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 Host/Peripheral Controller
Core Processor	CY16
Program Memory Type	ROM (8kB)
Controller Series	CY7C673xx
RAM Size	16K x 8
Interface	SPI Serial, USB, HPI
Number of I/O	32
Voltage - Supply	3V ~ 3.6V
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy7c67300-100axat

Table 1. Interface Options for GPIO Pins

GPIO Pins	HPI	IDE	PWM	HSS	SPI	UART	I2C	OTG
GPIO31							SCL/SDA	
GPIO30							SCL/SDA	
GPIO29								OTGID
GPIO28						TX		
GPIO27						RX		
GPIO26			PWM3	CTS[1]				
GPIO25								
GPIO24	INT	IOREADY						
GPIO23	nRD	IOR						
GPIO22	nWR	IOW						
GPIO21	nCS							
GPIO20	A1	CS1						
GPIO19	A0	CS0						
GPIO18		A2	PWM2	RTS[1]				
GPIO17		A1	PWM1	RXD[1]				
GPIO16		A0	PWM0	TXD[1]				
GPIO15	D15	D15						
GPIO14	D14	D14						
GPIO13	D13	D13						
GPIO12	D12	D12						
GPIO11	D11	D11			MOSI[1]			
GPIO10	D10	D10			SCK[1]			
GPIO9	D9	D9			nSSI[1]			
GPIO8	D8	D8			MISO[1]			
GPIO7	D7	D7						
GPIO6	D6	D6						
GPIO5	D5	D5						
GPIO4	D4	D4						
GPIO3	D3	D3						
GPIO2	D2	D2						
GPIO1	D1	D1						
GPIO0	D0	D0						

Table 2. Interface Options for External Memory Bus Pins

MEM Pins	HPI	IDE	PWM	HSS	SPI	UART	I2C	OTG
D15				CTS[2]				
D14				RTS[2]				
D13				RXD[2]				
D12				TXD[2]				
D11					MOSI[2]			
D10					SCK[2]			
D9					nSSI[2]			
D8					MISO[2]			
D[7:0]								
A[18:0]								
CONTROL								

Notes
1. Default interface location.
2. Alternate interface location.

SPI Pins

The SPI port has a few different pin location options as shown in Table 9. The port location is selectable via the GPIO control register [0xC006].

Table 9. SPI Interface Pins

Pin Name	Pin Number
Default Location	
nSSI	56 or 65
SCK	61
MOSI	60
MISO	66
Alternate Location	
nSSI	73
SCK	72
MOSI	71
MISO	74

High-Speed Serial Interface

EZ-Host provides an HSS interface. The HSS interface is a programmable serial connection with baud rate from 9600 baud to 2.0M baud. The HSS interface supports both byte and block mode operations and also hardware and software handshaking. Complete control of EZ-Host can be accomplished through this interface via an extensible API and communication protocol. The HSS interface can be exposed through GPIO pins or the External Memory port.

HSS Features

- 8 bits, no parity code
- Programmable baud rate from 9600 baud to 2M baud
- Selectable 1- or 2-stop bit on transmit
- Programmable inter-character gap timing for Block Transmit
- 8-byte receive FIFO
- Glitch filter on receive
- Block mode transfer directly to/from EZ-Host internal memory (DMA transfer)
- Selectable CTS/RTS hardware signal handshake protocol
- Selectable XON/XOFF software handshake protocol
- Programmable Receive interrupt, Block Transfer Done interrupts
- Complete access to internal memory

HSS Pins

The HSS port has a few different pin location options as shown in Table 10. The port location is selectable via the GPIO control register [0xC006].

Table 10. HSS Interface Pins

Pin Name	Pin Number
Default Location	
CTS	44
RTS	53
RXD	54
TXD	55
Alternate Location	
CTS	67
RTS	68
RXD	69
TXD	70

Programmable Pulse/PWM Interface

EZ-Host has four built in PWM output channels. Each channel provides a programmable timing generator sequence that can be used to interface to various image sensors or other applications. The PWM interface is exposed through GPIO pins.

Programmable Pulse/PWM Features

- Four independent programmable waveform generators
- Programmable predefined frequencies ranging from 5.90 KHz to 48 MHz
- Configurable polarity
- Continuous and one-shot mode available

Programmable Pulse/PWM Pins.

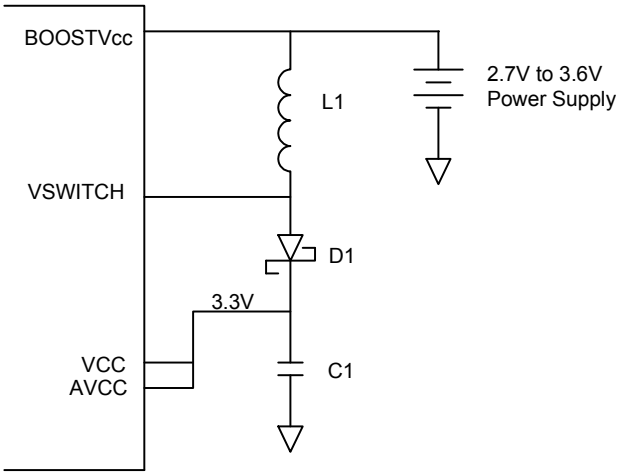
Table 11. PWM Interface Pins

Pin Name	Pin Number
PWM3	44
PWM2	53
PWM1	54
PWM0	55

Booster Interface

EZ-Host has an on chip power booster circuit for use with power supplies that range between 2.7V and 3.6V. The booster circuit boosts the power to 3.3V nominal to supply power for the entire chip. The booster circuit requires an external inductor, diode, and capacitor. During power down mode, the circuit is disabled to save power. Figure 6 shows how to connect the booster circuit.

Figure 6. Power Supply Connection With Booster

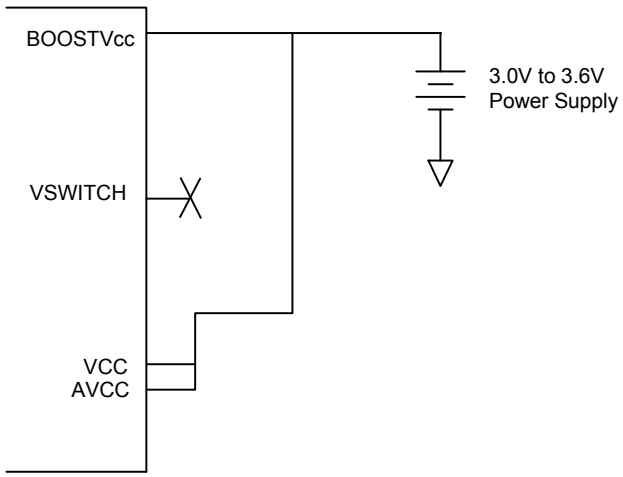


Component details:

- L1: Inductor with inductance of 10 µH and a current rating of at least 250 mA
- D1: Schottky diode with a current rating of at least 250 mA
- C1: Tantalum or ceramic capacitor with a capacitance of at least 2.2 µF

Figure 7 shows how to connect the power supply when the booster circuit is not being used.

Figure 7. Power Supply Connection Without Booster



Booster Pins

Table 17. Charge Pump Interface Pins

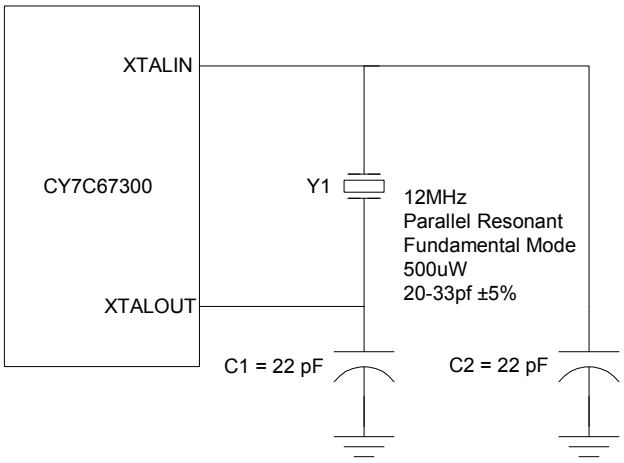
Pin Name	Pin Number
BOOSTVcc	16
VSWITCH	14

Crystal Interface

The recommended crystal circuit to be used with EZ-Host is shown in Figure 8 If an oscillator is used instead of a crystal circuit, connect it to XTALIN and leave XTALOUT unconnected. For further information about the crystal requirements, see Table 132 on page 90.

Noted that the CLKSEL pin (pin 38) is sampled after reset to determine what crystal or clock source frequency is used. For normal operation, 12 MHz is required so the CLKSEL pin must have a 47K ohm pull up resistor to V_{CC}.

Figure 8. Crystal Interface



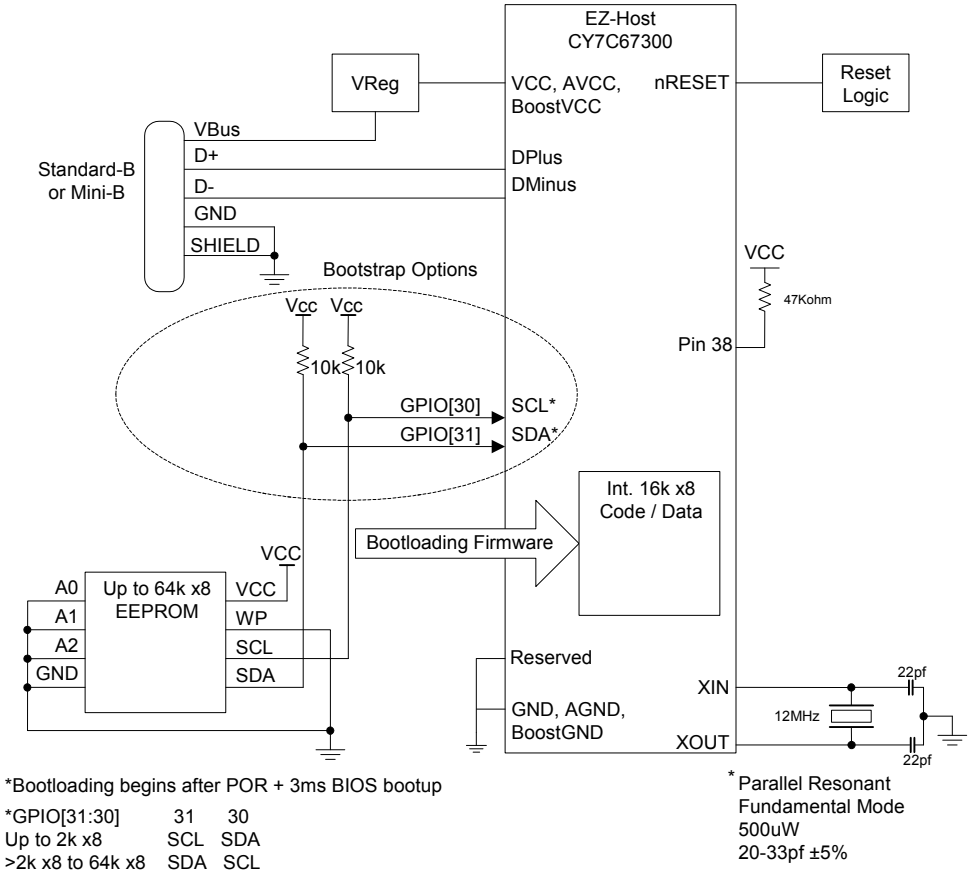
Crystal Pins

Table 18. Crystal Pins

Pin Name	Pin Number
XTALIN	29
XTALOUT	28

Minimum Hardware Requirements for Standalone Mode – Peripheral Only

Figure 9. Minimum Standalone Hardware Configuration – Peripheral Only



Power Savings and Reset Description

This sections describes the different modes for resetting the chip and ways to save power.

Power Saving Mode Description

EZ-Host has one main power saving mode, Sleep. For detailed information about Sleep mode, see the Sleep section that follows.

Sleep mode is used for USB applications to support USB suspend and non USB applications as the main chip power down mode.

In addition, EZ-Host is capable of slowing down the CPU clock speed through the CPU Speed register [0xC008] without affecting other peripheral timing. Reducing the CPU clock speed from 48 MHz to 24 MHz reduces the overall current draw by around 8 mA while reducing it from 48 MHz to 3 MHz reduces the overall current draw by approximately 15 mA.

Sleep

Sleep mode is the main chip power down mode and is also used for USB suspend. Sleep mode is entered by setting the Sleep Enable (bit 1) of the Power control register [0xC00A]. During Sleep mode (USB Suspend) the following events and states are true:

- GPIO pins maintain their configuration during sleep (in suspend)
- External Memory address pins are driven low
- XTALOUT is turned off
- Internal PLL is turned off
- Ensure that firmware disables the charge pump (OTG Control register [0xC098]) thereby causing OTGVBUS to drop below 0.2V. Otherwise OTGVBUS only drops to V_{CC} – (2 schottky diode drops).
- Booster circuit is turned off
- USB transceivers is turned off
- CPU goes into suspend mode until a programmable wakeup event

Registers

Some registers have different functions for a read vs. a write access or USB host vs. USB device mode. Therefore, registers of this type have multiple definitions for the same address.

The default register values listed in this data sheet may be altered to some other value during the BIOS initialization. Refer to the BIOS documentation for register initialization information.

Processor Control Registers

There are nine registers dedicated to general processor control. Each of these registers are covered in this section and are summarized in Table 21.

Table 21. Processor Control Registers

Register Name	Address	R/W
CPU Flags Register	0xC000	R
Register Bank Register	0xC002	R/W
Hardware Revision Register	0xC004	R
CPU Speed Register	0xC008	R/W
Power Control Register	0xC00A	R/W
Interrupt Enable Register	0xC00E	R/W
Breakpoint Register	0xC014	R/W
USB Diagnostic Register	0xC03C	W
Memory Diagnostic Register	0xC03E	W

CPU Flags Register [0xC000] [R]

Table 22. CPU Flags Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved...							
Read/Write	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	...Reserved			Global Interrupt Enable	Negative Flag	Overflow Flag	Carry Flag	Zero Flag
Read/Write	-	-	-	R	R	R	R	R
Default	0	0	0	X	X	X	X	X

Register Description

The CPU Flags register is a read only register that gives processor flags status.

Global Interrupt Enable (Bit 4)

The Global Interrupt Enable bit indicates if the Global Interrupts are enabled.

1: Enabled

0: Disabled

Negative Flag (Bit 3)

The Negative Flag bit indicates if an arithmetic operation results in a negative answer.

1: MS result bit is ‘1’

0: MS result bit is not ‘1’

Overflow Flag (Bit 2)

The Overflow Flag bit indicates if an overflow condition occurred. An overflow condition can occur if an arithmetic result was either larger than the destination operand size (for addition) or smaller than the destination operand must allow for subtraction.

1: Overflow occurred

0: Overflow did not occur

Carry Flag (Bit 1)

The Carry Flag bit indicates if an arithmetic operation resulted in a Carry for addition, or Borrow for subtraction.

1: Carry/Borrow occurred

0: Carry/Borrow did not occur

Zero Flag (Bit 0)

The Zero Flag bit indicates if an instruction execution resulted in a ‘0’.

1: Zero occurred

0: Zero did not occur

CPU Speed Register [0xC008] [R/W]

Table 26. CPU Speed Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved...							
Read/Write	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	...Reserved				CPU Speed			
Read/Write	-	-	-	-	R/W	R/W	R/W	R/W
Default	0	0	0	0	1	1	1	1

Register Description

The CPU Speed register allows the processor to operate at a user selected speed. This register only affects the CPU, all other peripheral timing is still based on the 48 MHz system clock (unless otherwise noted).

CPU Speed (Bits[3:0])

The CPU Speed field is a divisor that selects the operating speed of the processor as defined in Table 27.

Table 27. CPU Speed Definition

CPU Speed [3:0]	Processor Speed
0000	48 MHz/1
0001	48 MHz/2
0010	48 MHz/3
0011	48 MHz/4
0100	48 MHz/5
0101	48 MHz/6
0110	48 MHz/7
0111	48 MHz/8
1000	48 MHz/9
1001	48 MHz/10
1010	48 MHz/11
1011	48 MHz/12
1100	48 MHz/13
1101	48 MHz/14
1110	48 MHz/15
1111	48 MHz/16

Reserved

Write all reserved bits with ’0’.

HSS Interrupt Enable (Bit 7)

The HSS Interrupt Enable bit enables or disables the following High-speed Serial Interface hardware interrupts: HSS Block Done and HSS RX Full.

1: Enable HSS interrupt

0: Disable HSS interrupt

In Mailbox Interrupt Enable (Bit 6)

The In Mailbox Interrupt Enable bit enables or disables the HPI: Incoming Mailbox hardware interrupt.

1: Enable MBXI interrupt

0: Disable MBXI interrupt

Out Mailbox Interrupt Enable (Bit 5)

The Out Mailbox Interrupt Enable bit enables or disables the HPI: Outgoing Mailbox hardware interrupt.

1: Enable MBXO interrupt

0: Disable MBXO interrupt

UART Interrupt Enable (Bit 3)

The UART Interrupt Enable bit enables or disables the following UART hardware interrupts: UART TX, and UART RX.

1: Enable UART interrupt

0: Disable UART interrupt

GPIO Interrupt Enable (Bit 2)

The GPIO Interrupt Enable bit enables or disables the General Purpose IO pins interrupt (see the GPIO Control Register [0xC006] [R/W] on page 56). When the GPIO bit is reset, all pending GPIO interrupts are also cleared

1: Enable GPIO interrupt

0: Disable GPIO interrupt

Timer 1 Interrupt Enable (Bit 1)

The Timer 1 Interrupt Enable bit enables or disables the TImer1 Interrupt Enable. When this bit is reset, all pending Timer 1 interrupts are cleared.

1: Enable TM1 interrupt

0: Disable TM1 interrupt

Timer 0 Interrupt Enable (Bit 0)

The Timer 0 Interrupt Enable bit enables or disables the TImer0 Interrupt Enable. When this bit is reset, all pending Timer 0 interrupts are cleared.

1: Enable TM0 interrupt

0: Disable TM0 interrupt

Reserved

Write all reserved bits with ’0’.

Breakpoint Register [0xC014] [R/W]

Table 30. Breakpoint Register

Bit #	15	14	13	12	11	10	9	8
Field	Address...							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	...Address							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register Description

The Breakpoint register holds the breakpoint address. When the program counter matches this address, the INT127 interrupt occurs. To clear this interrupt, write a zero value to this register.

Address (Bits [15:0])

The Address field is a 16-bit field containing the breakpoint address.

Memory Diagnostic Register [0xC03E] [W]

Table 33. Memory Diagnostic Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved					Memory Arbitration Select		
Read/Write	-	-	-	-	-	W	W	W
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	Reserved							Monitor Enable
Read/Write	-	-	-	-	-	-	-	W
Default	0	0	0	0	0	0	0	0

Register Description

The Memory Diagnostic register provides control of diagnostic modes.

Memory Arbitration Select (Bits[10:8])

The Memory Arbitration Select field is defined in Table 34.

Table 34. Memory Arbitration Select

Memory Arbitration Select [3:0]	Memory Arbitration Timing
111	1/8, 7 of every 8 cycles dead
110	2/8, 6 of every 8 cycles dead
101	3/8, 5 of every 8 cycles dead
100	4/8, 4 of every 8 cycles dead
011	5/8, 3 of every 8 cycles dead
010	6/8, 2 of every 8 cycles dead
001	7/8, 1 of every 8 cycles dead
000	8/8, all cycles available

Monitor Enable (Bit 0)

The Monitor Enable bit enables or disables monitor mode. In monitor mode the internal address bus is echoed to the external address pins.

1: Enable monitor mode

0: Disable monitor mode

Reserved

Write all reserved bits with ’0’.

External Memory Registers

There are four registers dedicated to controlling the external memory interface. Each of these registers are covered in this section and are summarized in Table 35.

Table 35. External Memory Control Registers

Register Name	Address	R/W
Extended Page 1 Map Register	0xC018	R/W
Extended Page 2 Map Register	0xC01A	R/W
Upper Address Enable Register	0xC038	R/W
External Memory Control Register	0xC03A	R/W

Extended Page n Map Register [R/W]

- Extended Page 1 Map Register 0xC018
- Extended Page 2 Map Register 0xC01A

Table 36. Extended Page n Map Register

Bit #	15	14	13	12	11	10	9	8
Field	Address...							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	...Address							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register Description

The Extended Page n Map register contains the Page n high-order address bits. These bits are always appended to accesses to the Page n Memory mapped space.

Address (Bits [15:0])

The Address field contains the high-order bits 28 to 13 of the Page n address. The address pins [8:0] (Page n address [21:13]) reflect the content of this register when the CPU accesses the address 0x8000-0x9FFF. For the SRAM mode, the address pin on [4:0] (Page n address [17:13]) is used.

Set bit [8] (Page n address [21]) to '0', so that Page n reads/writes access external areas (SRAM, ROM or peripherals). nXMEMSEL is the external chip select for this space.

Upper Address Enable Register [0xC038] [R/W]

Table 37. External Memory Control Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved							
Read/Write	-	-	-	-	-	-	-	-
Default	X	X	X	X	X	X	X	X

Bit #	7	6	5	4	3	2	1	0
Field	Reserved				Upper Address Enable	Reserved		
Read/Write	-	-	-	-	R/W			
Default	X	X	X	X	0	X	X	X

Register Description

The Upper Address Enable register enables/disables the four most significant bits of the external address A[18:15]. This register defaults to having the Upper Address disabled. Note that on power up, pins A[18:15] are driven high.

Upper Address Enable (Bit 3)

The Upper Address Enable bit enables/disables the four most significant bits of the external address A[18:15].

1: Enable A[18:15] of the external memory interface for general addressing.

0: Disable A[18:15], not available.

Reserved

Write all reserved bits with '0'.

Device n Endpoint n Status Register [R/W]

- Device n Endpoint 0 Status Register [Device 1: 0x0206 Device 2: 0x0286]
- Device n Endpoint 1 Status Register [Device 1: 0x0216 Device 2: 0x0296]
- Device n Endpoint 2 Status Register [Device 1: 0x0226 Device 2: 0x02A6]
- Device n Endpoint 3 Status Register [Device 1: 0x0236 Device 2: 0x02B6]
- Device n Endpoint 4 Status Register [Device 1: 0x0246 Device 2: 0x02C6]
- Device n Endpoint 5 Status Register [Device 1: 0x0256 Device 2: 0x02D6]
- Device n Endpoint 6 Status Register [Device 1: 0x0266 Device 2: 0x02E6]
- Device n Endpoint 7 Status Register [Device 1: 0x0276 Device 2: 0x02F6]

Table 66. Device n Endpoint n Status Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved				Overflow Flag	Underflow Flag	OUT Exception Flag	IN Exception Flag
Read/Write	-	-	-	-	R/W	R/W	R/W	R/W
Default	X	X	X	X	X	X	X	X

Bit #	7	6	5	4	3	2	1	0
Field	Stall Flag	NAK Flag	Length Exception Flag	Setup Flag	Sequence Flag	Timeout Flag	Error Flag	ACK Flag
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	X	X	X	X	X	X	X	X

Register Description

The Device n Endpoint n Status register provides packet status information for the last transaction received or transmitted. This register is updated in hardware and does not need to be cleared by firmware. There are a total of eight endpoints for each of the two ports. All endpoints have the same definition for their Device n Endpoint n Status register.

The Device n Endpoint n Status register is a memory based register that must be initialized to 0x0000 before USB Device operations are initiated. After initialization, do not write to this register again.

Overflow Flag (Bit 11)

The Overflow Flag bit indicates that the received data in the last data transaction exceeded the maximum length specified in the Device n Endpoint n Count register. The Overflow Flag must be checked in response to a Length Exception signified by the Length Exception Flag set to '1'.

1: Overflow condition occurred

0: Overflow condition did not occur

Underflow Flag (Bit 10)

The Underflow Flag bit indicates that the received data in the last data transaction was less then the maximum length specified in the Device n Endpoint n Count register. The Underflow Flag must be checked in response to a Length Exception signified by the Length Exception Flag set to '1'.

1: Underflow condition occurred

0: Underflow condition did not occur

OUT Exception Flag (Bit 9)

The OUT Exception Flag bit indicates when the device received an OUT packet when armed for an IN.

1: Received OUT when armed for IN

0: Received IN when armed for IN

IN Exception Flag (Bit 8)

The IN Exception Flag bit indicates when the device received an IN packet when armed for an OUT.

1: Received IN when armed for OUT

0: Received OUT when armed for OUT

Stall Flag (Bit 7)

The Stall Flag bit indicates that a Stall packet was sent to the host.

1: Stall packet was sent to the host

0: Stall packet was not sent

NAK Flag (Bit 6)

The NAK Flag bit indicates that a NAK packet was sent to the host.

1: NAK packet was sent to the host

0: NAK packet was not sent

Length Exception Flag (Bit 5)

The Length Exception Flag bit indicates the received data in the data stage of the last transaction does not equal the maximum Endpoint Count specified in the Device n Endpoint n Count register. A Length Exception can either mean an overflow or

EP5 Interrupt Enable (Bit 5)

The EP5 Interrupt Enable bit enables or disables endpoint five (EP5) Transaction Done interrupt. An EPx Transaction Done interrupt triggers when any of the following responses or events occur in a transaction for the device's supplied Endpoint: send/receive ACK, send STALL, Timeout occurs, IN Exception Error, or OUT Exception Error. In addition, the NAK Interrupt Enable bit in the Device n Endpoint Control register can also be set so that NAK responses trigger this interrupt.

1: Enable EP5 Transaction Done interrupt

0: Disable EP5 Transaction Done interrupt

EP4 Interrupt Enable (Bit 4)

The EP4 Interrupt Enable bit enables or disables endpoint four (EP4) Transaction Done interrupt. An EPx Transaction Done interrupt triggers when any of the following responses or events occur in a transaction for the device's supplied Endpoint: send/receive ACK, send STALL, Timeout occurs, IN Exception Error, or OUT Exception Error. In addition, the NAK Interrupt Enable bit in the Device n Endpoint Control register can also be set so that NAK responses trigger this interrupt.

1: Enable EP4 Transaction Done interrupt

0: Disable EP4 Transaction Done interrupt

EP3 Interrupt Enable (Bit 3)

The EP3 Interrupt Enable bit enables or disables endpoint three (EP3) Transaction Done interrupt. An EPx Transaction Done interrupt triggers when any of the following responses or events occur in a transaction for the device's supplied Endpoint: send/receive ACK, send STALL, Timeout occurs, IN Exception Error, or OUT Exception Error. In addition, the NAK Interrupt Enable bit in the Device n Endpoint Control register can also be set so that NAK responses trigger this interrupt.

1: Enable EP3 Transaction Done interrupt

0: Disable EP3 Transaction Done interrupt

EP2 Interrupt Enable (Bit 2)

The EP2 Interrupt Enable bit enables or disables endpoint two (EP2) Transaction Done interrupt. An EPx Transaction Done interrupt triggers when any of the following responses or events occur in a transaction for the device's supplied Endpoint: send/receive ACK, send STALL, Timeout occurs, IN Exception Error, or OUT Exception Error. In addition, the NAK Interrupt Enable bit in the Device n Endpoint Control register can also be set so that NAK responses trigger this interrupt.

1: Enable EP2 Transaction Done interrupt

0: Disable EP2 Transaction Done interrupt

EP1 Interrupt Enable (Bit 1)

The EP1 Interrupt Enable bit enables or disables endpoint one (EP1) Transaction Done interrupt. An EPx Transaction Done interrupt triggers when any of the following responses or events occur in a transaction for the device's supplied Endpoint: send/receive ACK, send STALL, Timeout occurs, IN Exception Error, or OUT Exception Error. In addition, the NAK Interrupt Enable bit in the Device n Endpoint Control register can also be set so that NAK responses trigger this interrupt.

1: Enable EP1 Transaction Done interrupt

0: Disable EP1 Transaction Done interrupt

EP0 Interrupt Enable (Bit 0)

The EP0 Interrupt Enable bit enables or disables endpoint zero (EP0) Transaction Done interrupt. An EPx Transaction Done interrupt triggers when any of the following responses or events occur in a transaction for the device's supplied Endpoint: send/receive ACK, send STALL, Timeout occurs, IN Exception Error, or OUT Exception Error. In addition, the NAK Interrupt Enable bit in the Device n Endpoint Control register can also be set so that NAK responses trigger this interrupt.

1: Enable EP0 Transaction Done interrupt

0: Disable EP0 Transaction Done interrupt

Reserved

Write all reserved bits with '0'.

Device n Address Register [W]

- Device 1 Address Register 0xC08E
- Device 2 Address Register 0xC0AE

Table 70. Device n Address Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved...							
Read/Write	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	...Reserved	Address						
Read/Write	-	W	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0

Register Description

The Device n Address register holds the device address assigned by the host. This register initializes to the default

Done1 Flag (Bit 2)

In host mode the Done 1 Flag bit is a read only bit that indicates if a host packet done interrupt occurs on Host 1. In device mode this read only bit indicates if an any of the endpoint interrupts occur on Device 1. Firmware needs to determine which endpoint interrupt occurred.

1: Interrupt triggered

0: Interrupt did not trigger

Reset1 Flag (Bit 1)

The Reset1 Flag bit is a read only bit that indicates if a USB Reset interrupt occurs on either Host/Device 1.

1: Interrupt triggered

0: Interrupt did not trigger

Mailbox Out Flag (Bit 0)

The Mailbox Out Flag bit is a read only bit that indicates if a message is ready in the outgoing mailbox. This interrupt clears when the external host reads from the HPI Mailbox register.

1: Interrupt triggered

0: Interrupt did not trigger

SPI Registers

There are twelve registers dedicated to SPI operation. Each of these registers is covered in this section and summarized in Table 104.

Table 104. SPI Registers

Register Name	Address	R/W
SPI Configuration Register	0xC0C8	R/W
SPI Control Register	0xC0CA	R/W
SPI Interrupt Enable Register	0xC0CC	R/W
SPI Status Register	0xC0CE	R
SPI Interrupt Clear Register	0xC0D0	W
SPI CRC Control Register	0xC0D2	R/W
SPI CRC Value	0xC0D4	R/W
SPI Data Register	0xC0D6	R/W
SPI Transmit Address Register	0xC0D8	R/W
SPI Transmit Count Register	0xC0DA	R/W
SPI Receive Address Register	0xC0DC	R/W
SPI Receive Count Register	0xC0DE	R/W

SPI Data Register [0xC0D6] [R/W]

Table 114. SPI Data Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved							
Read/Write	-	-	-	-	-	-	-	-
Default	X	X	X	X	X	X	X	X

Bit #	7	6	5	4	3	2	1	0
Field	Data							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	X	X	X	X	X	X	X	X

Register Description

The SPI Data register contains data received on the SPI port when read. Reading it empties the eight byte receive FIFO in PIO byte mode. This receive data is valid when the Receive Interrupt Bit of the SPI Status register is set to '1' (RxIntVal triggers) or the Receive Data Ready bit of the SPI Control register is set to '1'. Writing to this register in PIO byte mode initiates a transfer of data, the number of bits defined by Transmit Bit Length field in the SPI Control register.

Data (Bits [7:0])

The Data field contains data received or to be transmitted on the SPI port.

Reserved

Write all reserved bits with '0'.

SPI Transmit Address Register [0xC0D8] [R/W]

Table 115. SPI Transmit Address Register

Bit #	15	14	13	12	11	10	9	8
Field	Address...							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	...Address							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register Description

The SPI Transmit Address register is used as the base address for the SPI transmit DMA.

Address (Bits [15:0])

The Address field sets the base address for the SPI transmit DMA.

Register Description

The SPI Receive Count register designates the block byte length for the SPI receive DMA transfer.

Count (Bits [10:0])

The Count field sets the count for the SPI receive DMA transfer.

Reserved

Write all reserved bits with ’0’.

UART Registers

There are three registers dedicated to UART operation. Each of these registers is covered in this section and summarized in Table 119.

Table 119. UART Registers

Register Name	Address	R/W
UART Control Register	0xC0E0	R/W
UART Status Register	0xC0E2	R
UART Data Register	0xC0E4	R/W

UART Control Register [0xC0E0] [R/W]

Table 120. UART Control Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved...							
Read/Write	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	...Reserved			Scale Select	Baud Select			UART Enable
Read/Write	-	-	-	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	1	1	1

Register Description

The UART Control register enables or disables the UART, allowing GPIO28 (UART_TXD) and GPIO27 (UART_RXD) to be freed up for general use. This register must also be written to set the baud rate, which is based on a 48 MHz clock.

Scale Select (Bit 4)

The Scale Select bit acts as a prescaler that divide the baud rate by eight.

1: Enable prescaler

0: Disable prescaler

Baud Select (Bits [3:1])

Refer to Table 121 for a definition of this field.

Table 121. UART Baud Select Definition

Baud Select [3:1]	Baud Rate w/ DIV8 = 0	Baud Rate w/ DIV8 = 1
000	115.2 KBaud	14.4 KBaud
001	57.6 KBaud	7.2 KBaud
010	38.4 KBaud	4.8 KBaud
011	28.8 KBaud	3.6 KBaud
100	19.2 KBaud	2.4 KBaud
101	14.4 KBaud	1.8 KBaud
110	9.6 KBaud	1.2 KBaud
111	7.2 KBaud	0.9 KBaud

UART Enable (Bit 0)

The UART Enable bit enables or disables the UART.

1: Enable UART

0: Disable UART. This allows GPIO28 and GPIO27 to be used for general use.

Reserved

Write all reserved bits with ’0’.

PWM n Start Register [R/W]

- PWM 0 Start Register 0xC0EA
- PWM 1 Start Register 0xC0EE
- PWM 2 Start Register 0xC0F2
- PWM 3 Start Register 0xC0F6

Table 128. PWM n Start Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved						Address...	
Read/Write	-	-	-	-	-	-	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	...Address							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register Description

The PWM n Start register designates where in the window defined by the PWM Maximum Count register to start the PWM pulse for a supplied channel.

Address (Bits [9:0])

The Address field designates when to start the PWM pulse. If this start value is equal to the Stop Count Value then the output stays at false.

Reserved

Write all reserved bits with ’0’.

PWM n Stop Register [R/W]

- PWM 0 Stop Register 0xC0EC
- PWM 1 Stop Register 0xC0F0
- PWM 2 Stop Register 0xC0F4
- PWM 3 Stop Register 0xC0F8

Table 129. PWM n Stop Register

Bit #	15	14	13	12	11	10	9	8
Field	Reserved						Address...	
Read/Write	-	-	-	-	-	-	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit #	7	6	5	4	3	2	1	0
Field	...Address							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register Description

The PWM n Stop register designates where in the window defined by the PWM Maximum Count register to stop the PWM pulse for a supplied channel.

Address (Bits [9:0])

The Address field designates when to stop the PWM pulse. If the PWM Start value is equal to the PWM Stop value then the output stays at ‘0’. If the PWM Stop value is greater then the PWM Maximum Count value then the output stays at true.

Reserved

Write all reserved bits with ’0’.

Table 133. DC Characteristics (Continued)[17]

Parameter	Description	Conditions	Min	Typ.	Max	Unit
I_{SLEEP}	Sleep Current	USB Peripheral: includes 1.5K internal pull up		210	500	µA
		Without 1.5K internal pull up		5	30	µA
I_{SLEEPB}	Sleep Current with Booster Enabled	USB Peripheral: includes 1.5K internal pull up		190	500	µA
		Without 1.5K internal pull up		5	30	µA

Table 134. DC Characteristics: Charge Pump

Parameter	Description	Conditions	Min	Typ.	Max	Unit
$V_{A\_VBUS\_OUT}$	Regulated OTGVBUS Voltage	8 mA< I_{LOAD} < 10 mA	4.4		5.25	V
$T_{A\_VBUS\_RISE}$	V_{BUS} Rise Time	I_{LOAD} = 10 mA			100	ms
$I_{A\_VBUS\_OUT}$	Maximum Load Current		8		10	mA
C_{DRD_VBUS}	OUTVBUS Bypass Capacitance	4.4V< V_{BUS} < 5.25V	1.0		6.5	pF
$V_{A\_VBUS\_LKG}$	OTGVBUS Leakage Voltage	OTGVBUS not driven			200	mV
$V_{DRD\_DATA\_LKG}$	Dataline Leakage Voltage				342	mV
I_{CHARGE}	Charge Pump Current Draw	I_{LOAD} = 8 mA		20	20	mA
		I_{LOAD} = 0 mA		0	1	mA
$I_{CHARGEB}$	Charge Pump Current Draw with Booster Active	I_{LOAD} = 8 mA		30	45	mA
		I_{LOAD} = 0 mA		0	5	mA
$I_{B\_DSCHG\_IN}$	B-Device (SRP Capable) Discharge Current	0V< V_{BUS} < 5.25V			8	mA
$V_{A\_VBUS\_VALID}$	A-Device VBUS Valid		4.4			V
$V_{A\_SESS\_VALID}$	A-Device Session Valid		0.8		2.0	V
$V_{B\_SESS\_VALID}$	B-Device Session Valid		0.8		4.0	V
$V_{A\_SESS\_END}$	B-Device Session End		0.2		0.8	V
E	Efficiency When Loaded	I_{LOAD} = 8 mA, V_{CC} = 3.3V		75		%
R_{PD}	Data Line Pull Down		14.25		24.8	Ω
$R_{A\_BUS\_IN}$	A-device V_{BUS} Input Impedance to GND	V_{BUS} is not being driven	40		100	kΩ
$R_{B\_SRP\_UP}$	B-device V_{BUS} SRP Pull Up	Pull up voltage = 3.0V	281			Ω
$R_{B\_SRP\_DWN}$	B-device V_{BUS} SRP Pull Down		656			Ω

USB Transceiver

USB 2.0 certified in full- and low-speed modes.

Notes
18. I_{CC} and I_{CCB} values are the same regardless of USB host or peripheral configuration.
19. There is no appreciable difference in I_{CC} and I_{CCB} values when only two transceivers are powered.

SRAM Write Cycle [26]

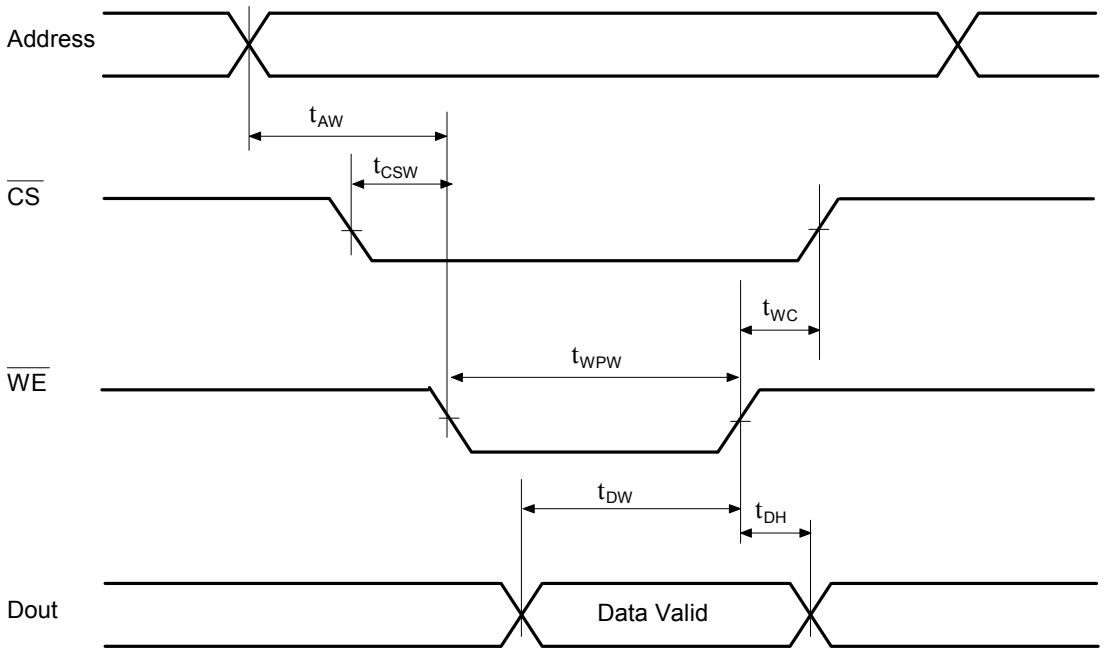


Table 138. SRAM Write Cycle Parameters

Parameter	Description	Min	Typical	Max	Unit
t_{AW}	Write Address Valid to $\overline{WE}$ LOW	7			ns
t_{CSW}	$\overline{CS}$ LOW to $\overline{WE}$ LOW	7			ns
t_{DW}	Data Valid to $\overline{WE}$ HIGH	15			ns
t_{WPW}[25]	$\overline{WE}$ Pulse Width	15			ns
t_{DH}	Data Hold from $\overline{WE}$ HIGH	4.5			ns
t_{WC}	$\overline{WE}$ HIGH to $\overline{CS}$ HIGH	13			ns

Notes
25. t_{WPW} The write pulse width = 18.8 ns min. for zero and one wait states. The write pulse = 18.8 ns + (n – 1)*T for wait states = n, n > 1, T = 48 MHz clock period.
26. Write timing is applicable for nXMEMSEL, nXRAMSEL and nXROMSEL.

9. BIOS USB Peripheral Mode: Descriptor Length

■ **Problem Definition**

The BIOS will not properly return a descriptor or set of descriptors, if the length is a multiple of the control endpoint's maximum packet size.

■ **Parameters Affected**

Control Endpoint maximum packet size.

■ **Trigger Condition(S)**

Get Descriptor requests.

■ **Scope of Impact**

If the descriptor length is a multiple of the maximum packet size, the BIOS will respond with a STALL instead of a zero-length data packet for the final IN request.

■ **Workaround**

If the requested descriptor length is a multiple of the maximum packet size, then either the maximum packet size or the descriptor length needs to change. A descriptor length can be increased by simply adding a padded byte to the end of a descriptor and increasing the descriptor Length byte by one. Section 9.5 (Descriptor) of the USB 2.0 specification allows a descriptor length to be larger than the value defined in the specification.

■ **Fix Status**

No silicon revision planned, use workaround.

12.Code fails to load from EEPROM

■ **Problem Definition**

If, while the BIOS is loading firmware, the part is reset and at that time the EEPROM is driving the SDA line low, the BIOS will configure the part for co-processor mode instead of standalone mode.

■ **Parameters Affected**

Initialization.

■ **Trigger Condition(S)**

Reset the part while firmware is being loaded from the EEPROM.

■ **Scope of Impact**

The firmware download process will not finish, leaving the part configured in co-processor mode.

■ **Workaround**

There is no workaround. Cycle power to the EEPROM to download firmware.

■ **Fix Status**

No silicon revision planned, use workaround.

13.ISOCH Endpoint Descriptor error

■ **Problem Definition**

Setting any bit other than 1:0 in the bmAttributes field will cause the ISOCH Endpoint Descriptors to be reported incorrectly.

■ **Parameters Affected**

Isochronous transfers.

■ **Trigger Condition(S)**

Setting any bit other than 1:0 in the bmAttributes field.

■ **Scope of Impact**

If the bmAttributes field in the Endpoint Descriptor is using bits 5...2, the BIOS will not correctly parse the endopoint and set up the part correctly for ISO transfers.

■ **Workaround**

There are two methods that can be used.

A.Mask these bits before the BIOS parses the descriptros using the SET_INTERFACE handler. An example of this is given in the “Using Multiple Interfaces to Implement a USB Isochronous Composite Peripheral with EZ-Host™ and EZ-OTG™.”

B.Replace the BIOS delta config interrupt and modify the USB-parse code to mask off all but the lower two bits of the bmAttribute. A possilble solution might look like this.

```
@@test b[r8+bEPAttribute], 0x01 ; check ISO
rz
test b[r8+bEPAttribute], 0x02
rnz
;if we get here, then the lower two bits
;of bEPAttribute = 01 meaning it is ISO or r2,EP_ISO
ret
```

■ **Fix Status**

No silicon revision planned, use workaround.

14.Missing Endpoint Interrupts in USB Peripheral Mode

■ **Problem Definition**

USB peripheral designs may miss endpoint interrupts when receiving Endpoint 0 (EP0) Control transfer requests mixed with other endpoint transfer type transactions.

■ **Parameters Affected**

All USB peripheral mode endpoint communication may potentially be affected.

■ **Trigger Condition(S)**

An endpoint interrupt may be missed when a non-EP0 transaction completes (with ACK) during an EP0 Control transfer or within ~200 μs before or after the EP0 Control transfer. Other processor activity and interrupts may influence the likelihood of this failure occurring.

■ **Scope of Impact**

This errata item applies to USB peripheral mode only. All USB peripheral designs that mix Standard, Vendor, or Class EP0 requests with other transfer types are potential candidates for this issue. If this problem occurs and an endpoint interrupt is missed, the endpoint will likely not be rearmed and therefore endpoint communication will be halted and the host system may reset the device.

■ **Workaround**

A. The PC Application and/or driver must be developed to ensure at least ~200 μs of idle time is given before and after any EP0 transfers. The driver must ensure that no other transfer type transactions occur during the EP0 transfer or during this idle time before and after the EP0 transfer. Therefore, the driver cannot submit asynchronous Control transfers. The driver must submit Control transfer requests synchronously with other transfer requests. In addition, the driver must be aware of any interrupt endpoint scheduling (which is under control of the host controller driver) when submitting Control transfers. This generally means that a vendor-specific driver is required.

B. USB endpoint communication stress testing of any USB peripheral designs that mix EP0 Control transfers with other transfer types is recommended.

■ **Fix Status**

No silicon revision planned, use workaround.