Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

Details	
Product Status	Obsolete
Applications	USB Microcontroller
Core Processor	M8C
Program Memory Type	OTP (8kB)
Controller Series	CY7C641xx
RAM Size	256 x 8
Interface	I²C, USB, HAPI
Number of I/O	36
Voltage - Supply	4V ~ 5.25V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	48-BSSOP (0.295", 7.50mm Width)
Supplier Device Package	48-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy7c64113c-pvxc

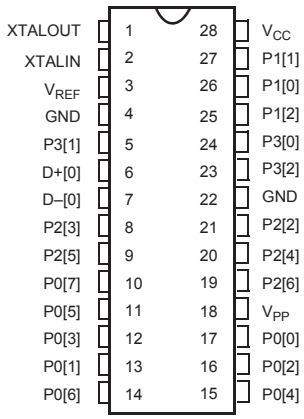
TABLE OF CONTENTS

TABLE OF CONTENTS

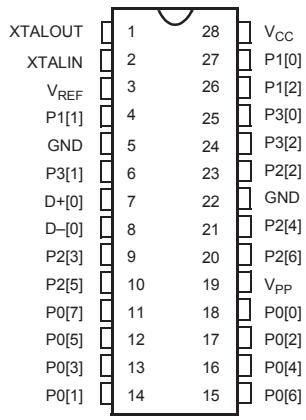
3.0 Pin Configurations

TOP VIEW

CY7C64013C
28-pin SOIC



CY7C64013C
28-pin PDIP



CY7C64113C
48-pin SSOP

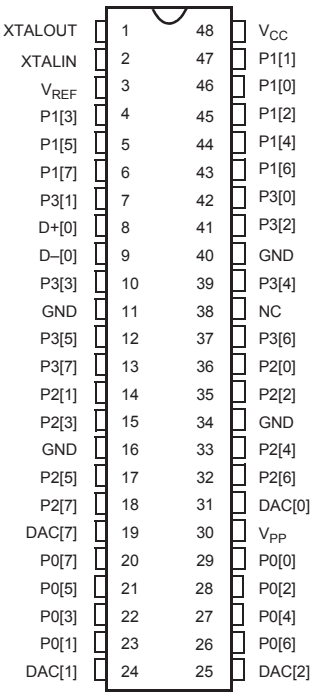


Table 4-2. I/O Register Summary (continued)

Register Name	I/O Address	Read/Write	Function	Page
GPIO Configuration	0x08	R/W	GPIO Port Configurations	20
HAPI and I^2C Configuration	0x09	R/W	HAPI Width and I^2C Position Configuration	24
USB Device Address A	0x10	R/W	USB Device Address A	34
EP A0 Counter Register	0x11	R/W	USB Address A, Endpoint 0 Counter	35
EP A0 Mode Register	0x12	R/W	USB Address A, Endpoint 0 Configuration	34
EP A1 Counter Register	0x13	R/W	USB Address A, Endpoint 1 Counter	35
EP A1 Mode Register	0x14	R/W	USB Address A, Endpoint 1 Configuration	35
EP A2 Counter Register	0x15	R/W	USB Address A, Endpoint 2 Counter	35
EP A2 Mode Register	0x16	R/W	USB Address A, Endpoint 2 Configuration	35
USB Status & Control	0x1F	R/W	USB Upstream Port Traffic Status and Control	34
Global Interrupt Enable	0x20	R/W	Global Interrupt Enable	29
Endpoint Interrupt Enable	0x21	R/W	USB Endpoint Interrupt Enables	29
Timer (LSB)	0x24	R	Lower 8 Bits of Free-running Timer (1 MHz)	23
Timer (MSB)	0x25	R	Upper 4 Bits of Free-running Timer	24
WDT Clear	0x26	W	Watchdog Timer Clear	18
I^2C Control & Status	0x28	R/W	I^2C Status and Control	25
I^2C Data	0x29	R/W	I^2C Data	25
DAC Data	0x30	R/W	DAC Data	22
DAC Interrupt Enable	0x31	W	Interrupt Enable for each DAC Pin	23
DAC Interrupt Polarity	0x32	W	Interrupt Polarity for each DAC Pin	23
DAC Isink	0x38-0x3F	W	Input Sink Current Control for each DAC Pin	22
Reserved	0x40		Reserved	
EP A3 Counter Register	0x41	R/W	USB Address A, Endpoint 3 Counter	35
EP A3 Mode Register	0x42	R/W	USB Address A, Endpoint 3 Configuration	34
EP A4 Counter Register	0x43	R/W	USB Address A, Endpoint 4 Counter	35
EP A4 Mode Register	0x44	R/W	USB Address A, Endpoint 4 Configuration	35
Reserved	0x48		Reserved	
Reserved	0x49		Reserved	
Reserved	0x4A		Reserved	
Reserved	0x4B		Reserved	
Reserved	0x4C		Reserved	
Reserved	0x4D		Reserved	
Reserved	0x4E		Reserved	
Reserved	0x4F		Reserved	
Reserved	0x50		Reserved	
Reserved	0x51		Reserved	
Processor Status & Control	0xFF	R/W	Microprocessor Status and Control Register	26

5.0 Programming Model

5.1 14-Bit Program Counter (PC)

The 14-bit program counter (PC) allows access to up to 8 KB of PROM available with the CY7C64x13C architecture. The top 32 bytes of the ROM in the 8 Kb part are reserved for testing purposes. The program counter is cleared during reset, such that the first instruction executed after a reset is at address 0x0000h. Typically, this is a jump instruction to a reset handler that initializes the application (see Interrupt Vectors on page 30).

The lower eight bits of the program counter are incremented as instructions are loaded and executed. The upper six bits of the program counter are incremented by executing an XPAGE instruction. As a result, the last instruction executed within a 256-byte “page” of sequential code should be an XPAGE instruction. The assembler directive “XPAGEON” causes the assembler to insert XPAGE instructions automatically. Because instructions can be either one or two bytes long, the assembler may occasionally need to insert a NOP followed by an XPAGE to execute correctly.

The address of the next instruction to be executed, the carry flag, and the zero flag are saved as two bytes on the program stack during an interrupt acknowledge or a CALL instruction. The program counter, carry flag, and zero flag are restored from the program stack during a RETI instruction. Only the program counter is restored during a RET instruction.

The program counter cannot be accessed directly by the firmware. The program stack can be examined by reading SRAM from location 0x00 and up.

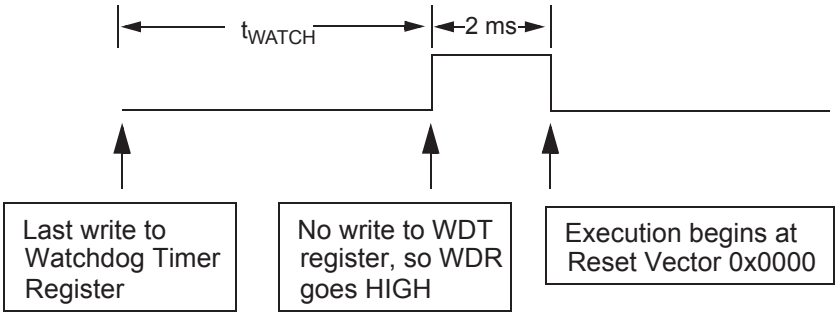


Figure 7-1. Watchdog Reset (WDR)

The USB transmitter is disabled by a Watchdog Reset because the USB Device Address Register is cleared (see Section 18.1). Otherwise, the USB Controller would respond to all address 0 transactions.

It is possible for the WDR bit of the Processor Status and Control Register (0xFF) to be set following a POR event. The WDR bit should be ignored If the firmware interrogates the Processor Status and Control Register for a Set condition on the WDR bit and if the POR (bit 3 of register 0xFF) bit is set.

8.0 Suspend Mode

The CY7C64x13C can be placed into a low-power state by setting the Suspend bit of the Processor Status and Control register. All logic blocks in the device are turned off except the GPIO interrupt logic and the USB receiver. The clock oscillator and PLL, as well as the free-running and Watchdog timers, are shut down. Only the occurrence of an enabled GPIO interrupt or non-idle bus activity at a USB upstream or downstream port wakes the part out of suspend. The Run bit in the Processor Status and Control Register must be set to resume a part out of suspend.

The clock oscillator restarts immediately after exiting suspend mode. The microcontroller returns to a fully functional state 1 ms after the oscillator is stable. The microcontroller executes the instruction following the I/O write that placed the device into suspend mode before servicing any interrupt requests.

The GPIO interrupt allows the controller to wake-up periodically and poll system components while maintaining a very low average power consumption. To achieve the lowest possible current during suspend mode, all I/O should be held at V_{CC} or Gnd. This also applies to internal port pins that may not be bonded in a particular package.

Typical code for entering suspend is shown below:

```
...          ; All GPIO set to low-power state (no floating pins)
...          ; Enable GPIO interrupts if desired for wake-up
mov a, 09h   ; Set suspend and run bits
iowr FFh     ; Write to Status and Control Register - Enter suspend, wait for USB activity (or GPIO Interrupt)
nop          ; This executes before any ISR
...          ; Remaining code for exiting suspend routine
```

9.0 General-Purpose I/O (GPIO) Ports

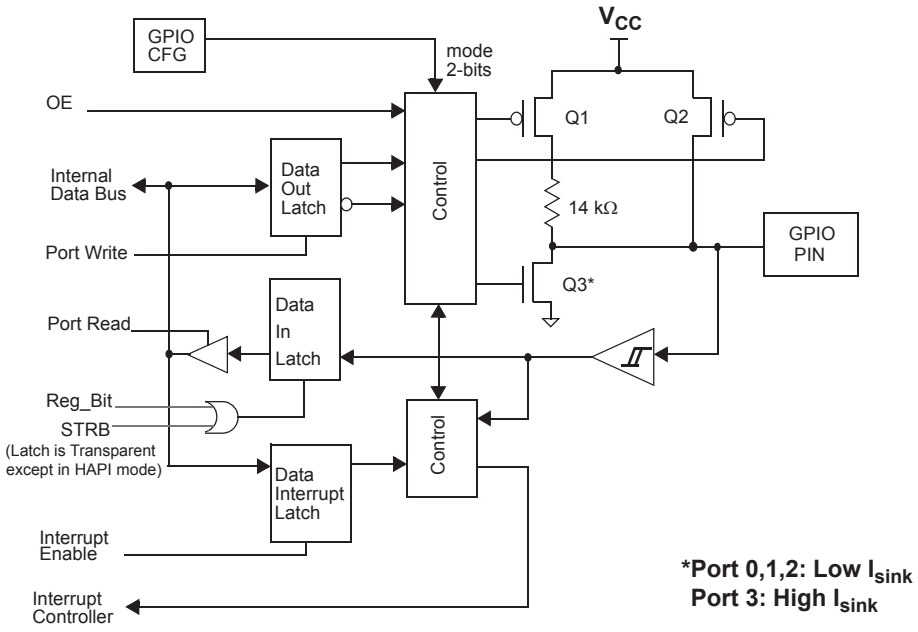


Figure 9-1. Block Diagram of a GPIO Pin

There are up to 32 GPIO pins (P0[7:0], P1[7:0], P2[7:0], and P3[7:0]) for the hardware interface. The number of GPIO pins changes based on the package type of the chip. Each port can be configured as inputs with internal pull-ups, open drain outputs, or traditional CMOS outputs. Port 3 offers a higher current drive, with typical current sink capability of 12 mA. The data for each GPIO port is accessible through the data registers. Port data registers are shown in *Figure 9-2* through *Figure 9-5*, and are set to 1 on reset.

Port 0 Data **ADDRESS 0x00**

Bit #	7	6	5	4	3	2	1	0
Bit Name	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Figure 9-2. Port 0 Data

Port 1 Data **ADDRESS 0x01**

Bit #	7	6	5	4	3	2	1	0
Bit Name	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Figure 9-3. Port 1 Data

Port 2 Data **ADDRESS 0x02**

Bit #	7	6	5	4	3	2	1	0
Bit Name	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Figure 9-4. Port 2 Data

Timer MSB **ADDRESS 0x25**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved	Reserved	Reserved	Reserved	Timer Bit 11	Timer Bit 10	Timer Bit 9	Timer Bit 8
Read/Write	-	-	-	-	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Figure 11-2. Timer MSB Register

Bit [3:0]: Timer higher nibble

Bit [7:4]: Reserved

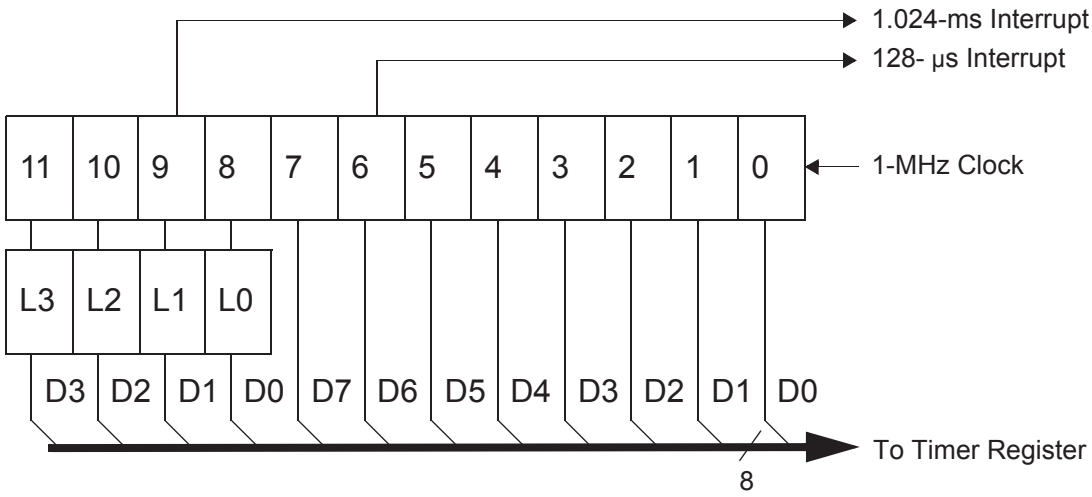


Figure 11-3. Timer Block Diagram

12.0 I²C and HAPI Configuration Register

Internal hardware supports communication with external devices through two interfaces: a two-wire I²C-compatible interface, and a HAPI for 1, 2, or 3 byte transfers. The I²C-compatible interface and HAPI functions, discussed in detail in Sections 13.0 and 14.0, share a common configuration register (see *Figure 12-1*). All bits of this register are cleared on reset.

I²C Configuration **ADDRESS 0x09**

Bit #	7	6	5	4	3	2	1	0
Bit Name	I²C Position	Reserved	LEMPTY Polarity	DRDY Polarity	Latch Empty	Data Ready	HAPI Port Width Bit 1	HAPI Port Width Bit 0
Read/Write	R/W	-	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 12-1. HAPI/I²C Configuration Register

Note: *I²C-compatible function must be separately enabled as described in Section 13.0.*

Bits [7,1:0] of the HAPI/I²C Configuration Register control the pin out configuration of the HAPI and I²C-compatible interfaces. Bits [5:2] are used in HAPI mode only, and are described in Section 14.0. *Table 12-1* shows the HAPI port configurations, and *Table 12-2* shows I²C pin location configuration options. These I²C-compatible options exist due to pin limitations in certain packages, and to allow simultaneous HAPI and I²C-compatible operation.

HAPI operation is enabled whenever either HAPI Port Width Bit (Bit 1 or 0) is non-zero. This affects GPIO operation as described in Section 14.0. I²C-compatible blocks must be separately enabled as described in Section 13.0.

Table 16-1. Interrupt Vector Assignments

Interrupt Vector Number	ROM Address	Function
Not Applicable	0x0000	Execution after Reset begins here
1	0x0002	USB Bus Reset interrupt
2	0x0004	128- µs timer interrupt
3	0x0006	1.024-ms timer interrupt
4	0x0008	USB Address A Endpoint 0 interrupt
5	0x000A	USB Address A Endpoint 1 interrupt
6	0x000C	USB Address A Endpoint 2 interrupt
7	0x000E	USB Address A Endpoint 3 interrupt
8	0x0010	USB Address A Endpoint 4 interrupt
9	0x0012	Reserved
10	0x0014	DAC interrupt
11	0x0016	GPIO interrupt
12	0x0018	I^2C interrupt

16.2 Interrupt Latency

Interrupt latency can be calculated from the following equation:

Interrupt latency = (Number of clock cycles remaining in the current instruction) + (10 clock cycles for the CALL instruction) + (5 clock cycles for the JMP instruction)

For example, if a 5 clock cycle instruction such as JC is being executed when an interrupt occurs, the first instruction of the Interrupt Service Routine executes a minimum of 16 clocks (1+10+5) or a maximum of 20 clocks (5+10+5) after the interrupt is issued. For a 12-MHz internal clock (6-MHz crystal), 20 clock periods is 20 / 12 MHz = 1.667 µs.

16.3 USB Bus Reset Interrupt

The USB Controller recognizes a USB Reset when a Single Ended Zero (SE0) condition persists on the upstream USB port for 12–16 µs (the Reset may be recognized for an SE0 as short as 12 µs, but is always recognized for an SE0 longer than 16 µs). SE0 is defined as the condition in which both the D+ line and the D– line are LOW. Bit 5 of the Status and Control Register is set to record this event. The interrupt is asserted at the end of the Bus Reset. If the USB reset occurs during the start-up delay following a POR, the delay is aborted as described in Section 7.1. The USB Bus Reset Interrupt is generated when the SE0 state is deasserted.

A USB Bus Reset clears the following registers:

SIE Section:USB Device Address Registers (0x10, 0x40)

16.4 Timer Interrupt

There are two periodic timer interrupts: the 128- µs interrupt and the 1.024-ms interrupt. The user should disable both timer interrupts before going into the suspend mode to avoid possible conflicts between servicing the timer interrupts first or the suspend request first.

16.5 USB Endpoint Interrupts

There are five USB endpoint interrupts, one per endpoint. A USB endpoint interrupt is generated after the USB host writes to a USB endpoint FIFO or after the USB controller sends a packet to the USB host. The interrupt is generated on the last packet of the transaction (e.g., on the host's ACK during an IN, or on the device ACK during on OUT). If no ACK is received during an IN transaction, no interrupt is generated.

16.6 DAC Interrupt

Each DAC I/O pin can generate an interrupt, if enabled. The interrupt polarity for each DAC I/O pin is programmable. A positive polarity is a rising edge input while a negative polarity is a falling edge input. All of the DAC pins share a single interrupt vector, which means the firmware needs to read the DAC port to determine which pin or pins caused an interrupt.

If one DAC pin has triggered an interrupt, no other DAC pins can cause a DAC interrupt until that pin has returned to its inactive (non-trigger) state or the corresponding interrupt enable bit is cleared. The USB Controller does not assign interrupt priority to different DAC pins and the DAC Interrupt Enable Register is not cleared during the interrupt acknowledge process.

Bits[6..0] :Device Address

Firmware writes this bits during the USB enumeration process to the non-zero address assigned by the USB host.

Bit 7 :Device Address Enable

Must be set by firmware before the SIE can respond to USB traffic to the Device Address.

Bit 7 (Device Address Enable) in the USB Device Address Register must be set by firmware before the SIE can respond to USB traffic to this address. The Device Addresses in bits [6:0] are set by firmware during the USB enumeration process to the non-zero address assigned by the USB host.

18.2 USB Device Endpoints

The CY7C64x13C controller supports one USB device address and five endpoints for communication with the host. The configuration of these endpoints, and associated FIFOs, is controlled by bits [7,6] of the USB Status and Control Register (0x1F). Bit 7 controls the size of the endpoints and bit 6 controls the number of endpoints. These configuration options are detailed in *Table 18-1*. The “unused” FIFO areas in the following table can be used by the firmware as additional user RAM space.

Table 18-1. Memory Allocation for Endpoints

USB Status And Control Register (0x1F) Bits [7, 6]											
[0,0]			[1,0]			[0,1]			[1,1]		
Label	Start Address	Size	Label	Start Address	Size	Label	Start Address	Size	Label	Start Address	Size
unused	0xD8	8	unused	0xA8	8	EPA4	0xD8	8	EPA4	0xB0	8
unused	0xE0	8	unused	0xB0	8	EPA3	0xE0	8	EPA3	0xA8	8
EPA2	0xE8	8	EPA0	0xB8	8	EPA2	0xE8	8	EPA0	0xB8	8
EPA1	0xF0	8	EPA1	0xC0	32	EPA1	0xF0	8	EPA1	0xC0	32
EPA0	0xF8	8	EPA2	0xE0	32	EPA0	0xF8	8	EPA2	0xE0	32

When the SIE writes data to a FIFO, the internal data bus is driven by the SIE; not the CPU. This causes a short delay in the CPU operation. The delay is three clock cycles per byte. For example, an 8-byte data write by the SIE to the FIFO generates a delay of 2 µs (3 cycles/byte * 83.33 ns/cycle * 8 bytes).

18.3 USB Control Endpoint Mode Register

All USB devices are required to have a Control Endpoint 0 (EPA0) that is used to initialize and control each USB address. Endpoint 0 provides access to the device configuration information and allows generic USB status and control accesses. Endpoint 0 is bidirectional to both receive and transmit data. The other endpoints are unidirectional, but selectable by the user as IN or OUT endpoints.

The endpoint mode register is cleared during reset. The endpoint zero EPA0 mode register uses the format shown in *Figure 18-2*.

USB Device Endpoint Zero Mode **ADDRESSES 0x12)**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Endpoint 0 SETUP Received	Endpoint 0 IN Received	Endpoint 0 OUT Received	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 18-2. USB Device Endpoint Zero Mode Registers

Bits[3..0] : Mode

These sets the mode which control how the control endpoint responds to traffic.

Bit 4 : ACK

This bit is set whenever the SIE engages in a transaction to the register’s endpoint that completes with an ACK packet.

Bit 5: Endpoint 0 OUT Received

1= Token received is an OUT token. 0= Token received is not an OUT token. This bit is set by the SIE to report the type of token received by the corresponding device address is an OUT token. The bit must be cleared by firmware as part of the USB processing.

Bit 6: Endpoint 0 IN Received

1= Token received is an IN token. 0= Token received is not an IN token. This bit is set by the SIE to report the type of token received by the corresponding device address is an IN token. The bit must be cleared by firmware as part of the USB processing.

Bit 7: Endpoint 0 SETUP Received

1= Token received is a SETUP token. 0= Token received is not a SETUP token. This bit is set ONLY by the SIE to report the type of token received by the corresponding device address is a SETUP token. Any write to this bit by the CPU will clear it (set it to 0). The bit is forced HIGH from the start of the data packet phase of the SETUP transaction until the start of the ACK packet returned by the SIE. The CPU should not clear this bit during this interval, and subsequently, until the CPU first does an IORD to this endpoint 0 mode register. The bit must be cleared by firmware as part of the USB processing.

Bits[6:0] of the endpoint 0 mode register are locked from CPU write operations whenever the SIE has updated one of these bits, which the SIE does only at the end of the token phase of a transaction (SETUP... Data... ACK, OUT... Data... ACK, or IN... Data... ACK). The CPU can unlock these bits by doing a subsequent read of this register. Only endpoint 0 mode registers are locked when updated. The locking mechanism does not apply to the mode registers of other endpoints.

Because of these hardware locking features, firmware must perform an IORD after an IOWR to an endpoint 0 register. This verifies that the contents have changed as desired, and that the SIE has not updated these values.

While the SETUP bit is set, the CPU cannot write to the endpoint zero FIFOs. This prevents firmware from overwriting an incoming SETUP transaction before firmware has a chance to read the SETUP data. Refer to *Table 18-1* for the appropriate endpoint zero memory locations.

The Mode bits (bits [3:0]) control how the endpoint responds to USB bus traffic. The mode bit encoding is shown in*Table 19-1*. Additional information on the mode bits can be found in*Table 19-2*.

18.4 USB Non-Control Endpoint Mode Registers

The format of the non-control endpoint mode register is shown in *Figure 18-3*.

USB Non-Control Device Endpoint Mode **ADDRESSES 0x14, 0x16, 0x42, 0x44**

Bit #	7	6	5	4	3	2	1	0
Bit Name	STALL	Reserved	Reserved	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 18-3. USB Non-Control Device Endpoint Mode Registers

Bits[3..0] : Mode

These sets the mode which control how the control endpoint responds to traffic. The mode bit encoding is shown in *Table 19-1*

Bit 4 : ACK

This bit is set whenever the SIE engages in a transaction to the register's endpoint that completes with an ACK packet.

Bits[6..5] : Reserved

Must be written zero during register writes.

Bit 7 : STALL

If this STALL is set, the SIE stalls an OUT packet if the mode bits are set to ACK-IN, and the SIE stalls an IN packet if the mode bits are set to ACK-OUT. For all other modes, the STALL bit must be a LOW.

18.5 USB Endpoint Counter Registers

There are five Endpoint Counter registers, with identical formats for both control and non-control endpoints. These registers contain byte count information for USB transactions, as well as bits for data packet status. The format of these registers is shown in *Figure 18-4*:

USB Endpoint Counter **ADDRESSES 0x11, 0x13, 0x15, 0x41, 0x43**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 18-4. USB Endpoint Counter Registers

Bits[5..0] : Byte Count

These counter bits indicate the number of data bytes in a transaction. For IN transactions, firmware loads the count with the number of bytes to be transmitted to the host from the endpoint FIFO. Valid values are 0 to 32, inclusive. For OUT or SETUP transactions, the count is updated by hardware to the number of data bytes received, plus 2 for the CRC bytes. Valid values are 2 to 34, inclusive.

Bit 6 : Data Valid

This bit is set on receiving a proper CRC when the endpoint FIFO buffer is loaded with data during transactions. This bit is used OUT and SETUP tokens only. If the CRC is not correct, the endpoint interrupt occurs, but Data Valid is cleared to a zero.

Bit 7 : Data 0/1 Toggle

This bit selects the DATA packet's toggle state: 0 for DATA0, 1 for DATA1. For IN transactions, firmware must set this bit to the desired state. For OUT or SETUP transactions, the hardware sets this bit to the state of the received Data Toggle bit.

Whenever the count updates from a SETUP or OUT transaction on endpoint 0, the counter register locks and cannot be written by the CPU. Reading the register unlocks it. This prevents firmware from overwriting a status update on incoming SETUP or OUT transactions before firmware has a chance to read the data. Only endpoint 0 counter register is locked when updated. The locking mechanism does not apply to the count registers of other endpoints.

18.6 Endpoint Mode/Count Registers Update and Locking Mechanism

The contents of the endpoint mode and counter registers are updated, based on the packet flow diagram in *Figure 18-5*. Two time points, UPDATE and SETUP, are shown in the same figure. The following activities occur at each time point:

SETUP:

The SETUP bit of the endpoint 0 mode register is forced HIGH at this time. This bit is forced HIGH by the SIE until the end of the data phase of a control write transfer. The SETUP bit can not be cleared by firmware during this time.

The affected mode and counter registers of endpoint 0 are locked from any CPU writes once they are updated. These registers can be unlocked by a CPU read, only if the read operation occurs after the UPDATE. The firmware needs to perform a register read as a part of the endpoint ISR processing to unlock the effected registers. The locking mechanism on mode and counter registers ensures that the firmware recognizes the changes that the SIE might have made since the previous IO read of that register.

UPDATE:

1. Endpoint Mode Register – All the bits are updated (except the SETUP bit of the endpoint 0 mode register).
2. Counter Registers – All bits are updated.
3. Interrupt – If an interrupt is to be generated as a result of the transaction, the interrupt flag for the corresponding endpoint is set at this time. For details on what conditions are required to generate an endpoint interrupt, refer to *Table 19-2*.
4. The contents of the updated endpoint 0 mode and counter registers are locked, except the SETUP bit of the endpoint 0 mode register which was locked earlier.

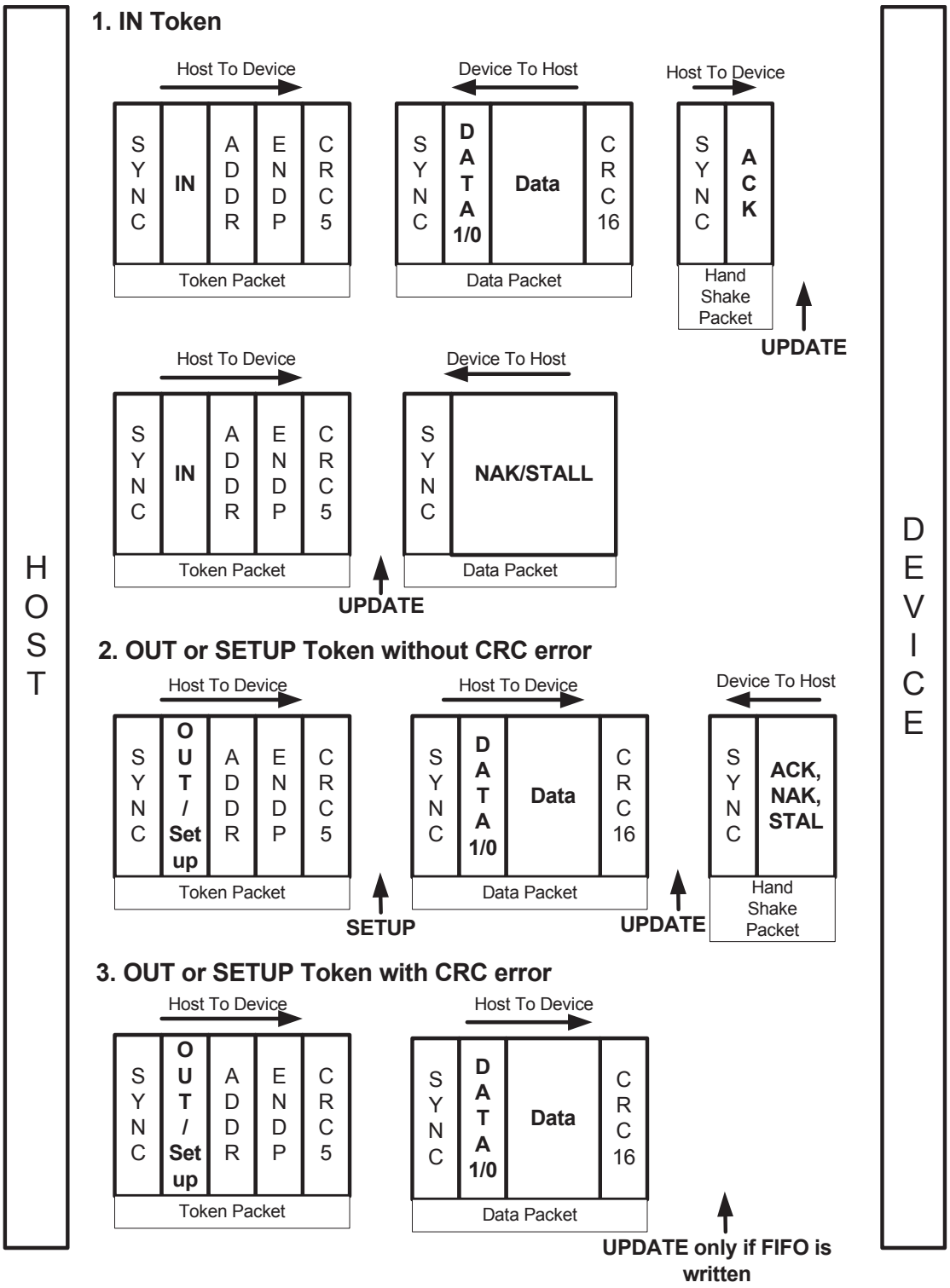


Figure 18-5. Token/Data Packet Flow Diagram

the firmware recognizes the changes that the SIE might have made during the previous transaction. Note that the setup bit of the mode register is NOT locked. This means that before writing to the mode register, firmware must first read the register to make sure that the setup bit is not set (which indicates a setup was received, while processing the current USB request). This read will of course unlock the register. So care must be taken not to overwrite the register elsewhere.

Table 19-2. Details of Modes for Differing Traffic Conditions (see *Table 19-1* for the decode legend)

SETUP (if accepting SETUPs)																			
Properties of Incoming Packet								**Changes made by SIE to Internal Registers and Mode Bits**											
Mode Bits				**token**	**count**	**buffer**	**dval**	**DTOG**	**DVAL**	**COUNT**	**Setup**	**In**	**Out**	**ACK**	**Mode Bits**				**Response**	**Intr**
See *Table 19-1*				Setup	<= 10	data	valid	updates	1	updates	1	UC	UC	1	0	0	0	1	ACK	yes
See *Table 19-1*				Setup	> 10	junk	x	updates	updates	updates	1	UC	UC	UC	NoChange				ignore	yes
See *Table 19-1*				Setup	x	junk	invalid	updates	0	updates	1	UC	UC	UC	NoChange				ignore	yes
Properties of Incoming Packet								**Changes made by SIE to Internal Registers and Mode Bits**											
Mode Bits				**token**	**count**	**buffer**	**dval**	**DTOG**	**DVAL**	**COUNT**	**Setup**	**In**	**Out**	**ACK**	**Mode Bits**				**Response**	**Intr**
DISABLED																				
0	0	0	0	x	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
Nak In/Out																				
0	0	0	1	Out	x	UC	x	UC	UC	UC	UC	UC	1	UC	NoChange				NAK	yes
0	0	0	1	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	NoChange				NAK	yes
Ignore In/Out																				
0	1	0	0	Out	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
0	1	0	0	In	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
Stall In/Out																				
0	0	1	1	Out	x	UC	x	UC	UC	UC	UC	UC	1	UC	NoChange				Stall	yes
0	0	1	1	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	NoChange				Stall	yes
CONTROL WRITE																				
Properties of Incoming Packet								**Changes made by SIE to Internal Registers and Mode Bits**											
Mode Bits				**token**	**count**	**buffer**	**dval**	**DTOG**	**DVAL**	**COUNT**	**Setup**	**In**	**Out**	**ACK**	**Mode Bits**				**Response**	**Intr**
Normal Out/premature status In																				
1	0	1	1	Out	<= 10	data	valid	updates	1	updates	UC	UC	1	1	1	0	1	0	ACK	yes
1	0	1	1	Out	> 10	junk	x	updates	updates	updates	UC	UC	1	UC	NoChange				ignore	yes
1	0	1	1	Out	x	junk	invalid	updates	0	updates	UC	UC	1	UC	NoChange				ignore	yes
1	0	1	1	In	x	UC	x	UC	UC	UC	UC	1	UC	1	NoChange				TX 0	yes
NAK Out/premature status In																				
1	0	1	0	Out	<= 10	UC	valid	UC	UC	UC	UC	UC	1	UC	NoChange				NAK	yes
1	0	1	0	Out	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
1	0	1	0	Out	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
1	0	1	0	In	x	UC	x	UC	UC	UC	UC	1	UC	1	NoChange				TX 0	yes
Status In/extra Out																				
0	1	1	0	Out	<= 10	UC	valid	UC	UC	UC	UC	UC	1	UC	0	0	1	1	Stall	yes
0	1	1	0	Out	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
0	1	1	0	Out	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
0	1	1	0	In	x	UC	x	UC	UC	UC	UC	1	UC	1	NoChange				TX 0	yes
CONTROL READ																				
Properties of Incoming Packet								**Changes made by SIE to Internal Registers and Mode Bits**											
Mode Bits				**token**	**count**	**buffer**	**dval**	**DTOG**	**DVAL**	**COUNT**	**Setup**	**In**	**Out**	**ACK**	**Mode Bits**				**Response**	**Intr**
Normal In/premature status Out																				
1	1	1	1	Out	2	UC	valid	1	1	updates	UC	UC	1	1	NoChange				ACK	yes
1	1	1	1	Out	2	UC	valid	0	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes
1	1	1	1	Out	!=2	UC	valid	updates	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes
1	1	1	1	Out	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
1	1	1	1	Out	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
1	1	1	1	In	x	UC	x	UC	UC	UC	UC	1	UC	1	1	1	1	0	ACK (back)	yes
Nak In/premature status Out																				
1	1	1	0	Out	2	UC	valid	1	1	updates	UC	UC	1	1	NoChange				ACK	yes
1	1	1	0	Out	2	UC	valid	0	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes

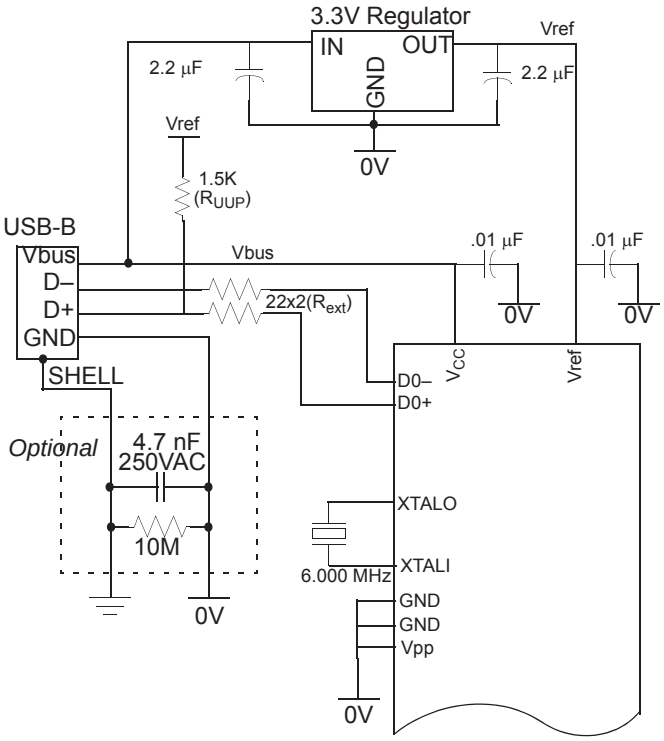
20.0 Register Summary

	Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/Write/ Both/-	Default/ Reset
GPIO CONFIGURATION PORTS 0, 1, 2 AND 3	0x00	Port 0 Data	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	BBBBBBBB	11111111
	0x01	Port 1 Data	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	BBBBBBBB	11111111
	0x02	Port 2 Data	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	BBBBBBBB	11111111
	0x03	Port 3 Data	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	BBBBBBBB	11111111
	0x04	Port 0 Interrupt Enable	P0.7 Intr Enable	P0.6 Intr Enable	P0.5 Intr Enable	P0.4 Intr Enable	P0.3 Intr Enable	P0.2 Intr Enable	P0.1 Intr Enable	P0.0 Intr Enable	WWWWWWWW	00000000
	0x05	Port 1 Interrupt Enable	P1.7 Intr Enable	P1.6 Intr Enable	P1.5 Intr Enable	P1.4 Intr Enable	Reserved	P1.2 Intr Enable	P1.1 Intr Enable	P1.0 Intr Enable	WWWWWWWW	00000000
	0x06	Port 2 Interrupt Enable	P2.7 Intr Enable	P2.6 Intr Enable	P2.5 Intr Enable	P2.4 Intr Enable	P2.3 Intr Enable	Reserved	Reserved	Reserved	WWWWWWWW	00000000
	0x07	Port 3 Interrupt Enable	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	P3.1 Intr Enable	P3.0 Intr Enable	WWWWWWWW	00000000
	0x08	GPIO Configuration	Port 3 Config Bit 1	Port 3 Config Bit 0	Port 2 Config Bit 1	Port 2 Config Bit 0	Port 1 Config Bit 1	Port 1 Config Bit 0	Port 0 Config Bit 1	Port 0 Config Bit 0	BBBBBBBB	00000000
HAPI I^2C	0x09	HAPI/I^2C Configuration	I^2C Position	Reserved	Reserved	Reserved	Reserved	Reserved	I^2C Port Width	Reserved	BBBBBBBB	00000000
ENDPOINT A0, AI AND A2 CONFIGURATION	0x10	USB Device Address A	Device Address A Enable	Device Address A Bit 6	Device Address A Bit 5	Device Address A Bit 4	Device Address A Bit 3	Device Address A Bit 2	Device Address A Bit 1	Device Address A Bit 0	BBBBBBBB	00000000
	0x11	EP A0 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x12	EP A0 Mode Register	Endpoint0 SETUP Received	Endpoint0 IN Received	Endpoint0 OUT Received	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000
	0x13	EP A1 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x14	EP A1 Mode Register	STALL	-	-	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000
	0x15	EP A2 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x16	EP A2 Mode Register	STALL	-	-	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000
USB CS	0x1F	USB Status and Control	Endpoint Size	Endpoint Mode	D+ Upstream	D– Upstream	Bus Activity	Control Bit 2	Control Bit 1	Control Bit 0	BBRRBBBB	-0xx0000
INTERRUPT	0x20	Global Interrupt Enable	Reserved	I^2C Interrupt Enable	GPIO Interrupt Enable	Reserved	USB Hub Interrupt Enable	1.024-ms Interrupt Enable	128-µs Interrupt Enable	USB Bus RESET Interrupt Enable	-BBBBBBB	-0000000
	0x21	Endpoint Interrupt Enable	Reserved	Reserved	Reserved	EPB1 Interrupt Enable	EPB0 Interrupt Enable	EPA2 Interrupt Enable	EPA1 Interrupt Enable	EPA0 Interrupt Enable	---BBBBB	---00000
TIMER	0x24	Timer (LSB)	Timer Bit 7	Timer Bit 6	Timer Bit 5	Timer Bit 4	Timer Bit 3	Timer Bit 2	Timer Bit 1	Timer Bit 0	RRRRRRRR	00000000
	0x25	Timer (MSB)	Reserved	Reserved	Reserved	Reserved	Timer Bit 11	Timer Bit 10	Time Bit 9	Timer Bit 8	----rrrr	----0000
	0x26	WDT Clear	x	x6	x	x	x 3	x2	x	x	WWWWWWWW	xxxxxxxx
I^2C	0x28	I^2C Control and Status	MSTR Mode	Continue/ Busy	Xmit Mode	ACK	Addr	ARB Lost/ Restart	Received Stop	I^2C Enable	BBBBBBBB	00000000
	0x29	I^2C Data	I^2C Data 7	I^2C Data 6	I^2C Data 5	I^2C Data 4	I^2C Data 3	I^2C Data 2	I^2C Data 1	I^2C Data 0	BBBBBBBB	XXXXXXXX
DAC PORT	0x30	DAC Data	Timer Bit 7	Timer Bit 6	Timer Bit 5	Timer Bit 4	Timer Bit 3	Timer Bit 2	Timer Bit 1	Timer Bit 0	RRRRRRRR	00000000
	0x31	DAC Interrupt Enable)	Reserved	Reserved	Reserved	Reserved	Timer Bit 11	Timer Bit 10	Time Bit 9	Timer Bit 8	----rrrr	----0000
	0x32	DAC Interrupt Polarity										
	0x38- 0x3F	DAS Isink	x	x6	x	x	x 3	x2	x	x	WWWWWWWW	
ENDPOINT A3, A4 CONFIGURATION	0x40	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	BBBBBBBB	00000000
	0x41	EP A3 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x42	EP A3 Mode Register	Endpoint 0 SETUP Received	Endpoint 0 IN Received	Endpoint 0 OUT Received	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000
	0x43	EP A4 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x44	EP A4 Mode Register	STALL	-	-	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000

	Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/Write/ Both/-	Default/ Reset
RESERVED	0x48	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x49	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4A	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4B	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4C	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	--000000
	0x4D	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4E	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4F	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x50	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x51	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x52	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0xFF	Process Status & Control	IRQ Pending	Watchdog Reset	USB Bus Reset Interrupt	Power-On Reset	Suspend	Interrupt Enable Sense	Reserved	Run	RBBBBRBB	00010001

Note:
B: Read and Write
W: Write
R: Read

21.0 Sample Schematic



23.0 Electrical Characteristics (continued)

f_{OSC} = 6 MHz; Operating Temperature = 0 to 70°C, V_{CC} = 4.0V to 5.25V

Parameter	Description	Conditions	Min.	Max.	Unit
	DAC Interface				
R_{up}	DAC Pull-up Resistance (typical 14 kΩ)		8.0	24.0	kΩ
$I_{sink0(0)}$	DAC[7:2] Sink current (0)	V_{out} = 2.0V DC	0.1	0.3	mA
$I_{sink0(F)}$	DAC[7:2] Sink current (F)	V_{out} = 2.0V DC	0.5	1.5	mA
$I_{sink1(0)}$	DAC[1:0] Sink current (0)	V_{out} = 2.0V DC	1.6	4.8	mA
$I_{sink1(F)}$	DAC[1:0] Sink current (F)	V_{out} = 2.0V DC	8	24	mA
I_{range}	Programmed Isink Ratio: max/min	V_{out} = 2.0V DC[6]	4	6	
T_{ratio}	Tracking Ratio DAC[1:0] to DAC[7:2]	V_{out} = 2.0V[7]	14	22	
$I_{sinkDAC}$	DAC Sink Current	V_{out} = 2.0V DC	1.6	4.8	mA
I_{lin}	Differential Nonlinearity	DAC Port[8]		0.6	LSB

24.0 Switching Characteristics (f_{OSC} = 6.0 MHz)

Parameter	Description	Min.	Max.	Unit
	Clock Source			
f_{OSC}	Clock Rate	6 ±0.25%		MHz
t_{cyc}	Clock Period	166.25	167.08	ns
t_{CH}	Clock HIGH time	0.45 t_{CYC}		ns
t_{CL}	Clock LOW time	0.45 t_{CYC}		ns
	USB Full Speed Signaling[9]			
t_{rfs}	Transition Rise Time	4	20	ns
t_{ffs}	Transition Fall Time	4	20	ns
t_{rfmfs}	Rise / Fall Time Matching; (t_r/t_f)	90	111	%
$t_{dratefs}$	Full Speed Date Rate	12 ±0.25%		Mb/s
	DAC Interface			
t_{sink}	Current Sink Response Time		0.8	µs
	HAPI Read Cycle Timing			
t_{RD}	Read Pulse Width	15		ns
t_{OED}	OE LOW to Data Valid[10, 11]		40	ns
t_{OEZ}	OE HIGH to Data High-Z[11]		20	ns
t_{OEDR}	OE LOW to Data_Ready Deasserted[10, 11]	0	60	ns
	HAPI Write Cycle Timing			
t_{WR}	Write Strobe Width	15		ns
t_{DSTB}	Data Valid to STB HIGH (Data Set-up Time)[11]	5		ns
t_{STBZ}	STB HIGH to Data High-Z (Data Hold Time)[11]	15		ns
t_{STBLE}	STB LOW to Latch_Empty Deasserted[10, 11]	0	50	ns
	Timer Signals			
t_{watch}	Watchdog Timer Period	8.192	14.336	ms

Notes:

6. Irange: $I_{sinkn}(15)/ I_{sinkn}(0)$ for the same pin.
7. $T_{ratio} = I_{sink1}[1:0](n)/I_{sink0}[7:2](n)$ for the same n, programmed.
8. I_{lin} measured as largest step size vs. nominal according to measured full scale and zero programmed values.
9. Per Table 7-6 of revision 1.1 of USB specification.
10. For 25-pF load.
11. Assumes chip select $\overline{CS}$ is asserted (LOW).

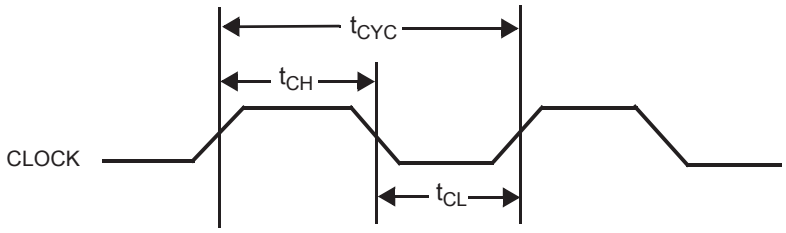


Figure 24-1. Clock Timing

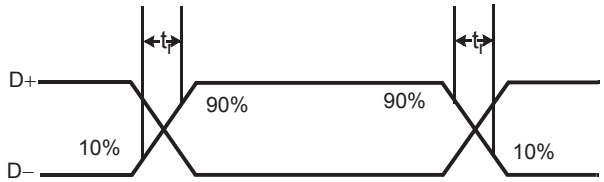


Figure 24-2. USB Data Signal Timing

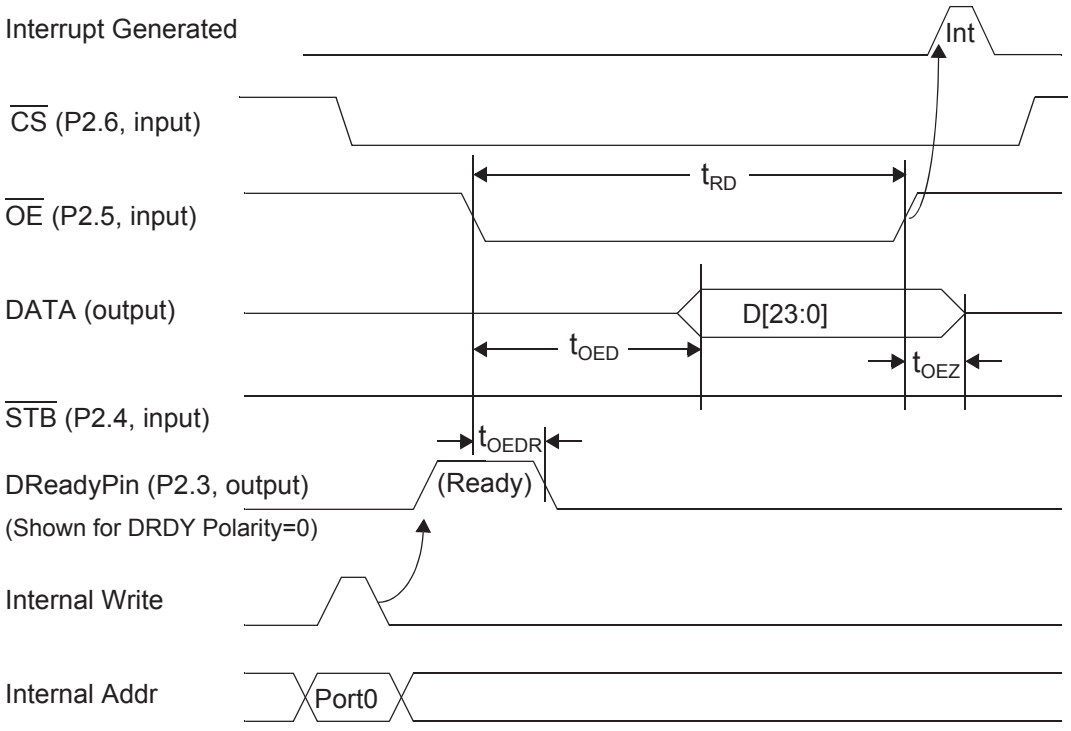
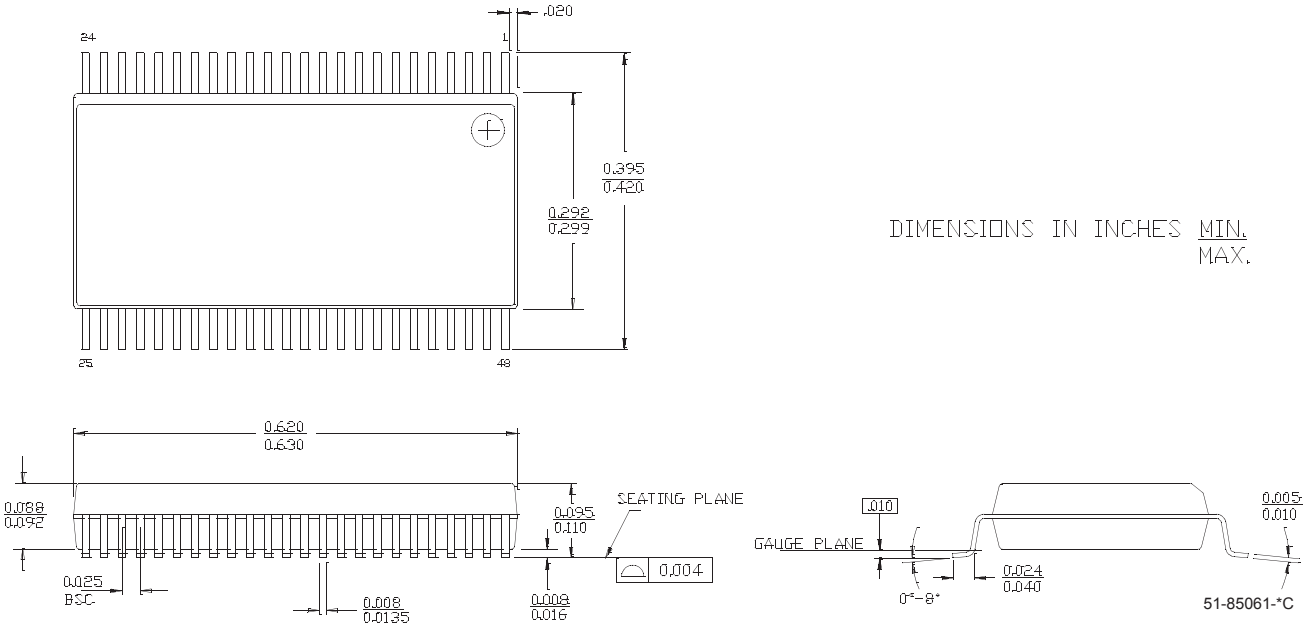


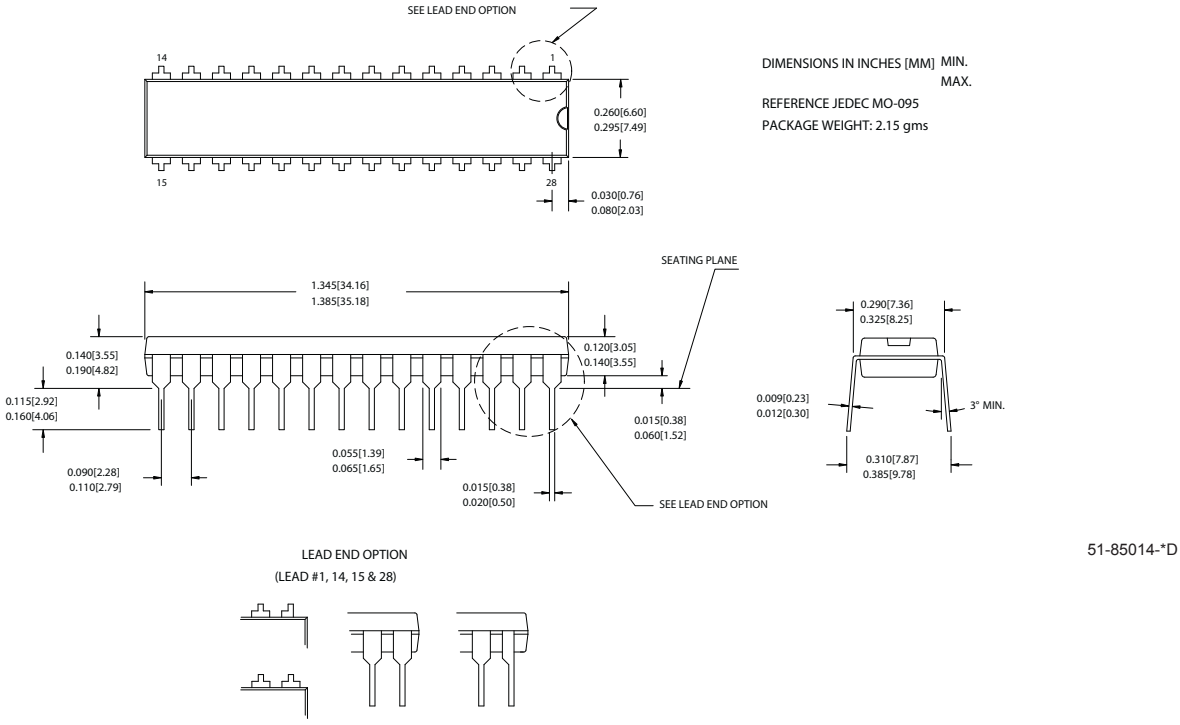
Figure 24-3. HAPI Read by External Interface from USB Microcontroller

26.0 Package Diagrams

48-Lead Shrunk Small Outline Package



28-Lead (300-Mil) PDIP



Document History Page

Document Title: CY7C64013C, CY7C64113C Full-Speed USB (12 Mbps) Function Document Number: 38-08001				
REV.	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	109962	12/16/01	SZV	Change from Spec number: 38-00626 to 38-08001
*A	129715	02/05/04	MON	Added register bit definitions Added default bit state of each register Corrected the Schematic (location of the Pull up on D+) Added register summary Modified tables 19-1 and 19-2 Provided more explanation regarding locking/unlocking mechanism of the mode register.
*B	429099	See ECN	TYJ	Changed part numbers to the ‘C’ types. Included ‘Cypress Perform’ logo. Updated part numbers in the Ordering section.