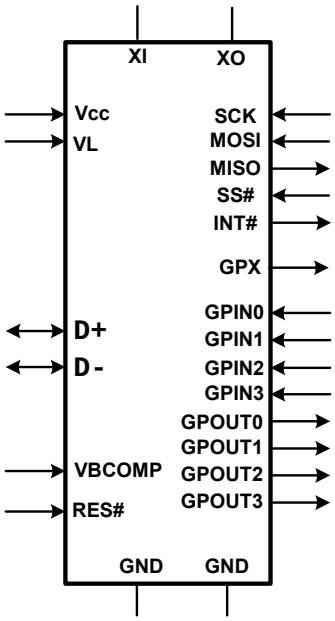
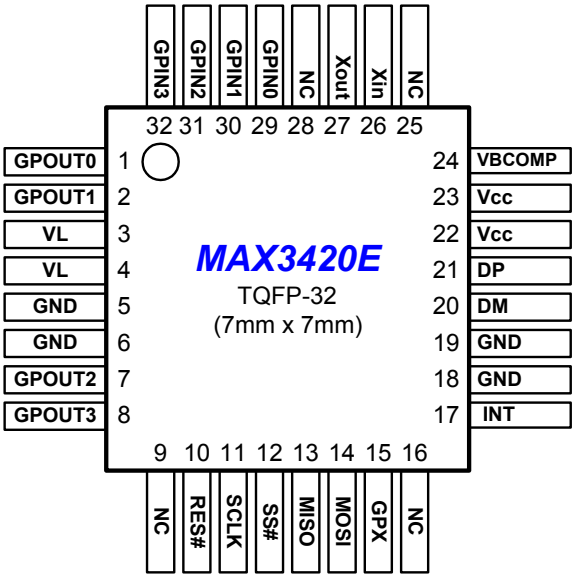
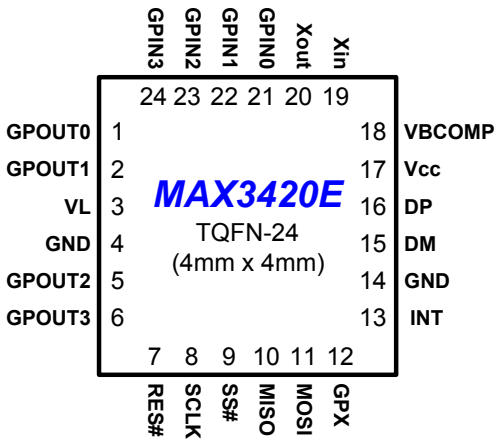
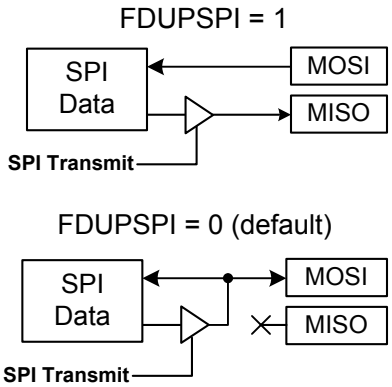
MAX3420E USB Interface Information Card





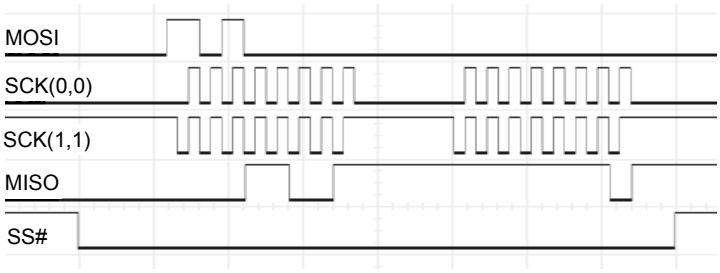


SPI Data



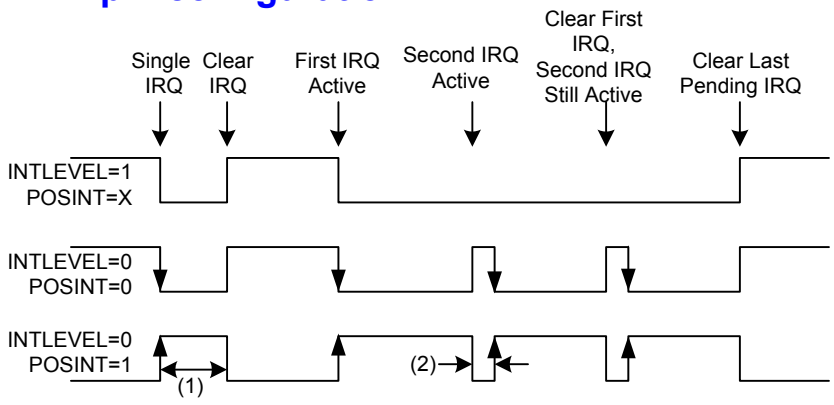
The FDUPSPI bit configures the SPI interface for full-duplex or half-duplex operation. In half-duplex mode (FDUPSPI = 0) the MISO pin can be left unconnected.

SPI Modes



The MAX3420E SPI interface works in modes (0,0) and (1,1) without alteration. The difference is the inactive SCK level. In both modes the data is changed on the SCK falling edge and sampled on the SCK rising edge. Full-duplex mode (FDUPSPI=1) is shown here.

INT pin configuration



(1) Width determined by time taken to clear the IRQ. (2)~10 usec.

INT pin waveforms for the three settings of the INTLEVEL and POSINT bits. In level mode the INT pin stays low until no interrupt requests are pending. In edge mode the INT pin delivers an edge whenever a new interrupt request occurs or an interrupt request bit is cleared while others are pending. The INTLEVEL mode is open-drain and requires a pullup resistor to VL.

Visit http://www.maxim-ic.com/usb for:

- A downloadable copy of this reference card
- Additional information on USB
- Maxim USB power and interface products

Rev. August 16, 2005

	Reg	Name	b7	b6	b5	b4	b3	b2	b1	b0
0	R0	**EP0FIFO**	b7	b6	b5	b4	b3	b2	b1	b0
1	R1	**EP1OUTFIFO**	b7	b6	b5	b4	b3	b2	b1	b0
2	R2	**EP2INFIFO**	b7	b6	b5	b4	b3	b2	b1	b0
3	R3	**EP3INFIFO**	b7	b6	b5	b4	b3	b2	b1	b0
4	R4	**SUDFIFO**	b7	b6	b5	b4	b3	b2	b1	b0
5	R5	**EP0BC**	0	b6	b5	b4	b3	b2	b1	b0
6	R6	**EP1OUTBC**	0	b6	b5	b4	b3	b2	b1	b0
7	R7	**EP2INBC**	0	b6	b5	b4	b3	b2	b1	b0
8	R8	**EP3INBC**	0	b6	b5	b4	b3	b2	b1	b0
9	R9	**EPSTALLS**	0	ACKSTAT	STLSTAT	STLEP3IN	STLEP2IN	STLEP1OUT	STLEP0OUT	STLEP0IN
A	R10	**CLRTOGS**	EP3DISAB	EP2DISAB	EP1DISAB	CTGEP3IN	CTGEP2IN	CTGEP1OUT	0	0
B	R11	**EPIRQ**	0	0	SUDAVIRQ	IN3BAVIRQ	IN2BAVIRQ	OUT1DAVIRQ	OUT0DAVIRQ	IN0BAVIRQ
C	R12	**EPIEN**	0	0	SUDAVIE	IN3BAVIE	IN2BAVIE	OUT1DAVIE	OUT0DAVIE	IN0BAVIE
D	R13	**USBIRQ**	URESDNIRQ	VBUSIRQ	NOVBUSIRQ	SUSPIRQ	URESIRQ	BUSACTIRQ	RWUDNIRQ	OSCOKIRQ
E	R14	**USBIEN**	URESDNIE	VBUSIE	NOVBUSIE	SUSPIE	URESIE	BUSACTIE	RWUDNIE	OSCOKIE
F	R15	**USBCTL**	HOSCSTEN	VBGATE	CHIPRES	PWRDOWN	CONNECT	SIGRWU	0	0
10	R16	**CPUCTL**	0	0	0	0	0	0	0	IE
11	R17	**PINCTL**	EP3INAK	EP2INAK	EP0INAK	FDUPSPI	INTLEVEL	POSINT	GPXB	GPXA
12	R18	**REVISION**	0	0	0	0	0	0	0	1
13	R19	**FNADDR**	0	b6	b5	b4	b3	b2	b1	b0
14	R20	**IOPINS**	GPIN3	GPIN2	GPIN1	GPIN0	GPOUT3	GPOUT2	GPOUT1	GPOUT0

Power On	RES# Pin or CHIPRES = 1	USB Bus Reset
IN3BAVIRQ=1	IN3BAVIRQ=1	IN3BAVIRQ=1
IN2BAVIRQ=1	IN2BAVIRQ=1	IN2BAVIRQ=1
IN0BAVIRQ=1	IN0BAVIRQ=1	IN0BAVIRQ=1
	CHIPRES	CHIPRES
	CONNECT	CONNECT
		EP0FIFO
		EP1OUTFIFO
		EP2INFIFO
		EP3INFIFO
		GPOUT[3:0]
	GPX[B:A]	GPX[B:A]
	HOSCSTEN	HOSCSTEN
		IE
	INTLEVEL	INTLEVEL
	POSINT	POSINT
	PWRDOWN	PWRDOWN
	SIGRWU	SIGRWU
		URESDNIE
		URESIE

MAX3420E Resets. For the three reset types, all register bits are cleared except the indicated bits in this table. The indicated BAVIRQ bits are set to 1, and the remaining bits shown in the table retain their states. The Revision register is read-only and is never cleared.

To Send an IN Packet

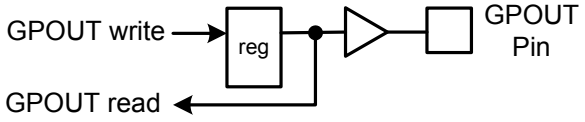
1. Load an INFIFO with data.
2. Load the EPINBC register with the byte count. This arms the transfer (next IN gets the data instead of a NAK).
3. INBAVIRQ asserts when packet is successfully sent.
4. Go to step 1.

To Read an OUT Packet

1. OUTDAV IRQ asserts when new OUT data is available.
2. Read the EPOUTBC register to determine how many bytes are available.
3. Read that number of bytes from the EPOUTFIFO.
4. Re-arm the endpoint by clearing the OUTDAV IRQ bit.

GPXB	GPXA	GPX Pin
0	0	Operate
0	1	Vbus Comparator
1	0	BusAct
1	1	SOF

GPOUT Pins



Writing a GPOUT bit updates theGPOUT pin state.
Reading a GPOUT bit returns the register output.

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

SPI Command Byte. All SPI transfers are MSB-first.

b7	b6	b5	b4	b3	b2	b1	b0
SUSP IRQ	URES IRQ	SUDAV IRQ	IN3BAV IRQ	IN2BAV IRQ	OUT1DAV IRQ	OUT0DAV IRQ	IN0BAV IRQ

Full-duplex SPI mode (FDUPSPI=1) only: These register bits simultaneously clock out on the MISO pin as the command byte is clocked into the MOSI pin.