or leave it unconnected. When NTEST is low, it combines with PWRGOOD for 2 8051 modes, otherwise it is a NOP.	IP

TQFP PINS	SYMBOL	PIN DESCRIPTION	BUFFER TYPE
15	PWRGOOD	Power good input. It keeps the SMSC USB97C196 in power down mode when low. Must be high for normal operation.	I
16	CLKOUT	**Clock output -** The 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

BUFFER TYPE DESCRIPTIONS

Table 2 - SMSC USB97C196 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.
ICLKx	XTAL clock input
OCLKx	XTAL clock output
ICLK	Clock input (TTL levels)
IOUSB	Defined in USB specification; uses VDD3.3

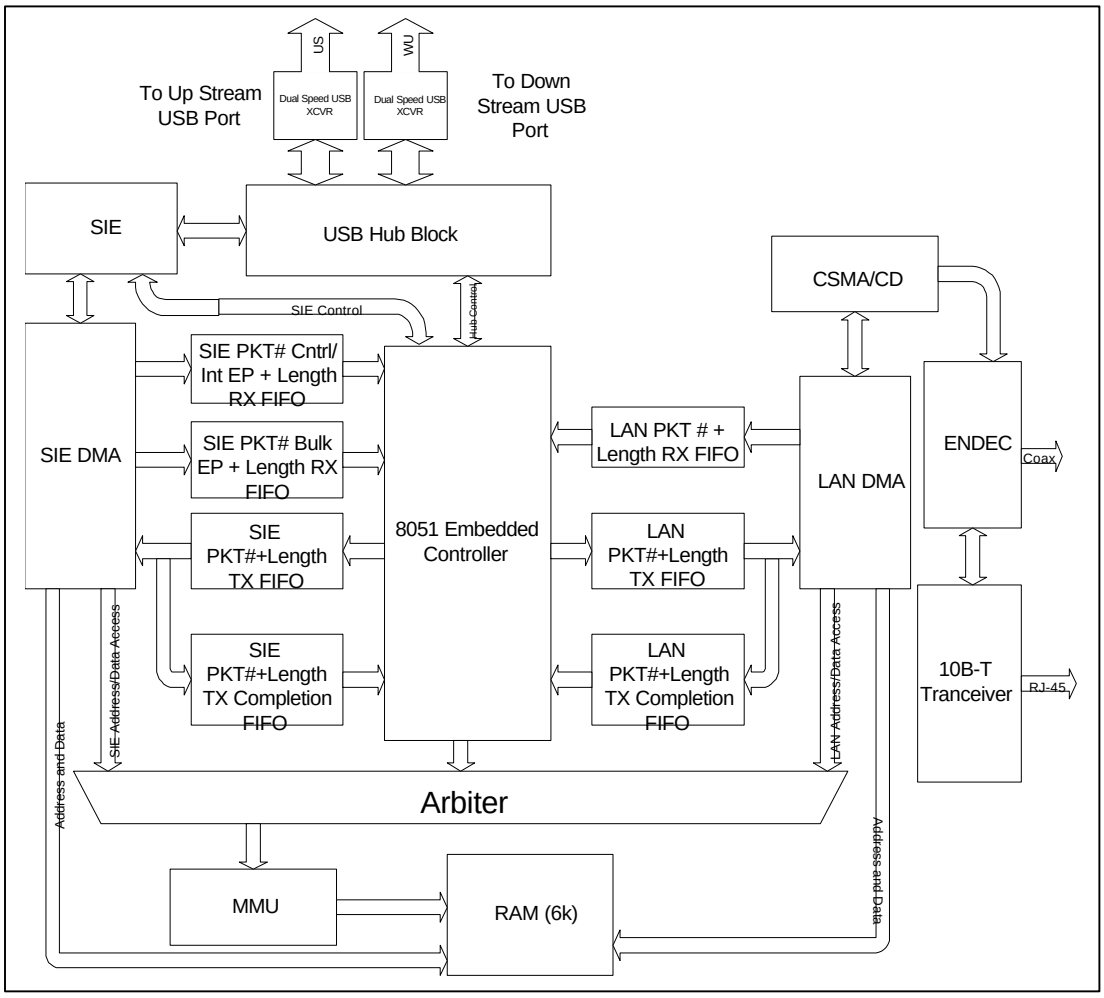


FIGURE 1 – USB97C196 BLOCK DIAGRAM

FUNCTIONAL DESCRIPTION

The SMSC USB97C196 incorporates a USB Serial Interface Engine (SIE), 8051 Micro-controller Unit (MCU), Serial Interface Engine DMA (SIEDMA), Ethernet EPH DMA engine and an Embedded 2 port (One Up Stream and one down stream) USB Hub. The device has 4096 bytes of internal SRAM for data stream buffering, and an MMU (Memory Management Unit) to dynamically manage buffer allocation. The semi-automatic nature of the SIEDMA, CSMA/CD engine 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, CSMA/CD, 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 five transmit, and five receive endpoints (EP2 is transmit only, and EP3 is receive only), for 1 or 2 independent 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 SMSC USB97C196 blocks are mapped into the external memory space of the MCU.

A detailed description of the micro-controller's internal registers and instruction set can be found in the "SMSC USB97C196 Programmer's Reference Guide".

SIEDMA

This is a simplified DMA engine, which automatically transfers data between SIE and SRAM via MMU control. The SIEDMA places a status block containing frame number, endpoint, and byte count to each incoming logical packet (an Ethernet packet consists of multiple USB packets) before notifying the MCU of its arrival. This block operation is transparent to the firmware.

Memory Management Unit (MMU) Register Description

This MMU consists of a 4096 buffer RAM that consists of 32 pages of 128 bytes. Allocations can be done with up to 12 pages each (1536 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 6 independent transmit FIFO queues (one for each endpoint), and one 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.

Specific Support for USB Based SAR as Defined in the USB CDC Specification

This section will describe the additional Device class requirements associated with the specific features needed to implement a USB to Ethernet LAN Bridge.

Data Transfer Management via a USB Bulk Endpoint Pair

Below, shown in figures **FIGURE 2 – USB TO LAN DATA FLOW** through FIGURE 5 – SEGMENT DELINEATION, shows the data flow that would be required to implement a USB to Ethernet bridge device. If the implementation is based on the USB Communications Device Class Specification (USB CDC) that is currently under development in the USB Device Working Group, there are a number of methods that need to be implemented.

In the case of when the device is receiving a sequence of bulk transactions, the data flow shown in **FIGURE 2 – USB TO LAN DATA FLOW**, is depicted below. A set of receive FIFO's is required to store the sequence of USB packets/transactions and upon completion, queue this buffer onto the USB RX FIFO to allow the embedded 8051 to eventually queue to send over the Ethernet LAN. The requirements for a device of this class as defined in the USB Device class specification are:

1. All USB transactions are received in order
2. The bulk endpoint pipe will only receive out bulk data transactions for data that will be sent over Ethernet.
1. If bulk packets / transactions are not transmitted successfully by the host, the host will retry or eventually clear the endpoint to reset the false condition.
2. If the MMU runs out of memory while the USB bulk endpoint is receiving, ALL USB Bulk packets will be de-allocated (Lost).

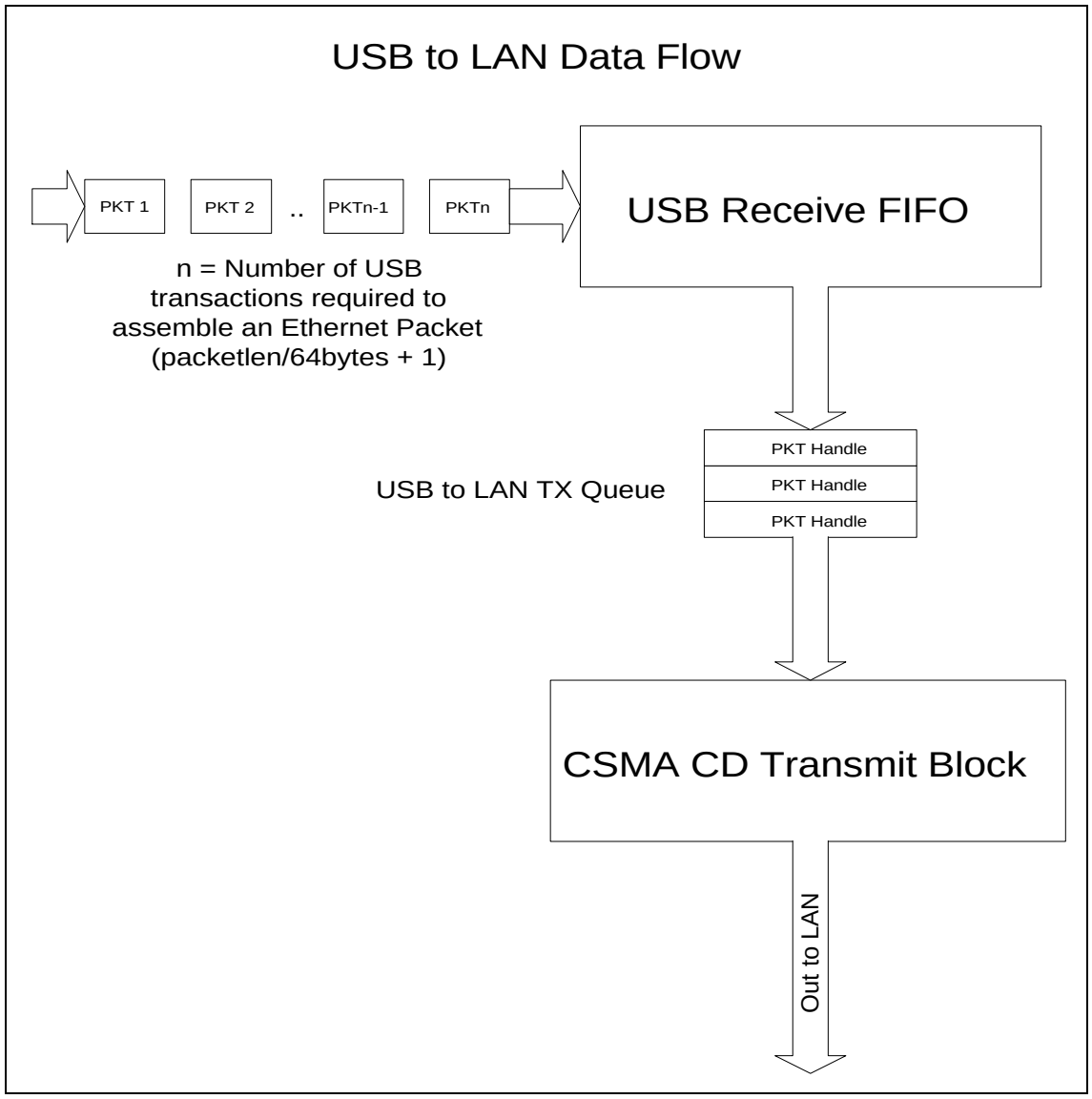


FIGURE 2 – USB TO LAN DATA FLOW

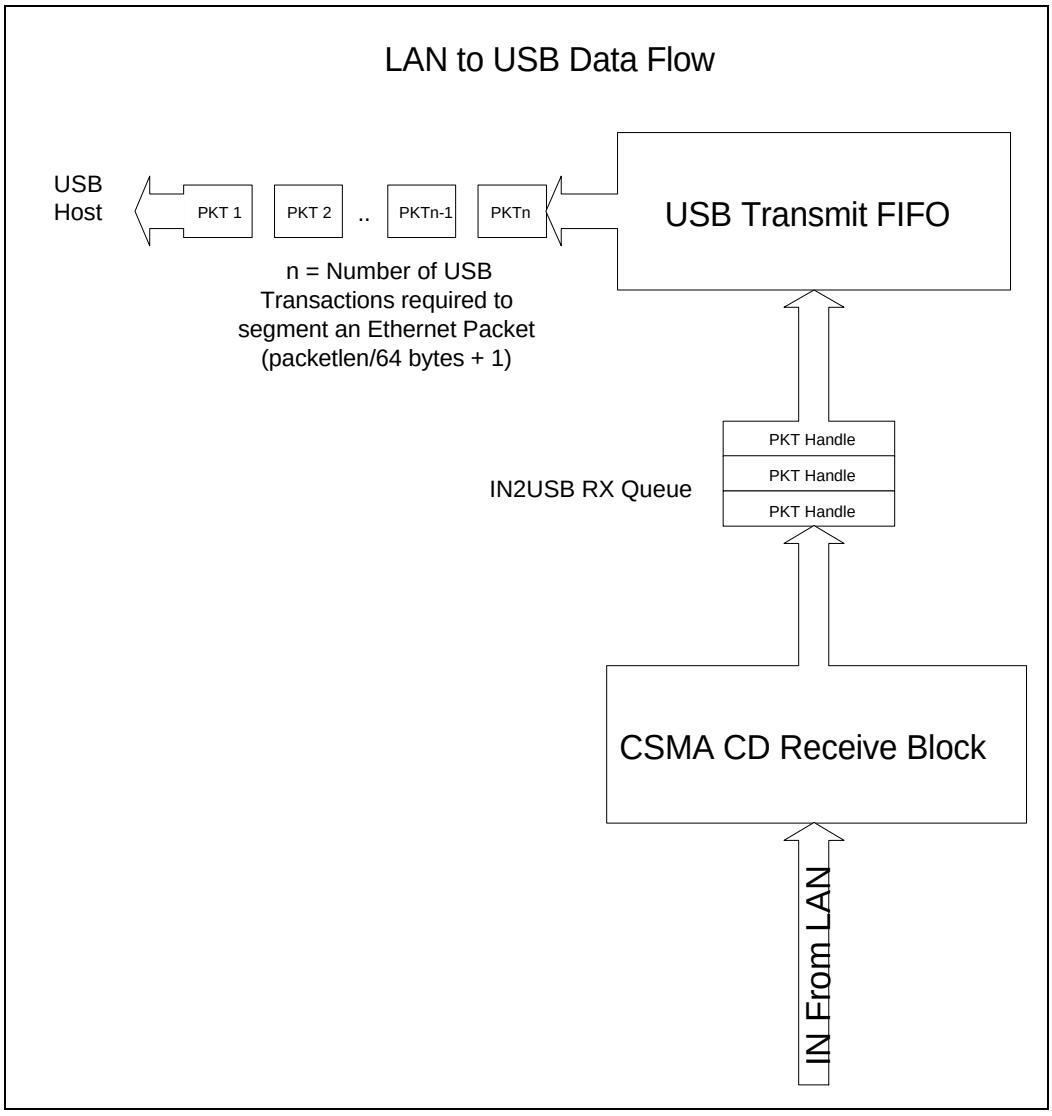


FIGURE 3 - LAN TO USB DATA FLOW

USB–based Networking devices will use BULK transfers (Isochronous is also possible) to exchange data between a host and the USB device.

The Ethernet connection between the host computer and the Ethernet media will be “tunneled” through the USB channel. To do this, the host must formulate an Ethernet packet and provide it to a driver. The driver must then segment the Ethernet packet into multiple USB packets for transport via the USB bus. The USB receive SIE block in conjunction with the SIE DMA engine and MMU must assemble the Ethernet packet and send it out onto the LAN. When an Ethernet packet from the LAN arrives at the device it must be segmented into multiple USB packets by the device for transport over the USB to the host. The driver executing on the host must then reconstruct the original Ethernet packet from these fragments.

Both the driver and the device must be capable of segmenting Ethernet packets into USB packets as well as re-combining USB packets into Ethernet packets.

Assembly of Multiple USB Bulk Packets into an Ethernet Frame

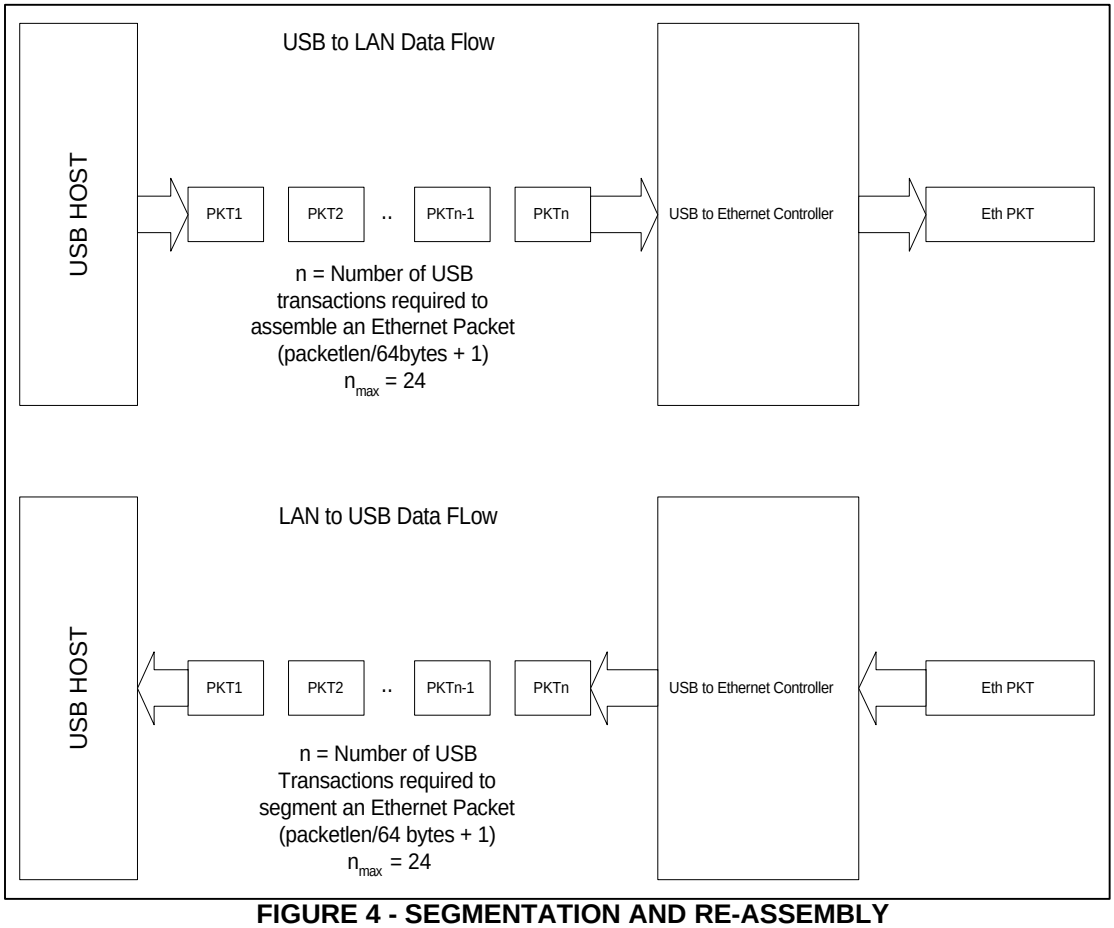


FIGURE 4 - SEGMENTATION AND RE-ASSEMBLY

Ethernet frames are sent from the host to the USB97C196 in 64 byte bulk packets over USB. See the diagram above. The USB97C196 will recognize / decode the start and ending of a logical Ethernet Packet by receiving an out USB packet that is less than 64 bytes or Zero (0) length in the case of the Ethernet packet being an exact multiple of 64 bytes.

Once the USB97C196 is synchronized, it must allocate sufficient memory to buffer the USB packets while the full Ethernet frame is reconstructed. Two possible techniques are:

In the case that insufficient memory exists to re-construct the complete Ethernet frame, the USB97C196 will employ flow control to restrict the host, or ignore the packet and stall the end point, causing the host to re-try.

Otherwise, the USB97C196 will continue accumulating USB bulk packets until the end of frame is detected. There are at least two methods for detecting the end of the frame**.**

Segmentation of an Ethernet Frame into Multiple USB Bulk Packets

The Diagram below shows the method the USB97C196 USB to Ethernet Bridge controller determines the start and end of the Logical Ethernet packet as it is being transmitted over USB via a Bulk OUT or IN end point.

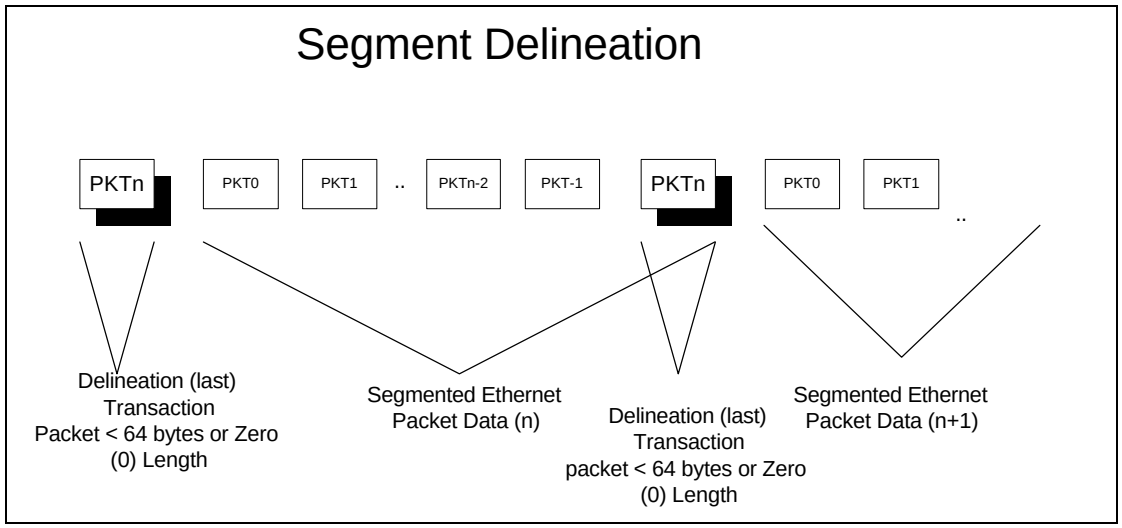


FIGURE 5 – SEGMENT DELINEATION

Ethernet Packet Reception

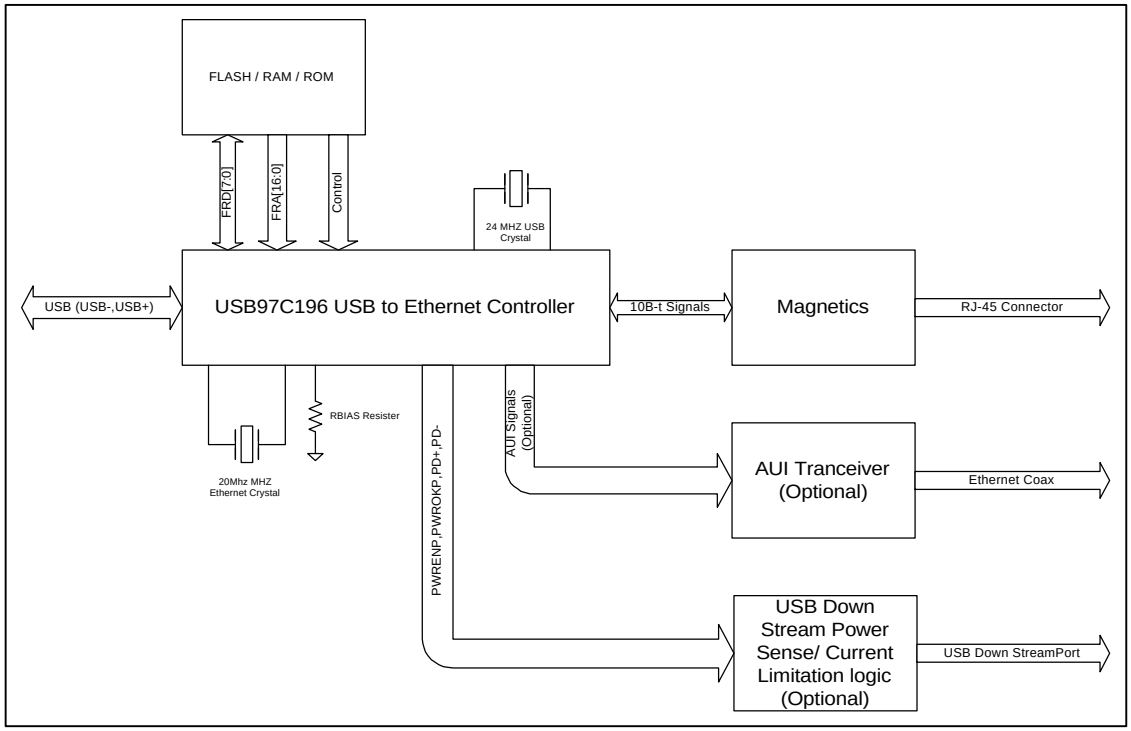
Ethernet frames being received from the LAN must be buffered. The received Ethernet packet is then segmented into USB packets as described above, and transported to the host via USB. The device detects the arrival of an inbound Ethernet frame and receives it into a buffer. The contents of that buffer are delivered on demand over the USB to the host via a bulk IN endpoint.

The MMU of the device is “multiplexed” such that it can be read or written too by an Ethernet frame buffer, or a USB packet buffer chain. Since buffer sizes are different for USB and Ethernet and some control and status information needs to be embedded into each memory buffer, an additional Status Queue is defined to store envelope encapsulation information.

APPLICATIONS

The USB97C196 allows the designer to implement a connectivity solution based on the USB data model from/to the standard 802.x Ethernet data flow medium. This USB97C196 allows a designer the ability to utilize an embedded USB Hub to allow an additional USB port. This allows the “Use one – Get one back” paradigm, allowing the user the ability to gain an Ethernet connection while retaining the current USB port.

TYPICAL SYSTEM IMPLEMENTATION



This section will describe the requirements relating to the USB Communications device class specification relating to how Endpoint resources will be allocated. The next section titled “Specific Support for USB Based SAR as Defined in the USB CDC Specification” on page 24.

The Data-In/Data-Out pipe mechanism is the same for all networking device models supported by the CDC specification. It is independent of the media type (e.g., Cable, xDSL, Ethernet) or media data type (e.g. Ethernet frames).Typical USB-based Networking devices will support bulk transfers as the default configuration to exchange data between a host and the USB device. While each data packet of a bulk endpoint is limited to the maximum packet size defined in the associated endpoint descriptor, it should be noted that a host might request multiple bulk USB protocol packets within a single USB frame. The USB97C196 will support this requirement. For maximum throughput, a Networking device must be prepared to transfer multiple bulk packets within a single USB frame. Some USB-based Networking device implementations may support isochronous data transfers in addition to (or instead of) bulk transfers as in the USB97C196. Isochronous transfers guarantee data throughput and bounded latency, consistent with the needs of real-time streams. Isochronous data errors are reported to receiver, but no data integrity (i.e.,retransmission) is provided by the USB link. The Data Class Interface Descriptor protocol code for all Networking Control Models is 00h.

USB provides no inherent flow control mechanism for isochronous pipes, and the CDC specification defines no higher level mechanism for doing so. Instead, it is assumed that the host software is responsible for doing traffic shaping as necessary to match any end-to-end negotiation. If the networking device is performing traffic shaping, then either a bulk endpoint should be used, or the flow control methods should be provided using vendor-specific methods. The Data Class interface of a networking device shall have a minimum of two interface settings. The first setting (the default interface setting) includes no endpoints and therefore no networking traffic is exchanged whenever the default interface setting is selected. One or more additional interface settings are used for normal operation, and therefore each includes a pair of endpoints (one IN, and one OUT) to exchange network traffic. The Firmware will select an alternate interface setting to initialize the network aspects of the device and to enable the exchange of network traffic. To recover the network aspects of a device to known states, select the default interface setting (with no endpoints) and then select the appropriate alternate interface setting. This action will flush device buffers, clear any filters or statistics counters and will cause NETWORK_CONNECTION and CONNECTION_SPEED_CHANGE notifications to be sent to the host.

As stated previously for almost any type of USB attached networking device, a mechanism is needed where both the networking device and the Host can delineate the beginning and ending of a *segment* within the data stream delivered by an endpoint pipe. This positive delineation is done using a USB short packet mechanism. When a segment spans *N* USB packets, the first packet through packet *N-1* shall be the maximum packet size defined for the USB endpoint. If the *N*th packet is less than maximum packet size the USB transfer of this short packet will identify the end of the segment. If the *N*th packet is exactly maximum packet size, it shall be followed by a zero-length packet (which is a short packet) to assure the end of segment is properly identified. When transmitting data to the networking device, it is assumed that the client of the host USB driver

takes the appropriate actions to cause a short packet to be sent to the networking device. For segments with lengths that are an even multiple of the pipe's "max packet size", the ability to write a buffer of zero length is required to generate this short packet.

The host and the attached network device must negotiate to establish the maximum segment size. The upper limit for this is usually a function of the buffering capacity of the attached device, but there may be other factors involved as well. For networking devices that exchange Ethernet frames, the size of a segment is also negotiable. Typical Ethernet frames are 1514 bytes or less in length (not including the CRC), but this could be longer (e.g., 802.1Q VLAN tagging).

The Ethernet Networking Control Model is used for exchanging Ethernet framed data between the device and host. A Communication Class interface is used to configure and manage various Ethernet functions, where an "Ethernet Networking Control Model" SubClass code is indicated in the descriptor definition of its Communication Class interface. A Data Class interface is used to exchange Ethernet encapsulated frames sent over USB. These frames shall include everything from the Ethernet destination address (DA) up to the end of the data field. The CRC checksum must not be included for either send or receive data. It is the responsibility of the device hardware to generate and check CRC as required for the specific media. Receive frames that have a bad checksum must not be forwarded to the host. This implies that the device must be able to buffer at least one complete Ethernet frame.

Although a typical USB Networking device stays in an "always connected" state, some Networking device management requests are required to properly initialize both the device and the host networking stack. There also may be occasional changes of device configuration or state, e.g., adding multicast filters.

USB TO ETHERNET CDC DEVICE CLASS ENDPOINT ALLOCATION

The SMSC USB97C196 is a specialized Device in the sense that its USB Endpoints have specific attributes that are unique to the USB to Ethernet bridge design paradigm. The USB97C100 USB endpoints of which it has 6, are not the same. The Endpoints and their functionality is shown in on page 24. The Table describes the USB97C196 Endpoint definition and usage relative to the section above Specific Support for USB Based SAR as Defined in the USB CDC Specification as defined starting on page 13.

USB ENDPOINT NUMBER	ENDPOINT ATTRIBUTES
Endpoint 0	Control Endpoint – This Endpoint corresponds to the Standard USB Common Device Class definition of the “Control Endpoint.” All of the requirements of the “Control Endpoint” as defined in the USB 1.1 specification must be adhered too. Please refer to the USB 1.1 Revision specification for additional details
Endpoint 1	Interrupt Endpoint – This Endpoint is a general purpose Endpoint that conforms to the standard protocols defined in the USB specification. Bulk, Isochronous and Interrupt options are available. This Endpoint however will be initialized as an “Interrupt Endpoint” based on the CDC Ethernet implementation standard. It is to the description of the Firmware Driver developer if the Endpoint is a bi-directional or single direction interrupt Endpoint.
Endpoint 2	CDC Compliant IN SAR (Segmentation and Re-assembly) Endpoint – This Endpoint corresponds to the Host to USB to Ethernet Bridge LAN Bulk Data Reception Pipe. This pipe is a special Pipe. The SIE adheres to the rules as defined in the USB CDC Class specification relating to the rules regarding to the delineation of logical Ethernet Packets as defined in the previous section tilted “Specific Support for USB Based SAR as Defined in the USB CDC Specification” on page 13
Endpoint 3	CDC Compliant OUT SAR (Segmentation and Re-assembly) Endpoint – This Endpoint corresponds to the Host to USB to Ethernet Bridge LAN Bulk Data transmission Pipe. This pipe is a special Pipe. The SIE adheres to the rules as defined in the USB CDC Class specification relating to the rules regarding to the delineation of logical Ethernet Packets as defined in the previous section tilted “Specific Support for USB Based SAR as Defined in the USB CDC Specification” on page 13
Endpoint 4	Generic USB Pipe – This Endpoint corresponds to a standard USB Pipe that has the ability to be a USB Bulk, Isochronous, Control, and Interrupt Pipe. This endpoint is optional for standard CDC Compliant applications. It is available for the system developer if the requirement is need to enhance feature support.
Endpoint 5	Generic USB Pipe – This Endpoint corresponds to a standard USB Pipe that has the ability to be a USB Bulk, Isochronous, Control, and Interrupt Pipe. This endpoint is optional for standard CDC Compliant applications. It is available for the system developer if the requirement is need to enhance feature support

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 page Via MEM_BANK reg. Select	External FLASH
0x8000-0xBFFF	Fixed 16k page 0x00000-0x03FFF FLASH	External FLASH
0x4000-0x7FFF	Movable 16k FLASH page; 1 of 8 16k pages in External FLASH (0x00000-0x1FFFF) selected by MEM_BANK Register Default: 0x04000- 0x07FFF FLASH Via MEM_BANK reg. Select	External FLASH External FLASH External FLASH External FLASH
0x0000-0x3FFF	Fixed 16k FLASH Page 0x00000-0x03FFF FLASH	External FLASH

Data Space

Table 4 - MCU Data Memory Map

8051 ADDRESS	DATA SPACE	ACCESS
0xC000-0xFFFF	Movable 16k page Default : 0x04000-0x07FFF FLASH Select Via MEM_BANK reg.	External FLASH
0x8000-0xBFFF	Fixed 16k page 0x00000-0x03FFF FLASH	External FLASH
0x7000-0x7FFF	0x7FD0-0x7FEF Ethernet Register I/O Window 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 0x6000-0x6FFF: entire MMU RAM is mapped here	Internal
0x5000-0x5FFF		Not used
0x4000-0x4FFF		Not used
0x3000-0x3FFF		Not used
0x2000-0x2FFF		Not used
0x0100-0x1FFF		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 SMSC USB97C196. These addresses will appear on the FLASH bus, but the read and write strobes will be inhibited.

Data Space and Code Space mappings

The diagrams shown in FIGURE 7 - MCU TO EXTERNAL CODE SPACE MAP DIAGRAM and FIGURE 8 - MCU TO EXTERNAL DATA MEMORY MAP DIAGRAM on pages 26 and 27 respectively, describe the Data Space and Code Space mappings between the External Flash and the internal 8051 MCU memory map.

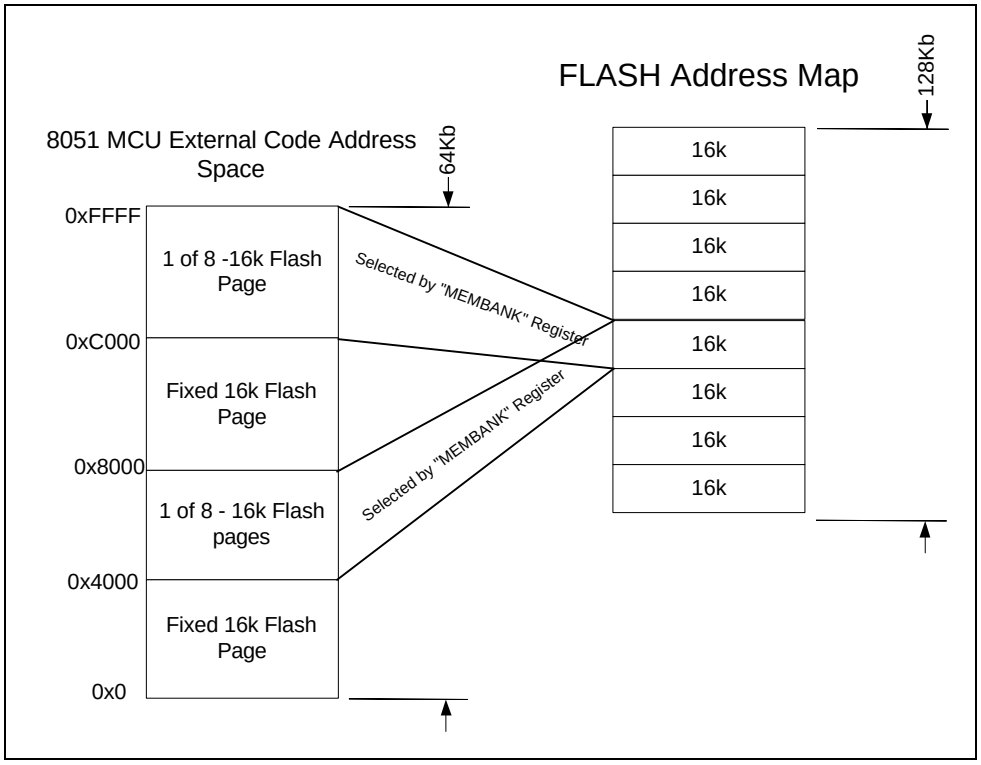


FIGURE 7 - MCU TO EXTERNAL CODE SPACE MAP DIAGRAM

Note 1: The USB97C196 presents a 17-bit address to the flash that is obtained by concatenating three bits with the lower 14 bits [13:0] of the address presented by the MCU. The three bits are Determined as follows:

a. If the page is movable, then the upper three bits are copied from MEM_BANK.
b. If the page is fixed, then the upper three bits are set to zero.

The lower 14 bits [13:0] of the 16-bit MCU address represent the offset into the 16K page which is being referenced. The upper two bits [15:14] represent the page being accessed in the MCU's address space; replacing these two bits by (the larger) MEM_BANK allows mappings to occur.

From the MCU's perspective though, it has one 64K-address space for code and one 64K-address space for data.

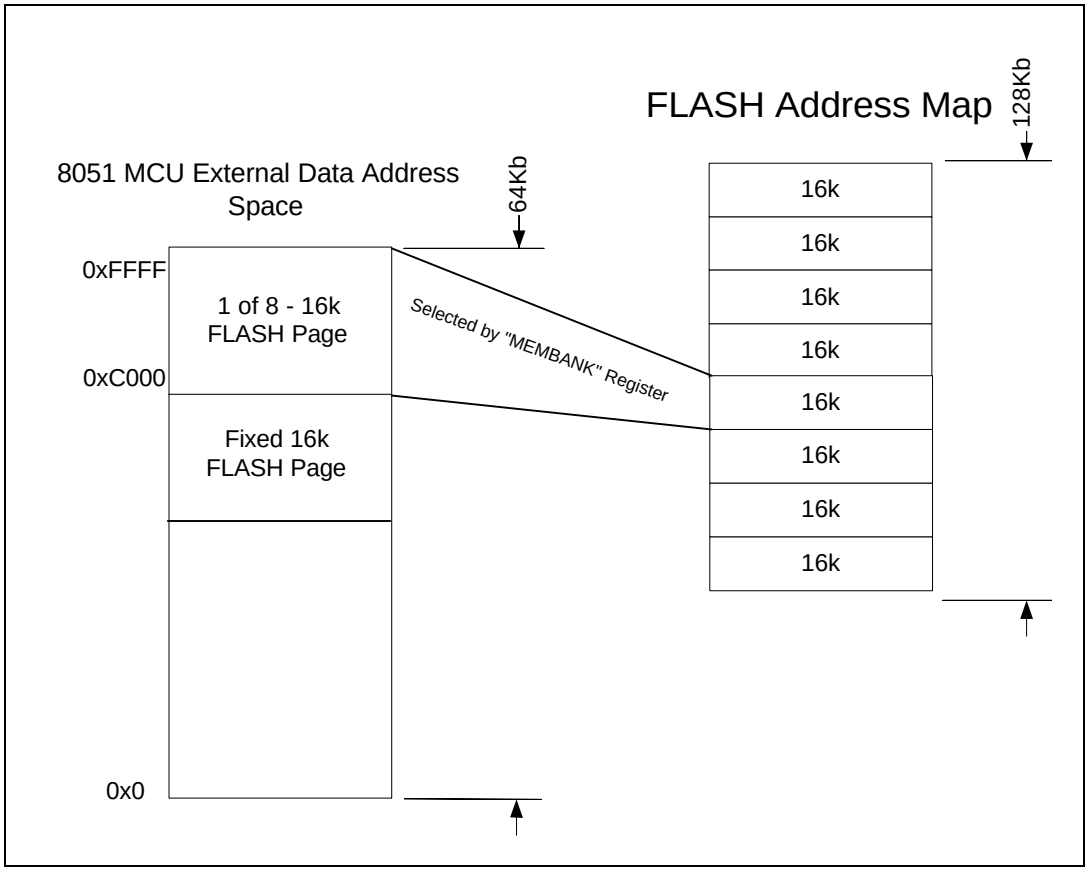


FIGURE 8 - MCU TO EXTERNAL DATA MEMORY MAP DIAGRAM

MCU Block Register Summary

Table 5 - MCU Block Register Summary

ADDRESS (HEX)	NAME	R/W	DESCRIPTION	PAGE
RUNTIME REGISTERS				
7F00	ISR_0	R	INT0 Source Register	33
7F01	IMR_0	R/W	INT0 Mask Register	36
7F02	ISR_1	R	INT1 Source Register	37
7F03	IMR_1	R/W	INT1 Mask Register	39
7F06	DEV_REV	R	Device Revision Register	39
7F07	DEV_ID	R	Device ID Register	39
UTILITY REGISTERS				
7F18	GPIOA_DIR	R/W	GPIO Configuration Register	40
7F19	GPIOA_OUT	R/W	GPIO Data Output Register	41
7F1A	GPIOA_IN	R	GPIO Data Input Register	41
7F1B	UTIL_CONFIG	R/W	Miscellaneous Configuration Register	42
POWER MANAGEMENT REGISTERS				
7F27	CLOCK_SEL	R/W	8051 Clock Select Register	46
7F29	MEM_BANK	R/W	Flash Bank Select	46
7F2A	WU_SRC_1	R	Wakeup Source	47
7F2B	WU_MSK_1	R/W	Wakeup Mask	47
7F2C	Reserved	R	Reserved	
7F2D	Reserved	R	Reserved	
MCU TEST REGISTERS				
7F7D	MCU_TEST3	N/A	Reserved for Test	
7F7E	MCU_TEST2	N/A	Reserved for Test	
7F7F	MCU_TEST1	N/A	Reserved for Test	

MMU Block Register Summary

Table 6 - MMU Block Register Summary

ADDRESS (HEX)	NAME	R/W	DESCRIPTION	PAGE
MMU REGISTERS				
6000	MMU_DATA	R/W	8051-MMU Data Window Register FIFO	59
7F50	PRL	R/W	8051-MMU Pointer Register (Low)	60
7F51	PRH	R/W	8051-MMU Pointer Register (High) & R/W	60
7F52	MMUTX_SEL	R/W	8051-MMU TX FIFO Select for Commands	61
7F53	MMUCR	W	8051-MMU Command Register	62
7F54	ARR	R	8051-MMU Allocation Result Register	65
7F55	PNR	R/W	8051-MMU Packet Number Register	66
7F56	MIR	R	MEMORY INFORMATION REGISTER	71
7F57	TX/RX_MGMT	R/W	USB TX/RX Management Register 1	76
7F58	USB_RX_FIFO	R	USB Receive Packet Number FIFO Register	72
7F59	USB_RX_FIFO_ Length_Low	R	USB Receive FIFO Packet Length Register Low	72
7F5A	USB_RX_FIFO_ Length_High	R	USB Receive FIFO Packet Length Register High	72
7F5B	Ethernet_RX_FIFO	R	ETHERNET Receive Packet Number FIFO Register	73
R	RX_FIFO_Length_ Low		ETHERNET Receive FIFO Packet Length Register Low	73
7F5D	RX_FIFO_Length_ High	R	ETHERNET Receive FIFO Packet Length Register High	73
7F60	USB_TXSTAT_A	R/W	USB TRANSMIT FIFO STATUS REGISTER A	74
7F61	USB_TXSTAT_B	R/W	TRANSMIT FIFO STATUS REGISTER B	75
7F62	PNBCRL	R/W	PACKET NUMBER Byte Count REGISTER Low	66
7F63	PNBCRH	R/W	PACKET NUMBER Byte Count REGISTER High	66
7F64	USB_POP_TX	R	USB POP TX FIFO	69

ADDRESS (HEX)	NAME	R/W	DESCRIPTION	PAGE
7F65	USB_POP_TX_Done	R	USB POP TX FIFO Done	70
7F66	ETH_TX	R	Ethernet TX FIFO	70
7F67	Eth_POP_TX_DONE	R	Ethernet POP Transmit FIFO Done Register	71
7F6C	PNCRH	R/W	Conversion Byte Count REGISTER Low	68
7F6D	PNCRL	R/W	Conversion Byte Count REGISTER High	68
7F6E	MMU_TESTx	N/A	Reserved for Test	
7F6F	MMU_TESTx	N/A	Reserved for Test	

SIE Block Register Summary

Table 7 - SIE Block Register Summary

ADDRESS (HEX)	NAME	R/W	DESCRIPTION	PAGE
SIE Control Registers				
7F80	EP_CTRL0	R/W	Endpoint 0 Control Register	82
7F81	EP_CTRL1	R/W	Endpoint 1 Control Register	82
7F82	EP_CTRL2	R/W	Endpoint 2 Control Register	82
7F83	EP_CTRL3	R/W	Endpoint 3 Control Register	82
7F84	EP_CTRL4	R/W	Endpoint 4 Control Register	82
7F85	EP_CTRL5	R/W	Endpoint 5 Control Register	82
7F86-7F8F	RESERVED	R	RESERVED	
7F90	FRAMEL	R	USB Frame Count Low	87
7F91	FRAMEH	R	USB Frame Count High	87
7F92	SIE_ADDR	R/W	USB Local Address Register	87
7F93	SIE_STAT	R	SIE Status Register	88
7F94	SIE_CTRL	R/W	SIE Control Register	89
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 1	R/W	SIE Configuration Register 1	90
7F99	ALT_ADDR	R/W	Secondary Local Address Register	88
7F9A	SIE_TST	R/W	Reserved Test Register	
7F9B	SIE_TST	R/W	Reserved Test Register	
7F9C	SIE_TST	R/W	Reserved Test Register	
7F9D	SIE_TST	R/W	Reserved Test Register	
7F9E	ALT_ADDR2	R/W	Reserved	
7F9F	ALT_ADDR3	R/W	Reserved	
7FA9	SIE_CTRL2	R/W	SIE Control Register 2	89
7FAA	EPCMD	W	Endpoint Command Register	85
7FAC	NONCTRL_EP	R/W	NonControl Endpoint Register	84
HUB BLOCK CONTROL REGISTERS				
7FA0	IdVendor-Low Byte	R/W	Low byte Vendor ID	94
7FA1	IdVendor-High Byte	R/W	High byte Vendor ID	94
7FA2	IdProduct-Low Byte	R/W	Low byte Product ID	94
7FA3	IdProduct-High Byte	R/W	High byte Product ID	94
7FA4	BcdDevice - Low Byte	R/W	USB device release number	94
7FA5	BcdDevice - High Byte	R/W	USB device release number	94
7FA6	HubControl1	R/W	Hub Control Register 1	95
7FA7	HubControl2	R/W	Hub Control Register 2	96

EPH (Ethernet) Block Register Summary

Table 8 - EPH (Ethernet) Block Register Summary

ADDRESS (HEX)	NAME	R/W	DESCRIPTION	PAGE
Ethernet Control Registers				
7FD0	ETCR1	R/W	Ethernet Transmit Control Register1	48
7FD1	ETCR2	R/W	Ethernet Transmit Control Register2	48
7FD2	EPH1	R/W	EPH STATUS REGISTER1	49
7FD3	EPH2	R/W	EPH STATUS REGISTER2	50
7FD4	RCR1	R/W	Ethernet Receive Control Register 1	51
7FD5	RCR2	R/W	Ethernet Receive Control Register 2	52
7FD6	ECR1	R	Ethernet counter Register 1	53
7FD7	ECR2	R	Ethernet counter Register 2	53
7FD8	EconfigR1	R/W	Ethernet CONFIGURATION REGISTER	54
7FD9	EconfigR2	R/W	Ethernet CONFIGURATION REGISTER	55
7FDA	IAR0	R/W	INDIVIDUAL ADDRESS REGISTER 0	55
7FDB	IAR1	R/W	INDIVIDUAL ADDRESS REGISTER 1	55
7FDC	IAR2	R/W	INDIVIDUAL ADDRESS REGISTER 2	56
7FDD	IAR3	R/W	INDIVIDUAL ADDRESS REGISTER 3	56
7FDE	IAR4	R/W	INDIVIDUAL ADDRESS REGISTER 4	56
7FDF	IAR5	R/W	INDIVIDUAL ADDRESS REGISTER 5	56
7FE0	MCT0	R/W	Multicast Table Register 0	56
7FE1	MCT1	R/W	Multicast Table Register 1	56
7FE2	MCT2	R/W	Multicast Table Register 2	56
7FE3	MCT3	R/W	Multicast Table Register 3	57
7FE4	MCT4	R/W	Multicast Table Register 4	57
7FE5	MCT5	R/W	Multicast Table Register 5	57
7FE6	MCT6	R/W	Multicast Table Register 6	57
7FE7	MCT7	R/W	Multicast Table Register 7	57

MCU REGISTER DESCRIPTIONS

Below describes the registers the 8051 MCU deals with in terms of handling the USB to Ethernet Bridge functions.

MCU Runtime Registers

Table 9 - Interrupt 0 Source Register

ISR_0 (0x7F00 – RESET=0x00)			INTERRUPT 0 SOURCE REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EPH	R	EPH INT - Set when the Ethernet Protocol Handler section indicates one out of various possible special conditions. This bit merges exception type of interrupt sources, whose service time is not critical to the execution speed of the 8051 Core MCU firmware. The exact nature of the interrupt can be obtained from the EPH Status Register (EPHSR), and enabling of these sources can be done via the Control Register. The possible sources are: 1. LINK_OK transition. 2. CTR_ROL - Statistics counter roll over. 3. TXENA cleared - A fatal transmit error occurred forcing TXENA to be cleared. TX_SUC will be low and the specific reason will be reflected by the bits: • TXUNRN – Transmit underrun. • SQET - SQE Error. • LOST CARR – Lost Carrier. • LATCOL - Late Collision. • 16COL – 16 collisions.
6	ETH_RX_PKT	R	1 = A Packet Number (PNR) has been successfully queued on the Ethernet RXFIFO and set (1) when a receive interrupt is generated. The first packet number to be serviced can be read from the FIFO PORTS register. The RCV INT bit is always the logic complement of the REMPTY bit in the FIFO PORTS register.
5	ETH_TX_EMPTY	R	1 = Whenever the Ethernet EPH TX FIFO becomes empty. This will occur when the last queued packet has been successfully been transmitted by the Ethernet EPH.
4	ETH_TX_PKT	R	Ethernet TX INT - Set when at least one packet transmission was completed. The first packet number to be serviced can be read from the Ethernet FIFO PORTS register. The Eth_TX_PKT bit is always the logic complement of the TEMPTY bit in the FIFO PORTS register.

ISR_0 (0x7F00 – RESET=0x00)			INTERRUPT 0 SOURCE REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
3	USB_RX_PKT	R	1 = A Packet Number (PNR) has been successfully queued on the RXFIFO.
2	USB_TX_EMPTY	R	1 = Whenever an enabled TX Endpoint's FIFO becomes empty. This will occur when the last queued packet in one of the 6 TX queues is successfully transferred to the Host.
1	USB_TX_PKT	R	1 = A Packet was successfully transmitted.
0	Eth_RCV_OVRN	R	Ethernet Receive Overrun Interrupt – This bit is set when the Ethernet EPH, in conjunction with the MMU does not have enough buffer memory to receive the packet.

These bits are automatically cleared each time this register is read. Therefore, each time this register is read all pending interrupts must be serviced before continuing normal operation.

Software 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.

2 – ETH_TX EMPTY - Set if the EPH TX FIFO goes empty. This bit can be used to generate a single interrupt at the end of a sequence of Ethernet packets en-queued for transmission. This bit latches the empty condition, and the bit will stay set until it is specifically cleared by writing the acknowledge register with the TX EMPTY INT bit set. If a real time reading of the FIFO empty is desired, the bit should be first cleared and then read.
The TX EMPTY INT ENABLE should only be set after the following steps:
a) A packet is enqueued for transmission
b) The previous empty condition is cleared (acknowledged).

3 - If the Firmware uses AUTO RELEASE (the default) mode then it should enable TX EMPTY INT as well as TX INT. The TX EMPTY INT will be set when the complete sequence of packets is transmitted. TX INT will be set if the sequence stops due to a fatal error on any of the packets in the sequence.

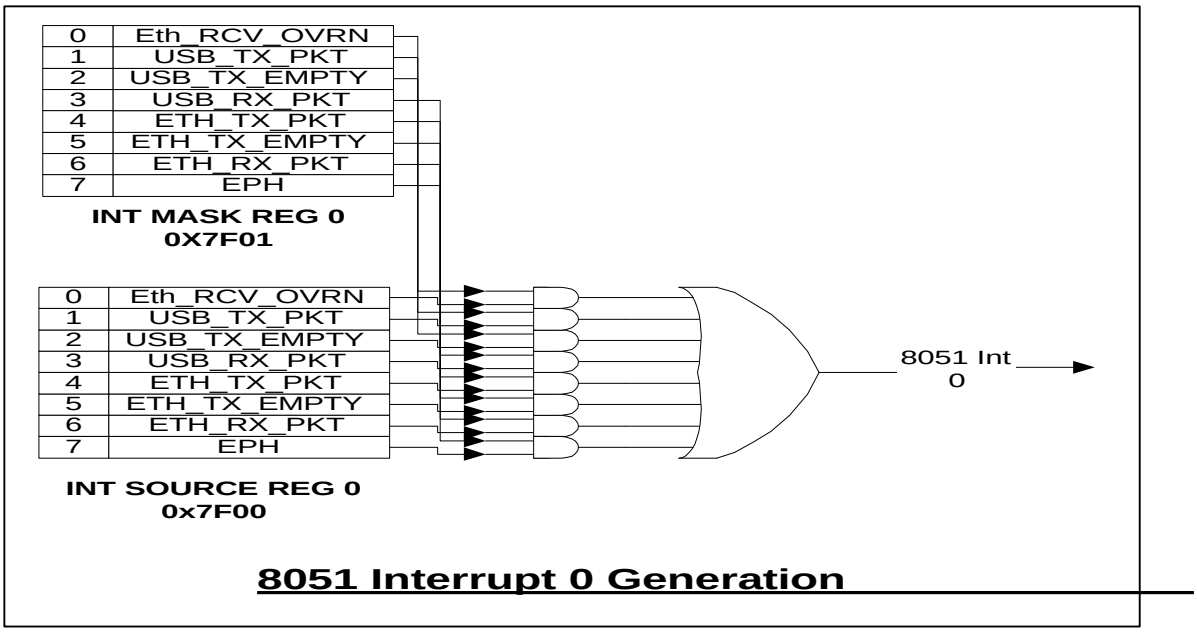

0 Eth_RCV_OVRN
1 USB_TX_PKT
2 USB_TX_EMPTY
3 USB_RX_PKT
4 ETH_TX_PKT
5 ETH_TX_EMPTY
6 ETH_RX_PKT
7 EPH
INT MASK REG 0
0X7F01
0 Eth_RCV_OVRN
1 USB_TX_PKT
2 USB_TX_EMPTY
3 USB_RX_PKT
4 ETH_TX_PKT
5 ETH_TX_EMPTY
6 ETH_RX_PKT
7 EPH
INT SOURCE REG 0
0x7F00
8051 Int 0
8051 Interrupt 0 Generation

Table 10 - Interrupt 0 Mask

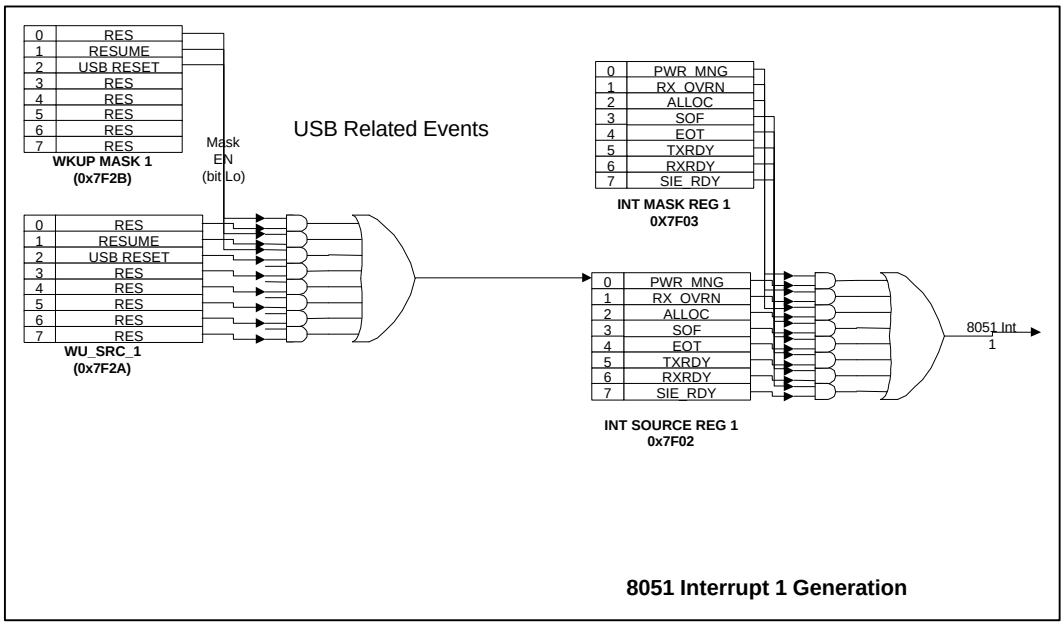
IMR_0 (0x7F01- RESET=0xFF)			INTERRUPT 0 MASK REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EPH	R/W	Ethernet Received Packet MMU Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
6	ETH_RX_PKT	R/W	Ethernet Received Packet MMU Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
5	ETH_TX_EMPTY	R/W	Ethernet Transmit Queue Empty MMU Interrupt 0 = Enable Interrupt 1 = Mask Interrupt
4	ETH_TX_PKT	R/W	Ethernet Transmit Packet MMU Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
3	USB_RX_PKT	R/W	Received Packet MMU Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
2	USB_TX_EMPTY	R/W	Transmit Queue Empty MMU Interrupt 0 = Enable Interrupt 1 = Mask Interrupt
1	USB_TX_PKT	R/W	Transmit Packet MMU Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
0	Eth_RCV_OVRN	R	Ethernet Receive Overrun Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt

Table 11 - Interrupt 1 Source Register

ISR_1 (0x7F02- RESET=0x00)			INTERRUPT 1 SOURCE REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	SIE_RDY	R	This is the READY signal, directly from the SIE block. Can be used in diagnostics.
6	RFRDY	R	This is the “Not Empty” signal from Rx SIE FIFO.
5	TXRDY	R	This is the “Not Full” signal from Tx SIE FIFO.
4	EOT	R	1 = The SIE returned to Idle State. Marks the end of each transaction.
3	SOF	R	1 = When a Start of Frame token is correctly decoded. Generated by the write strobe to the Frame Count register.
2	ALLOC	R	1 = MCU Software Allocation Request complete interrupt. This interrupt is not generated for hardware (SIEDMA or EPHDMA) allocation requests. This bit is the complement of the FAILED bit in the ALLOCATION RESULT register. The ALLOC INT ENABLE bit should only be set following an allocation command, and cleared upon servicing the interrupt.
1	USB_RX_OVRN	R	1 = A receive error has occurred within the domain of the USB SIE DMA Core. The hardware automatically recovers from this condition after its cause has been alleviated (e.g. any partially allocated packets will be released. See Note 2). The RX_OVRN bit of the EPHSR will also be set, but if a new packet is received it will be cleared. The RX_OVRN INT bit, however, latches the overrun condition for the purpose of being polled or generating an interrupt, and will only be cleared by writing the acknowledge register with the RX_OVRN INT bit set.
0	PWR_MNG	R	1 = A wakeup or power management event in the WU_SRC_1 or the Signature Wakeup register has gone active.

Note 1: These bits are cleared each time this register is read .

Note 2: The RX_OVRN interrupt should be considered by firmware as a general Receive Overrun for the Core, meaning that a packet destined for the Core 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, there may be too many transmit packets queued (thereby consuming memory) for the device to receive anything, or the last packet may have been corrupted on the wire, or then endpoint received a packet while stalled. If it is not empty, then one or more receive packets must be de-queued 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.



8051 Interrupt 1 Generation

Table 12 - Interrupt 1 Mask

IMR_1 (0x7F03- RESET=0xFF)			INTERRUPT 1 MASK REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	SIE_RDY	R/W	SIE Ready Interrupt mask 0 = Enable Interrupt 1 = Mask Interrupt
6	RFRDY	R/W	Rx SIE FIFO “Not Empty” Interrupt mask 0 = Enable Interrupt 1 = Mask Interrupt
5	TXRDY	R/W	Tx SIE FIFO “Not Full” Interrupt mask 0 = Enable Interrupt 1 = Mask Interrupt
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	USB_RX_OVRN	R/W	USB Receive Overrun Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt
0	PWR_MNG	R/W	Power management Interrupt Mask 0 = Enable Interrupt 1 = Mask Interrupt

Table 13 - Device Revision Register

DEV_REV (0x7F06- RESET=0x41)			DEVICE REVISION REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	ASCII 'A' HEX 0x41	R	This register should match the revision level stamped with the date code on the part.

Table 14 - Device Identification Register

DEV_ID (0x7F07- RESET=0x23)			DEVICE IDENTIFICATION REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	BCD '23'	R	This register should match the last two digits of the mask number.

GENERAL PURPOSE IO REGISTER DEFINITIONS

The registers described below allow the USB USB97C196 the ability to control external logic for multiple functions. Examples of possible functions is for Power Control , Power management, HID devices, Re-enumeration logic, etc. Additional functionality would include the possibility of external logic generating events or system status changes that the USB to LAN device would consider important such as LAN Link State change ro media change.

GPIO Direction Register

This register determines the input/output state of the associated GPIO Pin. Please refer to the pin definition in the Pin definition section of the specification. 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 15 - 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	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

GPIO Output Register

When the associated GPIO pins are configured as outputs via the GPIO direction register defined above, writing to this register sets the associated out put pin high or low. Note: When bits 0 through 3 are configured 8051 UART pins defined below, the state of the pin(s) are defined by the 8051 block.

Table 16 - 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	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

GPIO Input Register

When the associated GPIO pins are configured as inputs via the GPIO direction register defined above, reading to this register will latch the state of the associated input pin. Note: When bits 0 through 3 are configured 8051 UART pins defined below, the state of the pin(s) are defined by the 8051 block.

Table 17 - 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	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

Utility Configuration Register

This register defines the function of the associated GPIO pin(s) defined below. When bits 0 through 3 are configured 8051 UART pins defined below, the state of the pin(s) are defined by the 8051 block.

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.

Table 18 - Utility Configuration Register

UTIL_CONFIG (0x7F1B- RESET=0x00)			**UTILITY CONFIGURATION REGISTER**
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:4]	Reserved	R	Reserved
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'

GPIO MUXING BLOCK DIAGRAM

The diagram below describes the internal logic that is implemented to allow the pins defined below to be configured as either input pins, output pins or 8051 UART pins.

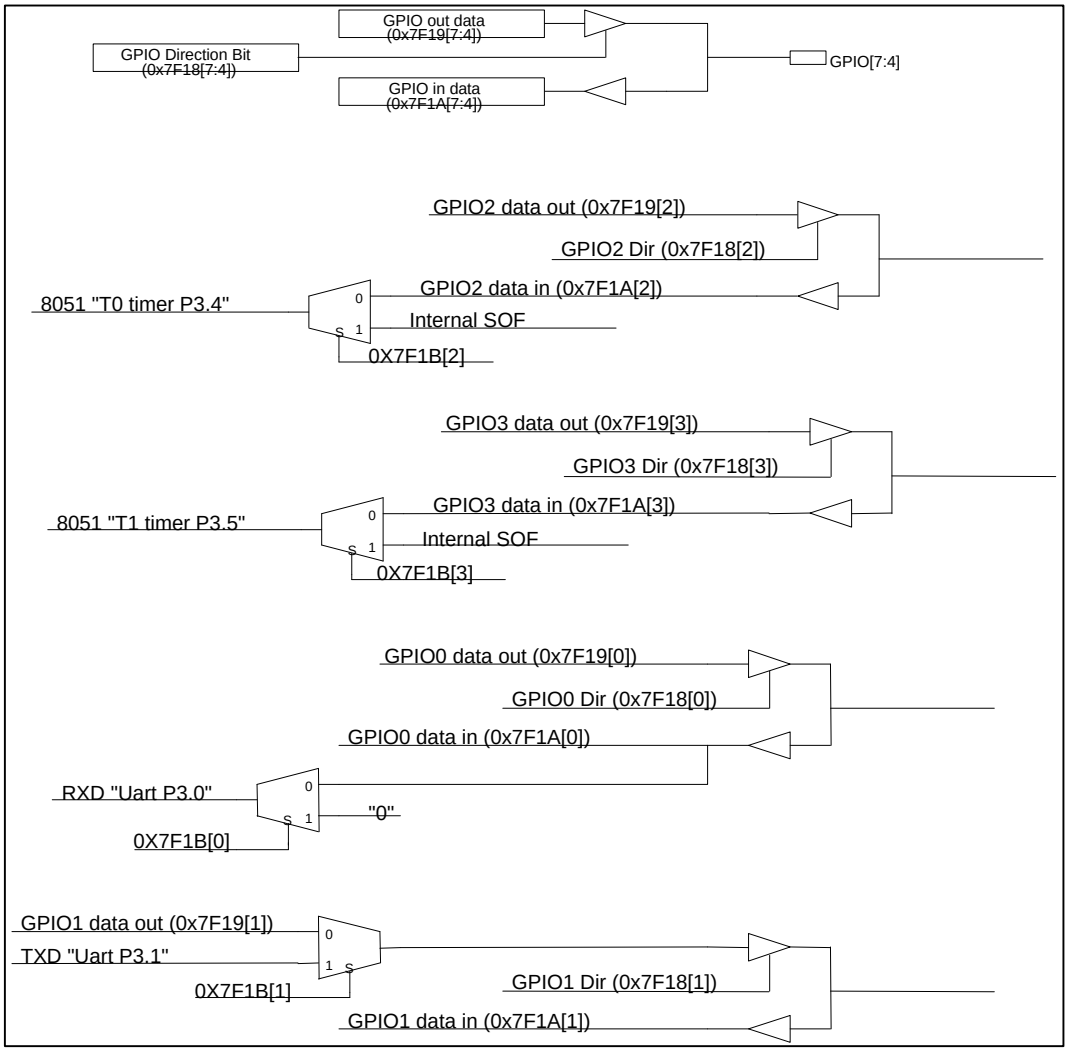


FIGURE 9 - GPIO MUXING BLOCK DIAGRAM

DEVICE POWER MANAGEMENT

The USB97C196 is required to support Intel and Microsoft's "OnNow" initiative as well as conform to the USB 1.1 Specification regarding Suspend and Resume Power Management. This requirements is derived from requirements to support the PC 98 and PC 99 design guide. These initiatives define the needed functionality for both a USB device as well as a 10Mb/Sec Ethernet device.

The SMC USB97C196 needs to support USB suspend and resume as well as power management in a sleep state being able to "wakeup" from a low power state by USB events and Ethernet Packet Signature detection.

USB97C196 Power States

The following are the USB97C196 power states:

ETHERNET Function

Table 19 - Ethernet Wakeup Modes

	MODE	DESCRIPTION
i	D0	• Fully operational
ii	D1	• Same as D3(Hot) (Systems Engineering Note)
Iii	D2	• Same as D3(Hot) (Systems Engineering Note)
Iv	D3(Hot)	• USB Suspend Mode (When device is powered up, this is the state the Ethernet function is initialized to) • External ENDEC mode supported in this state
v	D3(Cold)	• No Ethernet Transmit or Receive functionality possible • Internal function state not preserved • Clocks are off in Ethernet Function

USB SIE Function

The USB97C196 will initially default, on Reset, to low-power USB Function mode. Low-power devices are limited to 500uA of suspend current. The USB97C196 device, after it is configured by the MCU, will be configured to a High-power device and will be enabled as a remote wakeup source.

The USB97C196 SIE will:

- The USB97C196 will initially default, on Reset, to low-power USB Function mode. The device will be limited to 100ua (500uA – External System Power requirements) of suspend current.
- Once the Device configured, the USB97C196 is limited to draw up to 2.0 mA (2.5mA – External logic) during suspend.
- In Bus-Powered HUB mode, the USB97C196 internal configured bus-powered hub may also consume a maximum of 2.5mA, with 500uA allocated to the single external port.
- While in the Suspend state, the amplitude of the current spike will not exceed the device power allocation budget of 100mA. A maximum of 1.0 second is allowed for an averaging interval.
- The average current cannot exceed the average suspend current limit (500uA) during any 1.0s interval.

Network Device Power Conservation transition table (Systems Engineering Note)

The software firmware can place the SMSC USB USB97C196 Network function into the operational mode D0 or one of the low power modes D1, D2, or D3 via the Power management registers. The SMSC USB USB97C196 powers up in state D3. The effect of the USB Bus reset assertion on the controller power states is select-able.

MCU Power Management and Utility Registers

Table 20 - MCU 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
[2:0]	Reserved	R	Reserved

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.

Code FLASH Bank Select Register

Table 21 - FLASH Bank Select Register

CMEM_BANK (0x7F29 - RESET=0x01)			CODE FLASH BANK SELECT REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	R	Reserved
[2:0]	A[16:14]	R/W	These bits selects which 16k page resides at 0x4000-0x7FFF in Code Space. These bits also mirror the selection of the 16K page at 0xC000 – 0xFFFF in code space as well. The 0x0000-0x3FFF page will always reflect the 16K FLASH page 0 (0x00000-0x03FFF). The fixed page is also mirrored at 0x8000 through 0xBFFF.

Wakeup Source 1 Register

Table 22 - Wakeup Source 1 Register

WU_SRC_1 (0x7F2A - RESET=0x00)			WAKEUP SOURCE 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	R	Reserved
2	USB_Reset	R	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).
1	Resume	R	This bit is set on detection of Global Resume state (when there is a transition from the "J" state while in Global Suspend).
0	Reserved '0'	R	Reserved

Note 1: Only low to high transitions for the associated inputs sets these bits.
Note 2: These bits are cleared each time this register is read .
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.

Wakeup Mask 1 Register

Table 23 - 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 1A: Interrupt events enabled by these bits are routed to the PWR_MNG Bit 0 in the ISR_1 register.
Note 1B: Only low to high transitions for the associated inputs sets these bits.
Note 2: These bits are cleared each time this register is read .
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.

Ethernet Transmit Control Register1

Table 24 – Ethernet Transmit Control Register1

ETCR1 (0x7FD0 – RESET=0x00)			Ethernet Transmit Control Register 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	PAD_EN	R/W	When set, the USB97C196 will pad transmit frames shorter than 64 bytes with 00. Does not pad frames when reset.
[6:4]	Reserved	R	Reserved
3	PHY_SPEED	R/W	This bit configures the SMSC USB97C196 to operate at a 1Mbps Ethernet rate 1 = 1Mbps 0 = 10Mbps
2	FORCOL	R/W	When set the USB97C196 monitors carrier while transmitting. It must see its own carrier by the end of the preamble. If it is not seen, or if carrier is lost during transmission, the transmitter aborts the frame without CRC and turns itself off. When this bit is clear the transmitter ignores its own carrier. Defaults low.
1	LOOP	R/W	Local Loopback. When set, transmit frames are internally looped to the receiver after the encoder/decoder. Collision and Carrier Sense are ignored. No data is sent out. Defaults low to normal mode.
0	TXENA	R/W	Transmit enabled when set. Transmit is disabled if clear. When the bit is cleared the USB97C196 will complete the current transmission before stopping. When stopping due to an error, this bit is automatically cleared.

Ethernet Transmit Control Register2

Table 25 – Ethernet Transmit Control Register2

ETCR2 (0x7FD1 – RESET=0x00)			Ethernet Transmit Control Register 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:6]	Reserved	R	Reserved – Same as LAN91C94
5	EPH LOOP	R/W	Internal loopback at the EPH block. Does not exercise the encoder decoder. Serial data is looped back when set. Defaults low. Note: After exiting the loopback test, an SRESET in the ECOR or the SOFT_RST in the RCR must be set before returning to normal operation.
4	STP SQET	R/W	Stop transmission on SQET error. If set, stops and disables transmitter on SQE test error. Does not stop on SQET error and transmits next frame if clear. Defaults low.

ETCR2 (0x7FD1 – RESET=0x00)			Ethernet Transmit Control Register 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
3	Reserved	R	Reserved – Same as LAN91C94
2	MON_ CSN	R/W	When set the USB97C196 monitors carrier while transmitting. It must see its own carrier by the end of the preamble. If it is not seen, or if carrier is lost during transmission, the transmitter aborts the frame without CRC and turns itself off. When this bit is clear the transmitter ignores its own carrier. Defaults low.
1	Reserved	R	Reserved - Same as LAN91C94
0	NOCRC	R/W	Does not append CRC to transmitted frames when set, allows software to insert the desired CRC. Defaults to zero, namely CRC inserted.

EPH STATUS REGISTER1

Table 26 – EPH STATUS REGISTER1

EPH1 (0x7FD2 – RESET=0x00)			EPH STATUS REGISTER1
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	TX_DEFR	R	Transmit Deferred. When set, carrier was detected during the first 6.4 usec of the inter frame gap. Cleared at the end of every packet sent.
6	LTX_BRD	R	Last transmit frame was a broadcast. Set if frame was broadcast. Cleared at the start of every transmit frame.
5	SQET	R	Signal Quality Error Test. The transmitter opens a 1.6 us window 0.8 us after transmission is completed and the receiver returns inactive. During this window, the transmitter expects to see the SQET signal from the transceiver. The absence of this signal is a 'Signal Quality Error' and is reported in this status bit. Transmission stops and EPH INT is set if STP_SQET is in the TCR is also set when SQET is set. This bit is cleared by setting TXENA high.
4	16COL	R	16 collisions reached. Set when 16 collisions are detected for a transmit frame. TXENA bit in TCR is reset. Cleared when TXENA is set high.
3	LTX_MULT	R	Last transmit frame was a multicast. Set if frame was a multicast. Cleared at the start of every transmit frame.
2	MUL_COL	R	More than one collision occurred while transmitting the current Ethernet Packet. Cleared at the start of every transmit frame.
1	SNGL_COL	R	Single collision detected for the last transmit frame. Set when a collision is detected. Cleared when TX_SUC is high at the end of the packet being sent.

EPH1 (0x7FD2 – RESET=0x00)			EPH STATUS REGISTER1
BIT	**NAME**	**R/W**	**DESCRIPTION**
0	TX_SUC	R	Last transmit was successful. Set if transmit completes without a fatal error. This bit is cleared by the start of a new frame transmission or when TXENA is set high. Fatal errors are: 16 collisions SQET fail and STP_SQET = 1 FIFO Underrun Carrier lost and MON_CSN = 1 Late collision

EPH STATUS REGISTER2

Table 27 – EPH STATUS REGISTER2

EPH2 (0x7FD3 - RESET=0x00)			EPH STATUS REGISTER2
BIT	**NAME**	**R**	**DESCRIPTION**
7	TX UNRN	R	Transmit Under run. Set if Under run occurs, it also clears TXENA bit in TCR. Cleared by setting TXENA high. This bit should never be set under normal operation.
6	LINK_ OK	R	State of the 10BASE-T Link Integrity Test. A transition on the value of this bit generates an interrupt when the LE ENABLE bit in the Control Register is set.
5	RX_ OVRN	R	Upon FIFO overrun, the receiver asserts this bit and clears the FIFO. The receiver stays enabled. After a valid preamble has been detected on a subsequent frame, RX_OVRN is de-asserted. The RX_OVRN INT bit in the Interrupt Status Register will also be set and stay set until cleared by the CPU. Note that receive overruns could occur only if receive memory allocations fail.
4	CTR _ROL	R	Counter Roll over. When set one or more 4 bit counters have reached maximum count (15). Cleared by reading the ECR register.
3	EXC _DEF	R	Excessive deferral. When set last/current transmit was deferred for more than 1518 * 2 byte times. Cleared at the end of every packet sent.
2	LOST CARR	R	Lost carrier sense. When set indicates that Carrier Sense was not present at end of preamble. Valid only if MON_CSN is enabled. This condition causes TXENA bit in TCR to be reset. Cleared by setting TXENA bit in TCR.
1	LATCOL	R	Late collision detected on last transmit frame. If set a late collision was detected (later than 64 byte times into the frame). When detected the transmitter JAMs and turns itself off clearing the TXENA bit in ETCR. Cleared by setting TXENA in ETCR.

EPH2 (0x7FD3 - RESET=0x00)			EPH STATUS REGISTER2
BIT	**NAME**	**R**	**DESCRIPTION**
0	Reserved	R	Reserved – Read as 0

Ethernet Receive Control Register 1

Table 28 – Ethernet Receive Control Register 1

RCR1 (0x7FD4 – RESET=0x00)			RECEIVE CONTROL REGISTER1
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	Reserved	R	Reserved
6	Reserved	R	Reserved
5	Reserved	R	Reserved
4	Reserved	R	Reserved
3	Reserved	R	Reserved
2	ALMUL	R/W	When set, accepts all multicast frames (frames in which the first bit of DA is '1'). When clear accepts only the multicast frames that match the multicast table setting. Defaults low.
1	PRMS	R/W	Promiscuous mode. When set receives all frames.
0	RX_ ABORT	R/W	This bit is set if a receive frame was aborted due to length longer than 1532 bytes. The frame will not be received. The bit is cleared by RESET or by the 8051 embedded controller writing it low.

Ethernet Receive Control Register 2

Table 29 – Ethernet Receive Control Register 2

RCR2 (0x7FD5 – RESET=0x00)			RECEIVE CONTROL REGISTER2
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	SOFT RST	R/W	Software activated CSMA/CD Block Reset. Active high. Initiated by writing this bit high and terminated by writing the bit low. The USB97C196 configuration is not preserved, except for Configuration Registers.
6	FILT CAR	R/W	Filter Carrier. When set filters leading edge of carrier sense for 12 bit times. Otherwise recognizes a receive frame as soon as carrier sense is active.
5	Reserved	R	Reserved
4	Reserved	R	Reserved
3	Reserved	R	Reserved
2	Reserved	R	Reserved
1	STRIP CRC	R/W	When set it strips the CRC on received frames. When clear the CRC is stored in memory following the packet. Defaults low.
0	RXEN	R/W	Enables the receiver when set. If cleared, completes receiving current frame and then goes idle. Defaults low on reset.

Ethernet Counter Register1

Table 30 – Ethernet Counter Register1

ECR1 (0x7FD6 – RESET=0x00)			Ethernet Counter Register1
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:4]	MULTIPLE COLLISION COUNT	R	Four (4) bit counter of the number of multiple collisions.
[3:0]	SINGLE COLLISION COUNT	R	Four (4) bit counter of the number of Single Collisions

Each four bit counter is incremented every time the corresponding event, as defined in the EPH STATUS REGISTER bit description, occurs. Note that the counters can only increment once per enqueued transmit packet, never faster, limiting the rate of interrupts that can be generated by the counters. For example if a packet is successfully transmitted after one collision the SINGLE COLLISION COUNT field is incremented by one. If a packet experiences between 2 to 16 collisions, the MULTIPLE COLLISION COUNT field is incremented by one. If a packet experiences deferral the NUMBER OF DEFERRED TX field is incremented by one, even if the packet experienced multiple deferrals during its collision retries.

The COUNTER REGISTER facilitates maintaining statistics in the AUTO RELEASE mode where no transmit interrupts are generated on successful transmissions.

Reading the register in the transmit service routine will be enough to maintain statistics.

Ethernet Counter Register2

Table 31 – Ethernet Counter Register2

ECR2 (0x7FD7 - RESET=0x00)			Ethernet Counter Register2
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:4]	NUMBER OF EXC. DEFERRED TX	R	Four (4) bit counter of the number of excessive deferred transmit packets
[3:0]	NUMBER OF DEFERRED TX	R	Four (4) bit counter of the number of deferred transmit packets

Ethernet Configuration Register 1

Table 32 – Ethernet Configuration Register 1

EConfigR1 (0x7FD8 – RESET=0x00)			Ethernet Configuration Register 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	LE ENABLE	R/W	LE ENABLE - Link Error Enable. When set it enables the LINK_OK bit transition as one of the interrupts merged into the EPH INT bit. Defaults low (disabled). Writing this bit also serves as the acknowledge by clearing previous LINK interrupt conditions.
6	CR ENABLE	R/W	CR ENABLE - Counter Roll over Enable. When set it enables the CTR_ROL bit as one of the interrupts merged into the EPH INT bit. Defaults low (disabled).
5	TE ENABLE	R/W	TE ENABLE - Transmit Error Enable. When set it enables Transmit Error as one of the interrupts merged into the EPH INT bit. Defaults low (disabled). Transmit Error is any condition that clears TXENA with TX_SUC staying low as described in the EPHSR register.
4	RCV_BAD	R/W	RCV_BAD - When set (1), bad CRC packets are received. When clear bad CRC packets do not generate interrupts and their memory is released.
3	DIS LINK	R/W	DISABLE LINK – This bit is used to disable the 10BASE-T link test functions. When this bit is high the USB97C196 disables link test functions by not generating nor monitoring the network for link pulses. In this mode the USB97C196 will transmit packets regardless of the link test, the EPHSR LINK_OK bit will be set and the LINK LED will stay on. When low the link test functions are enabled. If the link status indicates FAIL, the EPHSR LINK_OK bit will be low, while transmit packets enqueued will be processed by the USB97C196, transmit data will not be sent out to the cable.
2	FULL STEP	R	FULL STEP - This bit is used to select the signaling mode for the AUI port. When set the AUI port uses full step signaling. Defaults low to half step signaling. This bit is only meaningful when AUI SELECT is high.
1	SET SQLCH	R	SET SQLCH - When set, the squelch level used for the 10BASE-T receive signal is 240mV. When clear the receive squelch level is 400mV. Defaults low.
0	AUI SELECT	R	AUI SELECT - When set the AUI interface is used, when clear the 10BASE-T interface is used. Defaults low.

Ethernet Configuration Register 2

Table 33 – Ethernet Configuration Register 2

EConfigR2 (0x7FD9 – RESET=0x00)			Ethernet Configuration Register 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:2]	Reserved	R	Reserved
1	CLK_DISABLE	R/W	When SIE is in suspend mode, stop 20MHz clock to the Ethernet and disable the ENDEC
0	Ethernet AUTO RELEASE	R/W	AUTO RELEASE - When cleared (0), transmit pages are released by transmit completion if the transmission was successful (when TX_SUC is set). In that case there is no status word associated with its packet number, and successful packet numbers are not even written into the TX COMPLETION FIFO. A sequence of transmit packets will only generate an interrupt when the sequence is completely transmitted (TX EMPTY INT will be set), or when a packet in the sequence experiences a fatal error (TX INT will be set). Upon a fatal error TXENA is cleared and the transmission sequence stops. The packet number that failed is the present in the FIFO PORTS register, and its pages are not released, allowing the CPU to restart the sequence after corrective action is taken.

INDIVIDUAL ADDRESS REGISTER 0 - 5

These registers should be initialized by the 8051 MCU firmware upon hardware reset. The values for these registers must be unique for each system this device is designed into. As a result, the data can be stored in the FLASH or ROM or downloaded from the Host via a Device Class descriptor. The option is application specific.

Table 34 – INDIVIDUAL ADDRESS REGISTER 0

IAR0 (0x7FDA – RESET=0x00)			INDIVIDUAL ADDRESS REGISTER 0
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	IAR[0]	R/W	0th byte value of a 6 byte Ethernet Address Bit 0 of Individual Address 0 register corresponds to the first bit of the address on the cable.

Table 35 – INDIVIDUAL ADDRESS REGISTER 1

IAR1 (0x7FDB – RESET=0x00)			INDIVIDUAL ADDRESS REGISTER 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	IAR[1]	R/W	1st byte value of a 6 byte Ethernet Address

Table 36 – INDIVIDUAL ADDRESS REGISTER 2

IAR2 (0x7FDC – RESET=0x00)			INDIVIDUAL ADDRESS REGISTER 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	IAR[2]	R/W	2nd byte value of a 6 byte Ethernet Address

Table 37 – INDIVIDUAL ADDRESS REGISTER 3

IAR3 (0x7FDD – RESET=0x00)			INDIVIDUAL ADDRESS REGISTER 3
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	IAR[3]	R/W	3rd byte value of a 6 byte Ethernet Address

Table 38 – INDIVIDUAL ADDRESS REGISTER 4

IAR4 (0x7FDE – RESET=0x00)			I NDIVIDUAL ADDRESS REGISTER 4
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	IAR[4]	R/W	4th byte value of a 6 byte Ethernet Address

Table 39 – INDIVIDUAL ADDRESS REGISTER 5

IAR5 (0x7FDF – RESET=0x00)			INDIVIDUAL ADDRESS REGISTER 5
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	IAR[5]	R/W	5th byte value of a 6 byte Ethernet Address

Table 40 – MULTICAST TABLE REGISTER 0

MCT0 (0x7FE0 – RESET=0x00)			MULTICAST TABLE REGISTER 0
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	MCT[0]	R/W	Multicast Table Register 0

Table 41 – MULTICAST TABLE REGISTER 1

MCT1 (0x7FE1 – RESET=0x00)			MULTICAST TABLE REGISTER 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	MCT[1]	R/W	Multicast Table Register 1

Table 42 – MULTICAST TABLE REGISTER 2

MCT2 (0x7FE2 – RESET=0x00)			MULTICAST TABLE REGISTER 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	MCT[2]	R/W	Multicast Table Register 2

Table 43 – MULTICAST TABLE REGISTER 3

MCT3 (0x7FE3 – RESET=0x00)			MULTICAST TABLE REGISTER 3
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	MCT[3]	R/W	Multicast Table Register 3

Table 44 – MULTICAST TABLE REGISTER 4

MCT4 (0x7FE4 – RESET=0x00)			MULTICAST TABLE REGISTER 4
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	MCT[4]	R/W	Multicast Table Register 4

Table 45 – MULTICAST TABLE REGISTER 5

MCT5 (0x7FE5 – RESET=0x00)			MULTICAST TABLE REGISTER 5
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	MCT[5]	R/W	Multicast Table Register 5

Table 46 – MULTICAST TABLE REGISTER 6

MCT6 (0x7FE6 – RESET=0x00)			MULTICAST TABLE REGISTER 6
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	MCT[6]	R/W	Multicast Table Register 6

Table 47 – MULTICAST TABLE REGISTER 7

MCT7 (0x7FE7 – RESET=0x00)			MULTICAST TABLE REGISTER 7
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	MCT[7]	R/W	Multicast Table Register 7

The 64 bit multicast table is used for group address filtering. The hash value is defined as the six most significant bits of the CRC of the destination addresses. The three msb's of the hash value determine the register to be used (MCT0-7), while the three lsb's determine the bit within the register.

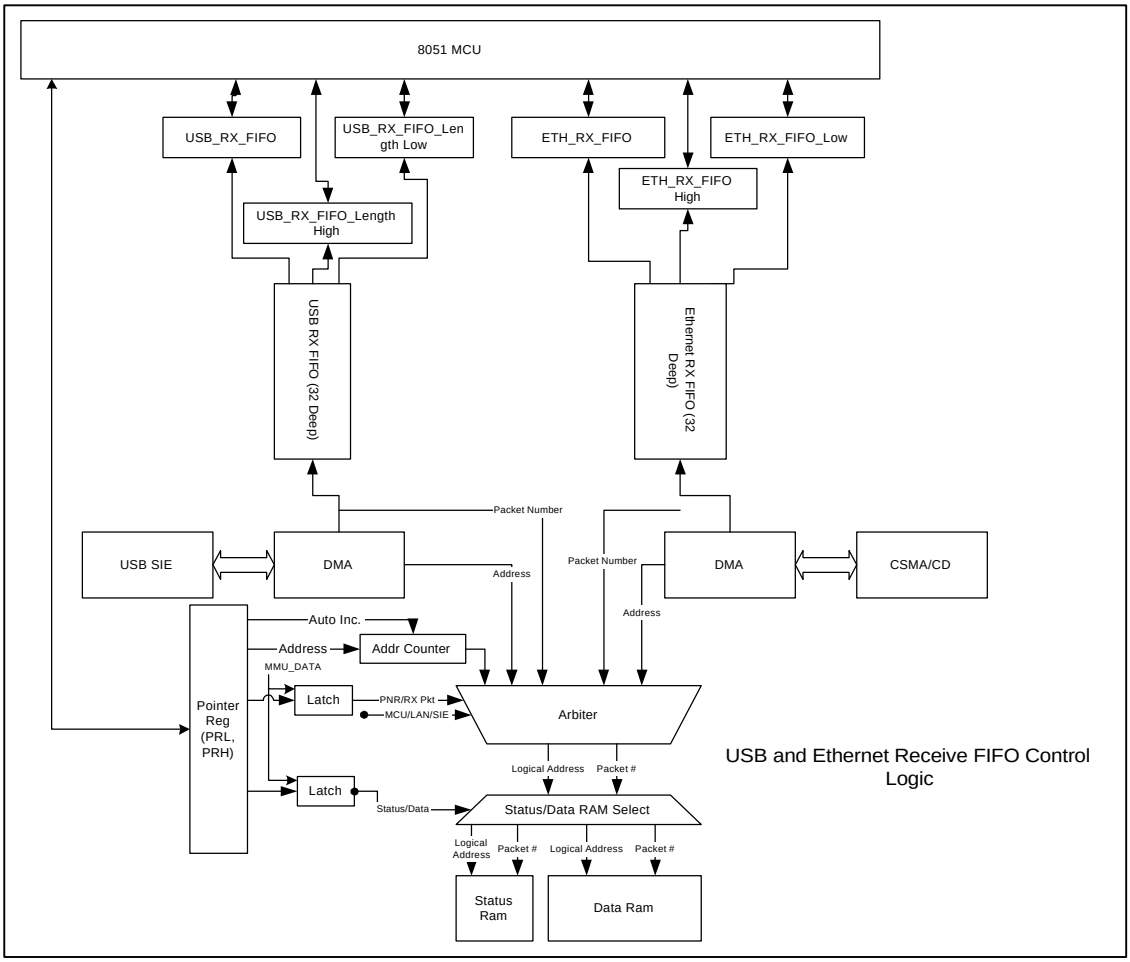
- If the appropriate bit in the table is set, the packet is received.
- If the ALMUL bit in the RCR1 register is set, all multicast addresses are received regardless of the multicast table values.

Hashing is only a partial group addressing filtering scheme, but being the hash value available as part of the receive status word, the receive routine can reduce the search time significantly. With the proper memory structure, the search is limited to comparing only the multicast addresses that have the actual hash value in question.

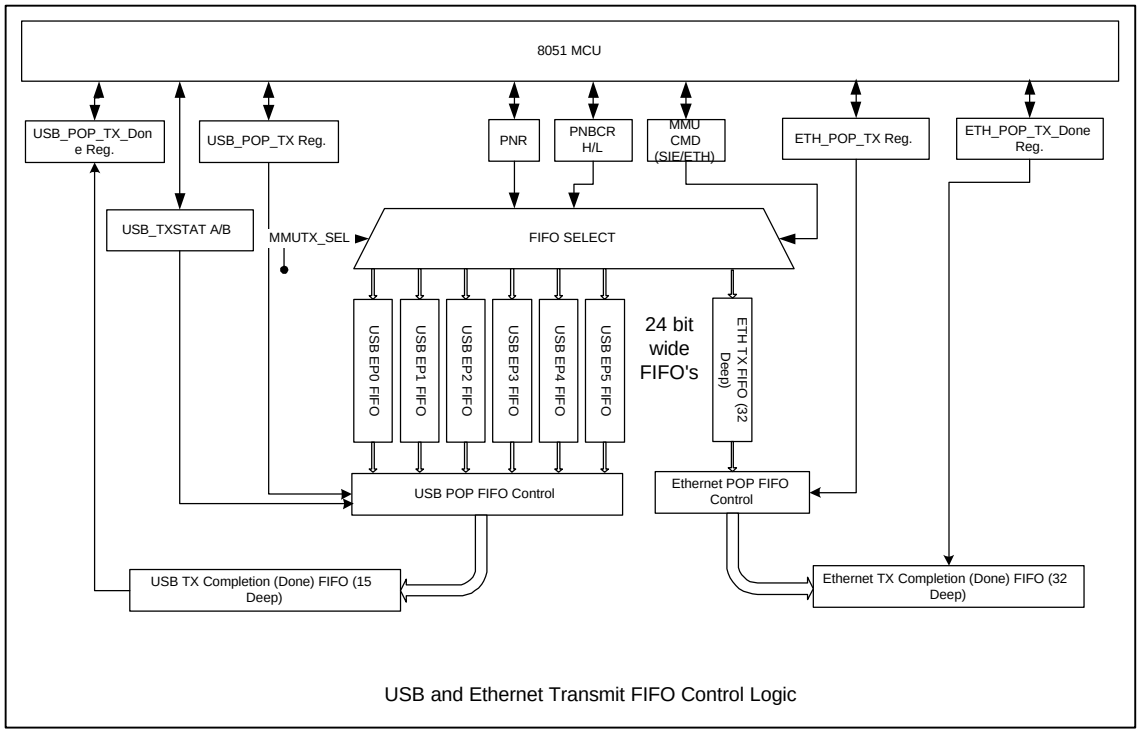
MEMORY MANAGEMENT UNIT (MMU) REGISTER DESCRIPTIONS

The embedded 8051 processor to support both Ethernet and USB transmission and reception of packets will access the registers described below.

The Following Diagrams describe the register and control blocks for USB and Ethernet Transmission and reception. The registers described are defined in the next section.



USB and Ethernet Receive FIFO Control Logic



USB and Ethernet Transmit FIFO Control Logic

USB / Ethernet MMU Data Window Register

This register is used to allow the embedded processor (8051 CORE) read / write access to the MMU data buffers presently addressed by the pointer register or Packet Number Register.

This register is mapped into two uni-directional FIFOs that allow moving bytes to and from the MMU regardless of whether the pointer address is even or odd. Data goes through the write FIFO into memory, and is pre-fetched from memory into the read FIFO.

Table 48 - 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 as selected by bit 4 of the MMUCR register, 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.

USB / Ethernet Pointer Register (Low)

Table 49 - Pointer Register (Low)

PRL (0x7F50)			POINTER REGISTER (LOW)
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	A[7:0]	R/W	LSB of the (0-1536 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.

USB / Ethernet Pointer Register (High)

This is the High byte portion of the Pointer Register. Under normal operation, the embedded 8051 MCU should load the low byte first and the high byte last.

When RCV is set, the address refers to the receive area and uses the output of either USB SIE or Ethernet RX FIFO as the packet number. When RCV is cleared, the address refers to the transmit area of either the USB SIE or Ethernet TX FIFO and uses the packet number in the PNR (Packet Number Register).

The READ bit determines the type of access to follow. If the READ bit is high, the operation is intended as a read. If the READ bit is low the operation is a write. Loading a new pointer value, with the READ bit high, generates a pre-fetch into the Data Register for read purposes.

Reading back of the pointer will indicate the value of the address last accessed by the 8051 (rather than the last pre-fetched). This allows any interrupt routine that uses the pointer, to save it and restore it without affecting the process being interrupted.

The Pointer Register should not be loaded until 400ns after the last write operation to the Data Register to ensure that the Data Register FIFO is empty.

Table 50 - 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 at 0x6000 is the packet pointed to by the RX packet number on the top of the RX Packet Number FIFO as selected by bit 4 of the MMUCR register..
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.

PRH (0x7F51)			POINTER REGISTER (HIGH)
BIT	**NAME**	**R/W**	**DESCRIPTION**
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	STATUS/DATA	R/W	STATUS/DATA Control Select – This bit selects whether the 8051 is accessing the STATUS HEADER or DATA area of the MMU's allocated buffer pointer too by the PNR (Packet Number Register). 1 = Status Buffer 0 = Data Buffer
3	Reserved	R	Reserved
[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.

SIE Transmit FIFO Select Register

This register selects which of the six (6) TX FIFO's (one FIFO for each associated Endpoint) is selected when the embedded 8051 processor is enqueueing a packet into one of the USB SIE FIFO's for transmission to the USB Host.

Table 51 – SIE Transmit FIFO Select Register

MMUTX_SEL (0x7F52)			TRANSMIT FIFO SELECT REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	R	Reserved
[2:0]	EP[2:0]	R/W	This register selects which Endpoint (6) of the USB SIE Control Block in terms of the MMU Command Register (0x7F53) Commands "110" and "111" will affect when issued to the MMU. See MMU Command Register for details.

MMU Command Register

This register is used by the embedded 8051 controller to control the memory allocation, de-allocation, TX FIFO and RX FIFO control of the MMU. The three command bits determine the command issued as described below:

Table 52 - MMU Command Register

MMUCR (0x7F53)			MMU COMMAND REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:5]	MMU_CMD	W	MMUCR COMMAND SET
4	SIE/EPH	R/W	USB SIE/Ethernet EPH Control Select – This bit, when set (1) will allow the MMU to execute an MMU Command, defined above targeted for the USB SIE. When this bit is cleared (0), will allow the MMU to execute an MMU command targeted for the Ethernet EPH.
[3:0]	N[3:0]/Busy	R/W	When these bits are read, the number of 128 byte Pages is allocated by the following formula as (N(3:0) + 1) * 128 bytes. For example, a value of N[3..0]=0000, indicates 1 page, and a value of N[3..0]=1011 indicates 12 pages, or 1536 bytes. When this bit is read, this bit indicates that the MMU is either busy allocating memory or it has completed its command. 1 = Busy 0 = Completed (successful)

MMU COMMAND Bits 7, 6, and 5 Description:

000 NOOP, No operation – This operation will be executed when the 8051 needs to switch between the Ethernet EPH and USB SIE RX and TX FIFO's.

001 Allocate Memory for TX: N3-0 specifies how many 128 byte pages to allocate for the USB or Ethernet packet (up to 12 pages allowed (1536 bytes) per packet.). Writing this command to this register, Immediately generates a "FAILED" code at the ARR and the code is cleared when the allocation is complete. This command can generate an ALLOC interrupt to the 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, EPH DMA engine or the SIEDMA) will cause the MMUCR to automatically continue the allocate command until all requested pages have been successfully allocated. The firmware software should never issue another allocate command until the previous allocate command has been successfully completed. The allocation time can take worst case (N[3:0] + 2) * 200ns**.**

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. This command removes the receive packet number from the RX FIFO and brings the next receive frame (if any) to the RX area (output of RX FIFO).

100 Remove and Release Top of RX FIFO : Same as (011), but also frees all memory used by the packet presently at the RX FIFO output. 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 (PACKET NUMBER REGISTER). This command should not be used for frames pending transmission. This command is typically used to remove transmitted frames from the TX Done FIFO, after reading their completion status.

110 Enqueue Packet into USB SIE Endpoint EP[3:0] or Ethernet EPH TX FIFO: 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 EP[3:0] or Ethernet EPH TX FIFO : Resets the associated 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.

Note 1: Only command "001" uses bits 3 through 0.

Note 2: When using the Reset TX Endpoint EP[3:0] or Ethernet EPH TX FIFO command, the CPU is responsible for releasing the memory associated with outstanding packets, or re-enqueuing them. Packet numbers in the completion FIFO can be read via the FIFO ports register before issuing the command.

Note 3: MMU commands releasing memory should only be issued if the corresponding packet number has memory allocated to it.

Note 4: MMU Commands 010 through 111 require the USB/ETH control bit to be set appropriately since the command can access the USB or Ethernet related FIFOs

MMU Command Sequencing

A second allocate command (command 2) should not be issued until the present one has completed. Completion is determined by reading the FAILED bit of the allocation result register or through the allocation interrupt.

A second release command should not be issued if the previous one is still being processed. The BUSY bit indicates that a release command is in progress. After issuing command A, the contents of the PNR should not be changed until BUSY goes low. After issuing command 8, command 6 should not be issued until BUSY goes low.

BUSY BIT - Readable at bit 0 of the MMU command register address. When set, indicates that MMU is still processing a release command. When clear, MMU has already completed last release command.

BUSY and FAILED bits are set upon the trailing edge of command.

Allocation Result Register

This register is updated upon an ALLOCATE MEMORY MMU command.

Table 53 - Allocation Result Register

ARR (0x7F54)			ALLOCATION RESULT REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	FAILED	R	FAILED - A zero indicates a successful allocation completion. If the allocation fails, the bit is set and only cleared when the pending allocation is satisfied. This bit defaults high, upon reset and reset MMU command. For polling purposes, the ALLOC_INT in the Interrupt Status Register should be used because it is synchronized to the read operation. Sequence: 1) Allocate Command 2) Poll ALLOC_INT bit until set 3) Read Allocation Result Register
[6:5]	Reserved	R	Reserved
[4:0]	P[4:0]	R	Packet number associated with the last memory allocation request. The value is only valid if the FAILED bit is clear. Returns Packet Number (0-31, 0x00-0x1F) from an allocation command. This can be written directly into the PNR register

Note: The value read from the ARR after an allocation request is intended to be written into the PNR as is, without masking higher bits (provided FAILED = 0).

Packet Number and Byte Count Registers

The value written into the two 8 bit registers defined below determines which packet number is accessible through the TX area. Also included is the Byte Length of the USB or Ethernet Packet. Some MMU commands use the number stored in this register as the packet number parameter. This register is cleared by a RESET or a RESET MMU Command.

Packet Number Register

The value written into this register determines which packet number is accessible through the USB SIE AND ETHERNET EPH TX area. Some MMU commands use the number stored in this register as the packet number parameter. This register is cleared by a RESET or a RESET MMU Command.

Table 54 - 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

Byte Count Register High

Table 55 - Packet Number Byte Count Register High

PNBCRL (0x7F63)			PACKET NUMBER Byte Count REGISTER High
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:3]	Reserved	R/W	Reserved
[2:0]	PacketByteCnt	R/W	PacketByteCnt – The High Three bits of the 11 bit Packet Byte Count

Byte Count Register Low

Table 56 - Byte Count Register Low

PNBCRH (0x7F62)			PACKET NUMBER Byte Count REGISTER Low
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	PacketByteCnt	R/W	PacketByteCnt – The low 8 bits of the 11 bit Packet Byte Count

Byte Count Conversion Registers

The two eight bit registers defined below are used to convert between the Ethernet and USB Byte count values. The MCU (8051) writes the received packet length into these registers which will convert the value as appropriate. The MCU(8051) can then read the converted length.

.

For Ethernet to USB:

1. Ethernet packet is received by EPH. The Ethernet packet length is always even.
2. If the Ethernet Data Packet Control Byte ODD bit is set (1), the actual packet length is the hardware packet length - 1. Therefore, Bit 6 of the PNCR High byte should be set.
3. If the Ethernet Data Packet Control Byte ODD bit is clear (0), the actual packet length is the hardware packet length - 2. Therefore, Bit 6 of the PNCR High byte should be cleared.
4. ETH2USB bit must be set (1).
5. The two registers are written with the packet length received from the RX packet length FIFO register AND the bits set defined above.
6. When this register set is read, the value of the two registers should be the value written into the USB TX FIFO packet length.

For USB to Ethernet:

1. USB packet is received from the SIE. The USB packet is the exact packet length.
2. ETH2USB bit must be cleared (0).
3. Bit 6 of the PNCR High byte is a don't care.
4. The two registers are written with the packet length received from the USB RX packet length FIFO register AND the bits set defined above.
5. When this register set is read, the value of the two registers should be the value written into the USB TX FIFO packet length.

Byte Count Conversion Register High

Table 57 - Byte Count Conversion Register High

PNCRH (0x7F6D)			PACKET NUMBER and Byte Count REGISTER High
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	ETH2USB	R/W	ETH2USB – This bit selects whether the conversion is for Ethernet to USB buffer transfers or USB to Ethernet Buffer transfers. ETH2USB = 0 -> USB to Ethernet Conversion ETH2USB = 1 -> Ethernet to USB Conversion
6	ODD/EVEN	R/W	ODD/EVEN – This bit controls the ODD/EVEN control byte in the Ethernet Data Buffer to USB Conversion. When set(1), indicates odd; when clear(0), indicates even.
[5:3]	Reserved	R	Reserved
[2:0]	PacketByteCnt	R/W	PacketByteCnt – The High Three bits of the 11 bit Packet Byte Count

Byte Count Conversion Register Low

Table 58 - Byte Count Conversion Register Low

PNCRL (0x7F6C)			PACKET NUMBER and Byte Count REGISTER Low
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	PacketByteCnt	R/W	PacketByteCnt – The Low Eight bits of the 11 bit packet Byte Count

USB and Ethernet POP Registers

The registers described below allow the 8051 MCU to manage the Ethernet and USB transmit FIFO's. The FIFO's can operate in two possible modes. The first mode is Automatic manage and release mode. In this mode, the SMSC USB97C196 will manage the completion and removal of transmit buffers upon transmit completion automatically. The buffers are removed and placed into the free buffer area of the MMU. This mode offers the most efficient method of buffer management and is the recommended mode of operation.

The second mode is manual release and removal. In this mode, the 8051 MCU can manage the TX done FIFO's manually, using the registers defined below. This method allows the MCU's firmware more flexibility in memory management, which may be a viable option in a particular application.

USB Tx FIFO POP Register

This register is used to help software manage USB 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 59 -USB POP TX FIFO

USB_POP_TX (0x7F64 – RESET=0x80)			USB POP TX FIFO
BIT	**NAME**	**R**	**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.

USB Tx Done FIFO POP Register

This register is used to help software manage USB TX Queues. All USB packets transmitted are are pushed onto this FIFO regardless of the state of the Auto Release bit in the **USB TX Management 1 Register.** When read, this register will return the Packet Number of the next packet on the TX Done queue, AND it will pop that Packet Number off of the selected TX DONE FIFO.

Table 60 -USB Tx DONE FIFO POP Register

USB_POP_TX_DONE (0x7F65 – RESET=0x80)			USB POP TX DONE FIFO
BIT	**NAME**	**R**	**DESCRIPTION**
7	POPTX_STAT	R	POP TX DONE FIFO empty status 0 = Has one or more TX packet 1 = Empty
6	FULL	R	When this bit is set (1), the FIFO is Full. When Clear (0), the FIFO is not Full
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 DONE FIFO pointer to by MMUTX_SEL. The TX DONE 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.

Ethernet Tx Awaiting FIFO Register

This register is used to help the 8051 MCU manage Ethernet TX Queues. This will provide a method to handle waiting Ethernet packet TX packet numbers if the Auto Release feature is not used. When read, this register will return the Packet Number of the next packet waiting on the Awaiting TX queue. Reading this register has no effect on the contents of the FIFO when this register is read by the MCU.

Table 61 – Ethernet Awaiting TX FIFO

Eth_TX_Awaiting (0x7F66 – RESET=0x80)			Eth_TX_Awaiting
BIT	**NAME**	**R**	**DESCRIPTION**
7	TX_Awaiting_ Stat	R	POP TX FIFO empty status 0 = Has one or more TX packet 1 = Empty
[6:5]	Reserved	R	Reserved
[4:0]	TX_Awaiting_ Value	R	This 5 bit value is the packet number or handle that is at the top of the TX FIFO. The TX FIFO is NOT popped when this register is read.

Ethernet Tx Done FIFO POP Register

This register is used to help software manage Ethernet TX Queues. This will provide a method to handle completed Ethernet packet TX packet numbers if the Auto Release feature is not used. When read, this register will return the Packet Number of the next packet waiting on the Completed TX queue AND it will pop that Packet Number off of the TX FIFO.

If the MCU programs the SMSC USB97C196 to not implement "Auto Release", it is the responsibility of the MCU to manually release the packet from memory.

Table 62 – Ethernet POP Done TX FIFO

Eth_POP_TX_Done (0x7F67 – RESET=0x80)			Eth_POP_TX_Done
BIT	**NAME**	**R**	**DESCRIPTION**
7	POPTXDone_ STAT	R	POP TX FIFO empty status 0 = Has one or more TX packet 1 = Empty
[6:5]	Reserved	R	Reserved
[4:0]	POPDone_TX	R	This 5 bit value is the packet number or handle that is at the top of the TX Done FIFO. The TX Done FIFO is popped when this register is read.

MEMORY INFORMATION REGISTER

Table 63 – MEMORY INFORMATION REGISTER

MIR (0x7F56 – RESET=0x20)			MEMORY INFORMATION REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
[5:0]	FREE PAGES AVAILABLE	R	FREE PAGES AVAILABLE - This register can be read at any time to determine the amount of free pages available. The register defaults 0x20 upon reset or upon the RESET MMU command. (Page size = 128 bytes)

USB Receive Data Packet Number and Length FIFO Registers

These registers, when read, will retrieve from the MMU the packet number pointing to the received data buffers Packet Number, Packet Length for processing. This packet which has come in on End Point 3 will then be forwarded to the Ethernet CSMA/CD engine to be forwarded onto the Ethernet network under normal operation.

USB Receive Packet Number FIFO Register

Table 64 - USB Receive Packet Number FIFO Register

USB_RX_FIFO (0x7F58)			USB Receive Packet Number FIFO Register
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	Empty	R	This bit is set when the Packet Number FIFO is empty
6	Full	R	This bit is set when the Packet Number FIFO is Full (Note: this bit may not be required since the FIFO will be 32 deep)
5	Bulk	R	When this bit is set (1), it indicates that the packet received is from the Bulk endpoint (3). When this bit is cleared (0), it indicates that the packet is on a Non-Bulk (control or general purpose) endpoint. This register allows the 8051 to determine if this is the dedicated Bulk Endpoint and as a result does not need to read the packet Status header.
[4:0]	Packet_Number	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.

USB Receive FIFO Packet Length Register Low

Table 65 – USB Receive FIFO Packet Length Register Low

USB_RX_FIFO_Length_Low (0x7F59)			USB Receive FIFO Packet Length Register Low
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	RX_FIFO_Length_ Low	R	This register defines the lower 8 bit packet size (in Bytes) of the packets received from the USB .

USB Receive FIFO Packet Length Register High

Table 66 – USB Receive FIFO Packet Length Register High

USB_RX_FIFO_Length_High (0x7F5A)			USB Receive FIFO Packet Length Register High
BIT	**NAME**	**R/W**	**DESCRIPTION**
[2:0]	RX_FIFO_Length_ High	R	This register defines the higher 3 bit packet size (in Bytes) of the packets received from the USB .

Ethernet Receive Data Packet Number and Length FIFO Registers

These registers, when read, will retrieve from the MMU the packet number pointing to the received data buffers Packet Number, Packet Length for processing. This packet which has come in on the Ethernet EPH Receive Block will then be forwarded to the USB SIE via 8051 control to be forwarded onto the USB under normal operation.

ETHERNET Receive Packet Number FIFO Register

Table 67 - Ethernet Receive Packet Number FIFO Register

Ethernet_RX_FIFO (0x7F5B)			ETHERNET Receive Packet Number FIFO Register
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	Empty	R	This bit is set when the Packet Number FIFO is empty.
6	Full	R	This bit is set when the Packet Number FIFO is Full (Note: this bit may not be required since the FIFO will be 32 deep).
5	Reserved	R	Reserved
[4:0]	Packet_Number	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.

ETHERNET Receive FIFO Packet Length Register Low

Table 68 – ETHERNET Receive FIFO Packet Length Register Low

RX_FIFO_Length_Low (0x7F5C)			ETHERNET Receive FIFO Packet Length Register Low
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:0]	RX_FIFO_Length_ Low	R	This register defines the lower 8 bit packet size (in Bytes) of the packets received from the ETHERNET EPH Block

ETHERNET Receive FIFO Packet Length Register High

Table 69 – ETHERNET Receive FIFO Packet Length Register High

RX_FIFO_Length_High (0x7F5D)			ETHERNET Receive FIFO Packet Length Register High
BIT	**NAME**	**R/W**	**DESCRIPTION**
[2:0]	RX_FIFO_Length_ High	R	This register defines the higher 3 bit packet size (in Bytes) of the packets received from the ETHERNET .

Note: Ethernet Byte Counts derived from the EPH block are always even. The odd/even determination of the Ethernet packet received must be determined by the ODD/EVEN bit in the Control word at the end of the Ethernet Receive packet buffer. Refer to the Ethernet Data buffer description in the following sections.

USB Transmit FIFO Status Registers

USB Transmit FIFO Status Register A

Table 70 – USB Transmit FIFO Status Register A

USB_TXSTAT_A (0x7F60 – RESET=0xAA)			USB 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	

USB Transmit FIFO Status Register B

Table 71 - Transmit FIFO Status Register B

USB_TXSTAT_B (0x7F61 - RESET=0x0A)			TRANSMIT FIFO STATUS REGISTER B
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:4]	Reserved	R	Reserved
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	

USB TX / RX Management Register 1

Table 72 – USB TX Management Register 1

TX/RX_MGMT (0x7F57 - RESET= 0x00)			USB TX/RX Management Register 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
[7:4]	Reserved	R	Reserved – Read as a zero (0)
3	SIE_RCV_Bad	R/W	SIE Receive Bad – This bit when set (1), will allow USB bulk transactions to be received that have an Invalid CRC. When this bit is cleared (0), the SIE will not receive any bulk packet that has an invalid CRC.
2	Nack_all_RX	R/W	Nack All USB received packets 0 = Normal Operation (Default) 1 = NACK all RX packets 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.
1	nUSB_auto_Gen _Zero	R/W	Auto Generate Zero Length TX Packets – Upon the termination of a sequence of USB Bulk packets to end a logical Ethernet Packet, the SMSC USB97C196 will generate a Zero length packet when the Logical length is a multiple of 64 (USB Bulk Length). When this bit is cleared (0), zero length packets are Automatically transmitted at end of USB Bulk String (logical Ethernet packet) if length is multiple of 64. When this bit is set (1), Zero Length Packets need to be allocated manually being put into the Bulk TX FIFO **immediately** after prior Bulk packet is enqueued.
0	USB_auto_ Release	R/W	Automatic TX Memory de-allocate Mode 0 = Auto TX Memory buffer de-allocation 1 = Manual de-allocation, but the TX FIFO Pop is still automatic. This control bit selects between Auto and Manual memory pages de-allocation. This bit should be statically set at the start of operation, and can not be changed during or if about to transmit. This bit should default to “0” for normal operation. When set, the MCU handles freeing up the memory pages.

PACKET HEADER DEFINITION

The following headers contain information to determine the status and length of the USB and Ethernet received or transmit packets. These headers are used for both the USB SIE and Ethernet CSMA/CD engine respectively. The fields are different for Ethernet and USB control and there is also a differentiation when they are used for USB/Ethernet transmission and reception.

USB Receive Frame Status Double Word (in Packet RAM. Offset 0 and 5)

The USB Status double word is used to obtain additional information relating to a USB Out Bulk packet received by the USB97C196.

USB Frame Counter [15:0] – This word value determines the number of USB SOF's (Start of frames) since the device has been reset. This information may be used to determine Isochronious timing information.

PACKET ID[3..0] – This 4 bit value shows the PID value of the current USB received packet.

Bad_CRC- This bit determines if the current received USB packet has an invalid CRC. This bit is only set if the Receive bad bit is set in the USB Tx Management Register 1.

SERIAL INTERFACE ENGINE (SIE) REGISTER DESCRIPTION

Packet Header Definition

The following header contains information to determine endpoint, status, and the payload "received data". (Length of the received packet is obtained from the USB Received FIFO Packet Length Registers High & Low).

Table 73 - Packet Header Definition

OFFSET	MSB 7	6	5	4	3	2	1	LSB 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	Same_TOG	0	ENDPOINT[3..0]			

Packet Description:

Offset 0 to 5 is the USB packet header. Offset 0x000 to 0x005 is generated by the SIE.

i) Offset 0x000, bit 5 - Same_TOG - This bit is set when the SIE receives the same toggle that was received in the previous packet. This is not necessarily an error condition, This bit could indicate a condition when the return handshake packet is lost. Note that Same_TOG is not set for isochronous transfers.

Last Packet Toggle Value	Current Packet Toggle Value	"SAME 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.

Ethernet Transmit and Receive packet Status RAM definition

The table shown below defines the structure of the packet buffer when an Ethernet packet is received or transmitted.

Table 74 - Ethernet/USB Transmit and Receive packet buffer RAM definition

	Status (Buffer) RAM							
Status RAM Offset (Decimal)	MSB - Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	LSB - Bit 0
0	Ethernet Status Byte Low							
1	Ethernet Status Byte High							

Ethernet RECEIVE FRAME STATUS WORD

This word, 2 bytes, is written at the beginning of each receive frame in memory. It is not available as a register. This information normally is not required under normal I/O operations. This information is available to the 8051 Embedded processor for informational or diagnostic content.

Status Byte Low	ALGN ERR	BROD CAST	BADCRC	ODDFRM	TOOLNG	TOO SHORT		

Status Byte High	HASH VALUE						MULT CAST
	5	4	3	2	1	0	

ALGNERR - Frame had alignment error.

BRODCAST - Receive frame was broadcast.

BADCRC - Frame had CRC error.

ODDFRM - This bit when set indicates that the received frame had an odd number of bytes.

TOOLNG - The received frame is longer than the 802.3 maximum size (1518 bytes on the cable).

TOOSHORT - The received frame is shorter than the 802.3 minimum size (64 bytes on the cable).

HASH VALUE - Provides the hash value used to index the Multicast Registers. Can be used by receive routines to speed up the group address search. The hash value consists of the six most significant bits of the CRC calculated on the Destination Address, and maps into the 64 bit multicast table. Bits 5,4,3 of the hash value select a byte of the multicast table, while bits 2,1,0 determine the bit within the byte selected.

Examples of the address mapping are shown in the table below:

ADDRESS	HASH VALUE 5-0	MULTICAST TABLE BIT
ED 00 00 00 00 00	000 000	MT-0 bit 0
0D 00 00 00 00 00	010 000	MT-2 bit 0
01 00 00 00 00 00	100 111	MT-4 bit 7
2F 00 00 00 00 00	111 111	MT-7 bit 7

MULTCAST - Receive frame was multicast. If hash value corresponds to a multicast table bit that is set, and the address was a multicast, the packet will pass address filtering regardless of other filtering criteria.

Ethernet Transmit FRAME STATUS WORD (in Status RAM)

This two byte value, stores the transmit status upon individual transmit packet completion. Packet interrupt processing should use this copy for status information, if needed. The value stored in these two bytes are copies of the EPH status register. This status word is a “Snap Shot” of the CSMA/CD transmit status for the associated packet. The EPH status register will be updated by subsequent packet transmissions. The EPH Status register can be used for real time values (like TXENA and LINK OK).

Status Byte Low	TX UNRN	LINK_ OK	RX_ OVRN	CTR _ROL	EXC _DEF	LOST CARR	LATCOL	0
	0	0	0	0	0	0	0	0
Status Byte High	TX DEFR	LTX BRD	SQET	16COL	LTX MULT	MUL COL	SNGL COL	TX_SUC
	0	0	0	0	0	0	0	0

TXUNRN - Transmit Under run. Set if Under run occurs, it also clears TXENA bit in TCR. Cleared by setting TXENA high. This bit should never be set under normal operation.

LINK_OK - State of the 10BASE-T Link Integrity Test. A transition on the value of this bit generates an interrupt when the LE ENABLE bit in the Control Register is set.

RX_OVRN - Upon FIFO overrun, the receiver asserts this bit and clears the FIFO. The receiver stays enabled. After a valid preamble has been detected on a subsequent frame, RX_OVRN is de-asserted. The RX_OVRN INT bit in the Interrupt Status Register will also be set and stay set until cleared by the CPU. Note that receive overruns could occur only if receive memory allocations fail.

CTR_ROL - Counter Roll over. When set one or more 4 bit counters have reached maximum count (15). Cleared by reading the ECR register.

EXC_DEF - Excessive deferral. When set last/current transmit was deferred for more than 1518 * 2 byte times. Cleared at the end of every packet sent.

LOST_CARR - Lost carrier sense. When set indicates that Carrier Sense was not present at end of preamble. Valid only if MON_CSN is enabled. This condition causes TXENA bit in TCR to be reset. Cleared by setting TXENA bit in TCR.

LATCOL - Late collision detected on last transmit frame. If set a late collision was detected (later than 64 byte times into the frame). When detected the transmitter JAMs and turns itself off clearing the TXENA bit in TCR. Cleared by setting TXENA in TCR.

WAKEUP - When this bit is set, it indicates that a receive packet was received that had a packet signature defined in the Packet Signature Control Registers. This bit indicates a valid detection for a packet signature match. - enabled WAKEUP_EN in CTR.

TX_DEFR - Transmit Deferred. When set, carrier was detected during the first 6.4 uSec of the inter frame gap. Cleared at the end of every packet sent.

LTX_BRD - Last transmit frame was a broadcast. Set if frame was broadcast. Cleared at the start of every transmit frame.

SQET - Signal Quality Error Test. The transmitter opens a 1.6 us window 0.8 us after transmission is completed and the receiver returns inactive. During this window, the transmitter expects to see the SQET signal from the transceiver. The absence of this signal is a 'Signal Quality Error' and is reported in this status bit. Transmission stops and EPH INT is set if STP_SQET is in the TCR is also set when SQET is set. This bit is cleared by setting TXENA high.

16COL - 16 collisions reached. Set when 16 collisions are detected for a transmit frame. TXENA bit in TCR is reset. Cleared when TXENA is set high.

LTX_MULT - Last transmit frame was a multicast. Set if frame was a multicast. Cleared at the start of every transmit frame.

MULCOL - Multiple collision detected for the last transmit frame. Set when more than one collision was experienced. Cleared when TX_SUC is high at the end of the packet being sent.

SNGLCOL - Single collision detected for the last transmit frame. Set when a collision is detected. Cleared when TX_SUC is high at the end of the packet being sent.

TX_SUC - Last transmit was successful. Set if transmit completes without a fatal error. This bit is cleared by the start of a new frame transmission or when TXENA is set high.
Fatal errors are:

16 collisions
SQET fail and STP_SQET = 1
FIFO Underrun
Carrier lost and MON_CSN = 1
Late collision

SERIAL INTERFACE ENGINE (SIE) REGISTER DESCRIPTION

The architecture of the SMSC USB97C196 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 1023 bytes. Each endpoint can be configured separately via the following register(s) – there are six in the USB97C196:

Endpoint Control Registers

Table 75 - Endpoint Control Registers

EP_CTRL[5..0] (0x7F85-0x7F80 - 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

EP_CTRL[5..0] (0x7F85-0x7F80 - RESET=0x00)			ENDPOINT CONTROL REGISTERS
BIT	**NAME**	**R/W**	**DESCRIPTION**
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)
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.
0	RX_TOGGLE	R	This bit reflects the last DATA0/DATA1 toggle.

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: 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.

Note 3: 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 4: 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 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.

NonControl Endpoint Register

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 register will correspond to the associated Endpoint. Bit 0 of the NonControl Endpoint register will correspond to Endpoint 0. 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 76 - NonControl Endpoint Register

NONCTRL_EP (0x7FAC – RESET=0x00)			NONCONTROL ENDPOINT REGISTER
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	Reserved	R/W	Reserved – This bit should be written to 0
6	Reserved	R/W	Reserved – This bit should be written to 0
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.

Endpoint Command Register

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, which 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 77 - Endpoint Command Register

EP_COMM (0x7FAA- RESET=0x00)			ENDPOINT Command Register				
BIT	**NAME**	**W**	**DESCRIPTION**				
7	TX/RX	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	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/OUT 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/OUT transaction directed at this EP, regardless of TX_QUEUE status.	
			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	
[3:0]	EP_Select	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	Reserved
			0	1	1	1	Reserved

EP_COMM (0x7FAA- RESET=0x00)			ENDPOINT Command Register				
BIT	**NAME**	**W**	**DESCRIPTION**				
			1	0	0	0	Reserved
			1	0	0	1	Reserved
			1	0	1	0	Reserved
			1	0	1	1	Reserved
			1	1	0	0	Reserved
			1	1	0	1	Reserved
			1	1	1	0	Reserved
			1	1	1	1	Reserved

For Example:

1) EP_CMD = cEPCMD_RX_ | cEPCMD_EP_BUSY_ | c_EPCMD_EP3_; /* cEPCMD_EP3_ is a manifest constant */
2) EP_CMD = cEPCMD_RX_ | cEPCMD_EP_BUSY_ | epNo; /* epNo is a BYTE variable */

By the Firmware implementing code structures this way, performance is enhanced by avoiding the read / modify / write. There is no issue involving any stepping on the wrong bit field, including toggles.

LSB FRAME Count Register

Table 78 - 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'.

MSB FRAME Count Register

Table 79 - 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'.

Local Address Register

Table 80 - 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.

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.

Alternate Address Register

Table 81 - Alternate Address Register

ALT_ADDR (0x7F99 - RESET=0x00)			ALTERNATE SIE ADDRESS
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	EN_ALTADDR	R/W	Alternate address. 1 = Enabled, this bit allows Endpoints 4 through 5 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

SIE Status Register

Table 82 - SIE Status Register

SIE_STAT (0x7F93 - RESET=0xXX)			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.
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.

SIE Control Register 1

Table 83 - SIE Control Register 1

SIE_CTRL1 (0x7F94 – RESET=0x00)			SIE CONTROL REGISTER
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: These bits must be set for normal operation. Altering these bits will cause an abnormal USB behavior.

SIE Control Register 2

Table 84 - 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	Reserved	R/W	Reserved – This bit should always be cleared (0)
0	Reserved	R/W	Reserved – This bit should always be cleared (0)

SIE Configuration Register 1

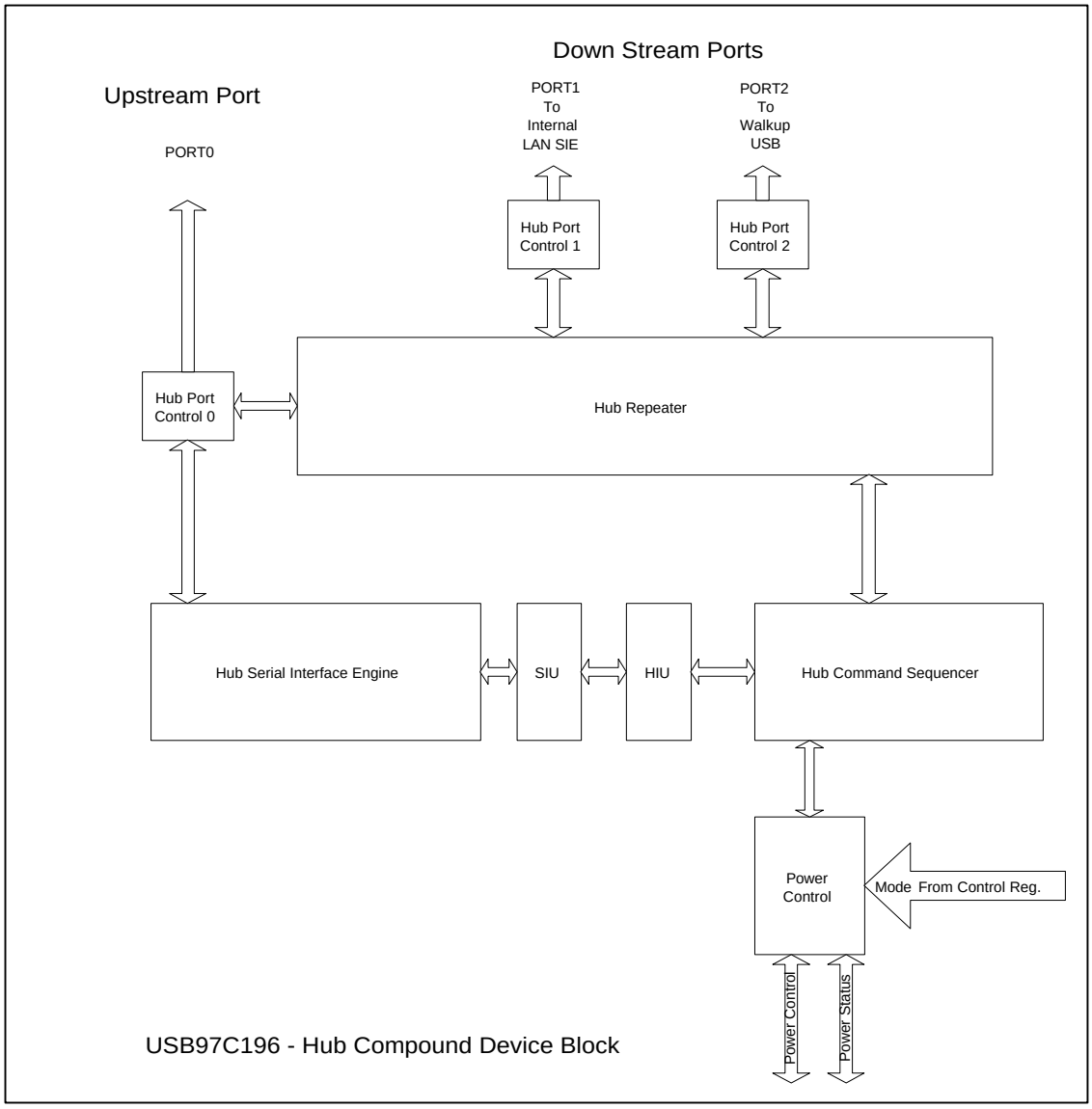
Table 85 - SIE Configuration Register 1.

SIE_CONFIG 1 (0x7F98 - RESET=0x40)			SIE CONFIGURATION REGISTER 1
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	FSEN	R/W	Controls SIE select and SIE Transceiver Slew Rate. 0 = 1.5Mbps 1 = 12Mbps
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 12Mbps bit times. This causes a USB_RESET Power management interrupt.

USB HUB BLOCK

The registers shown below interface the Internal 8051 MCU with the SMSC USB97C196 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.



USB97C196 - Hub Compound Device Block

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. Note: Low Speed down stream port support requires external transceivers
- Reflecting Remote Resume to Upstream and enabled down stream USB ports

HUB Block Register Summary

The Register definitions defined below are defined in Table 86 and. These registers are memory mapped into the 8051 MCU memory space defined in Table 4 - MCU Data Memory Map (page 7).

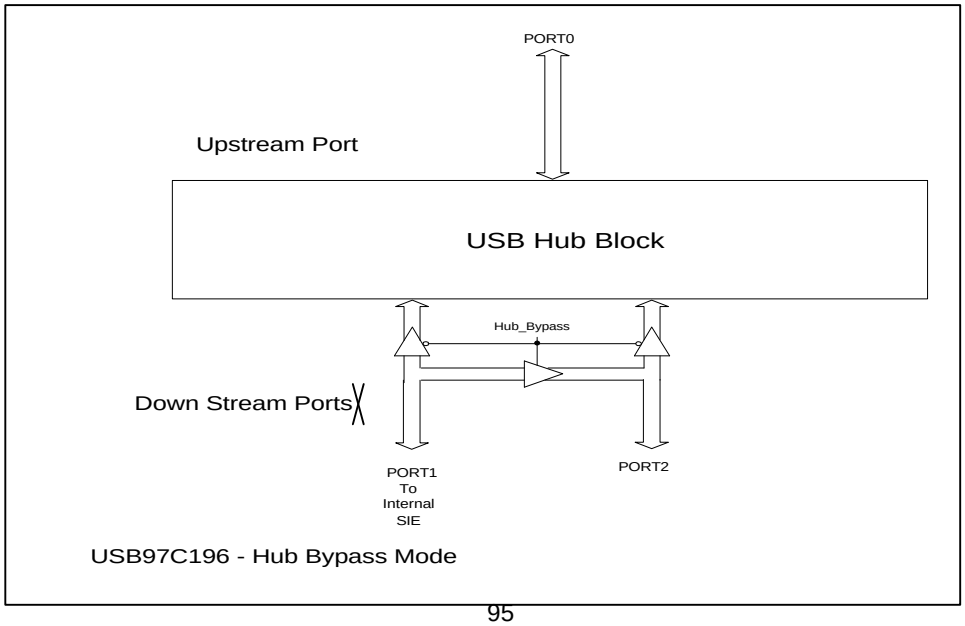
Table 86 - HUB Block Register Summary

ADDRESS	NAME	RESET VALUE	R/W	DESCRIPTION
7FA0	IdVendor-Low Byte	00	R/W	Low byte Vendor ID in little endian format (Bit 0 is the LSB)
7FA1	IdVendor-High Byte	00	R/W	High byte Vendor ID in little endian format (Bit 0 is the LSB)
7FA2	IdProduct-Low Byte	00	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	00	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	00	R/W	This 8-bit value defines the USB device release number, which is assigned by the system manufacture.
7FA5	BcdDevice – High Byte	00	R/W	This 8-bit value defines the USB device release number, which is assigned by the system manufacture.
7FA6	HubControl1	00	R/W	Hub Control Register1
7FA7	HubControl2	00	R/W	Hub Control Register 2

Hub Control Register 1

Table 87 – Hub Control Register 1

HubControl (0x7FA6- RESET=0x00)			HUB CONTROL REGISTER 1
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	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	Hub_Bypass	R/W	When this bit is set, Port 2 is no longer connected to the hub. The SIE is directly connected to Port 2, which becomes the upstream port. See Figure.
1	ForceSE0	R/W	Force Single Ended zero (SE0). – This bit will force a SE0 condition on the upstream port (as selected by the Hub_Bypass bit). 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 > 2.5us).
0	Reserved	R/W	Reserved – This bit should always be cleared (0)



USB97C196 - Hub Bypass Mode

Hub Control Register 2

Table 88 – Hub Control Register 2

HubControl (0x7FA7- RESET=0x00)			HUB CONTROL REGISTER 2
BIT	**NAME**	**R/W**	**DESCRIPTION**
7	SRTSTCLKEN	R/W	SRTSTCLKEN – used to by pass the divide-by-4 (USB 12MHz) clock to the suspend/resume logic (used for testing)
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	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 (0), the internal transceiver is used.
1	Reserved	R/W	Reserved – This bit should always be cleared (0)
0	Reserved	R/W	Reserved – This bit should always be cleared (0)

DC PARAMETERS

MAXIMUM GUARANTEED RATINGS

Operating Temperature Range..... 0^oC to $+70^oC$
Storage Temperature Range.....-55^o to $+150^oC$
Lead Temperature Range (soldering, 10 seconds) ..... $+325^oC$
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}
O8 Type Buffer						
Low Output Level	V_{OL}	2.4		0.4	V	I_{OL} = 8 mA @ V_{CC}
High Output Level	V_{OH}				V	I_{OH} = -4 mA @ V_{CC}
Output Leakage	I_{OL}	-10		+10	uA	V_{IN} = 0 to V_{CC} (Note 1)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	COMMENTS
I/O8 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA @ V_{CC}
High Output Level	V_{OH}	2.4			V	I_{OH} = -4 mA @ V_{CC}
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}
High Output Level	V_{OH}	2.4			V	I_{OH} = -8 mA @ V_{CC}
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to V_{CC} (Note 1)
I/O24 Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 24 mA @ V_{CC}
High Output Level	V_{OH}	2.4			V	I_{OH} = -12 mA @ V_{CC}
Output Leakage	I_{OL}	-10		+10	µA	V_{IN} = 0 to V_{CC} (Note 1)
IO-U Note 2						
Supply Current Active	I_{CC}		TBD	TBD	mA	
Supply Current Suspend	I_{SB}		TBD	TBD	uA	
Supply Current Standby	I_{CSBU}		TBD	TBD	uA	All outputs open.

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	

INPUT CLOCK TIMING

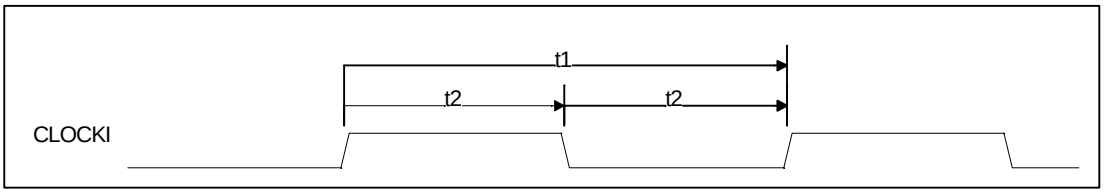


FIGURE 10 – INPUT USB CLOCK TIMING

TABLE 89 - 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	25/16.7		16.7/25	ns
t_r, t_f	Clock Rise Time/Fall Time (not shown)			5	ns

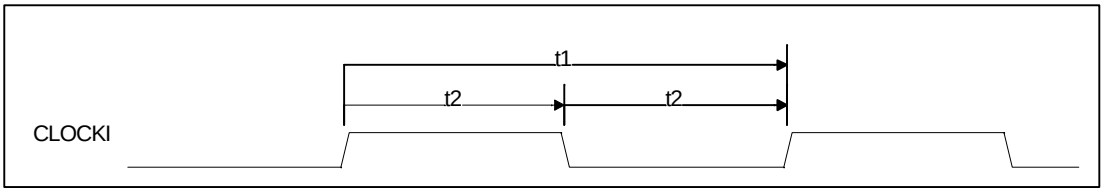


FIGURE 11 – INPUT ETHERNET CLOCK TIMING

TABLE 90 - INPUT CLOCK TIMING PARAMETERS

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t1	CLOCK CYCLE TIME FOR 20 MHZ		50		ns
t2	Clock High Time/Low Time	20/30		30/20	ns
t_r, t_f	Clock Rise Time/Fall Time (not shown)			5	ns

INPUT CLOCK TIMING

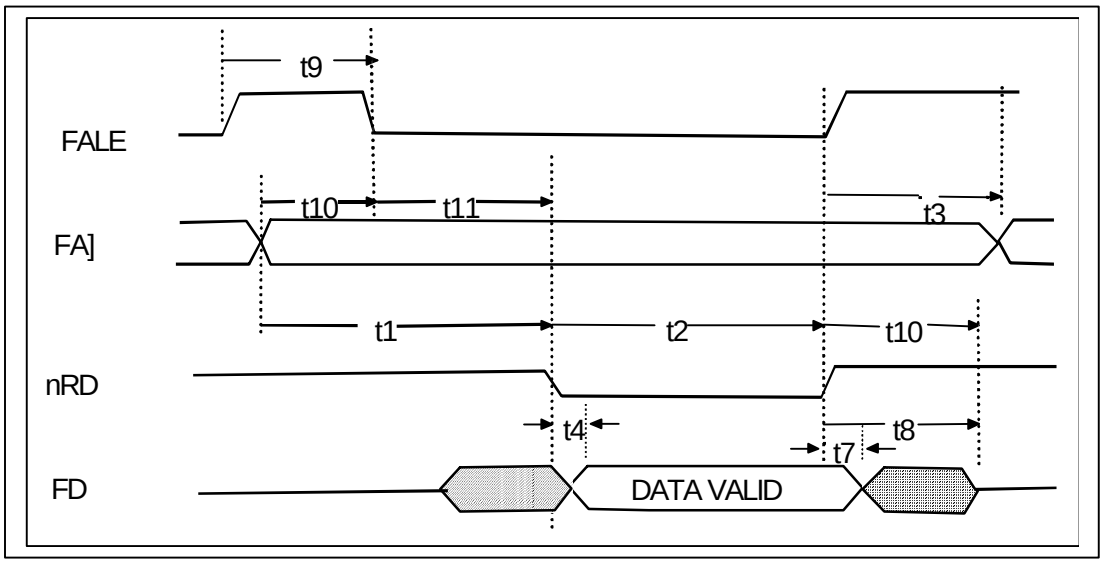


FIGURE 12 - 8051 FLASH PROGRAM FETCH TIMING

Table 91 - 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.

8051 FLASH MEMORY READ TIMING

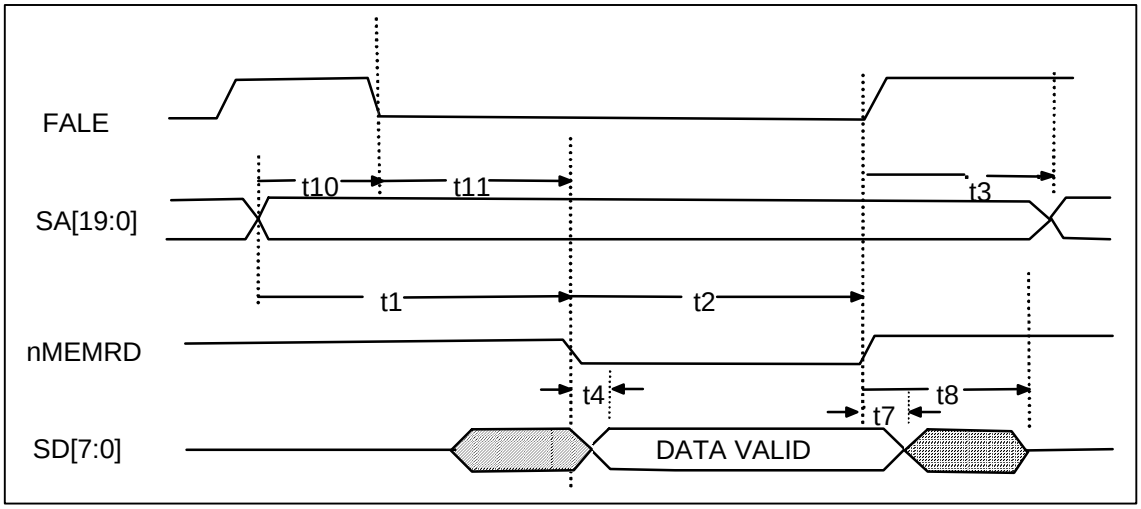


FIGURE 13 - 8051 FLASH MEMORY READ TIMING

Table 92 - 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.

8051 FLASH MEMORY WRITE TIMING

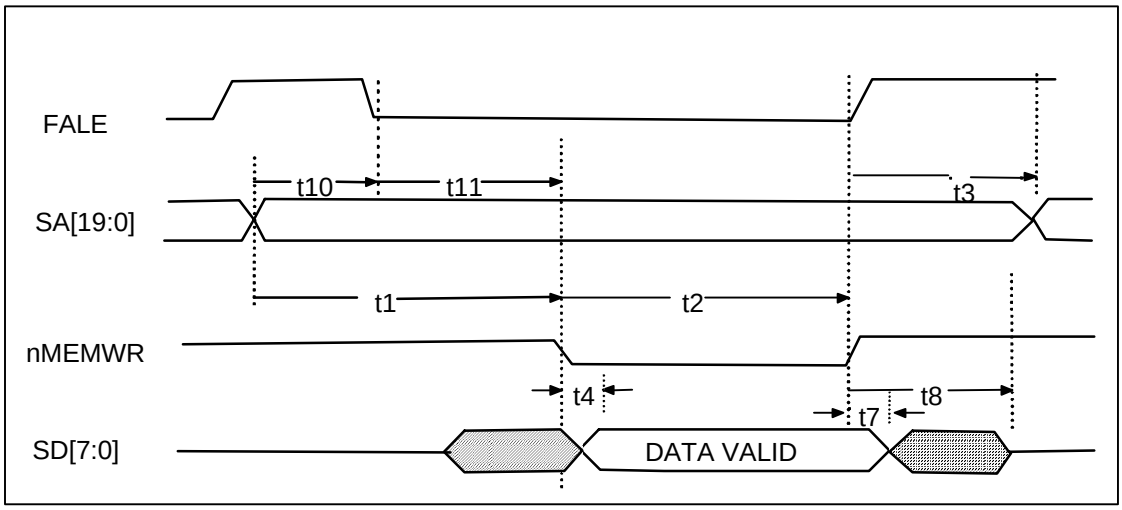


FIGURE 14 - 8051 FLASH MEMORY WRITE TIMING

Table 93 - 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.

RESET_IN TIMING

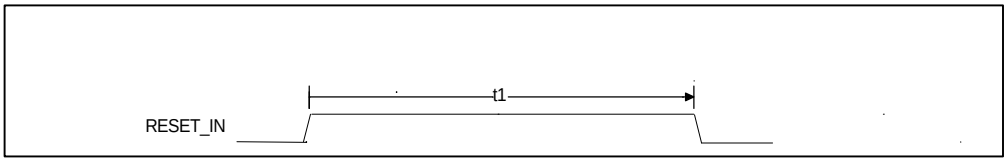
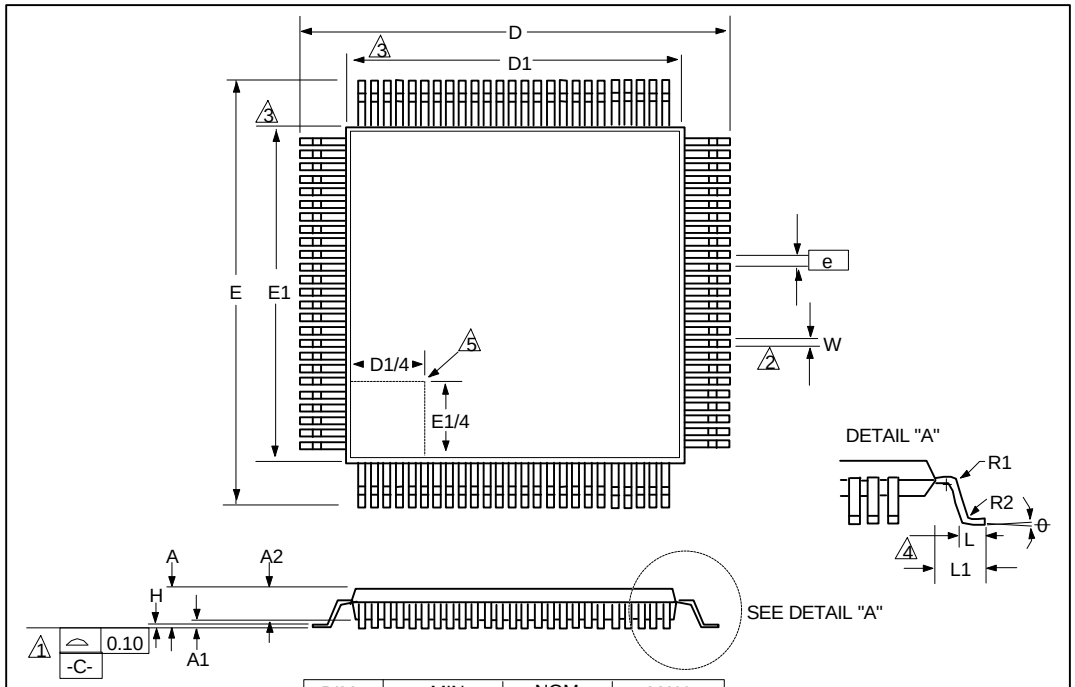


FIGURE 15 - RESET_IN TIMING

Table 94 - RESET_IN TIMING PARAMETERS

	PARAMETER	MIN	TYP	MAX	UNITS
t1	RESET_IN active pulse width	500			ns



DIM	MIN	NOM	MAX
A			1.20
A1	0.05	0.10	.15
A2	0.95	1.00	1.05
D	15.90	16.00	16.10
D1	13.90	14.00	14.10
E	15.90	16.00	16.10
E1	13.90	14.00	14.10
H	0.09		0.20
L	0.45	0.60	0.75
L1		1.00	
e		0.50 BSC	
θ	0°		7°
W	0.17	0.22	0.27
ccc			0.08

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

FIGURE 16 – 100 PIN TQFP 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.

USB97C196 Rev. 5/27/99