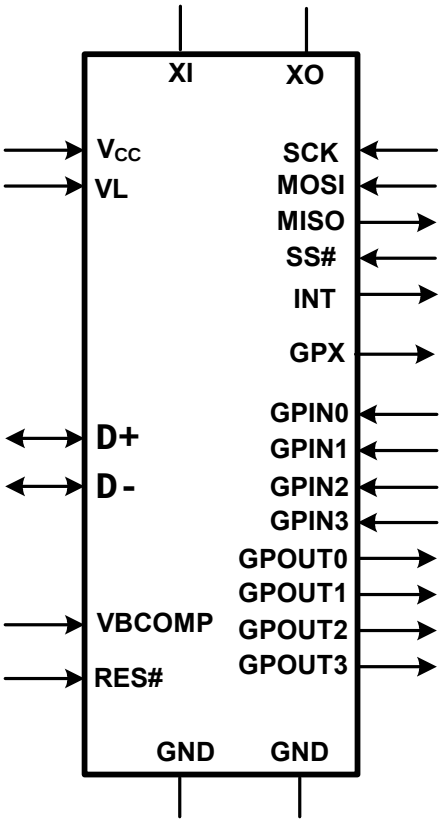
MAX3420E

USB Peripheral Controller with SPI Interface

Programming Guide



For more information on the MAX3420E, please visit http://www.maxim-ic.com/max3420e.
For more information on USB and Maxim's USB products, see http://www.maxim-ic.com/usb.
The Maxim logo is a registered trademark of Maxim Integrated Products, Inc.
The Dallas Semiconductor logo is a registered trademark of Dallas Semiconductor Corp.

Rev. Sept 28, 2005

Register Map

Bits are shown in normal font, registers are shown in italics.
Each highlighted cell is a link to a page describing the register or bit. Use the browser left arrow to return to this page. The Index at the end of this document is also linked to the descriptive pages.

Reg	Name	b7	b6	b5	b4	b3	b2	b1	b0	acc
R0	*EP0FIFO*	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R1	*EP1OUTFIFO*	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R2	*EP2INFIFO*	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R3	*EP3INFIFO*	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R4	*SUDFIFO*	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R5	*EP0BC*	0	b6	b5	b4	b3	b2	b1	b0	RSC
R6	*EP1OUTBC*	0	b6	b5	b4	b3	b2	b1	b0	RSC
R7	*EP2INBC*	0	b6	b5	b4	b3	b2	b1	b0	RSC
R8	*EP3INBC*	0	b6	b5	b4	b3	b2	b1	b0	RSC
R9	**EPSTALLS**	0	ACKSTAT	STLSTAT	STLEP3IN	STLEP2IN	STLEP1OUT	STLEP0OUT	STLEP0IN	RSC
R10	**CLRTOGS**	EP3DISAB	EP2DISAB	EP1DISAB	CTGEP3IN	CTGEP2IN	CTGEP1OUT	0	0	RSC
R11	**EPIRQ**	0	0	SUDAVIRQ	IN3BAVIRQ	IN2BAVIRQ	OUT1DAVIRQ	OUT0DAVIRQ	IN0BAVIRQ	RC
R12	**EPIEN**	0	0	SUDAVIE	IN3BAVIE	IN2BAVIE	OUT1DAVIE	OUT0DAVIE	IN0BAVIE	RSC
R13	**USBIRQ**	URESDNIRQ	VBUSIRQ	NOVBUSIRQ	SUSPIRQ	URESIRQ	BUSACTIRQ	RWUDNIRQ	OSCOKIRQ	RC
R14	**USBIEN**	URESDNIE	VBUSIE	NOVBUSIE	SUSPIE	URESIE	BUSACTIE	RWUDNIE	OSCOKIE	RSC
R15	**USBCTL**	HOSCSTEN	VBGATE	CHIPRES	PWRDOWN	CONNECT	SIGRWU	0	0	RSC
R16	**CPUCTL**	0	0	0	0	0	0	0	IE	RSC
R17	**PINCTL**	EP3INAK	EP2INAK	EP0INAK	FDUPSPI	INTLEVEL	POSINT	GPXB	GPXA	RSC
R18	**REVISION**	0	0	0	0	Rev3	Rev2	Rev1	Rev0	R
R19	*FNADDR*	0	b6	b5	b4	b3	b2	b1	b0	R
R20	**IOPINS**	GPIN3	GPIN2	GPIN1	GPIN0	GPOUT3	GPOUT2	GPOUT1	GPOUT0	RSC

Note: The acc (access) column indicates how the CPU can access the register. R=Read, RC=Read or Clear, RSC=Read, Set or Clear.

Accessing the MAX3420E Registers

An SPI™ master controls the MAX3420 by writing and reading twenty-one internal registers, R0-R20. The SPI master begins every register access by asserting the MAX3420E SS# (slave select, active low) pin, and clocking in eight bits that comprise the SPI command byte. Figure 1 shows the command byte format.

b7	b6	b5	b4	b3	b2	b1	b0
Reg4	Reg3	Reg2	Reg1	Reg0	0	DIR 1=wr 0=rd	ACKSTAT

Figure 1. SPI Command Byte. As for all SPI transfers, bit 7 is sent first.

Reg4:Reg0 set the register address, with valid values from 0-20. Values above 20 are ignored by the MAX3420E. The direction bit sets the direction for the data transfer. The ACKSTAT bit duplicates a USB control bit (R9 bit 6). ACKSTAT is provided in the control byte as a fast way to set this often-used register bit.

After sending the command byte, the SPI master transfers one or more bytes in the direction indicated by the DIR bit. Keeping SS# low, the SPI master provides additional bursts of eight SCLK pulses for each byte. When the byte transfers are complete, the SPI master de-asserts SS# (drives high) and the transfer terminates.

It is possible to truncate an SPI cycle after sending only the command byte. This feature allows the SPI master to set the ACKSTAT bit without doing a full SPI access.

Note: The ACKSTAT bit tells the MAX3420E that the SPI master has finished servicing a USB Control transfer. This causes the MAX3420E to ACKnowledge the next Control transfer STATUS stage.

To set the ACKSTAT bit using the SPI command byte, the SPI master sets the register field to a dummy value (it won't be used), sets the DIR bit for a read operation (for example, read Revision register R18), and sets the ACKSTAT bit. The SPI master then asserts SS#, clocks in the 8 command bits, and then truncates the SPI cycle by de-asserting the SS# signal. This is the fastest way to set the ACKSTAT bit.

The MAX3420E has two register types, FIFOS and control registers. Repeated reads or writes to a register has different effects, depending on the register type.

Registers R0-R4 access internal FIFOS. After selecting the register number R0-R4 with the command byte, the SPI master loads or unloads consecutive FIFO bytes by repeating reads or writes during the SPI transfer. For example, to read 8 bytes from the SUDFIFO, the SPI master would perform the following steps:

1. Set SS#=0.
2. Send 00100000. This command byte selects R4 (SUDFIFO), for a read operation (DIR=0).
3. Issue eight SCLK pulses, clock in a data byte, one bit per SCLK rising edge.
4. Repeat step 3 seven more times, clocking in and storing a total of 8 bytes.
5. Set SS#=1.

Registers R5-R20 are control registers. If the SPI master repeatedly reads or writes R5-R20 during the same SPI transfer (SS# low), every byte read or write automatically increments the register address. This allows reading or writing registers in consecutive groups without writing a new command byte to set each new register address. The register addresses continue to increment until the last register, R20 is reached, where the register address “sticks” at R20. This feature gives the uP a quick access method for the IO pins in R20. For example, to output a pre-stored waveform on a GPIO pin, the SPI master can write the command byte 10100010 (R20, Write) and then send multiple data bytes to R20 output the waveform.

Meaning: Acknowledge the STATUS stage of a CONTROL transfer.

Location: EPSTALLS.6

Set: The CPU sets this bit after it has finished servicing a CONTROL transfer request. This instructs the SIE to send the ACK handshake to the status stage of the current CONTROL transfer. Until the CPU sets this bit, the SIE responds to the status stage of the CONTROL transfer with a NAK handshake.

Clear: The SIE clears this bit whenever a SETUP token arrives.

POR: ACKSTAT=0
Chip Reset: ACKSTAT=0
Bus Reset: ACKSTAT=0
Pwr Down: Read-only

FYI: A fast way to set the ACKSTAT register bit is to set bit 0 of the SPI command byte. All Maxim example code uses this method.

Programming Notes:

When the CPU receives a Setup Data Available Interrupt Request (SUDAVIRQ bit, page 66), it clears the SUDAVIRQ bit by writing "1" to it, and then reads the eight data bytes from the SUDFIFO into memory. The CPU then inspects the eight bytes to determine the nature of the USB request. If the request is in error or unknown, the CPU sets the STLSTAT bit to answer the status stage with a STALL handshake (page 64).

If the CPU recognizes the request, it services the request, and when finished sets ACKSTAT=1 to tell the SIE to send the ACK handshake to the status stage to terminate the CONTROL transfer. Until the CPU either acknowledges or stalls the transfer, the SIE automatically returns the NAK handshake to the CONTROL transfer status stage.

A C program that interprets the 8 bytes of setup data usually consists of one or more **case** statements that check for all the legal combination of bytes in the setup packet. A convenient way to handle the STALL is to make the default case a statement that stalls the CONTROL transfer. (see the STLSTAT bit, page 64).

BUSACTIE

Meaning: Bus Active Interrupt Enable.

Location: USBIEN.2

Set: The CPU sets this bit to enable the BUSACT IRQ (page 2).

Clear: The CPU clears this bit to disable the BUSACT IRQ.

POR: BUSACTIE=0
Chip Reset: BUSACTIE=0
Bus Reset: BUSACTIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that turns on the interrupt enable bits should be called as part of servicing a USB bus reset.

Meaning: Bus Active Interrupt Request.

Location: USBIRQ.2

Set: The SIE sets this bit to indicate USB bus activity. An internal BUSACT signal is set when the SIE receives a SYNC field, and reset after 32 bit time of a J-state, or during a USB bus reset. The BUSACTIRQ bit is set when the internal BUSACT signal makes a 0-1 transition.

Clear: The CPU clears this bit by writing a "1" to it.

POR: BUSACTIRQ=0
Chip Reset: BUSACTIRQ=0
Bus Reset: BUSACTIRQ=0
Pwr Down: Read-only

CHIPRES

Meaning: Chip Reset.

Location: USBCTL.5

Set: The CPU sets this bit to reset the chip. Its effect is identical to driving the RES# pin low.

Clear: The CPU clears this bit to take the chip out of reset.

POR: CHIPRES=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

The CPU can clear this bit immediately after setting it.

Meaning: Connect to USB.

Location: USBCTL.3

Set: The CPU sets this bit to connect an internal 1500 Ohm resistor between the DPLUS line and V_{CC}.

Clear: The CPU clears this bit to disconnect an internal 1500 Ohm resistor between the DPLUS line and V_{CC}.

POR: CONNECT=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

The operation of the CONNECT bit depends on the setting of the VBGATE bit (page 76). If CONNECT=1 and VBGATE=1, internal logic will not connect the pullup resistor unless V_{BUS} is detected to be valid on the V_{BUS} pin. If VBGATE=0 the DPLUS pullup resistor is unconditionally connected when CONNECT=1.

Only a power-on reset clears the CONNECT bit. If, during operation, an external reset is applied to the MAX3420E via the INT pin, the CONNECT bit retains its state. This means that if a device is connected to USB (CONNECT=1) when the RES# is asserted, it remains connected throughout the reset and after the reset is removed (RES#=1).

CTGEP1OUT

Meaning: Clear Data Toggle for endpoint 1 OUT.

Location: CLRTOGS.2

Set: The CPU sets this bit to clear the data toggle for EP1-OUT to the DATA0 state.

Clear: The SIE automatically clears this bit.

POR: CTGEP1OUT=0
Chip Reset: CTGEP1OUT=0
Bus Reset: CTGEP1OUT=0
Pwr Down: Read-only

Programming Notes:

The SIE automatically clears all data toggles during a chip or USB bus reset. The CPU normally needs to clear an individual endpoint data toggle under two conditions:

- The host issues a Set_Configuration request.
- The host issues a Clear_Feature (endpoint stall) request.

Meaning: Clear Data Toggle for Endpoint 2 IN.

Location: CLRTOGS.3

Set: The CPU sets this bit to clear the data toggle for EP2-IN to the DATA0 state.

Clear: The SIE automatically clears this bit.

POR: CTGEP2IN=0
Chip Reset: CTGEP2IN=0
Bus Reset: CTGEP2IN=0
Pwr Down: Read-only

Programming Notes:

The SIE automatically clears all data toggles during a chip or USB bus reset. The CPU normally needs to clear an individual endpoint data toggle under two conditions:

- The host issues a Set_Configuration request.
- The host issues a Clear_Feature (endpoint stall) request.

CTGEP3IN

Meaning: Clear Data Toggle for endpoint 3 IN.

Location: CLRTOGS.4

Set: The CPU sets this bit to clear the data toggle for EP3-IN to the DATA0 state.

Clear: The SIE automatically clears this bit.

POR: CTGEP3IN=0
Chip Reset: CTGEP3IN=0
Bus Reset: CTGEP3IN=0
Pwr Down: Read-only

Programming Notes:

The SIE automatically clears all data toggles during a chip or USB bus reset. The CPU normally needs to clear an individual endpoint data toggle under two conditions:

- The host issues a Set_Configuration request.
- The host issues a Clear_Feature (endpoint stall) request.

Meaning: Endpoint 0 Byte Count Register. Since EP0 is a bi-directional endpoint, whereby both IN and OUT transfers share the same FIFO (EP0FIFO, page 10), the action of this register depends on the transfer direction.

Location: EP0BC[6:0]

Write (IN): For an IN transfer, the CPU writes the byte count to this register after loading the EP0FIFO with data. Valid values are 0-64. When the CPU writes this register the SIE arms the endpoint so that it returns a data packet instead of a NAK to the next IN request to the endpoint.

Read (OUT): For an OUT transfer, the SIE loads the byte count to indicate the number of bytes received in an OUT data transfer. When the OUT transfer is successful, the SIE ACKS the transfer, updates the byte count register, and asserts the OUT0DAV interrupt request bit (page 51).

POR: EP0BC=0
Chip Reset: EP0BC=0
Bus Reset: EP0BC=0
Pwr Down: No read or write

Programming Notes:

Bit 7 has no effect and reads as a 0.
The CPU writes the EP0FIFO as the response to the data stage of a CONTROL transfer.
The CPU reads the EP0FIFO to retrieve the data stage of a CONTROL transfer.

EP0FIFO

Meaning: Endpoint 0 FIFO. This 64 byte FIFO is used for OUT and IN transfers to and from the bi-directional endpoint 0.

Location: EP0FIFO[7:0]

Write (IN): For an IN transfer, the CPU writes a series of bytes to this FIFO to fill it with IN data. After filling the FIFO with a packet (0 to 64 bytes), the CPU writes the byte count register (page 9) to arm the IN transfer and to tell the SIE how many bytes to transfer when it receives the IN packet to endpoint 0.

Read (OUT): For an OUT transfer, the SIE fills the FIFO with USB data received from the host. When the OUT transfer is verified to be error-free, the SIE loads the byte count register (page 9) to indicate the number of bytes received in the OUT data transfer. For a successful transfer the SIE also ACKS the OUT transfer and asserts the OUT0DAV interrupt request bit (page 51).

POR: EP0FIFO[7:0]=0
Chip Reset: EP0FIFO[7:0]=0
Bus Reset: *Unchanged*
Pwr Down: No read or write

Programming Notes:

The SIE automatically retries packets that it finds to contain errors (CRC, bit stuff, etc.). This is invisible to the CPU—no interrupt flags or registers are updated with less than perfect transfers.

Meaning: EP0-IN NAK.

Location: PINCTL.5

Set: The SIE sets this bit when the EP0-IN endpoint receives an IN request and returns the NAK handshake.

Clear: The CPU clears this bit by writing a 1 to it.

POR: EP0IBN=0
Chip Reset: EP0IBN=0
Bus Reset: EP0IBN=0
Pwr Down: Read-write

Programming Notes:

This bit may be polled to discover that the host is asking for IN data which is not yet available from an IN endpoint because the CPU has not yet loaded and armed the endpoint. This bit is not included in the interrupt system.

Note: The EP0INAK bit is informational only. It is normally not used by USB device firmware.

EP1DISAB

Meaning: Disable Endpoint 1-OUT.

Location: CLRTOGS.5

Set: The CPU sets this bit to disable traffic to Endpoint 1-OUT.

Clear: The CPU clears this bit to enable traffic to Endpoint 1-OUT.

POR: EP1DISAB=0
Chip Reset: EP1DISAB=0
Bus Reset: EP1DISAB=0
Pwr Down: Read-only

Programming Notes:

A disabled endpoint does not respond to any traffic. A host normally will never send traffic to an endpoint that is not reported during enumeration, so this is a “safety” bit to guard against an errant host.

Endpoint 0 has no disable bit because as the default CONTROL endpoint it must always be active.

Meaning: Endpoint 1-0UT Byte Count.

Location: EP1OUTBC[6:0]

Write: After successfully receiving an OUT transfer over Endpoint 1, the SIE ACKS the transfer, updates this register with the received byte count, and asserts the OUT1DAV interrupt request (page 51).

Read: The CPU reads this register after receiving an OUT1DAV interrupt request to determine how many bytes to read from the EP1OUTFIFO (page 14).

POR: EP1OUTBC=0
Chip Reset: EP1OUTBC=0
Bus Reset: EP1OUTBC=0
Pwr Down: No read or write

Programming Notes:

EP1OUT is a double-buffered endpoint, meaning that there are two FIFOS and byte count registers. Double buffering allows USB data simultaneously to move into one FIFO while the CPU reads data from the other. This improves bandwidth performance in many systems. See the OUT1DAVIRQ bit discussion (page 51) for a description of how the double buffering works for an OUT endpoint.

The double buffering is invisible to the programmer because the OUT1DAVIRQ flag logic accommodates the double buffering. For example, assume that both buffers are available and therefore OUT1DAVIRQ=0. When an OUT packet arrives, OUT1DAVIRQ makes a 0-1 transition to indicate availability of the first packet. For a single-buffered endpoint, if another OUT packet arrived over EP1-OUT before the CPU had time to drain the FIFO, the SIE would respond with a NAK handshake to indicate that the endpoint was not available to accept data.

However with the double buffered endpoint, the second OUT packet is accepted and ACK'd because the second buffer is available for data. If a third OUT packet arrives before either FIFO is drained, the SIE NAKS the transfer to indicate that both FIFOS are full.

EP1OUTFIFO

Meaning: Double-buffered 64 byte FIFO for Endpoint 1-OUT.

Location: EP1OUTFIFO[7:0]

Write: The SIE fills the OUT FIFO with bytes transmitted from the host to endpoint 1-OUT. After successfully receiving the OUT transfer, the SIE ACKS the transfer, updates the byte count register (page 13), and asserts the OUT1DAV interrupt request (page 51).

Read: When the CPU receives an OUT1DAV interrupt request, it reads the byte count register to determine how many bytes are in the FIFO, and then reads that number of bytes from this register.

POR: EP1OUTFIFO=0
Chip Reset: EP1OUTFIFO=0
Bus Reset: *Unchanged*
Pwr Down: No read or write

Programming Notes:

EP1OUT is a double-buffered endpoint, meaning that there are two FIFOS and byte count registers. Double buffering allows USB data simultaneously to move into one FIFO while the CPU reads data from the other. This improves bandwidth performance in many systems. See the OUT1DAVIRQ bit discussion (page 53) for a description of how the double buffering works for an OUT endpoint.

The double buffering is invisible to the programmer because the OUT1DAVIRQ flag logic accommodates the double buffering. For example, assume that both buffers are available and therefore OUT1DAVIRQ=0. When an OUT packet arrives, OUT1DAVIRQ makes a 0-1 transition to indicate availability of the first packet. For a single-buffered endpoint, if another OUT packet arrived over EP1-OUT before the CPU had time to drain the FIFO, the SIE would respond with a NAK handshake to indicate that the endpoint was not available to accept data.

However with the double buffered endpoint, the second OUT packet is accepted and ACK'd because the second buffer is available for data. If a third OUT packet arrives before either FIFO is drained, the SIE NAKS the transfer to indicate that both FIFOS are full.

Meaning: Disable Endpoint 2.

Location: CLRTOGS.6

Set: The CPU sets this bit to disable traffic to endpoint 2-IN.

Clear: The CPU sets this bit to enable traffic to endpoint 2-IN.

POR: EP2DISAB=0
Chip Reset: EP2DISAB=0
Bus Reset: EP2DISAB=0
Pwr Down: Read-only

Programming Notes:

A disabled endpoint does not respond to any traffic. A host normally will never send traffic to an endpoint that is not reported during enumeration, so this is a “safety” bit to guard against an errant host.

Endpoint 0 has no disable bit because as the default CONTROL endpoint it must always be active.

EP2INAK

Meaning: Endpoint 2-IN NAK

Location: PINCTL.6

Set: The SIE sets this bit when the EP2-IN endpoint receives an IN request and the SIE returns the NAK handshake.

Clear: The CPU clears this bit by writing a 1 to it.

POR: EP2INAK=0
Chip Reset: EP2INAK=0
Bus Reset: EP2INAK=0
Pwr Down: Read-write

Programming Notes:

This bit may be polled to discover that the host is asking for IN data which is not yet available from an IN endpoint because the CPU has not yet loaded and armed the endpoint. This bit is not included in the interrupt system.

Note: The EP2INAK bit is informational only. It is normally not used by USB device firmware.

Meaning: Endpoint 2-IN Byte Count Register.

Location: EP2INBC[6:0]

Write: The CPU loads this register with the number of bytes it has loaded into the EP2INFIFO (page 18). This arms the endpoint for the next IN transfer.

Read: The SIE sends the data in this FIFO as the response to an EP2-IN host request.

POR: EP2INBC=0
Chip Reset: EP2INBC=0
Bus Reset: EP2INBC=0
Pwr Down: Read-only

EP2INFIFO

Meaning: Endpoint 2-IN FIFO (double-buffered 64 byte FIFOS).

Location: EP2INFIFO[7:0]

Write: The CPU loads bytes into this FIFO in preparation for sending to the host.

Read: The SIE sends these bytes over USB in response to an IN request to EP2-IN.

POR: EP2INFIFO=0
Chip Reset: EP2INFIFO=0
Bus Reset: *Unchanged*
Pwr Down: No read or write

Meaning: Disable Endpoint 3.

Location: CLRTOGS.7

Set: The CPU sets this bit to disable traffic to endpoint 3-IN.

Clear: The CPU clears this bit to enable traffic to endpoint 3-IN.

POR: EP3DISAB=0
Chip Reset: EP3DISAB=0
Bus Reset: EP3DISAB=0
Pwr Down: Read-only

Programming Notes:

A disabled endpoint does not respond to any traffic. A host normally will never send traffic to an endpoint that is not reported during enumeration, so this is a "safety" bit to guard against an errant host.

Endpoint 0 has no disable bit because as the default CONTROL endpoint it must always be active.

EP3INAK

Meaning: Endpoint 3-IN NAK

Location: PINCTL.7

Set: The SIE sets this bit when the EP3-IN endpoint receives an IN request and the SIE returns the NAK handshake.

Clear: The CPU clears this bit by writing a 1 to it.

POR: EP3INAK=0
Chip Reset: EP3INAK=0
Bus Reset: EP3INAK=0
Pwr Down: Read-write

Programming Notes:

This bit may be polled to discover that the host is asking for IN data which is not yet available from an IN endpoint because the CPU has not yet loaded and armed the endpoint. This bit is not included in the interrupt system.

Note: The EP0INAK bit is informational only. It is normally not used by USB device firmware.

Meaning: Endpoint 3-IN Byte Count Register.

Location: EP3INBC[6:0]

Write: The CPU loads this register with the number of bytes it has previously loaded into the EP3INFIFO (page 22). This arms the endpoint for the next IN transfer.

Read: The SIE sends the data in this FIFO as the response to an EP3-IN host request.

POR: EP3INBC=0
Chip Reset: EP3INBC=0
Bus Reset: EP3INBC=0
Pwr Down: Read-only

Programming Notes:

EP3INFIFO

Meaning: Endpoint 3-IN FIFO, a 64-byte FIFO.

Location: EP3INFIFO[7:0]

Write: The CPU loads bytes into this FIFO in preparation for sending to the host.

Read: The SIE sends these bytes over USB in response to an IN request to EP3-IN.

POR: EP3INFIFO=0
Chip Reset: EP3INFIFO=0
Bus Reset: *Unchanged*
Pwr Down: No read or write

Meaning: Full Duplex SPI port operation.

Location: PINCTL.4

Set: The CPU sets this bit to operate the SPI port in full-duplex mode.

Clear: The SIE clears this bit to operate the SPI port in half-duplex mode.

POR: FDUPSPI=0 (half-duplex)
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

Half-duplex SPI

In half-duplex mode (FDUPSPI=0), the MOSI (Master Out, Slave In) pin becomes a bi-directional IO pin, and the MISO (Master In, Slave Out) pin is tri-stated.

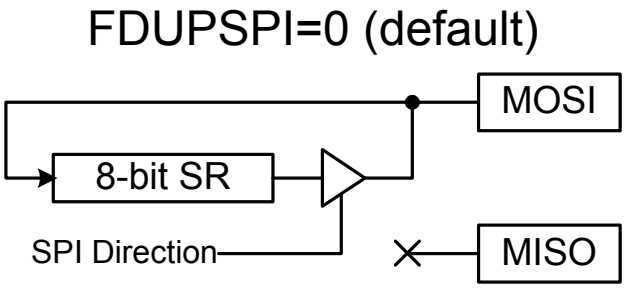


Figure 2. Half-duplex SPI interface.

Full-duplex SPI

Full-duplex mode (FDUPSPI=1) provides separate MOSI and MISO pins. This configuration has the added feature that while the first byte of every transfer (the command byte) is clocked in, eight bits of status bits are simultaneously clocked out, as shown in Figure 3.

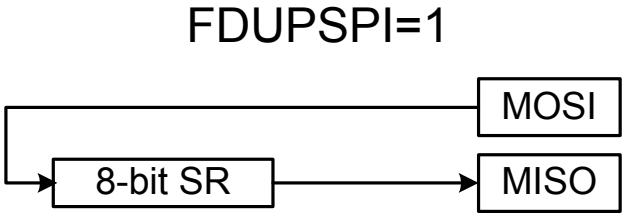


Figure 3. Full-duplex SPI interface.

MISO bit	Signal	page
7	SUSPIRQ	**69**
6	URESIRQ	**73**
5	SUDAVIRQ	**66**
4	IN3BAVIRQ	**44**
3	IN2BAVIRQ	**42**
2	OUT1DAVIRQ	**53**
1	OUT0DAVIRQ	**51**
0	IN0BAVIRQ	**40**

Figure 4. Full-duplex mode: These USB status bits are clocked out while the command byte is clocked in.

The SPI Command Byte

In either SPI mode, the first byte clocked into the SPI interface is a command byte that sets the register address, the direction, and a bit that directly sets the ACKSTAT bit. In all SPI transactions, in or out, the bit ordering is b7 first, b0 last.

MOSI bit	Signal
7	REG4
6	REG3
5	REG2
4	REG1
3	REG0
2	0
1	Direction (1=Wr, 0=Rd)
0	ACKSTAT (page 1)

Figure 5. The SPI command byte.

An SPI cycle starts with the SPI master driving CS# low, then driving eight SPI clocks whose rising edges strobe in the data shown in the Figure 5 command byte. REG[4:0] set the register address, and the direction bit sets the read or write direction for the SPI cycle. ACKSTAT writes the corresponding bit in the EPSTALLS register. If FDUPSPI=1, the data shown in Figure 4 is simultaneously clocked out on the MISO pin during these first 8 SCLK clocks.

Following the command byte, the SPI master issues one or more groups of 8-SCLK clocks to clock byte data into or out of the MAX3420E. As long as CS# remains low, the register address clocked in with the command remains in effect. This ability to burst bytes is convenient when reading or writing the endpoint FIFOS. For example, to load 37 bytes into the EP0FIFO, the SPI master writes the command byte 00000010 which selects R0 (EP0FIFO) for a write operation (direction bit is 1). Then it writes 37 bytes to the SPI port, and finally drives CS# high to complete the SPI cycle.

Note: Both MOSI and MISO data are sampled on the rising edge of SCLK. Data changes on the falling edge of SCLK.

The SPI cycle terminates when the SPI master return CS# to its high state.

SPI Modes

The SPI standard defines four clocking modes, reflecting two mode signals called CPOL (clock polarity) and CPHA (clock phase). These signals are represented in the form (CPOL, CPHA). It turns out that an interface that expects positive edge SCLKS and which also expects the MOSI data to be available before the first positive clock edge can operate in modes (0,0) and (1,1) without alteration. This property allows the MAX3420E to operate in mode (0,0) or (1,1) without requiring a mode pin.

The scope traces below illustrate identical data transfers between a microprocessor and the MAX3420E. Figure 6 uses SPI mode (0,0) and Figure 7 uses SPI mode (1,1). The difference is the inactive level of the SCLK signal, low for mode (0,0) and high for mode (1,1). In both modes the MOSI and MISO data as sampled by the rising edge of SCLK is the same.

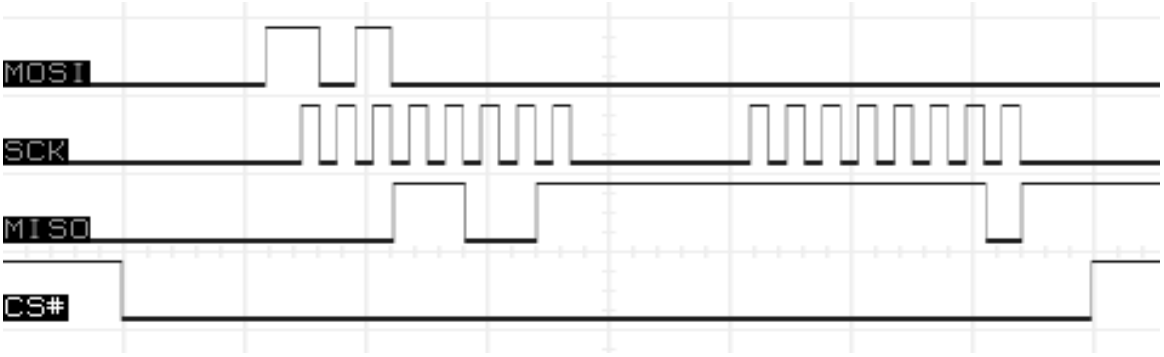


Figure 6. SPI interface operating in mode (0,0).

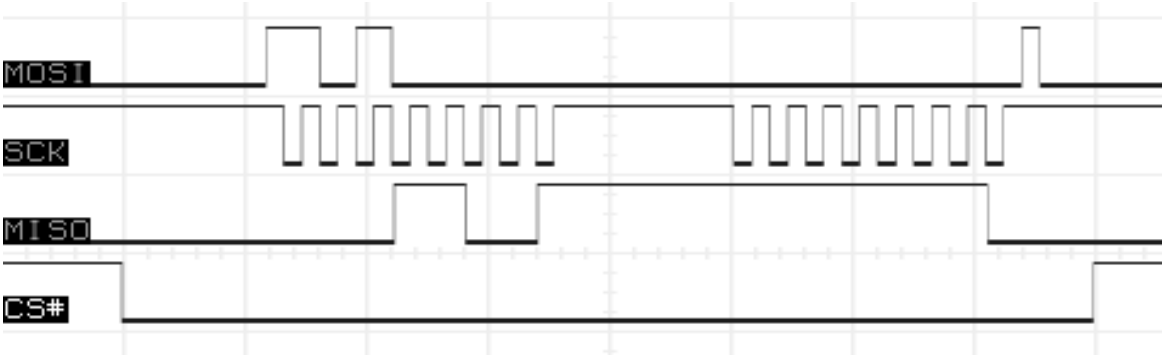


Figure 7. SPI interface operating in mode (1,1).

FNADDR

Meaning: The Function Address assigned to the USB peripheral by the host.

Location: FNADDR[6:0]

Set: The SIE updates this register after receiving the ACK handshake at the conclusion of a Set_Address request from the host

Clear: The SIE clears this register during a chip reset or a USB bus reset.

POR: FNADDR=0
Chip Reset: FNADDR=0
Bus Reset: FNADDR=0

Pwr Down: Read-only

Programming Notes:

CPU writes to this register have no effect.

Meaning: General Purpose Input Pin 0.

Location: IOPINS.4

Set: This pin indicates the state of the GPIN0 pin (referenced to V_L). This pin is pulled high with a weak pullup resistor, so if nothing is connected to the pin it indicates logic 1.

Clear: Writing this bit has no effect.

POR: n/a
Chip Reset: n/a
Bus Reset: n/a
Pwr Down: Read-only

GPIN1

Meaning: General Purpose Input Pin 1.

Location: IOPINS.5

Set: This pin indicates the state of the GPIN1 pin (referenced to V_L). This pin is pulled high with a weak pullup resistor, so if nothing is connected to the pin it indicates logic 1.

Clear: Writing this bit has no effect.

POR: n/a
Chip Reset: n/a
Bus Reset: n/a
Pwr Down: Read-only

Meaning: General Purpose Input Pin 2.

Location: IOPINS.6

Set: This pin indicates the state of the GPIN2 pin (referenced to V_L). This pin is pulled high with a weak pullup resistor, so if nothing is connected to the pin it indicates logic 1.

Clear: Writing this bit has no effect.

POR: n/a
Chip Reset: n/a
Bus Reset: n/a
Pwr Down: Read-only

GPIN3

Meaning: General Purpose Input Pin 3.

Location: IOPINS.7

Set: This pin indicates the state of the GPIN3 pin (referenced to V_L). This pin is pulled high with a weak pullup resistor, so if nothing is connected to the pin it indicates logic 1.

Clear: Writing this bit has no effect.

POR: n/a
Chip Reset: n/a
Bus Reset: n/a
Pwr Down: Read-only

Meaning: General Purpose Output Pin 0.

Location: IOPINS.0

Set: The CPU sets this bit to set the GPOUT0 pin high (referenced toV_L). The CPU can also read this bit. Reading the bit indicates the state of the output flipflop before the output buffer. Therefore if the output pin is driving a large load (e.g. an LED) that compromises the logic level, the CPU can still read the correct logic state of the output pin.

Clear: The CPU clears this bit to set the GPOUT0 pin state to 0.

POR: GPOUT0=0
Chip Reset: No change
Bus Reset: No change
Pwr Down: Read-write

GPOUT1

Meaning: General Purpose Output Pin 1.

Location: IOPINS.1

Set: The CPU sets this bit to set the GPOUT1 pin high (referenced toV_L). The CPU can also read this bit. Reading the bit indicates the state of the output flipflop before the output buffer. Therefore if the output pin is driving a large load (e.g. an LED) that compromises the logic level, the CPU can still read the correct logic state of the output pin.

Clear: The CPU clears this bit to set the GPOUT1 pin state to 0.

POR: GPOUT1=0
Chip Reset: No change
Bus Reset: No change
Pwr Down: Read-write

Meaning: General Purpose Output Pin 2.

Location: IOPINS.2

Set: The CPU sets this bit to set the GPOUT2 pin high (referenced toV_L). The CPU can also read this bit. Reading the bit indicates the state of the output flipflop before the output buffer. Therefore if the output pin is driving a large load (e.g. an LED) that compromises the logic level, the CPU can still read the correct logic state of the output pin.

Clear: The CPU clears this bit to set the GPOUT2 pin state to 0.

POR: GPOUT2=0
Chip Reset: No change
Bus Reset: No change
Pwr Down: Read-write

GPOUT3

Meaning: General Purpose Output Pin 3.

Location: IOPINS.3

Set: The CPU sets this bit to set the GPOUT3 pin high (referenced toV_L). The CPU can also read this bit. Reading the bit indicates the state of the output flipflop before the output buffer. Therefore if the output pin is driving a large load (e.g. an LED) that compromises the logic level, the CPU can still read the correct logic state of the output pin.

Clear: The CPU clears this bit to set the GPOUT3 pin state to 0.

POR: GPOUT3=0
Chip Reset: No change
Bus Reset: No change
Pwr Down: Read-write

Meaning: The two bits GPXB:GPXA determine the output of the GPX pin.

Location: PINCTL.0

Set: The CPU sets this bit.

Clear: The CPU clears this bit.

POR: GPXA=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

GPXB	GPXA	GPX Pin
0	0	OPERATE (complement of internal POR)
0	1	VBUS Detect
1	0	BUSACT
1	1	SOF (0-1 transition when SOF packet arrives, 50% duty cycle signal)

Meaning: The two bits GPXB:GPXA determine the output of the GPX pin.

Location: PINCTL.1

Set: The CPU sets this bit.

Clear: The CPU clears this bit.

POR: GPXB=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

GPXB	GPXA	GPX Pin
0	0	OPERATE (complement of internal POR)
0	1	VBUS Detect
1	0	BUSACT
1	1	SOF (0-1 transition when SOF packet arrives, 50% duty cycle signal)

Meaning: Host Oscillator Start Enable. This applies when the MAX3420E is in the power-down mode (PWRDOWN=1).

Location: USBCTL.7

Set: The CPU sets this bit to start the on-chip oscillator when the USB DPLUS signal makes a 1-0 transition (host resume signaling).

Clear: The CPU clears this bit to keep the chip in the low power state (inhibit the oscillator from starting) when the USB host resumes bus signaling.

POR: HOSCSTEN=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

Once the CPU has put the MAX3420E into the power-down state (PWRDOWN=1, page 55), there are three ways to restart the MAX3420E oscillator:

1. The SPI master sets PWRDOWN=0 (this is also achieved by a chip reset).
2. The SPI master sets SIGRESUME=1 (page 58).
3. The USB host resumes bus activity. The MAX3420E detects this as a low level on the DPLUS data line while in low power mode (oscillator is off).

The HOSCSTEN bit deals with a special case of item 3, when the MAX3420E is designed into a self-powered peripheral. Suppose a user plugs a self-powered peripheral which is in power-down mode into a PC that is turned off. The logic low on DPLUS (due to the DPLUS pulldown resistor in the host PC's root hub) looks just like the USB RESUME signal, which would normally restart the oscillator and wake up the chip. But in this situation the chip should not power on or power its DPLUS resistor until the PC turns on V_{BUS} to activate its USB ports.

Therefore in a self-powered system, to enter USB suspend the CPU should set PWRDOWN=1 and HOSCSTEN=0. This enters the power-down state but inhibits the 1-0 transition on the DPLUS line as a way to wake up the chip.

An easy way for a self-powered peripheral to detect that the host PC is powered and has turned on VBUS is to set the GPXB-A pins (page 36) to 01. This connects the V_{BUS} comparator to the GPX output pin. The GPX pin can then serve as a direct V_{BUS} detect pin for the CPU. While disconnected, the CPU should set CONNECT=0 (page 5), and then set it to 1 only after it senses a valid USB connection.

IE

Meaning: Interrupt Enable.

Location: CPUCTL.0

Set: The CPU sets this bit to activate the INT output pin. The characteristics of the INT output pin are programmed by the INTLEVEL (page 45) and POSINT (page 54) bits.

Clear: The CPU clears this bit to disable the INT output pin. When IE=0 the state of the INT# pin is inactive (open for level mode, high for negative edge, low for positive edge).

POR: IE=0
Chip Reset: IE=0
Bus Reset: *No change*
Pwr Down: Read-only

Meaning: Endpoint 0 IN Buffer Available Interrupt Enable.

Location: EPIEN.0

Set: The CPU sets this bit to enable the IN0BAV interrupt request (page 40).

Clear: The CPU clears this bit to disable the IN0BAV interrupt request.

POR: IN0BAVIE=0
Chip Reset: IN0BAVIE=0
Bus Reset: IN0BAVIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that sets upthe interrupt enables should be called as part of servicing a USB bus reset.

IN0BAVIRQ

Meaning: Endpoint 0 IN Buffer Available Interrupt Request.

Location: EPIRQ.0

Set: The SIE sets this bit after receiving an IN token directed to Endpoint 0, sending the data in the EP0FIFO (page 10) and receiving the ACK handshake from the host. This indicates that the EP0FIFO is again available for loading by the CPU.

Clear: The CPU resets this bit by writing a "1" to it, or by writing the byte count register EP0BC (page 9).

POR: IN0BAVIRQ=1
Chip Reset: IN0BAVIRQ=1
Bus Reset: IN0BAVIRQ=1
Pwr Down: Read-only

Meaning: Endpoint 2 IN Buffer Available Interrupt Enable.

Location: EPIEN.3

Set: The CPU sets this bit to enable the IN2BAV interrupt request (page 42).

Clear: The CPU clears this bit to disable the IN2BAV interrupt request.

POR: IN2BAVIE=0
Chip Reset: IN2BAVIE=0
Bus Reset: IN2BAVIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that sets up the interrupt enables should be called as part of servicing a USB bus reset.

IN2BAVIRQ

Meaning: Endpoint 2-IN Buffer Available Interrupt Request.

Location: EPIRQ.3

Set: The SIE sets this bit after receiving an IN token directed to Endpoint 2, sending the data in the EP2INFIFO (page 18) and receiving the ACK handshake from the host. This indicates that the EP2INFIFO is again available for loading by the CPU.

Clear: The CPU resets this bit by writing the byte count register EP2INBC (page17).

.

POR: IN2BAVIRQ=1
Chip Reset: IN2BAVIRQ=1
Bus Reset: IN2BAVIRQ=1
Pwr Down: Read-only

Programming Notes:

EP2IN is a double-buffered endpoint, meaning that it uses two FIFOS and byte count registers. Double buffering allows USB data simultaneously to move out of one IN FIFO while the CPU loads data into the other. This improves bandwidth performance in many systems.

The IN2BAVIRQ flag logic makes the double buffering invisible to the programmer. For example, assume that both buffers are available and therefore IN2BAVIRQ=1. The CPU writes N bytes to the EP2INFIFO, and then arms the transfer by writing N to the EP2INBC register. For a single buffered IN endpoint, nothing would happen until the host sent the IN token to the endpoint and accepted the IN data packet. But because the second buffer is available for CPU loading the IN2BAVIRQ bit goes invalid and immediately goes valid again, just as if the first packet had already been sent and accepted by the host.

The double buffering action also means that after the MAX3420E is reset, the IN2BAVIRQ bit remains set until the EP2INBC register is loaded twice.

Meaning: Endpoint 3 IN Buffer Available Interrupt Enable.

Location: EPIEN.4

Set: The CPU sets this bit to enable the IN3BAV interrupt request (page 44).

Clear: The CPU clears this bit to disable the IN3BAV interrupt request.

POR: IN3BAVIE=0
Chip Reset: IN3BAVIE=0
Bus Reset: IN3BAVIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that sets up the interrupt enables should be called as part of servicing a USB bus reset.

IN3BAVIRQ

Meaning: EP3-IN Buffer Available Interrupt Request.

Location: EPIRQ.4

Set: The SIE sets this bit after receiving an IN token directed to Endpoint 3, sending the data in the EP3INFIFO (page 22) and receiving the ACK handshake from the host. This indicates that the EP3INFIFO is again available for loading by the CPU.

Clear: The CPU clears this bit by writing the byte count register EP3INBC (page 21).

.

POR: IN3BAVIRQ=1
Chip Reset: IN3BAVIRQ=1
Bus Reset: IN3BAVIRQ=1
Pwr Down: Read-only

Programming Notes:

At power-on or reset, the IN FIFOS are available to accept CPU data and therefore indicate 1. IN FIFO Buffer Available bits are the only bits that default to 1.

Meaning: The INT output pin is level-active.

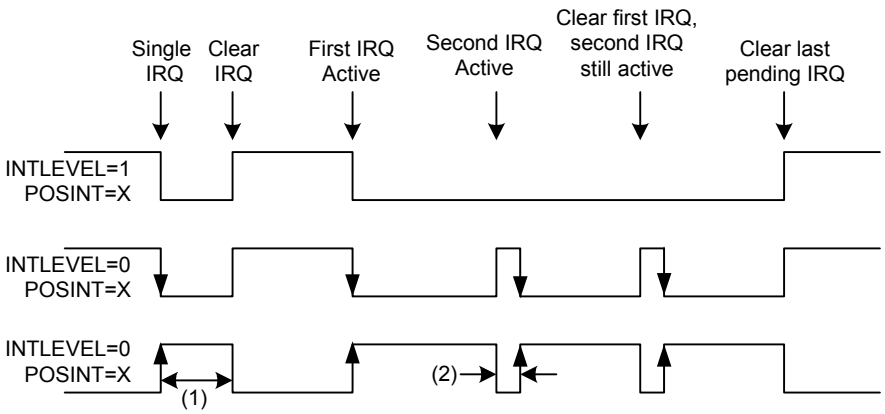
Location: PINCTL.3

Set: The CPU sets this bit to make the INT output pin level sensitive. When INTLEVEL=1 the output pin is active low, open-drain. When INTLEVEL=1 the system must include a pullup resistor to VL.

Clear: The CPU clears this bit to make the INT pin edge active. When INTLEVEL=0, the edge polarity is set by the POSINT bit (page 54).

POR: INTLEVEL=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:



(1) Width determined by clearing the IRQ. (2) Fixed at ~20 usec.

Figure 8. INT pin behavior depending on INTLEVEL and POSINT bits.

The waveforms in Figure 8 show the INT pin behavior for different settings of the INTLEVEL and POSINT (page 54) bits. In level mode (INTLEVEL=1), the INT pin stays low if any interrupts are pending.

NOVBUSIE

Meaning: No V_{BUS} Interrupt Enable.

Location: USBIEN.5

Set: Enables the NOVBUS Interrupt Request (page 47).

Clear: Disables the NOVBUS Interrupt Request.

POR: NOVBUSIE=0
Chip Reset: NOVBUSIE=0
Bus Reset: NOVBUSIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that turns on the interrupt enable bits should be called as part of servicing a USB bus reset.

Meaning: No V_{BUS} interrupt request.

Location: USBIRQ.5

Set: The SIE sets this bit when the V_{BUS} pin drops below the V_{BUS} detect threshold.

Clear: The CPU clears this bit by writing a "1" to it.

POR: NOVBUSIRQ=0
Chip Reset: NOVBUSIRQ=0
Bus Reset: NOVBUSIRQ=0
Pwr Down: Read-only

Programming Notes:

This IRQ bit provides an easy way for a self-powered device to determine that the USB peripheral implemented by the MAX3420E has just been disconnected from a USB host.

OSCOKIE

Meaning: Oscillator OK Interrupt Enable.

Location: USBIEN.0

Set: The CPU sets this bit to enable the OSCOK Interupt Request (page 49).

Clear: The CPU clears this bit to disable the OSCOK Interrupt Request.

POR: OSCOKIE=0
Chip Reset: OSCOKIE=0
Bus Reset: OSCOKIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that turns on the interrupt enable bits should be called as part of servicing a USB bus reset.

Meaning: Oscillator OK Interrupt Request.

Location: USBIRQ.0

Set: An internal OSCOK bit indicates that the internal 12 MHz oscillator is stable and the chip is ready to operate. The SIE sets the OSCOKIRQ bit when the OSCOK signal makes a 0-1 transition, indicating that the chip is ready to operate.

Clear: The CPU clears this bit by writing a "1" to it.

POR: OSCOKIRQ=0
Chip Reset: OSCOKIRQ=0
Bus Reset: OSCOKIRQ=0
Pwr Down: Read-only

OUT0DAVIE

Meaning: Endpoint 0 OUT Data Available Interrupt Enable.

Location: EPIEN.1

Set: The CPU sets this bit to enable the OUT0DAV interrupt request (page 53).

Clear: The CPU clears this bit to disable the OUT0DAV interrupt request.

POR: OUT0DAVIE=0
Chip Reset: OUT0DAVIE=0
Bus Reset: OUT0DAVIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that sets up the interrupt enables should be called as part of servicing a USB bus reset.

Meaning: Endpoint 0 OUT Data Available Interrupt Request.

Location: EPIRQ.1

Set: The SIE sets this bit when it has successfully received (and ACK'd) an OUT data packet to EP0.

Clear: The CPU clears this bit by writing a 1 to it. This also arms the endpoint for another transfer.

POR: OUT0DAVIRQ=0
Chip Reset: OUT0DAVIRQ=0
Bus Reset: OUT0DAVIRQ=0
Pwr Down: Read-only

OUT1DAVIE

Meaning: Endpoint 1 OUT Data Available Interrupt Enable.

Location: EPIEN.2

Set: The CPU sets this bit to enable the OUT1DAV Interrupt Request (page 51).

Clear: The CPU clears this bit to disable the OUT1DAV Interrupt Request.

POR: OUT1DAVIE=0
Chip Reset: OUT1DAVIE=0
Bus Reset: OUT1DAVIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that sets up the interrupt enables should be called as part of servicing a USB bus reset.

Meaning: Endpoint 1 OUT Data Available Interrupt Request.

Location: EPIRQ.2

Set: The SIE sets this bit when it has successfully received (and ACK'd) an OUT data packet to EP1-OUT.

Clear: The CPU clears this bit by writing a 1 to it. This also arms the endpoint for another transfer.

POR: OUT1DAVIRQ=0
Chip Reset: OUT1DAVIRQ=0
Bus Reset: OUT1DAVIRQ=0
Pwr Down: Read-only

Programming Notes:

EP1OUT is a double-buffered endpoint, meaning that there are two FIFOS and byte count registers. Double buffering allows USB data simultaneously to move into one FIFO while the CPU reads data from the other. This improves bandwidth performance in many systems.

The OUT1DAVIRQ flag logic makes the double buffering invisible to the programmer. For example, assume that both buffers are available and therefore OUT1DAVIRQ=0. When an OUT packet arrives, OUT1DAVIRQ makes a 0-1 transition to indicate availability of the first packet. For a single-buffered endpoint, if another OUT packet arrived over EP1-OUT before the CPU had time to drain the FIFO, the SIE would respond with a NAK handshake to indicate that the endpoint was not available to accept data.

However with the double buffered endpoint, the second OUT packet is accepted and ACK'd because the second buffer is available for data. If a third OUT packet arrives before either FIFO is drained, the SIE NAKS the transfer to indicate that both FIFOS are full. When the CPU finishes reading the first FIFO and clears the OUT1DAVIRQ bit (by writing one to it), it immediately makes another 0-1 transition to indicate data is available in the second FIFO.

Meaning: The INT output pin (if set for edge output) is positive-edge active. This bit takes effect only if INTLEVEL=0 (page 45).

Location: PINCTL.2

Set: The CPU sets this bit to cause the INT pin to make a 0-1 transition whenever an interrupt requires service.

Clear: The CPU clears this bit to cause the INT pin to make a 1-0 transition whenever an interrupt requires service.

POR: POSINT=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

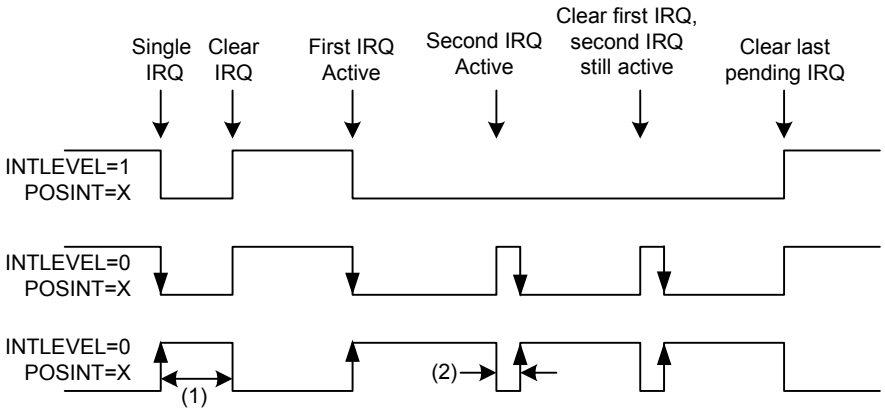


Figure 9. INT pin behavior depending on INTLEVEL and POSINT bits.

The setting of this bit has no effect if INTLEVEL=1 (page 45).

The edge sensitive mode supply an edge every time a new interrupt becomes active, or whenever the SPI master clears an IRQ while others are pending. As Figure 9 shows, the width of the edge-active pulse can vary. If only one IRQ is active, as shown by the first pulse, the width depends on how long the SPI master takes to clear the IRQ bit. If others are pending when a new IRQ asserts or when the SPI master clears an IRQ, the pulse width is fixed at about 20 microseconds.

Meaning: Power-Down the MAX3420E.

Location: USBCTL.4

Set: The CPU sets this bit to put the chip into the low power state required by a USP peripheral in the suspended state.

Clear: The CPU clears this bit to take the chip out of the low power state and resume operation.

POR: PWRDOWN=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

The power-down operation is triggered by the 0-1 transition of the PWRDOWN bit. Therefore if the chip wakes up while the bit PWRDOWN=1 (see below for the various wakeup methods), it will not immediately power down again. *Any wakeup routine should clear the PWRDOWN bit to ready it for the next 0-1 transition.*

The CPU normally puts the MAX3420E into its power-down mode by setting PWRDOWN=1 and HOSCSTEN=1 (page 37). When the CPU sets PWRDOWN=1, the SIE takes the following actions:

- Stops the internal 12 MHz oscillator.
- Monitors the USB DPLUS pin for bus activity.
- Monitors the SPI port for access to a limited set of registers (e.g. USBCTL).

Once in the low power state (PWRDOWN=1), the chip may wake up in two ways:

1. *The CPU clears the PWRDOWN bit.*

This starts the internal oscillator, and once stable, the SIE activates the OSCOK IRQ (page 49).

2. *The SIE detects activity on DPLUS and the bit HOSCSTEN=1.*

See page 37 for a description of the special case of a self-powered device that sets HOSCSTEN=0.

RWUDNIE

Meaning: Remote Wakeup Signaling Done Interrupt Enable.

Location: USBIEN.1

Set: The CPU sets this bit to enable the RWUDN Interrupt Request (page 57).

Clear: The CPU clears this bit to disable the RWUDN interrupt request.

POR: RWUDNIE=0
Chip Reset: RWUDNIE=0
Bus Reset: RWUDNIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that turns on the interrupt enable bits should be called as part of servicing a USB bus reset.

Meaning: Remote Wakeup Signaling Done Interrupt Request.

Location: USBIRQ.1

Set: The SIE sets this bit at the end of RWU signaling (10 ms of a K-state).

Clear: The CPU clears this bit by writing a “1” to it.

POR: RWUDNIRQ=0
Chip Reset: RWUDNIRQ=0
Bus Reset: RWUDNIRQ=0
Pwr Down: Read-only

Programming Notes:

The CPU signals a remote wakeup by setting the bit SIGRWU=1 (page 58).

SIGRWU

Meaning: Signal Remote Wakeup.

Location: USBCTL.2

Set: The CPU sets this bit to signal remote wakeup to the USB host.

Clear: The CPU clears this bit to terminate remote wakeup signaling.

POR: SIGRWU=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

The SPI master sets this bit to initiate USB Remote Wakeup signaling on the bus. When the SPI master sets SIGRWU=1 the MAX3420E waits for 5 milliseconds of a J-state, and then drives the bus with the K-state (D+ low, D- high) for 10 milliseconds. After the 10 millisecond interval, the MAX3420E stops driving the bus and asserts the RWUDNIRQ (page 57) interrupt bit. The SPI master should set SIGRWU=1 only when the bus is in the suspended state.

If the SPI master writes the USBCTL register with SIGRWU=1 while the MAX3420E is in its power-down state (PWRDOWN=1, page 55) the write to the USBCTL register should also have the PWRDOWN bit cleared to restart the internal oscillator. In this case SIE restarts its oscillator and waits for it to stabilize before initiating the RWU signaling.

The SPI master sets SIGRWU=1 and waits for the SIE to assert the RWUDNIRQ (page 57) to indicate that the signaling interval is over. When RWUDNIRQ asserts, the SPI master should set SIGRWU=0 to turn off the signaling. If the SPI master sets SIGRWU=0 during the 10 msec signaling interval, the RWU signal still terminates after the full 10 msec. If the SPI master does not clear the SIGRWU bit when the RWUDNIRQ interrupt asserts, the SIE continuously drives the bus with 5 msec float (J), 10 msec K.

FYI: A USB peripheral should signal remote wakeup only if two pre-conditions are met:

1. During enumeration the device firmware informs the host that the device is capable of signaling remote wakeup. Specifically, the bmAttributes field of the Configuration Descriptor uses bit 5 to indicate if a configuration supports remote wakeup: 1=yes, 0=no.
2. The host issues a Set_Feature request with the feature selector field set to "Device_Remote_Wakeup". The host would never issue this request if it found the first condition, remote-wakeup-capable, to be false.

The host can later inhibit remote wakeup signaling by issuing a Clear_Feature request with the selector field set to "Device_Remote_Wakeup".

Meaning: Return the STALL handshake in response to an IN request to endpoint 0.

Location: EPSTALLS.0

Set: The CPU sets this bit to instruct the SIE to return the STALL handshake for an IN request directed to endpoint 0.

Clear: The SIE clears this bit whenever a SETUP token arrives.

POR: STLEP0IN=0
Chip Reset: STLEP0IN=0
Bus Reset: STLEP0IN=0
Pwr Down: Read-only

Programming Notes:

The CPU sends the STALL handshake to indicate an illegal or unknown request to endpoint 0.

Endpoint 0 has three stall bits to account for the status stage, and the optional IN and OUT data stages that the transfer may use:

- STLSTAT (page 64)
- STLEP0IN
- STLEP0OUT (page 60)

If a CONTROL transfer is to be stalled, both the data stage (if present) and the status stage should receive the STALL handshake. Since all three stall bits are cleared with the arrival of the next SETUP token, the best way to stall any CONTROL transfer is to set all three endpoint 0 stall bits. This will correctly stall all stages of any CONTROL transfer that the host may send.

STLEP0OUT

Meaning: Return the STALL handshake in response to an OUT request to endpoint 0.

Location: EPSTALLS.1

Set: The CPU sets this bit to instruct the SIE to return the STALL handshake for an OUT request directed to endpoint 0.

Clear: The SIE clears this bit whenever a SETUP token arrives.

POR: STLEP0OUT=0
Chip Reset: STLEP0OUT=0
Bus Reset: STLEP0OUT=0
Pwr Down: Read-only

Programming Notes:

The CPU sends the STALL handshake to indicate an illegal or unknown request to endpoint 0.

Endpoint 0 has three stall bits to account for the status stage, and the optional IN and OUT data stages that the transfer may use:

- STLSTAT (page 64)
- STLEP0IN (page 59)
- STLEP0OUT

If a CONTROL transfer is to be stalled, both the data stage (if present) and the status stage should receive the STALL handshake. Since all three stall bits are cleared with the arrival of the next SETUP token, the best way to stall any CONTROL transfer is to set all three endpoint 0 stall bits. This will correctly stall all stages of any CONTROL transfer that the host may send.

Meaning: Stall EP1-OUT. Return the STALL handshake in response to an OUT request to endpoint 1.

Location: EPSTALLS.2

Set: The CPU sets this bit to instruct the SIE to return the STALL handshake for an OUT request directed to endpoint 1.

Clear: The CPU clears this bit to return EP1-OUT to normal ACK-NAK operation.

POR: STLEP1OUT=0
Chip Reset: STLEP1OUT=0
Bus Reset: STLEP1OUT=0
Pwr Down: Read-only

Programming Notes:

The CPU sets this bit when it receives a Set_Feature(HALT) directed to EP1-OUT.
The CPU clears this bit when it receives a Clear_Feature(HALT) directed to EP1-OUT.

STLEP2IN

Meaning: Stall EP2-IN. Return the STALL handshake in response to an IN request to endpoint 2.

Location: EPSTALLS.3

Set: The CPU sets this bit to instruct the SIE to return the STALL handshake for an IN request directed to endpoint 2.

Clear: The CPU clears this bit to return EP2-IN to normal ACK-NAK operation.

POR: STLEP2IN=0
Chip Reset: STLEP2IN=0
Bus Reset: STLEP2IN=0
Pwr Down: Read-only

Programming Notes:

The CPU sets this bit when it receives a Set_Feature(HALT) directed to EP2-IN.
The CPU clears this bit when it receives a Clear_Feature(HALT) directed to EP2-IN.

Meaning: Return the STALL handshake in response to an IN request to endpoint 3.

Location: EPSTALLS.4

Set: The CPU sets this bit to instruct the SIE to return the STALL handshake for an IN request directed to endpoint 3.

Clear: The CPU clears this bit when it receives a Clear_Feature(Halt) request directed to EP3-IN.

POR: STLEP3IN=0
Chip Reset: STLEP3IN=0
Bus Reset: STLEP3IN=0
Pwr Down: Read-only

Programming Notes:

The CPU sets this bit when it receives a Set_Feature(HALT) directed to EP3-IN.
The CPU clears this bit when it receives a Clear_Feature(HALT) directed to EP3-IN.

STLSTAT

Meaning: Return the STALL handshake in response to the STATUS stage of a CONTROL transfer.

Location: EPSTALLS.5

Set: The CPU sets this bit to send the STALL handshake as the response to the status stage of a CONTROL transfer. Until the CPU either acknowledges (ACKSTAT bit=1) or stalls (STLSTAT=1) the transfer, the SIE answers the status stage of the CONTROL transfer with the NAK handshake.

Clear: The SIE clears this bit whenever a SETUP token arrives.

POR: STLSTAT=0
Chip Reset: STLSTAT=0
Bus Reset: STLSTAT=0
Pwr Down: Read-only

Programming Notes:

Endpoint 0 has three stall bits to account for the status stage, and the optional IN and OUT data stages that the transfer may use:

- STLSTAT
- STLEP0IN (page 59)
- STLEP0OUT (page 60)

If a CONTROL transfer is to be stalled, both the data stage (if present) and the status stage should receive the STALL handshake. Since all three stall bits are cleared with the arrival of the next SETUP token, the best way to stall any CONTROL transfer is to set all three endpoint 0 stall bits. This will correctly stall all stages of any CONTROL transfer that the host may send.

Meaning: SETUP Data Available Interrupt Enable.

Location: EPIEN.5

Set: The CPU sets this bit to enable the SUDAV Interrupt Request (page 66).

Clear: The SIE clears this bit to disable the SUDAV Interrupt Request.

POR: SUDAVIE=0
Chip Reset: SUDAVIE=0
Bus Reset: SUDAVIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that sets up the interrupt enables should be called as part of servicing a USB bus reset.

SUDAVIRQ

Meaning: SETUP Data Available Interrupt Request.

Location: EPIRQ.5

Set: The SIE sets this bit after error-free reception of the eight setup data bytes in a CONTROL transfer.

Clear: The CPU clears this bit by writing a “1” to it.

POR: SUDAVIRQ=0
Chip Reset: SUDAVIRQ=0
Bus Reset: SUDAVIRQ=0
Pwr Down: Read-only

Programming Notes:

This IRQ signals the CPU to read the eight SETUP data bytes from the SUDFIFO (page 67) and interpret the host request.

Meaning: Setup Data FIFO.

Location: SUDFIFO[7:0]

Write: When the SIE receives a CONTROL transfer from the host, it copies the eight SETUP data bytes into this FIFO. When the bytes are verified to be error-free, the SIE asserts the Setup Data Available Interrupt Request (page 66).

Read: The CPU does eight reads to this register to retrieve the eight SETUP bytes.

POR: SUDFIFO=0
Chip Reset: SUDFIFO=0
Bus Reset: SUDFIFO=0
Pwr Down: No read or write

Programming Notes:

The MAX3420E provides this separate FIFO to avoid mixing data and command bytes in the EP0FIFO (page 10). This relieves the programmer of the task of figuring out the type of data in the FIFO—the EP0FIFO (page 10) always contains data, and the SUDFIFO always contains commands.

SUSPIE

Meaning: SUSPEND Interrupt Enable.

Location: USBIEN.4

Set: The CPU sets this bit to enable the SUSPEND Interrupt Request (page 69).

Clear: The CPU clears this bit to disable the SUSPEND Interrupt Request.

POR: SUSPIE=0
Chip Reset: SUSPIE=0
Bus Reset: SUSPIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that turns on the interrupt enable bits should be called as part of servicing a USB bus reset.

Meaning: SUSPEND Interrupt Request.

Location: USBIRQ.4

Set: The SIE sets this bit when it detects a USB suspend event (3 milliseconds of no bus traffic).

Clear: The CPU clears this bit by writing a "1" to it.

POR: SUSPIRQ=0
Chip Reset: SUSPIRQ=0
Bus Reset: SUSPIRQ=0
Pwr Down: Read-only

Programming Notes:

The CPU responds to this interrupt by shutting down system peripherals, and then putting the MAX3420E into a low power state (PWRDOWN=1, page 55).

URESDNIE

Meaning: USB Bus Reset Done Interrupt Enable.

Location: USBIEN.7

Set: The CPU sets this bit to enable the URESDN IRQ (page 68).

Clear: The CPU clears this bit to enable the URESDN IRQ.

POR: URESDNIE=0
Chip Reset: URESDNIE=0
Bus Reset: *No change*
Pwr Down: Read-only

Programming Notes:

This is one of two bits that are not cleared during a USB bus reset. The other one is URESIE (page 72). Both of these bits are normally used by the CPU during the bus reset.

Meaning: USB Bus Reset Done Interrupt Request.

Location: USBIRQ.7

Set: The SIE sets this bit when it has detected the end of a USB bus reset.

Clear: The CPU clears this bit by writing a “1” to it.
.

POR: URESDNIRQ=0
Chip Reset: URESDNIRQ=0
Bus Reset: No change
Pwr Down: Read-only

URESIE

Meaning: USB Reset Interrupt Enable.

Location: USBIEN.3

Set: The CPU sets this bit to enable the USB Reset Interrupt Request (page 73).

Clear: The CPU clears this bit to disable the USB Reset Interrupt Request.

POR: URESIE=0
Chip Reset: URESIE=0
Bus Reset: *No change*
Pwr Down: Read-only

Programming Notes:

This is one of two bits that are not cleared during a USB bus reset. The other one is URESDNIE (page 70). Both of these bits are normally used by the CPU during the bus reset.

Meaning: USB bus reset interrupt request.

Location: USBIRQ.3

Set: The SIE sets this bit when it detects a USB bus reset (at least 2.5 usec of the SE0 bus state).

Clear: The CPU clears this bit by writing a “1” to it.

POR: URESIRQ=0
Chip Reset: URESIRQ=0
Bus Reset: URESIRQ=1
Pwr Down: Read-only

VBUSIE

Meaning: V_{BUS} Detect Interrupt Enable.

Location: USBIEN.6

Set: The CPU sets this bit to enable the VBUS Interrupt Request (page 74).

Clear: The CPU clears this bit to disable the VBUS Interrupt Request.

POR: VBUSIE=0
Chip Reset: VBUSIE=0
Bus Reset: VBUSIE=0
Pwr Down: Read-only

Programming Notes:

Because most of the Interrupt Enable bits are cleared during a USB bus reset, the initialization routine that turns on the interrupt enable bits should be called as part of servicing a USB bus reset.

Meaning: V_{BUS} present interrupt request.

Location: USBIRQ.6

Set: The SIE sets this bit when the V_{BUS} comparator makes a 0-1 transition.

Clear: The CPU clears this bit by writing a "1" to it.

POR: VBUSIRQ=0
Chip Reset: VBUSIRQ=0
Bus Reset: VBUSIRQ=0
Pwr Down: Read-only

Programming Notes:

This IRQ bit provides an easy way for a self-powered device to determine that the USB peripheral implemented by the MAX3420E has just been connected to a USB host.

VBGATE

Meaning: V_{BUS} Gate.

Location: USBCTL.6

Set: The CPU sets this bit to make operation of the CONNECT bit (page 5) conditional on V_{BUS} being present on the V_{BUS} pin.

Clear: The CPU clears this bit to make operation of the CONNECT bit independent of V_{BUS} being valid.

POR: VBGATE=0
Chip Reset: *No change*
Bus Reset: *No change*
Pwr Down: Read-write

Programming Notes:

The USB specification states that a full-speed USB device must never power its DPLUS pullup resistor in the absence of V_{BUS}. This is most likely to occur in a self-powered device which has its own power supply and does not directly power the pullup resistor from the V_{BUS} pin. Only when the host activates the downstream port by turning on V_{BUS} is the peripheral allowed to power its pullup resistor to indicate that it is plugged into the bus.

The VBGATE bit enforces this requirement automatically by gating the action of the CONNECT bit with the presence of V_{BUS}, as shown by the following table:

CONNECT	VBGATE	VBUS_DET	PULL-UP
0	X	X	Not Connected
1	0	X	Connected
1	1	0	Not Connected
1	1	1	Connected

If VBGATE=1 and CONNECT=1, the pullup resistor will not be connected until the V_{BUS} detector indicates a valid V_{BUS} voltage is present.