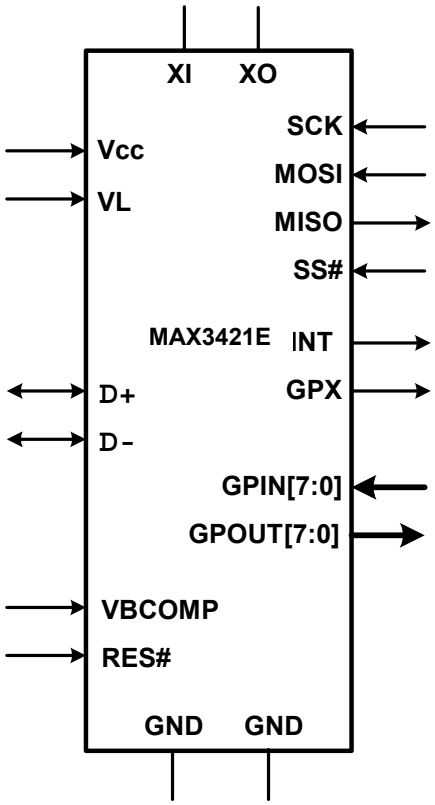
MAX3421E
Programming Guide

SPI インタフェース付き
USB ペリフェラル/ホストコントローラ

MAX3421E は、SPI インタフェースを備えたシステムに USB のホスト機能またはペリフェラル機能を追加するものです。このプログラミングガイドでは、ホスト動作の場合の各レジスタとビットについて説明します。ペリフェラルとしての MAX3421E の動作については、『MAX3420E プログラミングガイド』を参照してください。



MAX3421Eの詳細については、japan.maxim-ic.com/max3421eを参照してください。
USBおよびマキシムのUSB製品の詳細については、japan.maxim-ic.com/usbを参照してください。
SPI は Motorola, Inc.の登録商標です。
Maxim のロゴは Maxim Integrated Products, Inc.の登録商標です。
Dallas のロゴは Dallas Semiconductor Corp.の登録商標です。

December 2006

このプログラミングガイドについて

MAX3421E はデュアルロールの USB コントローラであり、USB ペリフェラルとしてプログラムすることもできれば、USB ホストとしてプログラムすることもできます。ペリフェラルとしての MAX3421E の動作は、MAX3420E (ペリフェラルのみの製品)と同じです。既存の MAX3420E のコードを実行する USB ペリフェラルとしてのみ MAX3421E を使用する予定の場合は、HOST ビットを 0 (デフォルト)に設定したままにして、プログラミングの詳細について、『*MAX3420E プログラミングガイド*』を参照してください。

MAX3421E を USB ペリフェラルとしてのみ使用し、ただし新しいペリフェラル機能を利用したい場合は、このプログラミングガイドの追加レジスタ R21～R24 と、追加ビット PULSEWID1/0、SEPIRQ、および HOST についてお読みになってから、『*MAX3420E プログラミングガイド*』を参照してください。

ほとんどのMAX3421Eユーザは、このガイドの目的であるホスト動作のプログラミングの詳細について知りたいと考えていることでしょう。このため、ホスト動作に適用されるビットとレジスタ(**表 1**) のみを説明する方が理に適っていますが、簡単に参照できるよう、**表 2** には、ホストとペリフェラルの両方の動作についてのレジスタビットもすべて記載しています。

表 1 は、このアプリケーションノートを電子形式で参照する場合のナビゲーションツールとしての役割を果たします。この表には、文書ページへのリンクが含まれており、リンクされたレジスタとビットについて記載されています。

表 1. MAX3421E の HOST レジスタ(HOST ビット = 1)

Reg	Name	b7	b6	b5	b4	b3	b2	b1	b0	acc
R0	—	0	0	0	0	0	0	0	0	—
R1	**RCVFIFO**	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R2	**SNDFIFO**	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R3	—	0	0	0	0	0	0	0	0	—
R4	**SUDFIFO**	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R5	—	0	0	0	0	0	0	0	0	RSC
R6	**RCVBC**	0	b6	b5	b4	b3	b2	b1	b0	RSC
R7	**SNDBC**	0	b6	b5	b4	b3	b2	b1	b0	—
R8	—	0	0	0	0	0	0	0	0	—
R9	—	0	0	0	0	0	0	0	0	—
R10	—	0	0	0	0	0	0	0	0	—
R11	—	0	0	0	0	0	0	0	0	—
R12	—	0	0	0	0	0	0	0	0	RSC
R13	**USBIRQ**	0	VBUSIRQ	NOVBUSIRQ	0	0	0	0	OSCOKIRQ	RC
R14	**USBIEN**	0	VBUSIE	NOVBUSIE	0	0	0	0	OSCOKIE	RSC
R15	**USBCTL**	0	0	CHIPRES	PWRDOWN	0	0	0	0	RSC
R16	**CPUCTL**	PULSEWID1	PULSEWID0	0	0	0	0	0	IE	RSC
R17	**PINCTL**	0	0	0	FDUPSPI	INTLEVEL	POSINT	GPXB	GPXA	RSC
R18	**REVISION**	b7	b6	b5	b4	b3	b2	b1	b0	R
R19	—	0	0	0	0	0	0	0	0	—
R20	**IOPINS1**	GPIN3	GPIN2	GPIN1	GPIN0	GPOUT3	GPOUT2	GPOUT1	GPOUT0	RSC
R21	**IOPINS2**	GPIN7	GPIN6	GPIN5	GPIN4	GPOUT7	GPOUT6	GPOUT5	GPOUT4	RSC
R22	**GPINIRQ**	GPINIRQ7	GPINIRQ6	GPINIRQ5	GPINIRQ4	GPINIRQ3	GPINIRQ2	GPINIRQ1	GPINIRQ0	RC
R23	**GPINIEN**	GPINIEN7	GPINIEN6	GPINIEN5	GPINIEN4	GPINIEN3	GPINIEN2	GPINIEN1	GPINIEN0	RSC
R24	**GPINPOL**	GPINPOL7	GPINPOL6	GPINPOL5	GPINPOL4	GPINPOL3	GPINPOL2	GPINPOL1	GPINPOL0	RSC
R25	**HIRQ**	HXFRDNIRQ	FRAMEIRQ	CONDETIRQ	SUSDNIRQ	SNDBAVIRQ	RCVDAVIRQ	RWUIRQ	BUSEVENTIRQ	RC
R26	**HIEN**	HXFRDNIE	FRAMEIE	CONDETIE	SUSDNIE	SNDBAVIE	RCVDAVIE	RWUIE	BUSEVENTIE	RSC
R27	**MODE**	DPPULLDN	DMPULLDN	DELAYISO	SEPIRQ	SOFKAENAB	HUBPRE	LOWSPEED	HOST	RSC
R28	**PERADDR**	0	b6	b5	b4	b3	b2	b1	b0	RSC
R29	**HCTL**	SNDTOG1	SNDTOG0	RCVTOG1	RCVTOG0	SIGRSM	SAMPLEBUS	FRMRST	BUSRST	LS
R30	**HXFR**	HS	ISO	OUTNIN	SETUP	EP3	EP2	EP1	EP0	LS
R31	**HRSL**	JSTATUS	KSTATUS	SNDTOGRD	RCVTOGRD	HRSLT3	HRSLT2	HRSLT1	HRSLT0	R

HOST ビット(R27 のビット 0)を 1 に設定すると、3 つの方法で MAX3420E のレジスタマップが変更されます。すなわち、レジスタ R0～R7 が再定義され、ホストの動作に適用されない特定のビットがクリアされ、レジスタ 25、26 および 28～31 が追加されます。ホストモードでは、**表 1** の「0」で示したビットに 0 が書き込まれます。明確にするため、未使用のペリフェラルレジスタとビットは 0 で示しています。

表 2. ペリフェラルモードとホストモードの両方における、MAX3421E の全レジスタビット

（網掛けで表されたビットは、モード(HOST ビット)の変更中に変化しません。）

Reg	Perip.Name	Host Name	b7	b6	b5	b4	b3	b2	b1	b0	acc
R0	**EP0FIFO**	—	b7	b6	b5	b4	b3	b2	b1	b0	—
R1	**EP1OUTFIFO**	**RCVFIFO**	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R2	**EP2INFIFO**	**SNDFIFO**	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R3	**EP3INFIFO**		b7	b6	b5	b4	b3	b2	b1	b0	—
R4	**SUDFIFO**	**SUDFIFO**	b7	b6	b5	b4	b3	b2	b1	b0	RSC
R5	**EP0BC**	—	0	b6	b5	b4	b3	b2	b1	b0	RSC
R6	**EP1OUTBC**	**RCVBC**	0	b6	b5	b4	b3	b2	b1	b0	RSC
R7	**EP2INBC**	**SNDBC**	0	b6	b5	b4	b3	b2	b1	b0	—
R8	**EP3INBC**	—	0	b6	b5	b4	b3	b2	b1	b0	—
R9	**EPSTALLS**	—	0	ACKSTAT	STLSTAT	STLEP3IN	STLEP2IN	STLEP1OUT	STLEP0OUT	STLEP0IN	—
R10	**CLRTOGS**	—	EP3DISAB	EP2DISAB	EP1DISAB	CTGEP3IN	CTGEP2IN	CTGEP1OUT	0	0	—
R11	**EPIRQ**	—	0	0	SUDAVIRQ	IN3BAVIRQ	IN2BAVIRQ	OUT1DAVIRQ	OUT0DAVIRQ	IN0BAVIRQ	—
R12	**EPIEN**	—	0	0	SUDAVIE	IN3BAVIE	IN2BAVIE	OUT1DAVIE	OUT0DAVIE	IN0BAVIE	—
R13	**USBIRQ**	**USBIRQ**	URESDNIRQ	VBUSIRQ	NOVBUSIRQ	SUSPIRQ	URESIRQ	BUSACTIRQ	RWUDNIRQ	OSCOKIRQ	RC
R14	**USBIEN**	**USBIEN**	URESDNIE	VBUSIE	NOVBUSIE	SUSPIE	URESIE	BUSACTIE	RWUDNIE	OSCOKIE	RSC
R15	**USBCTL**	**USBCTL**	HOSCSTEN	VBGATE	CHIPRES	PWRDOWN	CONNECT	SIGRWU	0	0	RSC
R16	**CPUCTL**	**CPUCTL**	PULSEWID1	PULSEWID0	0	0	0	0	0	IE	RSC
R17	**PINCTL**	**PINCTL**	EP3INAK	EP2INAK	EP1INAK	FDUPSPI	INTLEVEL	POSINT	GPXB	GPXA	RSC
R18	**REVISION**	**REVISION**	b7	b6	b5	b4	b3	b2	b1	b0	R
R19	**FNADDR**	—	0	b6	b5	b4	b3	b2	b1	b0	—
R20	**IOPINS1**	**IOPINS1**	GPIN3	GPIN2	GPIN1	GPIN0	GPOUT3	GPOUT2	GPOUT1	GPOUT0	RSC
R21	**IOPINS2**	**IOPINS2**	GPIN7	GPIN6	GPIN5	GPIN4	GPOUT7	GPOUT6	GPOUT5	GPOUT4	RSC
R22	**GPINIRQ**	**GPINIRQ**	GPINIRQ7	GPINIRQ6	GPINIRQ5	GPINIRQ4	GPINIRQ3	GPINIRQ2	GPINIRQ1	GPINIRQ0	RC
R23	**GPINIEN**	**GPINIEN**	GPINIE7	GPINIE6	GPINIE5	GPINIE4	GPINIE3	GPINIE2	GPINIE1	GPINIE0	RSC
R24	**GPINPOL**	**GPINPOL**	GPINPOL7	GPINPOL6	GPINPOL5	GPINPOL4	GPINPOL3	GPINPOL2	GPINPOL1	GPINPOL0	RSC
R25		**HIRQ**	HXFRDNIRQ	FRAMEIRQ	CONDETIRQ	SUSDNIRQ	SNDBAVIRQ	RCVDAVIRQ	RWUIRQ	BUSEVENTIRQ	RC
R26		**HIEN**	HXFRDNIE	FRAMEIE	CONDETIE	SUSDNIE	SNDBAVIE	RCVDAVIE	RWUIE	BUSEVENTIE	RSC
R27	**MODE**	**MODE**	DPPULLDN	DMPULLDN	DELAYISO	SEPIRQ	SOFKAENAB	HUBPRE	LOWSPEED	HOST	RSC
R28		**PERADDR**	0	b6	b5	b4	b3	b2	b1	b0	RSC
R29		**HCTL**	SNDTOG1	SNDTOG0	RCVTOG1	RCVTOG0	SIGRSM	SAMPLEBUS	FRMRST	BUSRST	LS
R30		**HXFR**	HS	ISO	OUTNIN	SETUP	EP3	EP2	EP1	EP0	LS
R31		**HRSL**	JSTATUS	KSTATUS	SNDTOGRD	RCVTOGRD	HRSLT3	HRSLT2	HRSLT1	HRSLT0	R

用語

CPU

MAX3421EのSPI™インタフェースは、たとえばマイクロコントローラ、DSP、またはASICなどのSPIマスタによって駆動することができます。この*プログラミングガイド*では、簡単にするため、MAX3421EのSPIポートに接続されたSPIマスタを「CPU」と呼びます。

SIE

SIE はシリアルインタフェースエンジン(Serial Interface Engine)のことです。MAX3421E の内部にあるこの論理ユニットは、USB のあらゆるアクティビティを管理します。いずれのエンティティ(CPU または MAX3421E)がレジスタビットをセットまたはクリアしているのかを明確に示すため、このアプリケーションノートでは、極めて厳密に、かつ一貫して用語を使用しています。すなわち、MAX3421E がビットをセットまたはクリアすることを示す場合には「SIE」を使用し、MAX3421E に接続された SPI マスタがビットをセットまたはクリアすることを示す場合には「CPU」を使用します。

モード

MAX3421E は、2 つのモード(ペリフェラルまたはホスト)で動作します。

- ペリフェラルのみのビットは、『*MAX3420E プログラミングガイド*』に掲載されています。
- 「**ホストおよびペリフェラル**」と示されたビットは両方のモードでアクティブです。
- 「**ホストのみ**」と示されたビットは、ホストモードでのみ有効です。

ISO

USB の 4 つの転送タイプの 1 つである ISOCHRONOUS の略語です。USB ホストとして動作するとき、MAX3421E はフルスピードの ISO 転送をサポートします。ペリフェラルとして動作するときは、ISO 転送をサポートしません。

リセット

MAX3421E には、以下の 3 つのリセットソースがあります。

1. 内部のパワーオンリセット(POR)回路
2. RES#ピン
3. CHIPRES と呼ばれるレジスタビット

リセットが発生する可能性のある 4 番目の要因があります。HOST ビットが状態を変更するときに(ホストまたはペリフェラルの動作に切り替えるとき)、SIE が特定のレジスタビットをクリアするというものです。

この項で、各リセットソースの影響について説明します。

MAX3421E のレジスタビットは、以下の 2 つのソースでクロック制御されます。

1. 12MHz の内部発振器
2. CPU が SPI インタフェースに供給する SCLK 信号

RES#または PWRDOWN のリセットをアサートすると、12MH の内部クロックが停止されます。ほとんどのレジスタビットは非同期でクリアされますが、SPI インタフェースでクロック制御されるレジスタビットはアクティブ状態のままであるため、CPU は、SPI の設定(たとえば FDUPSPI ビット)、USB バスのプルダウン抵抗器、および PWRDOWN ビットの状態を制御することができます。

パワーオンリセット

SIE は、あらゆるレジスタビットをクリアします。リセットが解除されると、SIE は、以下のビットをセットして利用可能なバッファを示します。

- IN3BAVIRQ (ペリフェラル)
- IN2BAVIRQ (ペリフェラル)
- IN0BAVIRQ (ペリフェラル)
- SNDBAVIRQ (ホスト)

注:POR は、HOST = 0 に設定します。これによってペリフェラル動作にデフォルト設定されます。

RES#ピンのアサートまたは CHIPRES = 1 の設定

これらのリセットは、12MHz の内部発振器を停止し、ほとんどのレジスタビットをクリアしますが、SPI によってクロック制御されるレジスタビットには影響せず、引き続き CPU がアクセス可能です。SPI によってクロック制御されるレジスタビットを以下に示します。

- HOSCSTEN
- VBGATE
- CHIPRES
- PWRDOWN
- CONNECT
- SIGRWU
- FDUPSPI

- INTLEVEL
- POSINT
- GPX[B:A]
- GPOUT[7:0]
- DPPULLUP/DN

注:これらのリセットは、HOST = 0 に設定し、ペリフェラルの動作を選択します。

MODE ビットの変更

ペリフェラルからホストに変更

SIE は、ホストの動作に適用されないレジスタビットをクリアします。HOST = 1 のとき、CPU がこれらのビットに 0 以外を書き込むことのないようにしてください。

- EPSTALLS レジスタ
- CLRTOGS レジスタ
- EPIRQ レジスタ
- EPIEN レジスタ
- USBIRQ レジスタ(VBUSIRQ ビットと NOVBUS IRQ ビットを除く。これらのビットは、ホストとしてアクティブのままになります)
- USBIEN レジスタ(VBUSIRQ ビットと NOVBUS IRQ ビットを除く。これらのビットは、ホストとしてアクティブのままになります)
- PINCTL レジスタの EP3/2/1 INAK ビット

ホストからペリフェラルに変更

CPU は、CHIPRES ビットをアサートすることによってホストモードからペリフェラルモードに戻ります。これによって、上記の規則にしたがって各ビットはクリアされ、HOST ビットは 0 に設定されます。

MAX3421E のレジスタにアクセスする方法

SPIマスタは、R0～R31 のレジスタ空間の 21 の内部レジスタ(表 1)の書込みと読出しを行うことによってMAX3421Eを制御します。SPIマスタは、MAX3421EのSS# (スレーブセレクト、アクティブロー)ピンをアサートし、SPIコマンドバイトを構成する 8 つのビットを同期入力することによって、あらゆるレジスタへのアクセスを開始します。**図 1** に、コマンドバイトのフォーマットを示します。

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

図 1. SPI コマンドバイト。すべての SPI 転送の場合と同様、ビット 7 が最初に送信されます。ACKSTAT ビットはペリフェラルモードでのみ有効です。ホストモードのとき、MAX3421E は ACKSTAT ビットを無視します。

Reg4～Reg0 は、0～31 の有効値を持つレジスタアドレスを設定します。MAX3421E は 31 を超えるレジスタの値を無視します。方向ビットは、それ以降のバイト転送の方向を設定します。ACKSTAT ビットは、USBの制御ビット(R9 のビット 6)をコピーしたものであり、MAX3421E が USB ペリフェラルとして動作するときにのみ有効です。HOST = 1 のとき、SIE は ACKSTAT ビットを無視します。

コマンドバイトを送信した後、SPI マスタは、DIR ビットで示された方向に 1 つまたは複数のバイトを転送します。SPI マスタは、SS#をローに保持したまま、各バイトに 8 つの SCK パルスを追加で提供します。バイトの転送が完了すると、SPI マスタは SS#をデアサートし(ハイに駆動)、転送は終了します。

MAX3421E には、2 つのタイプのレジスタがあります。すなわち FIFO レジスタと制御レジスタです。レジスタの読出しまたは書込みの繰り返しは、レジスタのタイプによって異なる結果をもたらします。

レジスタ R1、R2、および R4 は内部 FIFO にアクセスします。コマンドバイトによってレジスタ番号を選択した後、SPI マスタは、同じ SPI 転送(SS#をローに維持)の間に読出しまたは書込みを繰り返すことによって、連続する FIFO のバイトをロードまたはアンロードします。たとえば、SENDFIFO に 45 バイトを書き込むには、SPI マスタは、以下の手順を実施します。

1. SS# = 0 に設定し、転送を開始します。
2. 8 つの SCLK パルスを発行し、コマンドバイト 00010010 を送信します。このコマンドバイトは、書込み動作(DIR = 1)のために R2 (SNDFIFO)を選択します。
3. 8 つの SCK パルスを発行します。クロックのたびに、SNDFIFO レジスタに 1 データビットを同期入力します(SCK の立上りエッジごとに 1 ビット)。1 バイトごとに、SIE は FIFO のバイトを書き込んで、内部 FIFO のアドレスポインタを進めます。
4. 手順 3 をさらに 44 回繰り返し、合計 45 バイトを同期入力して SNDFIFO に格納します。
5. SS# = 1 に設定し、転送を終了します。

レジスタ R13～R31 は MAX3421E の制御レジスタです。SPI マスタが、同じ SPI 転送(SS#ロー)の間にR13～R20 の読出しまたは書込みを繰り返した場合、各バイトの読出しまたは書込みはすべて自動的に内部レジスタアドレスをインクリメントします。これによって、連続したレジスタの読出しまたは書込みが行えるようになり、新しいコマンドバイトを書き込んでそれぞれの新しいレジスタアドレスを設定する必要はありません。レジスタアドレスはこのように引き続きインクリメントされて R20 に到達すると、レジスタアドレス

は R20 で「固定」されます。この機能によって、µP は、R20 の IO ピンに迅速にアクセスすることができるようになります。たとえば、あらかじめ格納されている波形を GPIO ピン上に出力するには、SPI マスタは、コマンドバイト 10100010 (R20 の書込み)を書き込んでから、R20 に複数のデータバイトを送信し、波形を出力することができます。

SPI マスタが R21 以降の値を明示的にアドレス指定すると、レジスタアドレスは同じ SS#転送の間に再び自動的にインクリメントされ、R31 に到達すると、レジスタアドレスは R31 で再度「固定」されます。

フルデュープレクスモードでの最初の 8 ビットの MISO 値

FDUPSPI = 1 のとき、MAX3421Eはフルデュープレクスモードで SPIポートを動作します。つまり、SCLKの立上りエッジごとに、データのMOSIピンからの同期入力と、MISOピンへの同期出力が同時に行われます。SPIアクセスの最初の 8 ビットはコマンドバイトを構成し、MOSIピンに同期入力されます。MAX3421Eは、コマンドバイトの同期入力と同時にステータスデータをMISOピンに出力します。**図 2**に、ホストモードで動作中の場合のステータスビットを示します。

b7	b6	b5	b4	b3	b2	b1	b0
HXFRDN IRQ	FRAME IRQ	CONN IRQ	SUSDN IRQ	SNDBAV IRQ	RCVDAV IRQ	RSMREQ IRQ	BUSEVENT IRQ

図 2. FDUPSPI (HOST = 1)のときの MISO データ

ホスト転送のプログラミング

HOST = 1 のときは、ホストと同様に*考える*必要があります。ホストの要求に応答するのではなく、要求を生成することになります。つまり、送出するあらゆるパケットに以下が必要となります。

- PERADDR レジスタの関数アドレス
- HXFR レジスタの EP[3:0]のエンドポイント番号
- HXFR レジスタのパケット ID (PID)
- 一部のデータ。SNDFIFO の OUT データ(正確には SNDBC バイト)、または RCVFIFO の IN データ(RCVBC バイト)

関数アドレスとエンドポイント FIFO 内のデータは、CPU によって変更されるまで、MAX3421E のレジスタで保持されます。これは、たとえば OUT パケット(ペリフェラルによるエラーが原因)を再送するために、CPU は SNDFIFO を再度ロードする必要がないことを意味します。CPU は、HXFR レジスタに再び書き込むことによって、同じ転送を直ちに開始します。

MAX3421E は USB の管理作業を行います。上記のビットをセットアップし、HXFR レジスタに書き込むことによって転送を開始し、完了の割込みを待ち、HRSLT ビット(ホストの結果)をチェックして、転送がどのように進んでいるのかを確認します。MAX3421E は、16 のホストの結果状況を報告しますが、ほとんどの場合、**表 3** のいずれかに該当します。

表 3. USB の正常動作を表すホストの結果コード

HSRLT	ラベル	意味
0x00	hrSUCCESS	正常な転送
0x01	hrBUSY	SIEはビジー。転送は保留状態
0x04	hrNAK	ペリフェラルがNAKを返した
0x05	hrSTALL	ペリフェラルがSTALLを返した

16 の HRSL コードすべてを、36 ページに示します。表 3 の状況はいずれも電気的エラーや信号送出のエラーを表すものではありません。これらは、USB の正常な動作で発生するものです。hrBUSY の結果は、転送の完了時点を確認するのに HXFRDN 割込み要求を使用せずに、代わりに、HRSL レジスタをポーリングして、転送の完了の有無を確認するアプリケーションのためのものです。

ホストは、**表 4** に示す 7 つのタイプの転送を開始することができます。

表 4. さまざまな転送タイプのための HXFR レジスタビットの設定

Xfr Type	HS	ISO	OUTNIN	SETUP	hex
SETUP	**0**	**0**	**0**	**1**	**0x10**
BULK-IN	**0**	**0**	**0**	**0**	**0-ep**
BULK-OUT	**0**	**0**	**1**	**0**	**2-ep**
HS-IN	**1**	**0**	**0**	**0**	**0x80**
HS-OUT	**1**	**0**	**1**	**0**	**0xA0**
ISO-IN	**0**	**1**	**0**	**0**	**4-ep**
ISO-OUT	**0**	**1**	**1**	**0**	**6-ep**

CPU は、SIE が他の転送でビジーでないときであればいつでも、HXFR レジスタに書き込むことができます。SIE は、自動的に生成された SOF/KA フレームマーカと衝突しないように対処します。CPU の HXFR レジ

スタへの書込みがフレーム内で遅れて、次の転送がフレームマーカと衝突するようであれば、SIE は、次のフレームマーカの生成が完了するまで自動的にパケットの送信を延期します。

注:USB ホストは同一の方法を使用し、BULK と INTERRUPT のエンドポイントを介してデータを転送します。これら 2 つの転送タイプの唯一の違いは、パケットのスケジュールを設定するときの制御ファームウェアの関数です。このアプリケーションノートでは、BULK に関する考察はすべて INTERRUPT 転送にも当てはまるということを理解した上で、BULK 転送について説明します。

MAX3421E のデータトグルについて

USB プロトコルは、DATA0 および DATA1 と呼ばれる 2 つの PID (パケット ID)の 1 つを使用して、あらゆるデータパケットにタグを付けます。これらは、USB のエラー検出に役立ちます。データトグル値は、あらゆるエンドポイントに関連付けられており、これによって、使用する DATA PID が決まります。エンドポイントから、またはエンドポイントへの(リセット後の)最初のデータパケットは、DATA0 PID を使用して送信されます。送信側と受信側の両側が、(ACK ハンドシェイクを生成/受信することによって)データが正確であることを認めたとき、両方ともそのトグル値の補数をとります。したがって、エンドポイントに送信される、またはエンドポイントから受信される、連続したデータパケットには通常、*トグル*を行った PID の値(DATA0、DATA1、DATA0 など)が含まれます。

MAX3421Eには、データトグルを維持するための 4 つのビットがあります。エンドポイントへのデータ転送後、SIEはビットRCVTOGRDおよびSNDTOGRD (36ページ) を更新して、選択したエンドポイントのトグル値を示します。CPUは、これらのビットを読み出して格納し、必要であれば、この値で同じエンドポイントのトグル値に再初期化します。エンドポイントに転送する前にトグルの状態を初期化するため、CPUは、OUTデータについてはビットSNDTOG1～SNDTOG0 の 1 つを、INデータについてはRCVTOG1～RCVTOG0 の 1 つを設定します(59ページ)。これらのビットのペアを同時に 1 つだけ設定する必要があります。

CPU は、同じエンドポイントに対しての複数の**連続した**転送については、エンドポイントのトグル値を初期化する必要はありません。MAX3421E が、エンドポイントへの転送を実行するときにデータトグル値を更新します。CPU がエンドポイントを**切り替えた**ときにのみ、同じエンドポイントに最後に転送したときの値(すなわち、CPU が前回保存した値)にデータトグル値を復元する必要があります(RCVTOGRD ビットまたは SNDTOGRD ビットを読み出すことによる)。

BULK-OUT 転送のプログラミング

MAX3421E は、OUT パケット、SNDFIFO データ、およびハンドシェイクを使用して BULK データをペリフェラルに送信します。

CPUは、最初にSNDBAVIRQ = 1 (56ページ)であるかどうかをチェックし、送信バッファをロードに利用可能かどうかを確認します(SIEは、OUT転送が正常に完了するたびに、この割込み要求ビットをセットします)。バッファが利用可能な場合、CPUは、R2 の書込みを繰り返すことによって、SNDFIFO (58ページ)に最大 64 データバイトを書き込みます。次に、CPUは、SNDBCレジスタ(57ページ)にバイトカウント(CPUがSNDFIFOにロードしたバイト数)を書き込みます。SNDBCレジスタをロードすると、SIEはSNDBAVAIRQビットを非アクティブにします。

注:ダブルバッファ構造の SNDFIFO の 2 番目のバッファが空いている場合、SIE は直ちに SNDBAVIRQ ビットを再アサートします。

CPU は、所望のエンドポイントのデータトグル値を初期化することが必要となる*場合が*あります。

- 新しいエンドポイントへの転送を行う場合、CPU は、最後に保存したエンドポイントの値にデータトグルを初期化します。
- エンドポイントが前回の転送と同じ場合、CPU はデータトグル値を初期化する必要はありません。それぞれの転送の後に、SIE がデータトグルを更新します。

最後に、CPUはHXFRレジスタに値 0010eeeeをロードします(表 4)。ここで、eeeeは、データの送信先のエンドポイント番号です。HXFRレジスタはロードセンシティブです。つまり、CPUがHXFRレジスタをロードすると、SIEが転送を開始します。

SIE は OUT トークン、PERADDR レジスタのアドレス、EP[3:0]のエンドポイント番号、CRC5、および EOP を送信します。SIE は、この直後に(トグルビットの状態に応じて) DATA0 または DATA1 の PID、SNDFIFO の SNDBC バイト、および CRC16 を送信します。次に、SIE はペリフェラルの応答を 6.5 ビット時間だけ待ちます。

注:SNDBC = 0 の場合、SIE はデータバイトを送信しませんが、DATA0/1 の PID を送信します。

SIEは、ハンドシェイクまたはバスタイムアウトを受信すると、HXFRDNIRQ = 1 に設定し、HRSLT[3:0]ビット(36ページ)に結果を示します。

また、SIE は、ペリフェラルが ACK ハンドシェイクを返した場合には、データトグルの補数をとって、転送成功(HRSLT[3:0] = 0000)を示し、「送信バッファ利用可能」を示す SNDBAVIRQ をアサートします。

ペリフェラルが転送に対して肯定応答しない場合、データの再送信が必要となる場合があります。たとえば、NAK ハンドシェイクは、ペリフェラルがデータを受け入れる準備ができていないことを示すために使用する共通の応答です。CPU は HRSLT ビットを調べることによって、肯定応答されない OUT 転送の原因を見つけます。CPU は、HXFR=0010eeee を再書き込みするだけで、OUT 転送を再開します。SIE は PERADDR および SNDBC の同じ値を使用して、SNDFIFO 内の同じデータを送信します。

BULK-IN 転送のプログラミング

CPU は、IN トークンを発行することによって、BULK データを CPU に送信するようペリフェラルに要求します。次に、SIE は RCVFIFO にデータを転送し、転送に ACK 応答します。

CPUは、HXFR = 0000eeeeを書き込み(表 4)、IN転送を開始します。ここで、eeeeは、所望のエンドポイントのアドレスです。

SIE は、IN トークン、PERADDR レジスタのアドレス、EP[3:0]のエンドポイント番号、および CRC5 を送信します。次に、ペリフェラルの応答を 6.5 ビット時間だけ待ちます。ペリフェラルが DATA0 または DATA1 の PID に続いてデータで応答する場合、SIE は受信したデータバイトを RCVFIFO にロードし、バイトをカウントします。パケットの最後で、SIE はパケットにエラーがないかどうかを確認し、RCVBC レジスタおよび HRSLT ビットを更新してから、HXFRDNIRQ ビットをアサートします。転送結果に応じて、SIE は RCVDAVIRQ をアサートする場合とアサートしない場合があります。

IN データがエラーフリー(HRSLT = 0000)であった場合、SIE は ACK トークンを送信し、データトグルの補数をとり、RCVDAVIRQ をアサートして、新しい IN データが有効であることを示します。

IN データはエラーフリーだが、データトグルの不一致があった場合(ペリフェラルが送信した DATA0 または DATA1 の PID がエンドポイントのトグル値と一致しなかった)、SIE は ACK ハンドシェイクを送信しますが、データトグルの補数をとらず、また RCVDAVIRQ をアサートしません。SIE は、この状況を表す HRSL = 0110 (トグルエラー)を設定します。この状況は、ペリフェラルが前回の IN 転送から、破損した ACK ハンドシェイクを受信した場合に発生します。この場合、ホストは、RCVDATA FIFO のデータを無視します。RCVDATA FIFO のデータは、ペリフェラルが前回の ACK ハンドシェイクを取り損ねたときにペリフェラルが誤って再送したデータであるからです。SIE が転送に ACK 応答し、自身のトグルビットを更新しないようにすることによって、ペリフェラルはそのトグルビットの補数をとることになり、データトグルのメカニズムは同期に戻ります。

HRSLT ビットがデータエラーを示す場合、SIE は、ACK の送信、データトグルの補数の計算、あるいは RCVDAVIRQ ビットのアサートを行いません。

CPU は、HRSLT[3:0]の結果ビットを調べることによって、HXFRDNIRQ の指示に応答します。結果が 0000(成功)の場合、CPU は RCVBC を読み出してバイトカウントを確認してから、RCVFIFO レジスタの読出しを繰り返すことによって、指定のバイト数を読み出します(48 ページ)。CPU は、データを読み取った後、(1 を書き込むことによって) RCVDAVIRQ ビットをクリアします。ダブルバッファ構造の RCVFIFO に IN データのバッファがもう 1 つある場合、SIE は直ちに RCVDAVIRQ ビットを再アサートします。

注: SIE は、エラーを示す IN 転送を自動的に再試行することはありません。代わりに、SIE は (HXFRDNIRQ ビットおよび HRSLT ビットを介して)CPU にエラーを通知し、正しい USB 応答を生成します。通常、CPU は HXFR レジスタを再ロードすることによって、IN パケットを再送するだけです。

CONTROL 転送のプログラミング

ホストは、以下の 2 つあるいは 3 つのステージで CONTROL 転送を送信します。

1. SETUP パケット。ペリフェラルに「演算コード」の 8 バイトを送信します。
2. オプションのデータステージ。通常は BULK IN 要求です。
3. ステータスステージ

ホストは、ペリフェラルのデフォルトコントロールエンドポイントゼロに CONTROL 転送を送信します。

1. セットアップ

CPUは、8 バイトのSETUPデータをSUDFIFOに書き込みます。SETUPパケットのペイロードは常に 8 バイトであるため、SUDFIFOにはバイトカウントレジスタは付随していません。CPUは次に、HXFRレジスタに値 00010000 をロードし(表 4)、エンドポイントゼロにSETUPパケットを送信するようSIEに指示します。

SIEは、SETUP PID、PERADDRレジスタのアドレス、エンドポイント 0000、CRC5、およびEOPで構成されるSETUPパケットを送信し、続いて、SUDFIFOの 8 バイトを含んだDATA0 パケットを送信します。次に、SIEは、このデバイスの応答またはタイムアウトを 18 ビット時間だけ待ち、HXFRDNIRQビットをアサートし、HSRLTビット(36ページ)を更新することによって最終的に転送を終了します。USBの仕様には、ペリフェラルは必ずSETUPパケットに肯定応答する必要があると規定されています。

注:SIE は、内部データトグルの設定に関係なく、CONTROL 転送のさまざまなステージについて固定の DATA0 と DATA1 の PID トークンを送信します。

---

2. データ(オプション)

データステージが必要な場合、データステージは BULK-IN 転送または BULK-OUT 転送としてプログラムされます。Set_Address など一部の CONTROL 転送は、コマンドデータが SETUP パケットの 8 バイト内に納まるために、データステージは不要です。

3. ステータス

ステータスステージは、CONTROL 転送に特有なもので、これらの極めて重要な転送を保護する追加手段となります。ステータスステージは、直前のステージと反対方向の IN または OUT パケットで構成されます。

STATUS パケットは、データを含まず、DATA1 の PID を常に使用する BULK-OUT 転送または BULK-IN 転送です。プログラマは、標準的な BULK 転送をプログラミングすることによって、これらを転送することができますが、便宜上、MAX3421E はハンドシェイク用の特別な HS-OUT と HS-IN の開始コードを備えているため、SIE がこの作業を実施することが可能です。

HS-OUT

ホストはOUTハンドシェイクを送信し、Get_DescriptorなどのCONTROL-READ要求を終了します。OUTハンドシェイクのパケットを送信するには、CPUはHXFRレジスタに値 0xA0 をロードします(表 4)。

SIEは、OUT PID、PERADDRのアドレス、エンドポイントゼロ、およびCRC5 の値を送信します。SIEは、この直後にDATA1 のPIDを送信し、ペリフェラルの応答またはタイムアウトを 6.5 ビット時間だけ待ちます。これは、データを持たず、固定のDATA1 のPIを持つBULK-OUTパケットと同じであることに留意してください。SIEは、ハンドシェイクまたはバスタイムアウトを受信すると、HXFRDNIRQ = 1 に設定し、HRSLT[3:0]ビット(36ページ)に結果を示します。

HS-IN
ホストはINハンドシェイクを送信し、Set_AddressなどのCONTROL-WRITE要求を終了します。INハンドシェイクのパケットを送信するには、CPUはHXFRレジスタに値 0x80 をロードします(表 4)。

SIE は、IN PID、PERADDR のアドレス、エンドポイントゼロ、および CRC5 の値を送信してから、ペリフェラルの応答を 6.5 ビット時間だけ待ちます。次に、SIE は HRSLT ビットを更新し、HXFRDNIRQ ビットをアサートします。ペリフェラルが(データを持たない) DATA1 の PID を返して来た場合、SIE は CONTROL 転送の正常終了を示す ACK ハンドシェイクを自動的に送信します。

ISO 転送について

USB のアイソクロナス転送には、他のタイプの USB 転送に伴うハンドシェイクがない「オンタイムデリバリ」という特性があります。USB ホストがデバイスをエニュメレートし、その ISO 帯域幅要件(「帯域幅」とは、1 ミリ秒のフレームごとに ISO エンドポイントに割り当てられるバイト数を表す)を与えると、ホストは、USB の仕様によって、フレームごとにそのバイト数を供給または消費するよう要求されます。

これによって、ロースピードのSPIインタフェースまたはCPUに接続される可能性のあるMAX3421Eに関して特別な検討事項が生じます。すなわち、コントローラは、スケジューリングの要件に遅れないようにしなければなりません。速度が一致しない場合、MAX3421Eは、2 つの方法で問題を示すことができます。これはDELAYISOビットによって設定されます。

ダブルバッファリング

MAX3421E の SNDFIFO と RCVFIFO はダブルバッファ構造です。つまり、それぞれが 2 組の 64 バイト FIFO とバイトカウントレジスタを備えているということです。このダブルバッファリングは、1 パケットの最大サイズである 64 バイトより大きい ISO 転送を実施するためには不可欠です。ダブルバッファの機能はプログラマには意識されません。

- **IN**:CPU が、RCVFIFO のアンロード後に RCVDAVIRQ (Receive Data Available IRQ:受信データ利用可能 IRQ)ビットをクリアした場合で、もう一方のバッファで待機しているデータのパケットがもう 1 つあれば、SIE は RCVDAVIRQ ビットを直ちに再アサートします。

- **OUT**:CPU が、SNDFIFO に読み込んだバイト数を SNDBC レジスタにロードすることによって、SNDBAVIRQ (Send Buffer Available IRQ:送信バッファ利用可能 IRQ)をクリアした場合で、もう 1 つのバッファをロードに利用することができれば、SIE は SNDBAVIRQ ビットを直ちに再アサートします。

SNDFIFO および RCVFIFO のサイズは 64 バイトですが、CPU がデータを時間内に供給または消費する限り、SIE は、これらの FIFO に対して連続してロードするデータ、または連続して読み出したデータを 1 つの大きなデータパケットに連結することによって、任意のサイズ(USB 仕様の最大制限値 1023 バイト)の ISO データパケットを送受信することができます。

SIE は、EOP (End-Of-Packet:パケットの最後)のバス状況を検出することによって、ISO IN データパケットの最後を検出します。OUT 転送の場合、SIE は通常、CPU が FIFO に 64 バイト未満のデータをロードするときにデータペイロードの最後を検出します。これは、単一パケット(63 バイト以下)の場合でも、あるいは複数の 64 バイトバッファのロード(最後のバッファは 63 バイト以下)に及ぶデータペイロードでも機能します。ただし、データの最後のパケットがちょうど 64 バイトになる、特別なケースがあります。SIE は、これがマルチバッファのロードにさらなる 64 バイトが存在することを示すものなのか、あるいは長さが 64 バイトの整数

倍である転送の最後の 64 バイトなのかを判断する必要があります。CPU は、SNDBC = 0 をロードすることによって、SIE に 64 バイトのパケットが OUT 転送の最後であることを通知します。これは SIE への特殊な信号であるため、SNDBC レジスタがロードされるときにバッファを切り替えるという通常の処置は行われません。

ISO-IN 転送のプログラミング

CPU は、HXFR = 0100eeee を書き込んで、ISO IN 転送を開始します。

SIE は、PERADDR レジスタのアドレス、EP[3:0]のエンドポイント番号、CRC5、および EOP を使用した OUT トークンを送信します。次に、DATA0 の PID またはバスタイムアウトを 6.5 ビット時間だけ待ちます。SIE が DATA0 の PID を受信した場合、RCVFIFO へのデータのロードを開始します。FIFO が 64 バイトでいっぱいになったとき(または、end-of-packet 信号がバス上に示されたとき)、SIE は RCVBC にバイトカウントを書き込みます。また SNDBAVIRQ をアサートすることによって、必要に応じて CPU がさらにデータをロードするのに FIFO が利用可能であることを示します。ISO IN 転送の最後(EOP)に、SIE は HRSLT ビットを更新し、HXFRDNIRQ をアサートします。

ISO-OUT 転送のプログラミング

SIE マスタはペリフェラルに OUT パケットを送信した後、固定の DATA0 の PID を使用してデータパケットを送信します。3 つのケースを考慮する必要があります。

1. OUT データペイロードのサイズが 64 バイト未満の場合
2. 64 バイトの場合
3. 64 バイトを超える場合

1. 64 バイト未満の OUT ペイロード

CPUは、BULK転送と同様にこれをプログラミングします。CPUはSNDFIFOにバイトをロードしてから、SNDBCにバイトカウントを書き込みます。次に、HXFRレジスタに 0110eeee (表 4)を書き込んで転送を開始します。

SIE は、PERADDR レジスタのアドレス、EP[3:0]のエンドポイント番号、CRC5、および EOP を使用した OUT トークンを送信します。SIE はこの直後に、DATA0 の PID、SNDFIFO の SNDBC バイト、および CRC16 を送信します。次に、SNDBC < 64 であることを認識した上で、SIE は転送を終了し、SNDBAVIRQ をアサートします。ペリフェラルは、SIE がチェックするためのハンドシェイクを返信しません。結果として、CPU による転送スケジュールがフレーム内で極めて遅くなって自動的に生成される SOF パケットとの衝突を回避することができないため、パケットの切り捨てや取り損ないなどの重大なエラー状況が生じます。DELAYISO ビットは、このエラー状況に対する SIE の動作を制御します。

2. ちょうど 64 バイトの OUT ペイロード

このケースは、単一の 64 バイトパケットの場合に該当します。あるいは複数バッファによる転送の最終データペイロードがちょうど 64 バイトであった場合に該当します。CPU は SNDFIFO に 64 バイトをロードし、SNDBC = 64 を書き込み、SNDBAVIRQ = 1 を待ち、次に SNDBC = 0 を書き込んで転送の終了を示します。

3. 64 バイトを超える OUT ペイロード

CPU は SNDFIFO にデータの 64 バイトをロードしてから、SNDBC = 64 を書き込みます。SNDBAVIRQ = 1 の場合、CPU はデータパケットの 2 番目の部分を SNDFIFO にロードすることが可能で、その後、再び SNDBC レジスタにバイトカウントを書き込みます、転送を開始するには、CPU が HXFR レジスタに 0110eeee をロードします。

FIFO が利用可能になると、SIE は SNDBAVIRQ のアサートを継続し、ISO OUT データの次のチャンクを要求します。ダブルバッファリングによって、SIE が一方の FIFO から OUT データを送信し、同時に CPU が他方の FIFO にロードすることが可能になります。

SIE は、HRSLT ビットを更新し、HXFRDNIRQ をアサートすることによって、ISO OUT 転送を終了します。SIE が、関連 SNDBC が 64 未満である FIFO の最終バイトを送信した後、または CPU が SNDBC = 0 をロードしたときに、ISO OUT 転送は終了します。

ペリフェラル/ホストモードのアクティブなビット

いくつかの MAX3421E の機能はホストとペリフェラルの両方の動作で利用することができます。これらの機能は、SPI マスタの構成方法、割込みピンの動作方法、および IO ピンの状態など、非 USB システムのさまざまな面に関連します。これらの機能は、以下に示すようにグループ分けすることができます。

- インタフェースの構成
  - FDUPSPI
  - GPX[B:A]
  - INTLEVEL
  - POSINT
- IO ピンの構成および出力値
  - GPIN ピン。割込みの極性とイネーブル
  - GPOUT ピンの状態
  - VBCOMP ピンの検出レベル
- グローバルチップの動作
  - CHIPRES
  - PWRDOWN
  - OSCOK (Oscillator OK：発振器 OK)
- 特定の IRQ ビットおよび IE ビット

表 2は、SPIマスタがMAX3421Eのモード変更(ペリフェラルからホストへ、またはホストからペリフェラルへのいずれか)を命令するときに保持されるレジスタおよびビットを示しています。したがって、SPIインタフェースの構成やGPOUTピンの値は、モード変更によって影響を受けません。IEビットとGPINの割込みビット(GPINIRQおよびGPINIE)は、モード変更を行っても保持されます。つまり、モード変更前にGPIN割込みが保留にされている場合、モード変更後も保留のままということです。

注：ホストからペリフェラルの動作に切り替えるよう MAX3421E に命令する方法は 2 つあります。1 つ目は、SPI マスタが MODE レジスタ R27 の HOST ビットに 0 を書き込む方法です。2 つ目は、SPI マスタが USBCTL レジスタ R16 の CHIPRES ビットをセットしてからクリアする方法です。「クリーン」なペリフェラルで開始したい場合は、CHIPRES を使用する方法をお勧めします。

BUSEVENTIRQ、BUSEVENTIE

意味: **BUSEVENTIRQ**: 2 つのバスイベントのうちの 1 つが発生した
BUSEVENTIE: BUSEVENTIRQ をイネーブルにする

モード: **ホストのみ**

SIE は、以下の 2 つの USB バスイベントのうちの 1 つの信号送出を完了すると、BUSEVENTIRQ ビットをセットします。

- バスリセット(BUSRST が 1→0 のとき)。
- バスレジューム(BUSRSM が 1→0 のとき)。

CPU は、1 を書き込むことによって BUSEVENTIRQ ビットをクリアします。

CPU は BUSEVENTIE ビットをセットおよびクリアします。BUSEVENTIE = 1 のとき、BUSEVENTIRQ は、INT ピンをアクティブにするソースとしてイネーブルになります。

プログラミング上の注意:
この割込みは、バスリセット信号またはバスレジューム信号の送出完了という 2 つのソースによって共有されます。これは共有割込みであるため、2 つのバス信号のいずれかを開始する前に(BUSRST = 1 または BUSREM = 1 に設定する前に)、BUSEVENTIRQ ビットをクリアするのが賢明な方法です。

BUSRST

意味: USB ペリフェラルにバスリセットを発行する

モード: **ホストのみ**

CPU は、このビットをセットして、ペリフェラルに 50 ミリ秒のバスリセット(SE0)信号を開始します。

SIE は、バスリセット信号送出の最後にこのビットをクリアします。
CPU も、このビットをクリアして、50 ミリ秒の SE0 バス信号を早期に終了する場合があります。この方法でリセット信号を終了すると、BUSEVENTIRQ がアサートされます。

プログラミング上の注意:
CPU は、このビットをセットして、D+ラインと D-ライン上にバスリセットを発行するよう SIE に指示します。CPU は、このビットをセットした後、BUSRST ビットをポーリングして 0 を確認するか、BUSEVENTIRQ に応答することによって、50 ミリ秒間隔の終了を検出することができます。バスリセットをプログラミングするには、以下の一連の手順を実行します。

1. BUSRST = 1 に設定します。
2. BUSRST = 0 かどうかをテストするか、あるいは BUSEVENTIRQ に応答します。
3. SOFKAENAB = 1 に設定することによって、フレームマーカをオンにします。
4. 少なくとも 1 つの FRAMEIRQ を待ちます。

ステップ 4 は、ホストのロジックがホストの最初のトランザクションを生成する準備ができていることを保証するものです。

CHIPRES

意味: チップリセット

モード: **ペリフェラルおよびホスト**

CPU は、このビットをセットして、チップをリセットします。この効果は、RES#ピンをローに駆動するのと同じです。CPU は、このビットをクリアして、チップのリセットを解除します。

プログラミング上の注意:
CPU は、このビットをセットした直後にクリアすることができます。パワーオン時または CHIPRES ビットをセットすることによって MAX3421E をリセットすると、HOST ビットを含むほとんどのレジスタビットがクリアされるため、MAX3421E は USB ペリフェラルとして動作するようセットアップされます。

CHIPRES = 1 に設定すると、内部発振器が停止します。CHIPRES = 0 に設定することによってリセットを解除した後、CPU は OSCOK 割込み要求を確認する必要があります。CPU は、発振器と PLL が安定していることを示すこの割込みがアサートされた後でのみ、次に進むことができます。

CONDETIRQ、CONDETIE

意味: **CONDETIRQ:** ペリフェラルの接続/切断の割込み要求
CONDETIE: ペリフェラルの接続/切断の割込みイネーブル

モード: **ホストのみ**

SIE は、CONDETIRQ ビットをセットして、ペリフェラルが接続されたこと、あるいは接続が切断されたことを示します。CPU は、1 を書き込むことによって CONDETIRQ ビットをクリアします。CONDETIRQ ビットをアクティブにする条件は、以下のとおりです。

- バスが J 状態または K 状態から 25 マイクロ秒の SE0 に遷移したとき。これは、ペリフェラルの接続が切断されたことを示します。
- バスが 8 SE0 ビット時間から 25 マイクロ秒の J 状態または K 状態に遷移したとき。これは、ペリフェラルが接続されたことを示します。

CPU は CONDETIE ビットをセットおよびクリアします。CONDETIE = 1 のとき、CONDETIRQ は INT ピンをアクティブにするソースとしてイネーブルになります。

プログラミング上の注意:
ペリフェラルの接続または切断を検出するには、CPU は最初に HOST = 1 に設定して、MAX3421E をホストモードにし、DPPULLDN = 1 および DPPULLDN = 1 に設定して、D+信号および D-信号のロジックローレベルを確立する必要があります。

2 つのイベント(接続または切断)のいずれが発生したかを判断するには、CPU は HSRL レジスタを読み出し、JSTATUS ビットと KSTATUS ビットを確認します。通常、これらのステータスビットは、CPU が SAMPLEBUS ビットをセットすると更新されますが、CONDETIRQ 割込みがアサートされるときにも自動的に更新されます。

DELAYISO

意味: 次のフレームまで(次の SOF パケットの送信が完了するまで)、ISOCHRONOUS エンドポイントに対するデータ転送を遅延する。これは、パケットのスケジューリングがフレーム内で非常に遅いため、自動的に生成される次の SOF パケットとの衝突を回避することができない場合に行われます。

モード: **ホストのみ**

CPU がこのビットをセットおよびクリアします。

プログラミング上の注意:
USB の仕様では、ホストが、エニュメレートされた ISO エンドポイントに、1 ミリ秒ごとのフレームで ISO データを転送することができるだけのバス帯域幅(フレーム時間)を与えることを保証しています。つまり、ホストは、あらゆるフレーム内で *十分に早く* IN または OUT パケットを送信し(HXFR レジスタに書き込むことによる)、MAX3421E が自動的に次の SOF パケットを生成する前に転送が完了することができるようにしなければなりません。

MAX3421E はあらゆる速度の SPI マスタに接続することができますが、制御ファームウェアが、配信を保証できるほど、あらゆるフレーム内で十分に早く HXFR レジスタに書き込めるという保証がないため、パケットのスケジューリングがフレーム内で遅すぎるとき、SIE は DELAYISO ビットを提供して動作を制御します。パケット/SOF のタイミングが衝突したとき、DELAYISO ビットは、パケットデータと SOF パケットのいずれを優先すべきかを判断します。

DELAYISO = 1 の場合、CPU が HXFR レジスタを読み込むことによって ISO パケットを開始するたびに、SIE は 1 ミリ秒のフレーム内で残っている利用可能な時間をチェックします。SIE は、256 バイトの ISO パケットを送信または受信するのに十分な時間がフレームにないとわかった場合、次のフレームの開始まで ISO パケットの送信を遅延します。

注:一般的な ISO アプリケーションである CD 品質のオーディオに対応するため、256 バイトのウィンドウを選択しました。このアプリケーションでは、毎秒 44.1K サンプル x 2 チャネル x 16 ビット(チャネル当り)となり、ミリ秒フレーム当り 176 バイトが必要です。256 バイトのウィンドウならば、多少の余裕があります。

ISO OUT-SOF の衝突

送信中の ISO の OUT パケットデータと自動的に生成される SOF パケットとのスケジューリングの衝突に SIE が対処する方法は 2 つあります。SIE は、パケットの送信を開始してからデータの送信を途中で終了して SOF を生成するか、次の SOF パケットの生成が完了するまでパケット全体の送信を遅延するかです。

注:もう 1 つの ISO のエラー状況はデータのアンダランであり、結果コード HRSLT = 0x06 で示されます。データのアンダランは、ISO データパケットの送信中に SIE が SNDFIFO データを必要とし、ただし CPU が SNDFIFO を時間内にロードしなかった場合に発生します。

DELAYISO = 0 の場合の ISO OUT

SIE が次の SOF パケットを生成するときに ISO-OUT データパケットの転送が進行中の場合、SIE はデータパケットを切り捨てて SOF を生成します。SIE は、HRSLT[3:0] = 0x03 を設定し、その後に FRAMEIRQ を設定することによって、このデータ喪失の状況を示します。この場合、SIE は HXFRDNIRQ をアサートしません。

DELAYISO=1 の場合の ISO OUT

フレームの終了の前に SIE が 256 バイトを転送することができないときに CPU が HXFR レジスタに書き込んだ場合、SIE は OUT データのパケットの送信を遅延します。この場合、SIE は次の SOF パケットを送信します(および FRAMEIRQ をアサートします)。次に遅延された ISO パケットを送信し、転送を終了して、コード 0x00 (hrSUCCESS)で HXFRDNIRQ をアサートします。CPU は、FRAMEIRQ の直後に SNDBAVIRQ ビットをサンプリングすることによって、遅延されたパケットを検出します。SNDBAVIRQ が 0 の場合、SNDFIFO バッファは引き続き USB 転送に使用されるため、パケットは遅延されます。

ISO IN-SOF の衝突

SIE は、OUT 転送の場合とは少し異なる方法で IN 転送の衝突を処理します。その理由は、ホストがペリフェラルに対して、データの送信を停止して次の SOP パケットとの衝突を回避するよう命令することができないからです。したがって、衝突状況では、SIE は残りの ISO IN データを読み出して、SOP パケットを生成しないか、あるいは次の SOP パケットの生成が完了するまで IN 要求の送信を遅延します。

注:もう 1 つのエラー状況はデータのオーバランであり、結果コード HRSLT = 0x06 で示されます。データのオーバランは、利用可能な RCVFIFO がないため(CPU が時間内に RCVFIFO をアンロードしていない)、SIE が利用可能な ISO IN データがあっても、これを出力する場所がない場合に発生します。

DELAYISO = 0 の場合の ISO IN

ISO-IN 要求の結果、SOP 生成時間を超えるデータパケットをペリフェラルが送信した場合、SIE は引き続き IN データを受信し、そのフレームの SOP パケットの送信を抑制します。この場合、SIE は HRSLT[3:0] = 0x03 を設定し、IN 転送の完了時に HXFRDNIRQ をアサートしないで、そのフレームの FRAMEIRQ をアサートします。CPU は、HXFRDNIRQ がない FRAMEIRQ の発生を認識することによって、このエラーケースを検出することができます。

注:SIE は、SOP パケットが実際に送信されたかどうかにかかわらず、必ずフレームカウントをインクリメントします。

DELAYISO = 1 の場合の ISO IN

次の SOP と衝突するまでに 256 バイトのデータパケットを受信することができないときに CPU が ISO-IN 転送の準備を完了した場合、SIE は次のフレームまで IN パケットの送信を遅延します。この場合、SIE は次のフレームの FRAMEIRQ をアサートし、ISO-IN パケットを送信し、IN データを転送して、HRSLT[3:0] = 0x00 (hrSUCCESS)で HXFRDNIRQ をアサートします。

注:ISO のスケジューリング問題が検出された場合、おそらく SPI インタフェースの速度向上、ファームウェアの調整、あるいはその両方を行うことによって、各フレームで ISO のパケットをより早く送信するシステムを変更する必要があります。

DPPULLDN、DMPULLDN

意味: 15kΩ の内部抵抗器を D+および D-からグランドに接続する

モード: **ホストのみ**

CPU がこれらのビットをセットおよびクリアします。

プログラミング上の注意:
USB ホストは、これらの 2 つのプルダウン抵抗器を接続することによって、バスを休止状態にします。ロースピードのペリフェラルは D-から 3.3V に 1500kΩ を接続しており、フルスピードのペリフェラルは D+から 3.3V に 1500kΩ を接続しています。バスが弱く 15kΩ の抵抗でプルダウンされるため、SIE は、ペリフェラルが接続されたことだけではなく、その速度も検出することができます。

MAX3421E がペリフェラルとして動作している場合(HOST = 0)、これらのビットは両方とも 0 (デフォルト値)に設定する必要があります。

63ページで、2 つのUSBコネクタを実装し、ペリフェラルホストとして自動的に接続を検出する設計におけるDPPULLUPビットとDMPULLUPビットの役割について説明します。

FDUPSPI

意味: フルデュープレクス SPI ポートの動作

モード: **ペリフェラルおよびホスト**

CPU は、このビットをセットして、フルデュープレクスモードで SPI ポートを動作させます。
CPU は、このビットをクリアして、ハーフデュープレクスモードで動作させます。

パワーオンリセット: FDUPSPI = 0 (ハーフデュープレクス)。
チップリセット: *変化なし。*
バスリセット: *変化なし。*
パワーダウン: 読出し-書込み。

プログラミング上の注意:
ハーフデュープレクス SPI
ハーフデュープレクスモード(FDUPSPI = 0)では、MOSI (マスタアウトスレーブイン)ピンは双方向の IO ピンになり、MISO (マスタインスレーブアウト)ピンはトライステートになります。

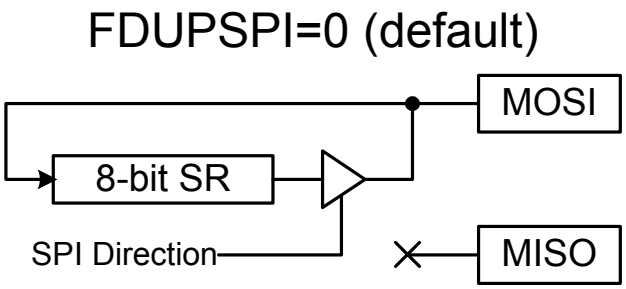


図 3. ハーフデュープレクス SPI インタフェース

フルデュープレクス SPI
フルデュープレクスモード(FDUPSPI = 1)では、MOSI ピンと MISO ピンが独立して用意されます。この構成には付加機能があり、各転送の最初のバイト(コマンドバイト)が同期入力されると同時に、ステータス情報の 8 つのビットが同期出力されます。

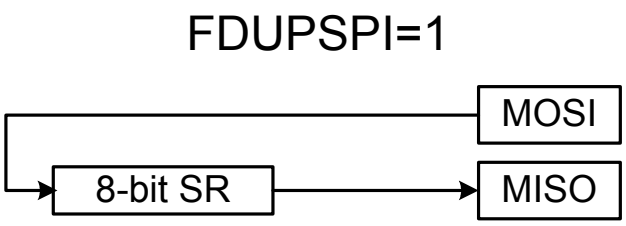


図 4. フルデュープレクス SPI インタフェース

SPI コマンドバイト
SPI インタフェースに同期入力される最初のバイトはコマンドバイトであり、レジスタのアドレスと方向を設定し、また USB ペリフェラルとして動作するときは、ACKSTAT ビットをじかにセットするビットを設定します。すべての SPI トランザクション(入力または出力)において、ビットの順序は、最初が b7 で最後が b0 です。

MOSI ビット	信号
7	REG4
6	REG3
5	REG2
4	REG1
3	REG0
2	0
1	方向(1 = 書込み、0 = 読出し)
0	ACKSTAT (ペリフェラル)

図 5. SPI コマンドバイト

SPI サイクルは、最初に SPI マスタが CS#をローに駆動することから始まり、次に 8 つの SPI クロックが駆動されます。このクロックの立上りエッジで、コマンドバイトがストローブされます。REG[4:0]はレジスタアドレスを設定し、方向ビットは SPI サイクルの読出しまたは書込みの方向を設定します。USB ペリフェラルの動作の場合、ACKSTAT ビットは、EPSTALLS レジスタの対応するビットを書き込みます。

コマンドバイトに続いて、SPI マスタは、8 つの SCK クロックからなる群を 1 つ以上送出して、MAX3421E に対してバイトデータを同期入力または同期出力します。FIFO にアクセスするとき、CS#がローの状態である限り、コマンドとともに同期入力されるレジスタアドレスは引き続き有効になります。バイトをまとめて送信するこの機能は、エンドポイント FIFO を読み出す、または書き込むときに便利です。たとえば、EP0FIFO に 37 バイトをロードするには(ペリフェラルモード)、SPI マスタはコマンドバイト 00000010 を書き込みます。これによって、書込み動作(方向ビットが 1)のための R0 (EP0FIFO)が選択されます。次に、SPI マスタは SPI ポートに 37 バイトを書き込み、最後に CS#をハイに駆動して SPI サイクルを終了します。

注:MOSI および MISO のデータはどちらも SCK の立上りエッジでサンプリングされます。データは SCK の立下りエッジで変化します。

SPI サイクルは、SPI マスタが CS#をハイ状態に戻すと終了します。

SPI モード
SPI 規格は、CPOL (クロック極性)および CPHA (クロックフェーズ)と呼ばれる 2 つのモード信号を反映した、4 つのクロックモードを規定しています。これらの信号は、(CPOL,CPHA)という書式で表現されます。これによって、インタフェースは正のエッジの SCK と MOSI の両方のデータが最初の正のクロックエッジの前に利用可能になることを想定しており、変更することなくモード(0,0)と(1,1)で動作可能であることがわかります。この特性によって、MAX3421E は、モードピンを必要とすることなくモード(0,0)または(1,1)で動作することが可能となります。

以下のスコープ波形は、マイクロプロセッサと MAX3421E 間の同一のデータ転送を示しています。**図 6** は SPI モード(0,0)を、**図 7** は SPI モード(1,1)を使用しています。違いは、SCK 信号の非アクティブレベルで

り、モード(0,0)ではローが、モード(1,1)ではハイが非アクティブレベルになります。どちらのモードでも、MOSIおよびMISOのデータはSCKの立上りエッジによってサンプリングされるため同じになります。

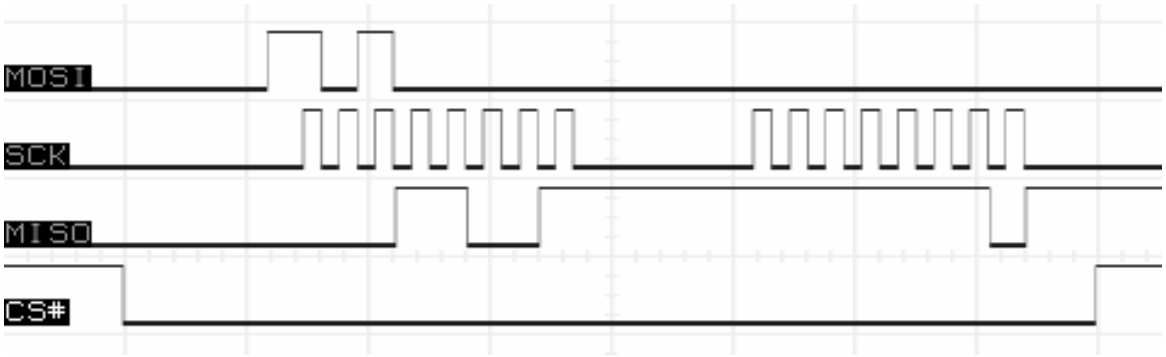


図 6. モード(0,0) で動作する SPI インタフェース

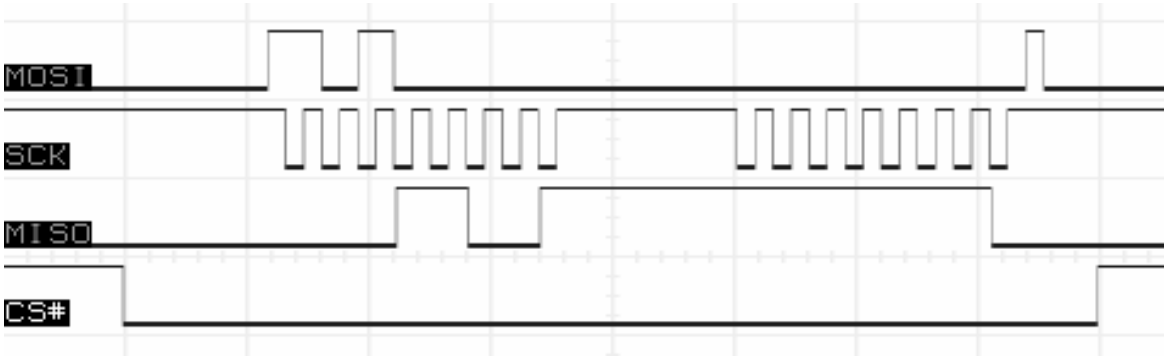


図 7. モード(1,1) で動作する SPI インタフェース

FRAMEIRQ、FRAMEIE

意味: **FRAMEIRQ:** フレームジェネレータの割込み要求
FRAMEIE: フレームジェネレータの割込みイネーブル

モード: **ホストのみ**

SIE は、1 ミリ秒のフレームマーカを生成するたびに FRAMEIRQ ビットをセットします。フレームマーカは、フルスピードの SOF パケットまたはロースピードのキープアライブパルスで構成されます。CPU は 1 を書き込むことによって FRAMEIRQ ビットをクリアします。

CPU は、FRAMEIE ビットをセットおよびクリアします。FRAMEIE = 1 のとき、FRAMEIRQ は INT ピンをアクティブにするソースとしてイネーブルになります。

プログラミング上の注意:
フルスピードのとき、SIE は SOF パケットの最初に FRAMEIRQ 割込みをセットします。

FRMRST

意味: SOF フレームカウンタをリセットする

モード: **ホストのみ**

CPU は、このビットをセットして、フレームカウンタをクリアします。
SIE は、このビットをクリアして、フレームカウンタリセットが終了したことを示します。

プログラミング上の注意:
このビットは、MAX3421E がフルスピードのホストとして動作しているときのみ意味があります(ロースピードのホストとして動作しているときには、SOP パケットやフレームカウントはありません)。FRMRST = 1 に設定した後、次の SOF パケットに 0 のフレームカウントが含まれます。

MAX3421E は、内部プリスケーラを使用してフレームカウンタをクロック制御します。FRMRST = 1 に設定することによって、フレームカウントはリセットされますが、プリスケーラはリセットされず、1 ミリ秒ごとに SOF パケットの規則的なストリームが保証されます。

GPIN(0-7)、GPINPOL(0-7) GPINIRQ(0-7)、GPINIE (0-7)

意味:

GPIN	汎用入力ピン 0～7
GPINPOL	汎用 IN 割込み極性 0～7
GPINIRQ	汎用 IN 割込み要求 0～7
GPINIE	汎用 IN の割込みイネーブル 0～7

モード: **ペリフェラルおよびホスト**

GPIN

CPUは、GPINレジスタビットを読み出すことによって、GPINピンの状態を読み出します。外部回路のみがこれらのレジスタビットの状態を制御することができます。8 つのGPINピンはすべて、弱い(約 20kΩ)内部抵抗によってV_{CC}にプルアップされます。

GPINPOL

CPUは、GPINPOLビットをセットして、以下の表にしたがって、各GPINピンに対する割込み要求のエッジ極性を設定します(**表 5**) 。これらのビットはパワーオン時にクリアされます。これらのビットは、CHIPRES = 1 のとき、またはモード変更の後に、その状態を保持します。

表 5. GPIN 割込みのエッジ極性

GPINPOL	極性
0	負エッジ
1	正エッジ

GPINIRQ

GPINピン上の信号が正または負に遷移するとき、MAX3421EはGPIN割込み要求ビットをセットします。GPINPOLビット(表 5)は、アクティブエッジ極性を制御します。GPINIRQビットは、個々のGPIN割込みがイネーブルであろうとなかろうと、あるいはIEビットがセットされているかどうかにかかわらず、アクティブです。

GPINIE

CPU は、GPIN の割込みイネーブルビットをセットして、対応する IRQ ビットを割込みロジックにパススルーして MAX3421E の INT ピンに供給します。IE = 1 の場合、イネーブルの IRQ は、INT ピン上に表われるか(SEPIRQ = 0 の場合)、あるいは GPX ピン上に表われます(SEPIRQ = 1 かつ GPX[B:A] = 10 の場合)。

注: 通常の動作中、GPIN割込みは、INTピン上に示される他のすべてのMAX3421Eの割込みソースとともにOR接続されます。8 つのGPIN割込み要求ビットをINTピンのソースとして分離し、これらをGPXピン上のグループとして個別に利用可能にすることもできます。このモードでは、GPXピンはMAX3421Eの 2 つ目のINT出力ピンとしての役割を果たします。このモードは、外部イベントの検出時間を最小限にする必要のあるシステムで好まれて使用されます。SEPIRQビットによって、割込みソースのこのような分離が可能になります。

GPOUT(0 through 7)

意味: 汎用出力ピン 0～7

モード: **ペリフェラルおよびホスト**

CPU がこれらのビットをセットおよびクリアします。

プログラミング上の注意:
CPU は、これらのビットを書き込んで、GPOUT ピンの状態を制御します。出力電圧は、VL ピン上の電圧を基準とします。CPU は、これらのビットを読み出すこともできます。このビットを読み出すと、出力バッファを駆動する出力フリップフロップの状態を示します。したがって、ロジックレベルを不安定にするような大きな負荷(たとえば LED)を出力ピンが駆動している場合、CPU は出力ピンの正しいロジック状態を読み出します。

GPXB、GPXA

意味: 2 つのビット GPXB および GPXA で GPX ピンの出力を決定する

モード: **ペリフェラルおよびホスト**

CPU がこれらのビットをセットおよびクリアします。

プログラミング上の注意:
GPX ピンは、以下の表に示すように、4 つの内部信号のうちの 1 つを出力することができます。

GPXB	GPXA	GPX Pin
0	0	OPERATE (内部 POR の補完)
0	1	VBUS 検出
1	0	BUSACT または INIRQ*
1	1	SOF (SOF パケットが到着したときに 0 から 1 に遷移。50%のデューティサイクル信号)

** IF SEPIRQ = 1 の場合*

GPXB = 1 かつGPXA = 0 のときのGPX出力のデフォルト設定はBUSACTです。ただし、SEPIRQビットを 1 に設定した場合、BUSACT信号は、8 つのGPINピンのうちの 1 つが 0 から 1、または 1 から 0 に遷移するたびに、アクティブな割込み要求信号に置き換えられます。この場合、GPXピンは(INTとともに) 2 つ目の割込みピンとしての役割を果たし、構成(レベルまたはエッジ、エッジ極性)はINTピンと同じです。SEPIRQビットの詳細については、53ページを参照してください。

HOST

意味: MAX3421E のホストまたはペリフェラルの動作

モード: **ペリフェラルおよびホスト**

CPU は、HOST = 1 に設定し、MAX3421E を USB ホストとして動作させます。
CPU は、HOST = 0 に設定し、MAX3421E を USB ペリフェラルとして動作させます(デフォルト)。

プログラミング上の注意:
パワーオン時、または RES#ピンのリセット後は、HOST ビットは 0 に設定され、MAX3421E はデフォルトのペリフェラル動作になります。このモードでは、MAX3421E は MAX3420E ペリフェラル専用コントローラとして動作します。

CPUは、HOST = 1 に設定して、MAX3421Eをホストとして動作させます。表 1に示すレジスタセットを使用します。

HOST = 0 のときの動作に関するプログラミング情報は、『*MAX3420E プログラミングガイド*』に記載されています。参考のため、ペリフェラルとホストの両方の動作についてのMAX3421Eの全レジスタビットを表 2に示します。

HRSL Register
HRSLT[3:0]
SNDTOGRD、RCVTOGRD

意味: これらのビットは、ホスト転送の結果を示す

- **HRSLT[3:0]**は、結果コードを示します。
- **SNDTOGRD** および **RCVTOGRD** は、それぞれ OUT 転送および IN 転送の結果として得られるデータトグル値を示します。

モード: **ホストのみ**

SIE がこれらのビットをセットおよびクリアします。

プログラミング上の注意:
SIEは、ホスト転送の完了時にこれらのビットを更新します。CPUは、HXFRDNIRQを受信した後、またはHRSLレジスタをポーリングし、hrBUSY以外のHRSLT値を読み出した後、HRSLレジスタを読み出します。**表 6**に示すように、HRSLTビットはホスト転送の結果を示します。

表 6. HRSLT[3:0]コード。

HSRLT	ラベル	意味
0x00	hrSUCCESS	正常な転送
0x01	hrBUSY	SIEはビジー。転送は保留状態
0x02	hrBADREQ	HXFRレジスタの不正な値
0x03	hrUNDEF	(予備)
0x04	hrNAK	ペリフェラルがNAKを返した
0x05	hrSTALL	ペリフェラルがSTALLを返した
0x06	hrTOGERR	トグルエラー/ISOオーバ(アンダ)ラン
0x07	hrWRONGPID	誤ったPIDの受信
0x08	hrBADBC	不正なバイトカウント
0x09	hrPIDERR	受信PIDが衝突している
0x0A	hrPKTERR	パケットエラー(スタッフ、EOP)
0x0B	hrCRCERR	CRCエラー
0x0C	hrKERR	応答がなくK状態
0x0D	hrJERR	応答がなくJ状態
0x0E	hrTIMEOUT	デバイスが時間内に応答しなかった
0x0F	hrBABBLE	デバイスの通話が長すぎる

SNDTOGRD (OUT転送の場合)およびRCVTOGRD (IN転送の場合)は、転送によって得られるデータトグル値を示します。CPUは、同じエンドポイントに対する転送の連続シーケンスを終了するたびに、これらの値を読み出して格納します。この結果、CPUは、次回に同じエンドポイントにデータを転送するとき、トグル値を復元することができます。CPUは、HCTLレジスタ内のSNDTOG0/1 ビットとRCVTOG0/1 ビット(59ページ)を使用して、エンドポイントのトグル値を初期化します。

HUBPRE

意味: USB ハブを通じて動作している LS デバイスに PRE PID を送信する

モード: **ホストのみ**

CPU がこのビットをセットおよびクリアします。

プログラミング上の注意:
ホストのファームウェアが、USB ハブを通じてロースピードのペリフェラルと通信していることを(エニュメレーション中に)検出した場合、HUBPRE = 1 を設定します。これは、SIE に、あらゆるロースピードパケットの前にフルスピードの PRE PID を送信し、USB の仕様で要求される必須の信号を送出するよう指示します。

HXFR レジスタ
EP[3:0]
HS、ISO、OUTNIN、SETUP

意味: CPU は、このレジスタに書き込んで、ホスト転送を開始する

モード: **ホストのみ**

CPU がこれらのビットをセットおよびクリアします。

プログラミング上の注意:
このレジスタは「ロードセンシティブ」です。つまり、CPU がこのレジスタに書き込むと、SIE が転送を開始します。CPU は、**表 7** に示す値をロードし、さまざまなタイプの USB 転送を開始します。

表 7. さまざまなタイプの転送のための HXFR レジスタビットの設定転送タイプ

Xfr Type	HS	ISO	OUTNIN	SETUP	hex
SETUP	0	0	0	1	10
BULK-IN	0	0	0	0	0-ep
BULK-OUT	0	0	1	0	2-ep
HS-IN	1	0	0	0	8-ep
HS-OUT	1	0	1	0	A-ep
ISO-IN	0	1	0	0	4-ep
ISO-OUT	0	1	1	0	6-ep

BULK-IN と BULK-OUT のエントリは、BULK または INTERRUPT のエンドポイントへの転送を求めます。これらの 2 つのタイプの転送は同一ですが、唯一違うのは、ホストのファームウェアが転送をスケジュール設定する時期です。

「hex」列の「-ep」フィールドは、EP[3:0]の値を示します。

これらの各転送タイプの詳細については、10ページの「ホスト転送のプログラミング」を参照してください。

HXFRDNIRQ、HXFRDNIE

意味: **HXFRDNIRQ:** ホスト転送完了の割込み要求
HXFRDNIE: ホスト転送完了の割込みイネーブル

モード: **ホストのみ**

SIE は、ホスト転送を完了すると、HXFRDNIRQ ビットをセットします。CPU は 1 を書き込むことによって HXFRDNIRQ をクリアします。

CPU は、HXFRDNIE ビットをセットおよびクリアします。HXFRDNIE = 1 のとき、HXFRDNIRQ は INT ピンをアクティブにするソースとしてイネーブルになります。

プログラミング上の注意:
この割込みがアサートされると、CPUは、HRSLレジスタのHSRLTビット(36ページ)を読み出すことによって、ホスト転送の結果を確認します。

IE

意味: INT ピンをイネーブルにする

モード: **ペリフェラルおよびホスト**

CPUは、このビットをセットして、INT出力ピンをアクティブにします。INT出力ピンの特性は、INTLEVELビット、POSINTビット、およびPULSEWID[1:0]ビットによって決まります(41ページ)。

CPU は、このビットをクリアして、INT 出力ピンをディセーブルにします。

プログラミング上の注意:
IE = 0 のとき、INT ピンの状態は非アクティブです(レベルモードの場合はオープン、負エッジの場合はハイ、正エッジの場合はロー)。

内部 IRQ ビットは、IE ビットの状態とは独立して動作します。IE ビットは、INT 出力ピンのアクティブ化のみを制御します。

INTLEVEL
POSINT
PULSEWID1、PULSEWID0

意味: **INTLEVEL**: INT 出力ピンをレベルアクティブ(1)またはエッジアクティブ(0)に設定する
POSINT: エッジアクティブな INT ピンのエッジ極性
PULSEWID: これらの 2 つのビットは、エッジモードでの IRQ の非アクティブ時間を設定する(以下を参照)。

モード: **ペリフェラルおよびホスト**

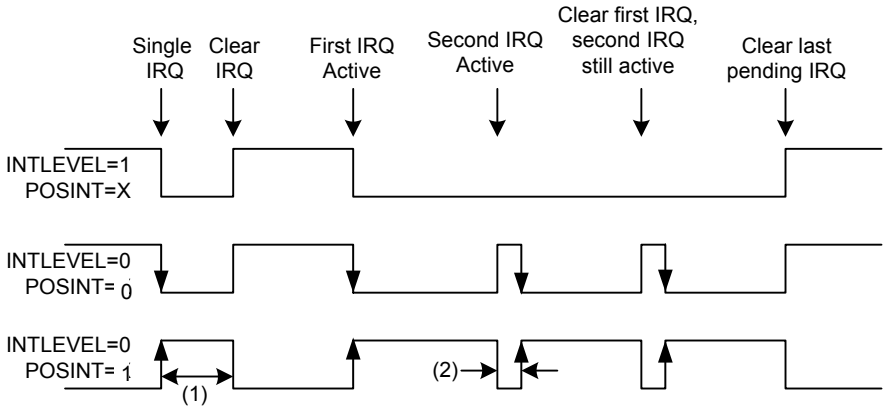
INTLEVEL
CPU は、INTLEVEL ビットをセットして、INT 出力ピンをレベルアクティブにします。INTLEVEL= 1 のとき、1 つ以上のイネーブルの割込みが保留中の場合、INT ピンはローに駆動します。このモードでは、INT ピンはオープンドレインであるため、システムは VL へのプルアップ抵抗器を搭載する必要があります。

CPU は、INTLEVEL ビットをクリアして、INT ピンをエッジアクティブにします。INTLEVEL = 0 のとき、エッジ極性は POSINT ビットによってセットされます。エッジ出力モードでは、INT ピンのドライバはプッシュ/プルであるため、VL へのプルアップ抵抗器は不要です。

POSINT
このビットは、CPU が INT ピンをエッジアクティブ(INTLEVEL = 0)にプログラミングしたときにのみ有効です。POSINT = 1 のとき、INT ピンは正エッジの保留状態の割込み信号を送出し、POSINT = 0 のとき、INT ピンは負エッジの保留状態の割込み信号を送出します。

PULSEWID1[0]
これらのビットは、CPUがINTピンをエッジアクティブ(INTLEVEL = 0)にプログラミングしたときにのみ有効です。これらのビットは、CPUによって 1 つのIRQビットがクリアされてから、1 つ以上の保留状態の割込みによってINTピンが再アサートされるまでの、INTピンが非アクティブな時間間隔を制御します(**図 8**)。

プログラミング上の注意:



(1)幅は IRQ をクリアすることによって決まります。(2)幅は PULSEWID[1:0]によって決まります。

図 8. INTLEVEL ビットと POSINT ビットに依存する INT ピンの動作

図 8の波形は、INTLEVELビットとPOSINTビットの各設定におけるINTピンの動作を示しています。

レベルモード(一番上の図)では、保留中の割込み要求がなくなるまで、INT ピンはローのままです。INTLEVEL モードはオープンドレインであるため、INT ピンから VL へのプルアップ抵抗器が必要です。

エッジモードでは、新しい割込み要求が発生すると必ず、あるいはある割込み要求ビットが他の割込み要求の保留中にクリアされると必ず、INTピンはエッジを出力します。エッジモードで、ある割込みが保留中の場合に他の割込みがクリアされると、INTピンは瞬間的に非アクティブになり、次に再びアクティブになってエッジを出力します。**図 8**の(2)に示すように、非アクティブな時間は、以下の 4 つの値にプログラム可能です。

PULSEWID1	PULSEWID0	INT Pulse Width uS
0	0	10.6
0	1	5.3
1	0	2.6
1	1	1.3

注: MAX3420E には、PULSEWID[1:0]ビットがありません。MAX3420E の時間は 10.6µs に固定されています。

SEPIRQ = 1 のとき、MAX3421Eは、8 つのGPINピンに対応する 8 つの割込みをINTピンから移動します。MAX3421Eは、GPXピンに割込みを転送します(GPXピンは、GPX[B:A] = 10 のとき 2 つ目の割込み出力ピンの役割を果たします)。GPXピンがこのように動作するとき、INT出力ピンとしてのGPXピンの特性はINTLEVELビット、POSINTビット、およびPULSEWID[1:0]ビットによって設定されます。

LOWSPEED

意味: ロースピード動作用にホストを設定する

モード: **ホストのみ**

CPU がこのビットをセットおよびクリアします。

プログラミング上の注意:
CPU は、このビットをセットして、ロースピードの USB ホストとしての動作を可能にします。CPU は通常、ペリフェラルが接続され(CONDETIRQ がアクティブ)、休止バスの状態が D+ = 0、D- = 1 であるときにこのビットをセットします。

この状態は、以下によって示されます。

- LOWSPEED = 1 かつ J 状態が検出される。
- LOWSPEED = 0 かつ K 状態が検出される。

バスの状態の検出の詳細については、50 ページを参照してください。

OSCOKIRQ、OSCOKIE

意味: **OSCOKIRQ**: 発振器の OK 割込み要求
OSCOKIE: 発振器の OK 割込みイネーブル

モード: **ペリフェラルおよびホスト**

内部 OSCOK 信号は、12MHz の内部発振器および 48MHz の PLL が安定し、チップの動作準備が完了したことを示します。OSCOK 信号が 0 から 1 に遷移すると、SIE は、OSCOKIRQ ビットをセットし、チップの動作準備が完了したことを示します。CPU は、「1」を書き込むことによって OSCOKIRQ ビットをクリアします。

CPU が OSCOKIE ビットをセットおよびクリアします。OSCOKIE = 1 のとき、OSCOKIRQ は INT ピンをアクティブにするソースとしてイネーブルになります。

プログラミング上の注意:
CPU は、チップをリセットすることによって(CHIPRES = 1 の後に CHIPRES = 0 を設定)、MAX3421E の発振器を停止した場合には必ず、動作を継続する前に OSCOKIRQ がアサートされるのを待つ必要があります。

PERADDR レジスタ

意味: パケット送信先のペリフェラルのアドレス

モード: **ホストのみ**

CPU がこのレジスタを書き込み、SIE がこのレジスタを読み出します。

プログラミング上の注意:
CPU が HXFR レジスタをロードした後に、SIE がトークンパケットを送信すると、SIE はこのレジスタからペリフェラルのアドレスを取得します。CPU が 1 つのデバイスのアドレスのみと通信する場合、このレジスタを一度初期化することで、そのデバイスへのすべての転送を行うことができます。CPU は通常、エニュメレーション時に Set_Address 要求をデバイスに発行してから、要求したアドレスをこのレジスタにロードします。

PWRDOWN

意味: MAX3421E をパワーダウンする

モード: **ペリフェラルおよびホスト**

CPU は、PWRDOWN ビットをセットしてチップを低電力状態にし、また PWRDOWN ビットをクリアして動作を再開します。

プログラミング上の注意:
このビットは、ペリフェラルモード専用として設計されていますが、ホストモードでもアクセス可能です。ホストとして動作しているときには、CPU が POWERDOWN = 1 を設定しないようにしてください。

RCVBC レジスタ

意味: 受信 FIFO バイトカウントレジスタ

モード: **ホストのみ**

プログラミング上の注意:
SIE は、バスから RCVFIFO にデータパケットをロードした後、受信したバイトカウントでこのレジスタを更新し、INDAVIRQ ビットをアサートします。

CPU は、RCVBC レジスタで示されるバイト数を読み出した後、1 を書き込むことによって RCVDAVIRQ ビットをクリアします。これによって、空のバッファを USB のアクセスに利用することができるようになります。

RCVDAVIRQ、RCVDAVIE

意味: **RCVDAVIRQ**: 受信 FIFO データ利用可能の割込み要求
RCVDAVIE: 受信 FIFO データ利用可能の割込みイネーブル

モード: **ホストのみ**

ホストからの IN 要求の結果として RCVFIFO にペリフェラルの新データが書き込まれると、SIE は RCVDAVIRQ ビットをセットします。この割込みを受けた後、CPU は RCVBC レジスタのバイト数を読み、次にデータを取り込むために RCVFIFO レジスタ(R1)から連続してデータを読み出します。最後に、CPU は RCVDAVIRQ ビットに 1 を書き込むことでこのビットをクリアします。SIE はすべてのリトライ(PID、CRC、データトグル、またはタイムアウトエラーによる)を行いますが、ペリフェラルに ACK ハンドシェイク生成するときのみ中断します。

CPU が RCVPAVIE をセットおよびクリアします。RCVDAVIE = 1 のとき、RCVDAVIRQ は INT ピンをアクティブにするソースとしてイネーブルになります。

プログラミング上の注意:
データ転送エラーが生じた場合、RCVDAVIRQ ではなく HXVRNIRQ がアサートされます。

RCVFIFO レジスタ

意味: 受信 FIFO

モード: **ホストのみ**

ペリフェラルはホストの IN 要求に応答してバスを介してデータを送信すると、SIE は内部 FIFO にデータを書き込みます。CPU は、RCVFIFO レジスタを繰り返し読み出すことによって、FIFO からバイトを読み出します。

受信したデータを破壊するおそれがあるため、CPU が RCVFIFO に書き込まないようにしてください。

プログラミング上の注意:
データパケットがエラーなしで到着後、SIE は RCVBC レジスタに受信バイト数をロードし、RCVDAVIRQ ビット(受信データ利用可能 IRQ)をアサートします。CPU は、まず RCVBC レジスタを読み出すことによって、RCVFIFO のバイト数を確認し、RCVDAVIRQ ビットをクリアしてから、RCVFIFO レジスタの読出しを繰り返してバイトを読み出します。

RCVFIFO レジスタは 2 つの 64 バイトの内部 FIFO に接続されます。2 つの FIFO を使用すると、CPU が一方の FIFO を空にしている間に、SIE はペリフェラルによって送信された IN データをもう一方の FIFO にロードすることが可能です。別のパケットがもう一方の FIFO で待機しているときに CPU が RCVDAVIRQ ビットをクリアした場合、SIE が RCVDAVIRQ ビットを直ちに再アサートします。

USB データが利用可能なとき、つまり RCVDAVIRQ が 1 のときだけ、CPU が RCVFIFO バイトを読み出すようにしてください。

REVISION レジスタ

意味: MAX3421E のレビジョン番号

モード: **ペリフェラルおよびホスト**

この読出し専用レジスタは、チップのレビジョンコードを示します。最新のレビジョン情報については、マキシムの web サイトを参照してください。このレジスタへの書込みは無効です。

RWUIRQ、RWUIE

意味: **RWUIRQ**: リモートウェイクアップの割込み要求
RWUIE: リモートウェイクアップの割込みイネーブル

モード: **ホストのみ**

SIE は、ペリフェラルからリモートウェイクアップ信号を受信すると、RWUIRQ ビットをセットします。CPU は「1」を書き込むことによって RWUIRQ ビットをクリアします。

CPU は RWUIE ビットをセットおよびクリアします。RWUIE = 1 のとき、RWUIRQ は INT ピンをアクティブにするソースとしてイネーブルになります。

プログラミング上の注意:
SOFKAEN = 0 に設定することによって CPU がバス信号送出をサスペンドした後、リモートウェイクアップ(RWU)が有効なペリフェラルが、RWU バスの状態をアサートして、ホストにバスのアクティビティを再開するよう要求することができます。SIE は、バスが K 状態の 10 ミリ秒の RWU 信号に続いて EOP を検出すると、RWUIRQ ビットをアサートします。

SAMPLEBUS
JSTATUS、KSTATUS

意味: USB バスの状態をサンプリングする

SAMPLEBUS: バスをサンプリングする
STATUS、KSTATUS: 状態を示す

モード: **ホストのみ**

CPU は、SAMPLEBUS ビットをセットして、D+ラインと D-ラインの状態をサンプリングするよう SIE に指示します。SIE は、動作が完了すると、SAMPLEBUS ビットをクリアします。

JSTATUS ビットは CPU からは読出し専用で、SIE によってセットおよびクリアされます

プログラミング上の注意:
JSTATUS ビットと KSTATUS ビットは、以下の 2 つの状況の下で更新されます。

1. CPU が SAMPLEBUS = 1 を設定する
2. CONDETIRQ がアサートされる

2 番目のケースは、あるデバイスの接続または切断のいずれかの状況を示します。CPU は CONDETIRQ に応答し、JSTATUS ビットと KSTATUS ビットを読み出して、どちらのイベントが発生したかを確認する必要があります。

JSTATUSビットとKSTATUSビットは、表 8に示すように符号化されています。

表 8. JSTATUS ビットと KSTATUS ビットの符号化

JSTATUS	KSTATUS	意味
0	0	SE0
0	1	K
1	0	J
1	1	N/A

表 8 の 4 行目は USB のバス状態では未定義です。

注: J 状態および K 状態の意味は、LOWSPEED ビットの設定によって異なります。LOWSPEED = 0 のとき、「J」は「D+がハイで D-がロー」であることを意味し、LOWSPEED = 1 のとき、「J」は「D+がローで D-がハイ」であることを意味します。

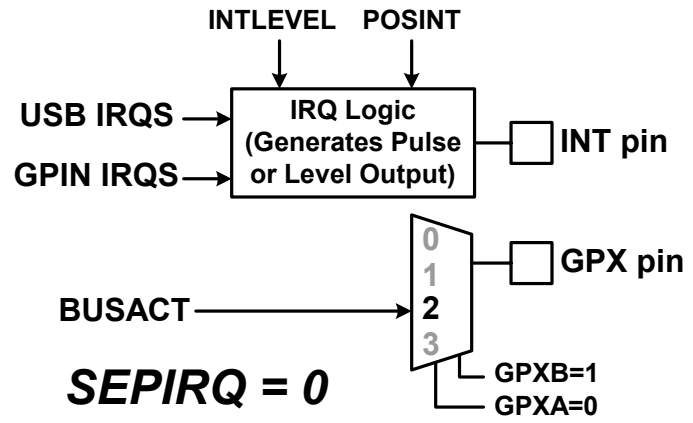
SEPIRQ

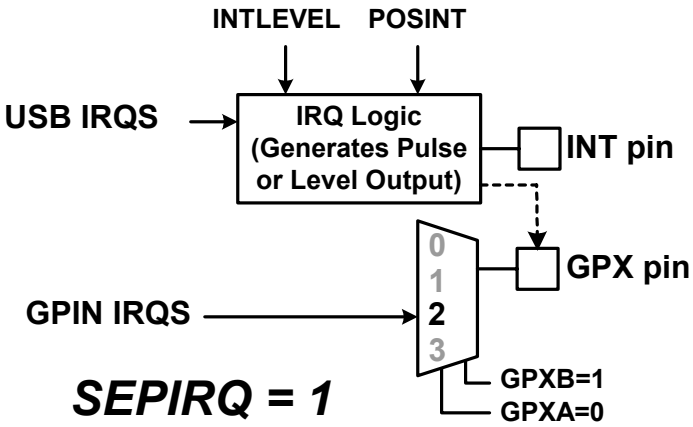
意味: 個別のピン(GPX)に GPIN の IRQ を提供する

モード: **ペリフェラルおよびホスト**

CPU がこれらのビットをセットおよびクリアします。

プログラミング上の注意:
MAX3421E の INT ピンは、内部割込み、USB、または GPIN のいずれかがアサートされると必ずアクティブになります(IRQ がイネーブルで IE = 1 であると仮定)。このデフォルトのケースは、上側の図(SEPIRQ = 0)に示されています。





下側の図に示すように、SPI マスタが SEPIRQ = 1 を設定すると、INT ピンへの割込みソースである 8 つの GPIN の IRQ 信号は移動され、MAX3421E の 2 つ目の割込みピンの役割を果たす GPX ピンに転送されます。GPX[B:A]を 2 に設定すると、通常、BUSACT 信号は GPX ピンに接続されます。SEPIRQ = 1 に

設定すると、この信号は、8 つの GPIN 割込みのいずれかを示す信号に置き換えられます。下側の図の点線は、GPX ピンがこの 2 つ目の割込みピンとして機能するとき、その特性(INTLEVEL と POSINT によって決まる)は INT ピンの場合と同じであることを表します。

SIGRSM

意味: バスレジュームイベントの信号を送出する

モード: **ホストのみ**

CPU は、このビットをセットして、バスレジュームイベントを生成します。
SIE は、このビットをクリアして、レジューム信号の送出を完了したことを示します。

プログラミング上の注意:
USB ホストは、バスのアクティビティを停止することによって、ペリフェラルをサスペンドし、これを低電力状態にします。CPU は、SOFKAENAB = 0 に設定することによってペリフェラルを低電力状態にし、これによって MAX23421E がフルスピードの SOF パケットまたはロースピードのキープアライブパルスを 1 ミリ秒ごとに自動的に生成するのを停止します。

バスのアクティビティを再開するには、ホストは、20 ミリ秒の K 状態と、その後に続くロースピードの EOP から構成される「レジューム」信号を送信します。CPU は、SIGRSM = 1 に設定してから SIGRSM = 0 であるかどうかを確認することによって、「レジューム」を行います。次に、CPU は、SOFKAENAB = 1 を設定することによって、ミリ秒のフレームマーカを再開します。

CPUは、BUSEVENT割込み要求ビットであるBUSEVENTIRQ (19ページ)を使用することによっても、MAX3421Eのレジューム信号送出の終了を確認することができます。SIGRSMビットが 1 から 0 に遷移するとき、SIEはこのビットをセットします。

SNDBAVIRQ、SNDBAVIE

意味: 送信バッファ利用可能の割込み要求
送信バッファ利用可能の割込みイネーブル

モード: **ホストのみ**

SIE が、SNDFIFO のデータをペリフェラルに送信し、以下の 2 つの状況が得られた場合、SNDBAVIRQ ビットをセットします。

1. ペリフェラルが ACK ハンドシェイクで応答した
2. 低水準のエラー(有効な PID、CRC、EOP、スタッフ)が発生しなかった

CPU は、SNDBC レジスタに書き込むことによって SNDBAVIRQ ビットをクリアします。

CPU は SNDBAVIE ビットをセットおよびクリアします。SNDBAVIE = 1 のとき、SNDBAVIRQ は INT ピンをアクティブにするソースとしてイネーブルになります。

プログラミング上の注意:
SIE は、ホスト転送の終了時に、FIFO ポインタをクリアします。したがって、IN 転送の後にホストがゼロ以外の HRSLT を読み出した場合(たとえば、ペリフェラルが NAK ハンドシェイクを返すときの HRSLT[3:0]=0x04)、CPU は、HXFR レジスタに適切な値を再ロードするだけで IN 転送を再試行することができます。FIFO データは再書き込みされるまで保持されるため、また SIE の FIFO ポインタがリセットされるため、ホストは、毎回 FIFO データを再ロードしなくても、この方法で同じ SNDFIFO データを繰り返して送信することができます。

CPU は、SNDBC レジスタに書き込むことによって SNDBAVIRQ をクリアします。**CPU が SNDBAVIRQ ビットをじかにクリアしないようにしてください。**

SNDBC レジスタ

意味: 送信 FIFO バイトカウントレジスタ

モード: **ホストのみ**

CPU はこのレジスタをロードして、SNDFIFO にロードしたバイト数を示し、バスを介した OUT データの送信を FIFO に委ねます。

プログラミング上の注意:
CPU が SNDBC レジスタをロードすると、SIE は SNDBAVIRQ ビットをクリアします。2 つ目の FIFO が利用可能な場合、SIE は直ちに SNDBAVIRQ ビットを再アサートします。

FIFO のアンロードの終了時に CPU がじかにクリアする RCVDAVIRQ ビットとは異なり、CPU は、SNDBC レジスタに書き込むことによって SNDBAVIRQ をクリアします。**CPU が SNDBAVIRQ ビットをじかにクリアしないようにしてください。**

SNDFIFO レジスタ

意味: 送信 FIFO

モード: **ホストのみ**

CPU は繰り返して SNDFIFO レジスタを書き込むことによって、OUT 転送として伝送するデータを内部 FIFO に書き込みます。

プログラミング上の注意:
CPU が FIFO に 64 バイトを書き込んだ後に SNDFIFO レジスタを読み出した場合、バイトが読み出されます。

SNDFIFO をロードした後、CPU は SNDBC(送信バイトカウント)レジスタにロードしたバイト数を書き込みます。CPU がバイトカウントレジスタに書き込むと、SIE は SNDBAVIRQ (送信バッファ利用可能の IRQ)を非アクティブにし、FIFO に USB 転送を委ねます。

SNDFIFO レジスタは 2 つの 64 バイトの内部 FIFO に接続されます。2 つの FIFO を使用すると、SIE が USB を介して一方の FIFO からデータを送信している間に、CPU がもう一方の FIFO に OUT データをロードすることが可能です。CPU が SNDBC レジスタに書き込み、もう一方のバッファが利用可能な場合、SIE は、SNDBAVIRQ ビットを非アクティブにした後、直ちに再アサートし、2 つ目のバッファが利用可能であることを示します。

SEND バッファが利用可能なとき、(つまり SNDBAVIRQ が 1 のとき)以外は、CPU が SNDFIFO をロードしないようにしてください。

SNDTOG1、SNDTOG0
RCVTOG1、RCVTOG0

意味: データ転送のデータトグル値をセットまたはクリアする

モード: **ホストのみ**

CPU は、これらのビットペアの 1 つに 1 を書き込んで、データ転送のデータトグル値を初期化します。これらのビットに 0 を書き込んでも無効です。一方のビットペアの両方のビットに 1 を書き込んでも無効です。

プログラミング上の注意:
MAX3421Eは 2 つのデータトグルフリップフロップを搭載し、これを使用して、SNDFIFOデータ転送時とRCVFIFOデータ転送時のUSB信号送出プロトコルを実現しています。エンドポイントにデータを転送する前に、CPUはこれらのビットを使用してエンドポイントのデータトグル値を初期化しました。エンドポイントにデータを転送した後、CPUはSNDTOGRDビットまたはRCVTOGRDビットのトグル値を読み出し(36ページ)、CPUのローカルストレージにこの値を格納します。複数のエンドポイントの場合、CPUはトグルビット値の配列(エンドポイントごとに 1 つ)を維持します。

CPU は、データをさらに転送するためにエンドポイントに戻ると、最初に、エンドポイント用として最後に保存された値にデータトグル値を初期化します。CPU は、SNDTOG1/0 ビットと RCVTOG1/0 ビットを使用して、エンドポイント用として最後に保存された値にデータトグルを設定します。

同じエンドポイントへの連続転送中、SIE はトグル値を維持します。CPU は、エンドポイントを切り替えたときにのみ、トグル値を保存して再初期化する必要があります。

データトグルの詳細については、「MAX3421E」(11ページ)を参照してください。

SOFKAENAB

意味: フルスピードの SOF パケットまたはロースピードのキープアライブパルスの自動生成をイネーブルにする

モード: **ホストのみ**

CPU がこのビットをセットおよびクリアします。

プログラミング上の注意:
CPU が SOFKAENAB = 1 を設定すると、SIE は 1 ミリ秒のフレームマーカを自動的に生成します。ビット LOWSPEED = 0 の場合、SIE は SOF パケットを生成し、LOWSPEED = 1 の場合、SIE はキープアライブパルスを生成します。

SOFKAENAB ビットが 1 ミリ秒間アサートされた後、SOF パルスまたは KA パルスが開始されます。SIE がフレームマーカを生成している間に、CPU が SOFKAENAB = 0 を設定した場合、SIE は信号送出を完了してからフレームマーカを閉じます。

SOF パケット

SIE は 1 ミリ秒ごとに SOF PID、11 ビットの内部フレームカウンタの内容、および CRC5 の値を送信します。パケットを送信した後、SIE はフレームカウンタをインクリメントし、FRAMEIRQ をアサートします。SIE は、HRSL レジスタ内のいずれの HRSLT ビットも更新しないため、CPU は前回の転送結果を常に読み出す必要があります。

フレームカウンタのパワーオン時のデフォルト値は 0 です。CPUは、ビットFRMRST = 1 (31ページ)を設定することによって、フレームカウンタをいつの時点でもリセットすることができます。

キープアライブパルス

SIE は、ロースピードのホストとして動作するとき、かつ SOFKAENAB = 1 のとき、キープアライブパルスを生成します。キープアライブパルスは、1 ミリ秒ごとに 1 つのロースピード EOP で構成されます。

SUDFIFO レジスタ

意味: セットアップデータ FIFO レジスタ

モード: **ホストのみ**

CPU は、SUDFIFO レジスタに 8 回書き込んで、8 バイトの内部 FIFO に、SETUP パケットに含めるべきデータをロードします。

プログラミング上の注意:
ペイロードは常に 8 バイトであるため、SUDFIFO にはバイトカウントレジスタは付随していません。

CPU が 8 バイトを書き込んだ後でこのレジスタを読み出した場合、FIFO のバイトが読み出されます。

SUSDNIRQ、SUSDNIE

意味: **SUSDNIRQ**: サスペンド動作処理 IRQ
SUSDNIE: サスペンド動作処理 IE

モード: **ホストのみ**

SIE は SUSDNIRQ ビットをアサートし、3 ミリ秒のバスアクティビティを示します。これは通常、SPU が SOFKAENAB = 0 を設定した後の 3 ミリ秒です。CPU は、1 を書き込むことによって SISDNIRQ ビットをクリアします。

CPU は SUSDNIE ビットをセットおよびクリアします。SUSDNIE = 1 のとき、SUSDNIRQ は INT ピンをアクティブにするソースとしてイネーブルになります。

VBUSIRQ、VBUSIE
NOVBUSIRQ、NOVBUSIE

意味:

VBUSIRQ:	V_{BUS}検出の割込み要求
VBUSIE:	V_{BUS}検出の割込みイネーブル
NOVBUSIRQ:	V_{BUS}欠如の割込み要求
NOVBUSIE:	V_{BUS}欠如の割込みイネーブル

モード: **ペリフェラルおよびホスト**

内部のV_{BUS}コンパレータの出力が 0 から 1 (VBUSIRQ)または 1 から 0 (NOVBUSIRQ)に遷移すると、SIEはVBUSIRQビットとNOVBUSIRQビットをセットします。このコンパレータは、VBCOMP (V_{BUS}コンパレータ)ピン上の電圧を示します。CPUは、1 を書き込むことによってVBUSIRQビットとNOVBUSIRQビットをクリアします。

CPU は VBUSIE ビットおよび NOVBUSIE ビットをセットおよびクリアします。これらのビットをセットすると、それぞれの IRQ ビットが INT ピンをアクティブにすることができます。

プログラミング上の注意:
これらのビットはペリフェラルモードとホストモードの両方でアクティブです。ペリフェラルモードでは、自己給電式の場合USBホストへの接続と切断を検出するために、これらのビットを使用することができます。ホストモードでは通常、V_{BUS}検出ビットとしてこれらのビットが要求されることはありません。ホストがV_{BUS}の電力をUSBコネクタに供給(および制御)するからです。この場合、VBCOMPピンを汎用入力として使用することが可能で、0 から 1 (VBUSIRQ)または 1 から 0 (NOVBUSIRQ)の、各エッジに個別の割込みを設けます。VBCOMピンはグランドへの弱い(約 100kΩ)プルダウン抵抗器を備えているため、VBCOMPピンを汎用入力ピンとして使用するときには終端抵抗器が不要です。

MAX3421Eのアプリケーションによっては、A (ホスト)およびB (ペリフェラル)のUSBコネクタの両方を実装し、どのコネクタにケーブルが接続されているかを反映するようMAX3421Eを自動的に構成する場合があります(USBペリフェラルへのAコネクタ、またはUSBホストへのBコネクタ)。この場合、AコネクタのV_{BUS}ピンはローカルの 5V電源によって給電することが可能で、またMAX3421EのD+プルダウン抵抗器とD-プルダウン抵抗器をオンにして、Aコネクタに接続されているペリフェラルを検出することができます。ホストの設計は、AコネクタのV_{BUS}ピン上でMAX4793 などの電流リミッタ/検出器/スイッチを使用する必要があります。BコネクタのVbusピンはMAX3421EのVBCOMPピンに接続することができるため、回路はV_{BUS}の有無を検出することによって、自分がUSBホストに接続されたときを検出することができます。この方法で、システムはペリフェラルまたはホストのいずれについても自己設定を行うことができます。

割込みイネーブルビットのほとんどは USB バスリセット時にクリアされるため、割込みイネーブルビットをオンにする初期化ルーチンを USB バスリセットのサービスの一部として呼び出す必要があります。