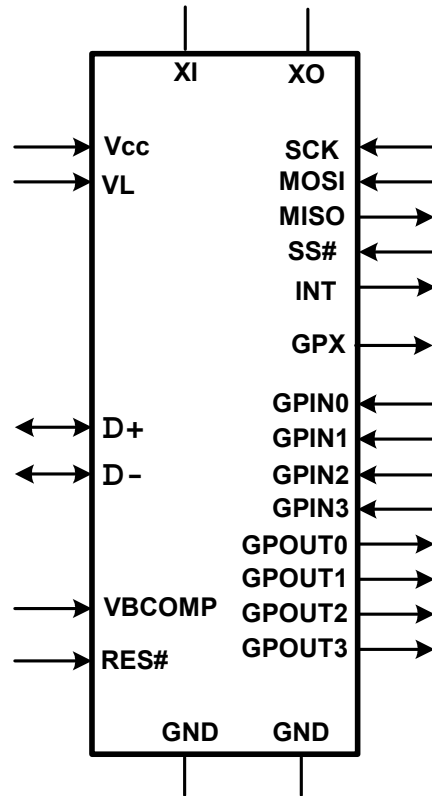


# MAX3420E

SPI インタフェース付き  
USB 周辺機器コントローラ

## プログラミングガイド



MAX3420E の詳細については、<http://japan.maxim-ic.com/max3420e> 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、<http://japan.maxim-ic.com/usb> を参照してください。  
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。

Copyright 2005 Maxim Integrated Products, Inc. All rights reserved.

2005 年 9 月 28 日改訂

## レジスタマップ

ビットは通常のフォントで表示し、レジスタは斜体で表示しています。  
 強調表示されている各セルは、そのレジスタまたはビットを説明しているページにリンクしています。ブラウザの左矢印を使用すると、このページに戻ります。  
 このアプリケーションノート巻末にある索引も、各項目の説明ページにリンクしています。

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

注: acc(アクセス)の欄は、CPUがレジスタにアクセス可能な方法を示しています。R = Read(読出し)、RC = Read(読出し)または Clear(クリア)、RSC = Read(読出し)、Set(セット)、または Clear(クリア)です。

## MAX3420E のレジスタにアクセスする方法

SPI™ マスタは、21 の内部レジスタ(R0~R20)の書込みと読出しを行うことによって MAX3420 を制御します。SPI マスタは、MAX3420E 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。

Reg4~Reg0 は、0~20 の有効値を持つレジスタアドレスをセットします。20 を超える値は MAX3420E によって無視されます。方向ビットは、データ転送の方向をセットします。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す。ACKSTAT は、よく使用されるこのレジスタビットを迅速にセットする方法として、制御バイトの中に用意されています。

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 SCLK 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。

コマンドバイトのみを送信した後、SPI サイクルを切り捨てることができます。この機能によって、SPI マスタは、SPI にフルアクセスすることなく ACKSTAT ビットをセットすることができます。

---

注:ACKSTAT ビットは、SPI マスタが USB の CONTROL 転送のサービスを終了したことを MAX3420E に通知します。これによって、MAX3420E は、次の CONTROL 転送の STATUS ステージに肯定応答します。

---

SPI コマンドバイトを使用して ACKSTAT ビットをセットするため、SPI マスタはレジスタフィールドにダミー値をセットし(このダミー値は使用されません)、読出し動作の DIR ビットをセットし(たとえば、Revision レジスタ R18 の読出し)、さらに ACKSTAT ビットをセットします。次に、SPI マスタは SS#をアサートし、8 つのコマンドビットを同期入力します。次に SPI マスタは、SS#信号をデアサートすることによって SPI サイクルを切り捨てます。これが、ACKSTAT ビットをセットするための最速の方法です。

MAX3420E には、2 つのレジスタのタイプがあります。すなわち FIFO と制御のレジスタです。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。

SPI は、Motorola, Inc.の登録商標です。

レジスタ R0~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 R0~R4 を選択した後、SPI マスタは、SPI 転送の間に読出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、SUDFIFO から 8 バイトを読み取るには、SPI マスタは、以下の手順を実施します：

1. SS# = 0 にセットします。
2. 00100000 を送信します。このコマンドバイトによって、読出し動作(DIR = 0)のための R4(SUDFIFO)が選択されます。
3. 8 つの SCLK パルスを発行し、1 データバイトを同期入力します(SCLK の立上りエッジごとに 1 ビット)。
4. 手順 3 をさらに 7 回繰り返し、合計 8 バイトを同期入力して格納します。
5. SS# = 1 に設定します。

レジスタ R5~R20 は制御レジスタです。SPI マスタが、同じ SPI 転送(SS#ロー)の間に R5~R20 の読出または書込みを繰り返した場合、各バイトの読出または書込みはすべて自動的に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、連続したグループのレジスタの読出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引き続きインクリメントされて、最後のレジスタ 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。

# ACKSTAT

---

**意味:** CONTROL 転送の STATUS ステージに肯定応答する

**位置:** EPSTALLS.6

**セット:** CPU は、CONTROL 転送要求のサービスを終了した後、このビットをセットします。このビットは、現在の CONTROL 転送のステータスステージに ACK ハンドシェイクを送信するように、SIE(シリアルインタフェースエンジン)に指示します。CPU がこのビットをセットするまで、SIE は、CONTROL 転送のステータスステージに NAK ハンドシェイクで応答します。

**クリア:** SIE は、SETUP トークンが到着するたびに、このビットをクリアします。

**パワーオンリセット:** ACKSTAT = 0

**チップリセット:** ACKSTAT = 0

**バスリセット:** ACKSTAT = 0

**パワーダウン:** 読出しのみ

**参考:** ACKSTAT レジスタビットをセットする迅速な方法は、SPI コマンドバイトのビット 0 をセットすることです。マキシムのサンプルコードはすべてこの方法を使用しています。

## プログラミング上の注意:

CPU は、「セットアップデータの利用可能な割込み要求」(66 ページの SUDAVIRQ ビット)を受信すると、SUDAVIRQ ビットに 1 を書き込むことによってこのビットをクリアし、次に 8 つのデータバイトを SUDFIFO からメモリに読み出します。次に、CPU はその 8 つのバイトを調べて、USB 要求の内容を判断します。その要求が誤っている、または不明である場合、CPU は STLSTAT ビットをセットし、STALL ハンドシェイクでステータスステージに応答します(64 ページ)。

CPU は、要求を認識すると、その要求を処理します。処理が終了すると、ACKSTAT = 1 をセットすることによって、ACK ハンドシェイクをステータスステージに送信して CONTROL 転送を終了するように SIE に指示します。CPU が肯定応答するまで、あるいは転送の機能を停止するまで、SIE は CONTROL 転送のステータスステージに NAK ハンドシェイクを自動的に返します。

8 バイトのセットアップデータを解釈する C プログラムは通常、セットアップパケット内のバイトがすべて正当な組み合わせかどうかを検査する 1 つ以上の **case** ステートメントから構成されています。STALL を処理する便利な方法は、CONTROL 転送の機能を停止するステートメントをデフォルトの case にすることです(64 ページの STLSTAT ビットを参照)。

# BUSACTIE

---

<b>意味:</b>	バスアクティブの割込みイネーブル
<b>位置:</b>	USBIEN.2
<b>セット:</b>	CPU は、このビットをセットして BUSACT IRQ(3 ページ)をイネーブルにします。
<b>クリア:</b>	CPU は、このビットをクリアして BUSACT IRQ をディセーブルにします。
<b>パワーオンリセット:</b>	BUSACTIE = 0
<b>チップリセット:</b>	BUSACTIE = 0
<b>バスリセット:</b>	BUSACTIE = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルビットを立てる初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# BUSACTIRQ

---

- 意味:** バスアクティブの割込み要求
- 位置:** USBIRQ.2
- セット:** SIE は、このビットをセットして、USB バスのアクティビティを示します。内部 BUSACT 信号は、SIE が SYNC フィールドを受信したときにセットされます。また、32 ビット時間の J 状態の後、あるいは USB バスリセットの間にリセットされます。BUSACTIRQ ビットは、内部 BUSACT 信号が 0 から 1 に遷移するときにセットされます。
- クリア:** CPU は、「1」を書き込むことによってこのビットをクリアします。
- パワーオンリセット:** BUSACTIRQ = 0
- チップリセット:** BUSACTIRQ = 0
- バスリセット:** BUSACTIRQ = 0
- パワーダウン:** 読出しのみ

# CHIPRES

---

意味:	チップリセット
位置:	USBCTL.5
セット:	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。
クリア:	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。
パワーオンリセット:	CHIPRES = 0
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:

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。

# CONNECT

---

意味:	USB に接続する
位置:	USBCTL.3
セット:	CPU は、このビットをセットして、DPLUS ラインと Vcc の間に 1500Ω の内部抵抗器を接続します。
クリア:	CPU は、このビットをクリアして、DPLUS ラインと Vcc との間の 1500Ω の内部抵抗器を切り離します。
パワーオンリセット:	CONNECT = 0
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:

CONNECT ビットの動作は、VBGATE ビット(76 ページ)の設定値によって決まります。CONNECT = 1 で VBGATE = 1 の場合、Vbus が Vbus ピン上で有効であることが検出されない限り、内部ロジックはプルアップ抵抗器を接続しません。VBGATE = 0 の場合、DPLUS プルアップ抵抗器は CONNECT = 1 のときに無条件で接続されます。

パワーオンリセットのみが CONNECT ビットをクリアします。動作中、INT ピンを介して外部リセットが MAX3420E に加えられると、CONNECT ビットはその状態を保持します。これは、RES#をアサートしたときにデバイスが USB に接続されている場合(CONNECT = 1)、デバイスは、リセット中およびリセットの解除後(RES# = 1)、接続が維持されたままであるということです。

# CTGEP1OUT

---

意味:	エンドポイント 1 OUT 用のデータグルをクリアする
位置:	CLRTOGS.2
セット:	CPU は、このビットをセットして、EP1-OUT 用のデータグルをクリアして DATA0 状態にします。
クリア:	SIE は、このビットを自動的にクリアします。
パワーオンリセット:	CTGEP1OUT = 0
チップリセット:	CTGEP1OUT = 0
バスリセット:	CTGEP1OUT = 0
パワーダウン:	読出しのみ

## プログラミング上の注意:

SIE は、チップリセットまたは USB バスリセットの間に、すべてのデータグルを自動的にクリアします。CPU は通常、以下の 2 つの条件下で、個々のエンドポイントのデータグルをクリアする必要があります:

- ホストが Set\_Configuration 要求を発行する
- ホストが Clear\_Feature(エンドポイントの機能停止)要求を発行する

<b>意味:</b>	エンドポイント 2 IN 用のデータトグルをクリアする
<b>位置:</b>	CLRTOGS.3
<b>セット:</b>	CPU は、このビットをセットして、EP2-IN 用のデータトグルをクリアして DATA0 状態にします。
<b>クリア:</b>	SIE は、このビットを自動的にクリアします。
<b>パワーオンリセット:</b>	CTGEP2IN = 0
<b>チップリセット:</b>	CTGEP2IN = 0
<b>バスリセット:</b>	CTGEP2IN = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

SIE は、チップリセットまたは USB バスリセットの間に、すべてのデータトグルを自動的にクリアします。CPU は通常、以下の 2 つの条件下で、個々のエンドポイントのデータトグルをクリアする必要があります:

- ホストが Set\_Configuration 要求を発行する
- ホストが Clear\_Feature(エンドポイントの機能停止)要求を発行する

# CTGEP3IN

---

意味:	エンドポイント 3 IN 用のデータゲルをクリアする
位置:	CLRTOGS.4
セット:	CPU は、このビットをセットして、EP3-IN 用のデータゲルをクリアして DATA0 状態にします。
クリア:	SIE は、このビットを自動的にクリアします。
パワーオンリセット:	CTGEP3IN = 0
チップリセット:	CTGEP3IN = 0
バスリセット:	CTGEP3IN = 0
パワーダウン:	読出しのみ

## プログラミング上の注意:

SIE は、チップリセットまたは USB バスリセットの間に、すべてのデータゲルを自動的にクリアします。CPU は通常、以下の 2 つの条件下で、個々のエンドポイントのデータゲルをクリアする必要があります:

- ホストが Set\_Configuration 要求を発行する
- ホストが Clear\_Feature(エンドポイントの機能停止)要求を発行する

- 意味:** エンドポイント 0 のバイトカウントレジスタ。EP0 は双方向のエンドポイントであるため、IN 転送と OUT 転送の両方が同じ FIFO(10 ページの EP0FIFO)を共有します。このレジスタの動作は、転送方向によって決まります。
- 位置:** EP0BC[6:0]
- 書込み(IN):** IN 転送の場合、CPU は、EP0FIFO にデータをロードした後、この EP0BC レジスタにバイトカウントを書き込みます。有効値は 0~64 です。CPU がこのレジスタに書き込むと、SIE は、エンドポイントへの次の IN 要求に対して、NAK ではなくデータパケットを返せるようにエンドポイントを準備します。
- 読出し(OUT):** OUT 転送の場合、SIE はバイトカウントをロードして、OUT データ転送で受信したバイトの数を示します。OUT 転送が成功すると、SIE はその転送に肯定応答してバイトカウントレジスタを更新し、OUT0DAV 割込み要求ビット(51 ページ)をアサートします。
- パワーオンリセット:** EP0BC = 0  
**チップリセット:** EP0BC = 0  
**バスリセット:** EP0BC = 0  
**パワーダウン:** 読出しも書込みもなし

## プログラミング上の注意:

ビット 7 は何の効力もなく、0 として読み出されます。

CPU は、CONTROL 転送のデータステージへの応答として EP0FIFO を書き込みます。

CPU は、EP0FIFO を読み出して、CONTROL 転送のデータステージを取り出します。

# EP0FIFO

---

- 意味:** エンドポイント 0 の FIFO。この 64 バイトの FIFO は、双方向のエンドポイント 0 に対する OUT 転送と IN 転送に使用されます。
- 位置:** EP0FIFO[7:0]
- 書込み(IN):** IN 転送の場合、CPU は一連のバイトをこの FIFO に書き込んで、FIFO を IN データで埋めます。パケット(0~64 バイト)で FIFO を埋めた後、CPU はバイトカウントレジスタ(9 ページ)を書き込んで IN 転送を準備し、SIE が IN パケットを受信したときに何バイトをエンドポイント 0 に転送するかを指示します。
- 読出し(OUT):** OUT 転送の場合、SIE はホストから受信した USB データで FIFO を埋めます。OUT 転送がエラーフリーであることが確認されると、SIE はバイトカウントレジスタ(9 ページ)をロードして、OUT データ転送で受信したバイトの数を示します。転送が成功した場合、やはり SIE は OUT 転送に肯定応答し、OUT0DAV 割込み要求ビット(ページ 51)をアサートします。
- パワーオンリセット:** EP0FIFO[7:0] = 0  
**チップリセット:** EP0FIFO[7:0] = 0  
**バスリセット:** 変化なし  
**パワーダウン:** 読出しも書込みもなし

## プログラミング上の注意:

SIE は、エラー(CRC やビットスタッフなど)が含まれたパケットを検出すると自動的にパケットを再試行します。これは、CPU には関与しません(転送が完全ではないために、割込みフラグや割込みレジスタは更新されません)。

<b>意味:</b>	EP0-IN の NAK
<b>位置:</b>	PINCTL.5
<b>セット:</b>	SIE は、EP0-IN エンドポイントが IN 要求を受信して NAK ハンドシェイクを返すときにこのビットをセットします。
<b>クリア:</b>	CPU は、1 を書き込むことによってこのビットをクリアします。
<b>パワーオンリセット:</b>	EP0IBN = 0
<b>チップリセット:</b>	EP0IBN = 0
<b>バスリセット:</b>	EP0IBN = 0
<b>パワーダウン:</b>	読出し-書込み

## プログラミング上の注意:

このビットをポーリングすることで、IN エンドポイントからまだ利用することができない IN データ(CPU がエンドポイントのロードと準備を完了していないため)をホストが要求しているかどうかを知ることができます。このビットは、割込みシステムには含まれていません。

注:EP0INAK ビットは単なる参考用です。通常、USB デバイスのファームウェアがこれを使用することはありません。

# EP1DISAB

---

<b>意味:</b>	エンドポイント 1-OUT をディセーブルにする
<b>位置:</b>	CLRTOGS.5
<b>セット:</b>	CPU は、このビットをセットして、エンドポイント 1-OUT へのトラフィックをディセーブルにします。
<b>クリア:</b>	CPU は、このビットをクリアして、エンドポイント 1-OUT へのトラフィックをイネーブルにします。
<b>パワーオンリセット:</b>	EP1DISAB = 0
<b>チップリセット:</b>	EP1DISAB = 0
<b>バスリセット:</b>	EP1DISAB = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

ディセーブルされたエンドポイントは、いかなるトラフィックにも応答しません。ホストは通常、エニュメレーションの間に報告されないエンドポイントにトラフィックを送信することはありません。したがってこれは、逸脱したホストを防止するための「安全」ビットとなります。

エンドポイント 0 にはディセーブルビットはありません。エンドポイント 0 は、デフォルトの CONTROL エンドポイントとして常にアクティブでなければならないからです。

# EP1OUTBC

<b>意味:</b>	エンドポイント 1-OUT のバイトカウント
<b>位置:</b>	EP1OUTBC[6:0]
<b>書込み:</b>	エンドポイント 1 上で OUT 転送の受信に成功した後、SIE はこの転送に肯定応答し、受信したバイトカウントでこのレジスタを更新して、OUT1DAV 割込み要求(53 ページ)をアサートします。
<b>読出し:</b>	CPU は、OUT1DAV 割込み要求を受信した後にこのレジスタを読み出して、EP1OUTFIFO(14 ページ)から読み出すバイト数を決定します。
<b>パワーオンリセット:</b>	EP1OUTBC = 0
<b>チップリセット:</b>	EP1OUTBC = 0
<b>バスリセット:</b>	EP1OUTBC = 0
<b>パワーダウン:</b>	読出しも書込みもなし

## プログラミング上の注意:

EP1OUT はダブルバッファ構造のエンドポイントです。つまり 2 つの FIFO と 2 つのバイトカウントレジスタがあるということです。ダブルバッファリングによって、1 つの FIFO に USB データが同時に入力される一方で、もう 1 つの FIFO から CPU がデータを読み出すことが可能になります。これによって、多くのシステムで帯域幅の性能が向上します。OUT エンドポイントに対するダブルバッファリングの動作の説明については、OUT1DAVIRQ ビットの説明(51 ページ)を参照してください。

OUT1DAVIRQ のフラグのロジックはダブルバッファリングに対応しているため、プログラマがダブルバッファリングを意識する必要はありません。たとえば、両バッファが利用可能で、したがって OUT1DAVIRQ = 0 であると仮定します。OUT パケットが到着すると、OUT1DAVIRQ は 0 から 1 に遷移して、最初のパケットが利用可能であることを示します。シングルバッファ構造のエンドポイントでは、CPU が FIFO を空にする時間がないうちに別の OUT パケットが EP1-OUT 上に到着した場合、SIE は NAK ハンドシェイクで応答し、データを受取りのためにエンドポイントを利用することができないことを示します。

しかし、ダブルバッファ構造のエンドポイントであれば、2 つ目のバッファをデータ用に利用することができるため、2 つ目の OUT パケットを受け取って肯定応答します。どちらかの FIFO が空になる前に 3 つ目の OUT パケットが到着した場合、SIE はその転送に否定応答し、FIFO が 2 つともいっぱいであることを示します。

# EP1OUTFIFO

---

<b>意味:</b>	エンドポイント 1-OUT 用のダブルバッファ構造の 64 バイト FIFO
<b>位置:</b>	EP1OUTFIFO[7:0]
<b>書込み:</b>	SIE は、ホストからエンドポイント 1-OUT に転送されるバイトで OUT FIFO を埋めます。OUT 転送の受信に成功した後、SIE はこの転送に肯定応答し、バイトカウントレジスタ(13 ページ)を更新して、OUT1DAV 割込み要求(53 ページ)をアサートします。
<b>読出し:</b>	CPU は、OUT1DAV 割込み要求を受信すると、バイトカウントレジスタを読み出して FIFO 内のバイト数を決定し、次にバイトカウントレジスタからそのバイト数を読み出します。
<b>パワーオンリセット:</b>	EP1OUTFIFO = 0
<b>チップリセット:</b>	EP1OUTFIFO = 0
<b>バスリセット:</b>	変化なし
<b>パワーダウン:</b>	読出しも書込みもなし

## プログラミング上の注意:

EP1OUT はダブルバッファ構造のエンドポイントです。つまり 2 つの FIFO と 2 つのバイトカウントレジスタがあるということです。ダブルバッファリングによって、1 つの FIFO に USB データが同時に入力される一方で、もう 1 つの FIFO から CPU がデータを読み出すことが可能になります。これによって、多くのシステムで帯域幅の性能が向上します。OUT エンドポイントに対するダブルバッファリングの動作の説明については、OUT1DAVIRQ ビットの説明(53 ページ)を参照してください。

OUT1DAVIRQ フラグのロジックはダブルバッファリングに対応しているため、プログラマがダブルバッファリングを意識する必要はありません。たとえば、両バッファが利用可能で、したがって OUT1DAVIRQ = 0 であると仮定します。OUT パケットが到着すると、OUT1DAVIRQ は 0 から 1 に遷移して、最初のパケットが利用可能であることを示します。シングルバッファ構造のエンドポイントでは、CPU が FIFO を空にする時間がないうちに別の OUT パケットが EP1-OUT 上に到着した場合、SIE は NAK ハンドシェイクで応答し、データの受け取りのためにエンドポイントを利用することができないことを示します。

しかし、ダブルバッファ構造のエンドポイントであれば、2 つ目のバッファをデータ用に利用することができるため、2 つ目の OUT パケットを受け取って肯定応答します。どちらかの FIFO が空になる前に 3 つ目の OUT パケットが到着した場合、SIE はその転送に否定応答し、FIFO が 2 つともいっぱいであることを示します。

# EP2DISAB

---

<b>意味:</b>	エンドポイント 2 をディセーブルにする
<b>位置:</b>	CLRTOGS.6
<b>セット:</b>	CPU は、このビットをセットして、エンドポイント 2-IN へのトラフィックをディセーブルにします。
<b>クリア:</b>	CPU は、このビットをクリアして、エンドポイント 2-IN へのトラフィックをイネーブルにします。
<b>パワーオンリセット:</b>	EP2DISAB = 0
<b>チップリセット:</b>	EP2DISAB = 0
<b>バスリセット:</b>	EP2DISAB = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

ディセーブルされたエンドポイントは、いかなるトラフィックにも応答しません。ホストは通常、エニユメレーションの間に報告されないエンドポイントにトラフィックを送信することはありません。したがってこれは、逸脱したホストを防止するための「安全」ビットとなります。

エンドポイント 0 にはディセーブルビットはありません。エンドポイント 0 は、デフォルトの CONTROL エンドポイントとして常にアクティブでなければならないからです。

# EP2INAK

---

<b>意味:</b>	エンドポイント 2-IN の NAK
<b>位置:</b>	PINCTL.6
<b>セット:</b>	SIE は、EP2-IN エンドポイントが IN 要求を受信して NAK ハンドシェイクを返すときにこのビットをセットします。
<b>クリア:</b>	CPU は、1 を書き込むことによってこのビットをクリアします。
<b>パワーオンリセット:</b>	EP2INAK = 0
<b>チップリセット:</b>	EP2INAK = 0
<b>バスリセット:</b>	EP2INAK = 0
<b>パワーダウン:</b>	読出し-書込み

## プログラミング上の注意:

このビットをポーリングすることで、IN エンドポイントからまだ利用することができない IN データ(CPU がエンドポイントのロードと準備を完了していないため)をホストが要求しているかどうかを知ることができます。このビットは、割込みシステムには含まれていません。

注: EP2INAK ビットは単なる参考用です。通常、USB デバイスのファームウェアがこれを使用することはありません。

- 意味:** エンドポイント 2-IN のバイトカウントレジスタ
- 位置:** EP2INBC[6:0]
- 書込み:** CPU は、EP2INFIFO(18 ページ)にロードされているバイト数をこのレジスタにロードします。これによって、次の IN 転送のためのエンドポイントが準備されます。
- 読出し:** SIE は、EP2-IN ホスト要求に対する応答として、この FIFO のデータを送信します。
- パワーオンリセット:** EP2INBC = 0
- チップリセット:** EP2INBC = 0
- バスリセット:** EP2INBC = 0
- パワーダウン:** 読出しのみ

# EP2INFIFO

---

- 意味:** エンドポイント 2-IN の FIFO(ダブルバッファ構造の 64 バイト FIFO)
- 位置:** EP2INFIFO[7:0]
- 書込み:** CPU は、ホストへの送信に備えてこの FIFO にバイトをロードします。
- 読出し:** SIE は、EP2-IN への IN 要求に応じて USB 上でこれらのバイトを送信します。
- パワーオンリセット:** EP2INFIFO = 0
- チップリセット:** EP2INFIFO = 0
- バスリセット:** 変化なし
- パワーダウン:** 読出しも書込みもなし

<b>意味:</b>	エンドポイント 3 をディセーブルにする
<b>位置:</b>	CLRTOGS.7
<b>セット:</b>	CPU は、このビットをセットして、エンドポイント 3-IN へのトラフィックをディセーブルにします。
<b>クリア:</b>	CPU は、このビットをクリアして、エンドポイント 3-IN へのトラフィックをイネーブルにします。
<b>パワーオンリセット:</b>	EP3DISAB = 0
<b>チップリセット:</b>	EP3DISAB = 0
<b>バスリセット:</b>	EP3DISAB = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

ディセーブルされたエンドポイントは、いかなるトラフィックにも応答しません。ホストは通常、エnumレーションの間に報告されないエンドポイントにトラフィックを送信することはありません。したがってこれは、逸脱したホストを防止するための「安全」ビットとなります。

エンドポイント 0 にはディセーブルビットはありません。エンドポイント 0 は、デフォルトの CONTROL エンドポイントとして常にアクティブでなければならないからです。

# EP3INAK

---

<b>意味:</b>	エンドポイント 3-IN の NAK
<b>位置:</b>	PINCTL.7
<b>セット:</b>	SIE は、EP3-IN エンドポイントが IN 要求を受信して NAK ハンドシェイクを返すときにこのビットをセットします。
<b>クリア:</b>	CPU は、1 を書き込むことによってこのビットをクリアします。
<b>パワーオンリセット:</b>	EP3INAK = 0
<b>チップリセット:</b>	EP3INAK = 0
<b>バスリセット:</b>	EP3INAK = 0
<b>パワーダウン:</b>	読出し-書込み

## プログラミング上の注意:

このビットをポーリングすることで、IN エンドポイントからまだ利用することができない IN データ(CPU がエンドポイントのロードと準備を完了していないため)をホストが要求しているかどうかを知ることができます。このビットは、割込みシステムには含まれていません。

注: EP3INAK ビットは単なる参考用です。通常、USB デバイスのファームウェアがこれを使用することはありません。

- 意味:** エンドポイント 3-IN のバイトカウントレジスタ
- 位置:** EP3INBC[6:0]
- 書込み:** CPU は、あらかじめ EP3INFIFO(22 ページ)にロードされているバイトの数をこのレジスタにロードします。これによって、次の IN 転送のためのエンドポイントが準備されます。
- 読出し:** SIE は、EP3-IN ホスト要求に対する応答として、この FIFO のデータを送信します。
- パワーオンリセット:** EP3INBC = 0
- チップリセット:** EP3INBC = 0
- バスリセット:** EP3INBC = 0
- パワーダウン:** 読出しのみ

# EP3INFIFO

---

**意味:** エンドポイント 3-IN の FIFO(64 バイト FIFO)

**位置:** EP3INFIFO[7:0]

**書込み:** CPU は、ホストへの送信に備えてこの FIFO にバイトをロードします。

**読出し:** SIE は、EP3-IN への IN 要求に応じて USB 上でこれらのバイトを送信します。

**パワーオンリセット:** EP3INFIFO = 0

**チップリセット:** EP3INFIFO = 0

**バスリセット:** 変化なし

**パワーダウン:** 読出しも書込みもなし

- 意味:** フルデュープレクス SPI ポートの動作
- 位置:** PINCTL.4
- セット:** 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。
- クリア:** SIE 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)ピンはトリステートになります。

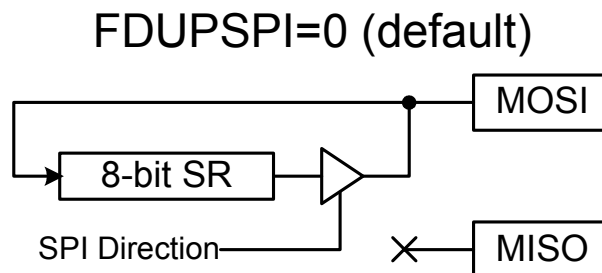


図 2. ハーフデュープレクスの SPI インタフェース

### 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、図 3 に示すように、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、ステータスビットの 8 つのビットが同期出力されます。

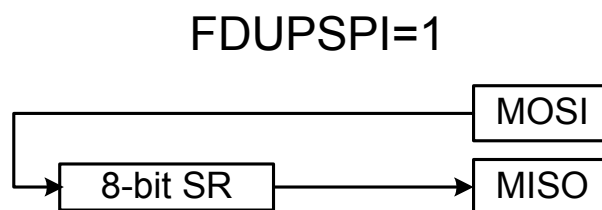


図 3. フルデュープレクス SPI インタフェース

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

図 4. フルデュプレクスモード:これらの USB ステータスビットは、コマンドバイトが同期入力される間に同期出力されます。

### SPI コマンドバイト

どちらの SPI モードでも、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、方向、および ACKSTAT ビットをじかにセットするビットをセットします。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 bit	Signal
7	REG4
6	REG3
5	REG2
4	REG1
3	REG0
2	0
1	Direction (1=W <sub>r</sub> , 0=R <sub>d</sub> )
0	ACKSTAT (page 1)

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、図 5 のコマンドバイトで示すデータがストローブされます。REG[4:0]はレジスタアドレスをセットし、方向ビットは SPI サイクルの読出または書込みの方向をセットします。ACKSTAT は、EPSTALLS レジスタの対応するビットを書き込みます。FDUPSPI = 1 の場合、8 つの最初の SCLK クロックの間に、図 4 に示すデータが同時に MISO ピンに同期出力されます。

コマンドバイトに続いて、SPI マスタは、8 つの SCLK クロックからなる群を 1 つ以上送出して、MAX3420E 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。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、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 SCLK の立上りエッジでサンプリングされます。データは SCLK 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(クロックの極性)および CPHA(クロックの位相)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 SCLK を想定していること、また変更することなく最初の正のクロックエッジがモード(0,0)と(1,1)で動作可能になる前に MOSI データが利用可能になることも想定していることがわかります。この特性によって、MAX3420E 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 MAX3420E 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、SCLK 信号の非アクティブレベルであり、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 SCLK 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。

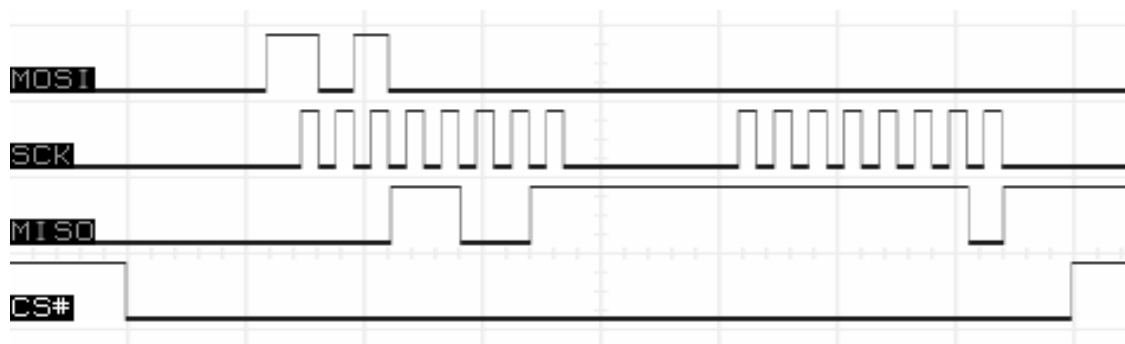


図 6. モード(0,0)での SPI インタフェースの動作

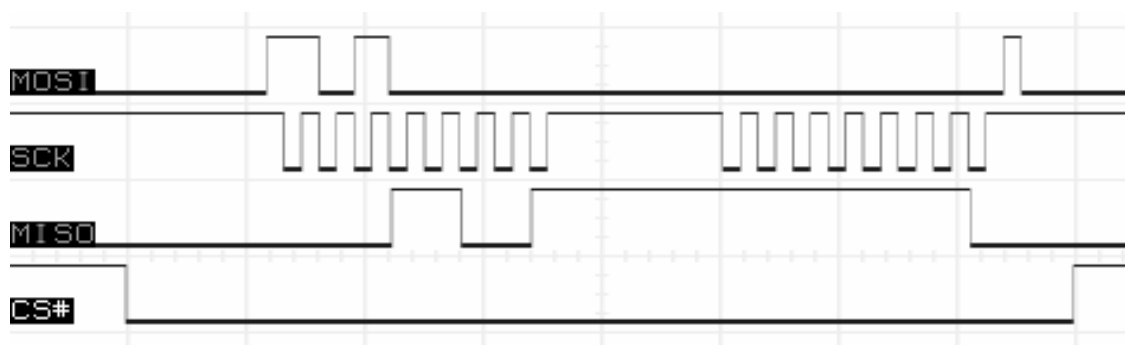


図 7. モード(1,1)での SPI インタフェースの動作

# FNADDR

---

- 意味:** ホストが USB 周辺機器に割り当てる関数アドレス
- 位置:** FNADDR[6:0]
- セット:** SIE は、ホストからの Set\_Address 要求の最後に ACK ハンドシェイクを受信した後、このレジスタを更新します。
- クリア:** SIE は、チップリセットまたは USB バスリセットの間にこのレジスタをクリアします。
- パワーオンリセット:** FNADDR = 0
- チップリセット:** FNADDR = 0
- バスリセット:** FNADDR = 0
- パワーダウン:** 読出しのみ

## プログラミング上の注意:

このレジスタへの CPU の書込みは無効です。

- 意味:** 汎用入力ピン 0
- 位置:** IOPINS.4
- セット:** このピンは GPIN0 ピン( $V_L$ を表す)の状態を示します。このピンは、小さなプルアップ抵抗器でハイにプルアップされます。このため、このピンに何も接続されていない場合、ピンはロジック 1 を示します。
- クリア:** このビットへの書込みは無効です。
- パワーオンリセット:** 非該当
- チップリセット:** 非該当
- バスリセット:** 非該当
- パワーダウン:** 読出しのみ

# GPIN1

---

意味:	汎用入力ピン 1
位置:	IOPINS.5
セット:	このピンは GPIN1 ピン( $V_L$ を表す)の状態を示します。このピンは、小さなプルアップ抵抗器でハイにプルアップされます。このため、このピンに何も接続されていない場合、ピンはロジック 1 を示します。
クリア:	このビットへの書込みは無効です。
パワーオンリセット:	非該当
チップリセット:	非該当
バスリセット:	非該当
パワーダウン:	読出しのみ

- 意味:** 汎用入力ピン 2
- 位置:** IOPINS.6
- セット:** このピンは GPIN2 ピン( $V_L$ を表す)の状態を示します。このピンは、小さなプルアップ抵抗器でハイにプルアップされます。このため、このピンに何も接続されていない場合、ピンはロジック 1 を示します。
- クリア:** このビットへの書込みは無効です。
- パワーオンリセット:** 非該当
- チップリセット:** 非該当
- バスリセット:** 非該当
- パワーダウン:** 読出しのみ

# GPIN3

---

- 意味:** 汎用入力ピン 3
- 位置:** IOPINS.7
- セット:** このピンは GPIN3 ピン( $V_L$ を表す)の状態を示します。このピンは、小さなプルアップ抵抗器でハイにプルアップされます。このため、このピンに何も接続されていない場合、ピンはロジック 1 を示します。
- クリア:** このビットへの書込みは無効です。
- パワーオンリセット:** 非該当
- チップリセット:** 非該当
- バスリセット:** 非該当
- パワーダウン:** 読出しのみ

- 意味:** 汎用出力ピン 0
- 位置:** IOPINS.0
- セット:** CPU は、このビットをセットして、GPOUT0 ピンをハイ( $V_L$ を表す)にセットします。CPU は、こ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、出力バッファの前の、出力フリップフロップの状態がわかります。したがって、ロジックレベルが不安定にな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でも、CPU は出力ピンの正しいロジック状態を読み出すことができます。
- クリア:** CPU は、このビットをクリアして、GPOUT0 ピンの状態を 0 にセットします。
- パワーオンリセット:** GPOUT0 = 0
- 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

# GPOUT1

---

**意味:** 汎用出力ピン 1

**位置:** IOPINS.1

**セット:** CPU は、このビットをセットして、GPOUT1 ピンをハイ( $V_L$ を表す)にセットします。CPU は、こ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、出力バッファの前の、出力フリップフロップの状態がわかります。したがって、ロジックレベルが不安定にな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でも、CPU は出力ピンの正しいロジック状態を読み出すことができます。

**クリア:** CPU は、このビットをクリアして、GPOUT1 ピンの状態を 0 にセットします。

**パワーオンリセット:** GPOUT1 = 0

**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

- 意味:** 汎用出力ピン 2
- 位置:** IOPINS.2
- セット:** CPU は、このビットをセットして、GPOUT2 ピンをハイ( $V_L$ を表す)にセットします。CPU は、こ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、出力バッファの前の、出力フリップフロップの状態がわかります。したがって、ロジックレベルが不安定にな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でも、CPU は出力ピンの正しいロジック状態を読み出すことができます。
- クリア:** CPU は、このビットをクリアして、GPOUT2 ピンの状態を 0 にセットします。
- パワーオンリセット:** GPOUT2 = 0
- 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

# GPOUT3

---

**意味:** 汎用出力ピン 3

**位置:** IOPINS.3

**セット:** CPU は、このビットをセットして、GPOUT3 ピンをハイ( $V_L$ を表す)にセットします。CPU は、こ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、出力バッファの前の、出力フリップフロップの状態がわかります。したがって、ロジックレベルが不安定にな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でも、CPU は出力ピンの正しいロジック状態を読み出すことができます。

**クリア:** CPU は、このビットをクリアして、GPOUT3 ピンの状態を 0 にセットします。

**パワーオンリセット:** GPOUT3 = 0

**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

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

**位置:** PINCTL.0

**セット:** CPU がこのビットをセットします。

**クリア:** CPU がこのビットをクリアします。

**パワーオンリセット:** GPXA = 0

**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:**

GPXB	GPXA	GPXピン
0	0	OPERATE(内部 POR の補完)
0	1	VBUS 検出
1	0	BUSACT
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)

# GPXB

---

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

**位置:** PINCTL.1

**セット:** CPU がこのビットをセットします。

**クリア:** CPU がこのビットをクリアします。

**パワーオンリセット:** GPXB = 0

**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:**

GPXB	GPXA	GPXピン
0	0	OPERATE(内部 POR の補完)
0	1	VBUS 検出
1	0	BUSACT
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)

- 意味:** ホストの発振器起動のイネーブル。これは、MAX3420E がパワーダウンモード (PWRDOWN = 1)にあるときに適用されます。
- 位置:** USBCTL.7
- セット:** USB の DPLUS 信号が 1 から 0 に遷移するとき(ホストのレジューム信号)、CPU は、このビットをセットしてオンチップ発振器を起動します。
- クリア:** USB ホストがバス信号の送出を再開すると、CPU は、このビットをクリアして、チップを低電力状態に維持します(発振器の起動を禁止)。

**パワーオンリセット:** HOSCSTEN = 0

**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:

CPU が MAX3420E をパワーダウン状態(PWRDOWN = 1, 55 ページ)にした後、MAX3420E の発振器を再起動するには、以下の 3 つの方法があります。

1. SPI マスタが PWRDOWN = 0 をセットする方法(これはチップリセットによっても達成されます)
2. SPI マスタが SIGRESUME = 1 をセットする方法(58 ページ)
3. USB ホストがバスのアクティビティを再開する方法。MAX3420E は、低電力モード(発振器がオフ)の間、DPLUS データライン上でローレベルのバスアクティビティを検出します。

HOSCSTEN ビットは、項目 3 の特殊なケースを取り扱います。このとき、MAX3420E は自己給電の周辺機器として設計されています。たとえば、ユーザが、パワーダウンモードにある自己給電の周辺機器を、電源が切れている PC に接続すると仮定します。DPLUS のロジックロー(ホスト PC のルートハブの DPLUS プルダウン抵抗器による)は、USB の RESUME 信号によく似ており、この信号によって通常、発振器を再起動してチップを起動します。ただし、今回の状況の場合、PC が Vbus をオンにして USB ポートをアクティブにするまで、チップはオンにならず、またチップが DPLUS 抵抗器に電力を供給することはありません。

したがって、自己給電のシステムでは、USB サスペンドの状態に移行するため、CPU は PWRDOWN = 1 および HOSCSTEN = 0 をセットします。これによってパワーダウン状態に移行しますが、チップを起動する方法として DPLUS ラインを 1 から 0 に遷移することは禁止されます。

ホスト PC に電力が供給されたことや、ホスト PC が VBUS をオンにしたことを自己給電の周辺機器が検出するための簡単な方法は、GPXB-A ピン(36 ページ)を 01 にセットすることです。これによって、Vbus コンパレータが GPX 出力ピンに接続されます。GPX ピンは、CPU のために Vbus を直接検出するピンとしての役割を果たすことができます。切断中、CPU は CONNECT = 0(5 ページ)をセットし、有効な USB 接続を検知したときにのみ CONNECT を 1 にセットする必要があります。

# IE

---

- 意味:** 割込みイネーブル
- 位置:** CPUCTL.0
- セット:** 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(45 ページ)と POSINT ビット(54 ページ)によってプログラムされます。
- クリア:** 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。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)。
- パワーオンリセット:** IE = 0
- チップリセット:** IE = 0
- バスリセット:** 変化なし
- パワーダウン:** 読出しのみ

- 意味:** エンドポイント 0 の IN バッファ利用可能の割込みイネーブル
- 位置:** EPIEN.0
- セット:** CPU は、このビットをセットして、IN0BAV 割込み要求(40 ページ)をイネーブルにします。
- クリア:** CPU は、このビットをクリアして、IN0BAV の割込み要求をディセーブルにします。

- パワーオンリセット:** IN0BAVIE = 0
- チップリセット:** IN0BAVIE = 0
- バスリセット:** IN0BAVIE = 0
- パワーダウン:** 読出しのみ

#### プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルをセットアップする初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# IN0BAVIRQ

---

- 意味:** エンドポイント 0 の IN バッファ利用可能の割込み要求
- 位置:** EPIRQ.0
- セット:** SIE は、エンドポイント 0 宛ての IN トークンを受信して、EP0FIFO(10 ページ)のデータを送信し、さらにホストから ACK ハンドシェイクを受信した後、このビットをセットします。これは、CPU が再び EP0FIFO をロードに利用することができることを示します。
- クリア:** CPU は、このビットに「1」を書き込むか、あるいはバイトカウントレジスタ EP0BC(9 ページ)に書き込むことによって、このビットをリセットします。
- パワーオンリセット:** IN0BAVIRQ = 1
- チップリセット:** IN0BAVIRQ = 1
- バスリセット:** IN0BAVIRQ = 1
- パワーダウン:** 読出しのみ

- 意味:** エンドポイント 2 の IN バッファ利用可能の割込みイネーブル
- 位置:** EPIEN.3
- セット:** CPU は、このビットをセットして、IN2BAV 割込み要求(42 ページ)をイネーブルにします。
- クリア:** CPU は、このビットをクリアして、IN2BAV 割込み要求をディセーブルにします。

- パワーオンリセット:** IN2BAVIE = 0
- チップリセット:** IN2BAVIE = 0
- バスリセット:** IN2BAVIE = 0
- パワーダウン:** 読出しのみ

#### プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルをセットアップする初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# IN2BAVIRQ

---

**意味:** エンドポイント 2 の IN バッファ利用可能の割込み要求

**位置:** EPIRQ.3

**セット:** SIE は、エンドポイント 2 宛ての IN トークンを受信して、EP2INFIFO(18 ページ)のデータを送信し、さらにホストから ACK ハンドシェイクを受信した後、このビットをセットします。これは、CPU が再び EP2INFIFO をロードに利用することができることを示します

**クリア:** CPU は、バイトカウントレジスタ EP2INBC(17 ページ)に書き込むことによって、このビットをリセットします。

**パワーオンリセット:** IN2BAVIRQ = 1

**チップリセット:** IN2BAVIRQ = 1

**バスリセット:** IN2BAVIRQ = 1

**パワーダウン:** 読出しのみ

## プログラミング上の注意:

EP2IN はダブルバッファ構造のエンドポイントです。つまり 2 つの FIFO と 2 つのバイトカウントレジスタを使用するという事です。ダブルバッファリングによって、1 つの IN FIFO から USB データが同時に出力される一方で、もう 1 つの FIFO に CPU がデータをロードすることが可能になります。これによって、多くのシステムで帯域幅の性能が向上します。

IN2BAVIRQ フラグのロジックによって、プログラマがダブルバッファリングを意識する必要はありません。たとえば、両バッファが利用可能で、したがって IN2BAVIRQ = 1 であると仮定します。CPU は N バイトを EP2INFIFO に書き込んでから、N を EP2INBC レジスタに書き込むことによって転送を準備します。シングルバッファ構造の IN エンドポイントでは、ホストが IN トークンをエンドポイントに送信し、IN データパケットを受け取るまで何も起こりません。しかし、2 つ目のバッファが CPU のロードに利用可能であるため、最初のパケットがホストによってすでに送受信されたかのように、IN2BAVIRQ ビットは無効になった後、直ちに再び有効になります。

また、ダブルバッファリングの動作によって、MAX3420E がリセットされた後も、EP2INBC レジスタが 2 度ロードされるまで IN2BAVIRQ ビットはセットされたままになります。

- 意味:** エンドポイント 3 の IN バッファ利用可能の割込みイネーブル
- 位置:** EPIEN.4
- セット:** CPU は、このビットをセットして、IN3BAV 割込み要求(44 ページ)をイネーブルにします。
- クリア:** CPU は、このビットをクリアして、IN3BAV 割込み要求をディセーブルにします。

- パワーオンリセット:** IN3BAVIE = 0
- チップリセット:** IN3BAVIE = 0
- バスリセット:** IN3BAVIE = 0
- パワーダウン:** 読出しのみ

#### プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルをセットアップする初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# IN3BAVIRQ

---

- 意味:** EP3-IN バッファ利用可能の割込み要求
- 位置:** EPIRQ.4
- セット:** SIE は、エンドポイント 3 宛ての IN トークンを受信して、EP3INFIFO(22 ページ)のデータを送信し、さらにホストから ACK ハンドシェイクを受信した後、このビットをセットします。これは、CPU が再び EP3INFIFO をロードに利用することができることを示します。
- クリア:** CPU は、バイトカウントレジスタ EP3INBC(21 ページ)を書き込むことによって、このビットをクリアします。

- パワーオンリセット:** IN3BAVIRQ = 1
- チップリセット:** IN3BAVIRQ = 1
- バスリセット:** IN3BAVIRQ = 1
- パワーダウン:** 読出しのみ

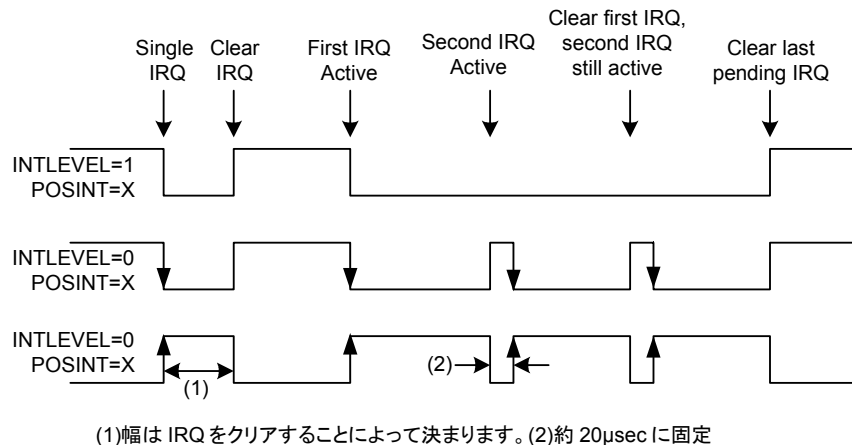
## プログラミング上の注意:

IN FIFO は、パワーオンまたはリセットにおいて、CPU データの受取りに利用することができるため 1 を示します。IN FIFO バッファの利用可能ビットは、デフォルトで 1 に設定される唯一のビットです。

# INTLEVEL

- 意味:** INT 出力ピンがレベルアクティブである
- 位置:** PINCTL.3
- セット:** CPU は、このビットをセットして、INT 出力ピンを「レベルセンシティブ」にします。INTLEVEL = 1 のとき、出力ピンはアクティブローのオープンドレインになります。INTLEVEL = 1 のとき、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。
- クリア:** CPU は、この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、エッジの極性は POSINT ビット(54 ページ)によってセットされます。
- パワーオンリセット:** INTLEVEL = 0
- 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:



(1)幅は IRQ をクリアすることによって決まります。(2)約 20μsec に固定

図 8. INTLEVEL ビットと POSINT ビットによって決まる INT ピンの動作

図 8 の波形は、INTLEVEL ビットと POSINT ビット(54 ページ)の異なる設定に対する INT ピンの動作を示しています。レベルモード(INTLEVEL = 1)では、保留中の割込みがあれば、INT ピンはローのままです。

# NOVBUSIE

---

- 意味:** Vbus 割込みイネーブルなし
- 位置:** USBIEN.5
- セット:** NOVBUS 割込み要求(47 ページ)をイネーブルにします。
- クリア:** NOVBUS 割込み要求をディセーブルにします。

- パワーオンリセット:** NOVBUSIE = 0
- チップリセット:** NOVBUSIE = 0
- バスリセット:** NOVBUSIE = 0
- パワーダウン:** 読出しのみ

## プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルビットを立てる初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# NOVBUSIRQ

---

- 意味:** Vbus 割込み要求なし
- 位置:** USBIRQ.5
- セット:** SIE は、Vbus ピンが Vbus 検出スレッシュホールドを下回ると、このビットをセットします。
- クリア:** CPU は、1 を書き込むことによってこのビットをクリアします。
- パワーオンリセット:** NOVBUSIRQ = 0
- チップリセット:** NOVBUSIRQ = 0
- バスリセット:** NOVBUSIRQ = 0
- パワーダウン:** 読出しのみ

## プログラミング上の注意:

この IRQ ビットを使用すると、MAX3420E によって組み込まれた USB 周辺機器が USB ホストから切り離されたことを、自己給電のデバイスが容易に確認できるようになります。

# OSCOKIE

---

- 意味:** 発振器の OK 割込みイネーブル
- 位置:** USBIEN.0
- セット:** CPU は、このビットをセットして、OSCOK 割込み要求(49 ページ)をイネーブルにします。
- クリア:** CPU は、このビットをクリアして、OSCOK 割込み要求をディセーブルにします。
- パワーオンリセット:** OSCOKIE = 0
- チップリセット:** OSCOKIE = 0
- バスリセット:** OSCOKIE = 0
- パワーダウン:** 読出しのみ

## プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルビットを立てる初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# OSCOKIRQ

---

- 意味:** 発振器の OK 割込み要求
- 位置:** USBIRQ.0
- セット:** 内部 OSCOK ビットは、12MHz の内部発振器が安定し、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、このビットをセットし、チップの動作準備が完了したことを示します。
- クリア:** CPU は、1 を書き込むことによってこのビットをクリアします。
- パワーオンリセット:** OSCOKIRQ = 0
- チップリセット:** OSCOKIRQ = 0
- バスリセット:** OSCOKIRQ = 0
- パワーダウン:** 読出しのみ

# OUT0DAVIE

---

- 意味:** エンドポイント 0 の OUT データ利用可能の割込みイネーブル
- 位置:** EPIEN.1
- セット:** CPU は、このビットをセットして、OUT0DAV 割込み要求(51 ページ)をイネーブルにします。
- クリア:** CPU は、このビットをクリアして、OUT0DAV 割込み要求をディセーブルにします。
- パワーオンリセット:** OUT0DAVIE = 0
- チップリセット:** OUT0DAVIE = 0
- バスリセット:** OUT0DAVIE = 0
- パワーダウン:** 読出しのみ

## プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルをセットアップする初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# OUT0DAVIRQ

---

- 意味:** エンドポイント 0 の OUT データ利用可能の割込み要求
- 位置:** EPIRQ.1
- セット:** EP0 への OUT データパケットの受信(および肯定応答)に成功したとき、CPU は、このビットをセットします。
- クリア:** CPU は、1 を書き込むことによってこのビットをクリアします。これによって、CPU は別の転送のためのエンドポイントも準備します。
- パワーオンリセット:** OUT0DAVIRQ = 0
- チップリセット:** OUT0DAVIRQ = 0
- バスリセット:** OUT0DAVIRQ = 0
- パワーダウン:** 読出しのみ

# OUT1DAVIE

---

- 意味:** エンドポイント 1 の OUT データ利用可能の割込みイネーブル
- 位置:** EPIEN.2
- セット:** CPU は、このビットをセットして、OUT1DAV 割込み要求(53 ページ)をイネーブルにします。
- クリア:** CPU は、このビットをクリアして、OUT1DAV 割込み要求をディセーブルにします。
- パワーオンリセット:** OUT1DAVIE = 0
- チップリセット:** OUT1DAVIE = 0
- バスリセット:** OUT1DAVIE = 0
- パワーダウン:** 読出しのみ

## プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルをセットアップする初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# OUT1DAVIRQ

---

- 意味:** エンドポイント 1 の OUT データ利用可能の割込み要求
- 位置:** EPIRQ.2
- セット:** EP1-OUT への OUT データパケットの受信(および肯定応答)に成功したとき、CPU は、このビットをセットします。
- クリア:** CPU は、1 を書き込むことによってこのビットをクリアします。これによって、CPU は別の転送のためのエンドポイントも準備します。

**パワーオンリセット:** OUT1DAVIRQ = 0

**チップリセット:** OUT1DAVIRQ = 0

**バスリセット:** OUT1DAVIRQ = 0

**パワーダウン:** 読出しのみ

## プログラミング上の注意:

EP1OUT はダブルバッファ構造のエンドポイントです。つまり 2 つの FIFO と 2 つのバイトカウントレジスタがあるということです。ダブルバッファリングによって、1 つの FIFO に USB データが同時に入力される一方で、もう 1 つの FIFO から CPU がデータを読み出すことが可能になります。これによって、多くのシステムで帯域幅の性能が向上します。

OUT1DAVIRQ フラグのロジックによって、プログラマがダブルバッファリングを意識する必要はありません。たとえば、両バッファが利用可能で、したがって OUT1DAVIRQ = 0 であると仮定します。OUT パケットが到着すると、OUT1DAVIRQ は 0 から 1 に遷移して、最初のパケットが利用可能であることを示します。シングルバッファ構造のエンドポイントでは、CPU が FIFO を空にする時間がないうちに別の OUT パケットが EP1-OUT 上に到着した場合、SIE は NAK ハンドシェイクで応答し、データの受取りのためにエンドポイントを利用することができないことを示します。

しかし、ダブルバッファ構造のエンドポイントであれば、2 つ目のバッファをデータ用に利用することができるため、2 つ目の OUT パケットを受け取って肯定応答します。どちらかの FIFO が空になる前に 3 つ目の OUT パケットが到着した場合、SIE はその転送に否定応答し、FIFO が 2 つともいっぱいであることを示します。CPU が最初の FIFO の読み出しを終了し、OUT1DAVIRQ ビットをクリアすると(OUT1DAVIRQ ビットに 1 を書き込む)、CPU は直ちに別の 0 から 1 への遷移を実行し、2 つ目の FIFO でデータが利用可能であることを示します。

# POSINT

**意味:** INT 出力ピンが正エッジアクティブである(エッジ出力用に設定されている場合)。このビットは、INTLEVEL = 0(45 ページ)の場合にのみ有効です。

**位置:** PINCTL.2

**セット:** CPU は、割込みがサービスを要求するたびに、このビットをセットして、INT ピンを 0 から 1 に遷移させます。

**クリア:** CPU は、割込みがサービスを要求するたびに、このビットをクリアして、INT ピンを 1 から 0 に遷移させます。

**パワーオンリセット:** POSINT = 0

**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:

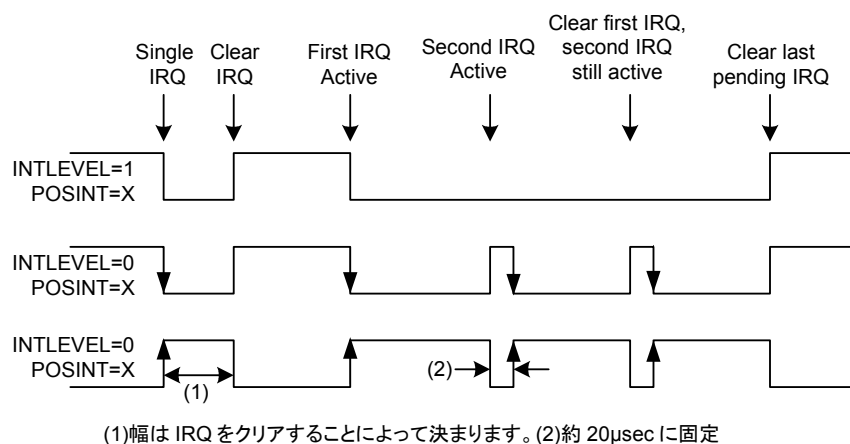


図 9. INTLEVEL ビットと POSINT ビットによって決まる INT ピンの動作

INTLEVEL = 1(45 ページ)の場合、このビットのセットは無効になります。

エッジセンシティブモードでは、新しい割込みがアクティブになるたびに、あるいは他の割込みの保留中に SPI マスタが IRQ をクリアするたびに、エッジが生成されます。図 9 に示すように、エッジアクティブのパルス幅は変化する可能性があります。最初のパルスが示すように、1 つの IRQ だけがアクティブである場合、SPI マスタが IRQ ビットをクリアするのにかかる時間によって幅が決まります。新しい IRQ がアサートされたとき、あるいは SPI マスタが IRQ をクリアしたときに他の IRQ が保留中の場合、パルス幅は約 20μ 秒に固定されます。

# PWRDOWN

---

意味:	MAX3420E をパワーダウンする
位置:	USBCTL.4
セット:	CPU は、このビットをセットして、チップを低電力状態にします。この状態は、サスペンド状態の USB 周辺機器によって要求されるものです。
クリア:	CPU は、このビットをクリアして、チップの低電力状態を解除し、動作を再開します。
パワーオンリセット:	PWRDOWN = 0
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:

パワーダウン動作は、PWRDOWN ビットを 0 から 1 に遷移させることによって起動されます。したがって、ビット PWRDOWN = 1 の間にチップが起動した場合(さまざまな起動方法については、以下を参照)、チップが直ちに再度パワーダウンすることはありません。いずれの起動ルーチンも、PWRDOWN ビットをクリアして次の 0 から 1 への遷移を準備する必要があります。

CPU は通常、PWRDOWN = 1 と HOSCSTEN = 1(37 ページ)をセットすることによって、MAX3420E をパワーダウンモードにします。CPU が PWRDOWN = 1 をセットすると、SIE は以下の措置を取ります:

- 12MHz の内部発振器を停止します。
- USB の DPLUS ピンをモニタし、バスアクティビティの有無を確認します。
- SPI ポートをモニタし、限定されたレジスタのセット(たとえば USBCTL)へのアクセスの有無を確認します。

チップはいったん低電力状態(PWRDOWN = 1)になると、次の 2 つの方法で起動することができます:

### 1. CPU が PWRDOWN ビットをクリアする

これによって内部発振器が起動され、内部発振器が安定すれば、SIE は OSCOK IRQ(49 ページ)をアクティブにします。

### 2. SIE が DPLUS 上のアクティビティとビット HOSCSTEN = 1 を検出する

HOSCSTEN = 0 をセットする自己給電デバイスの特異なケースの説明については、37 ページを参照してください。

# RWUDNIE

---

- 意味:** リモートウェイクアップ信号処理の割込みイネーブル
- 位置:** USBIEN.1
- セット:** CPU は、このビットをセットして、RWUDN 割込み要求(57 ページ)をイネーブルにします。
- クリア:** CPU は、このビットをクリアして、RWUDN 割込み要求をディセーブルにします。
- パワーオンリセット:** RWUDNIE = 0
- チップリセット:** RWUDNIE = 0
- バスリセット:** RWUDNIE = 0
- パワーダウン:** 読出しのみ

## プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルビットを立てる初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# RWUDNIRQ

---

- 意味:** リモートウェイクアップ信号処理の割込み要求
- 位置:** USBIRQ.1
- セット:** SIE は、RWU 信号の最後(K 状態の 10ms)にこのビットをセットします。
- クリア:** CPU は、1 を書き込むことによってこのビットをクリアします。
- パワーオンリセット:** RWUDNIRQ = 0
- チップリセット:** RWUDNIRQ = 0
- バスリセット:** RWUDNIRQ = 0
- パワーダウン:** 読出しのみ

**プログラミング上の注意:**

CPU は、ビット SIGRWU = 1(58 ページ)をセットすることによってリモートウェイクアップ信号を送出します。

# SIGRWU

---

- 意味:** リモートウェイクアップ信号を送出する
- 位置:** USBCTL.2
- セット:** CPU は、このビットをセットして、リモートウェイクアップ信号を USB ホストに送出します。
- クリア:** CPU は、このビットをクリアして、リモートウェイクアップ信号の送出を終了します。

**パワーオンリセット:** SIGRWU = 0

**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:

SPI マスタは、このビットをセットして、バス上への USB リモートウェイクアップ信号の送出を開始します。SPI マスタが SIGRWU = 1 をセットすると、MAX3420E は J 状態で 5 ミリ秒だけ待機した後、K 状態(D+ロー、D-ハイ)で 10 ミリ秒の間バスを駆動します。10 ミリ秒の間隔の後、MAX3420E はバスの駆動を停止して、RWUDNIRQ (57 ページ)割込みビットをアサートします。バスがサスペンド状態のときにのみ、SPI マスタは SIGRWU = 1 にセットする必要があります。

MAX3420E がパワーダウン状態(PWRDOWN = 1、55 ページ)にある間に、SPI マスタが SIGRWU = 1 にして USBCTL レジスタに書き込んだ場合、USBCTL レジスタへのこの書込みはさらに、PWRDOWN ビットをクリアして内部発振器を再起動します。この場合、SIE は発振器を再起動し、発振器が安定するのを待ってから RWU 信号の送出を開始します。

SPI マスタは SIGRWU = 1 をセットし、SIE が RWUDNIRQ(57 ページ)をアサートするのを待ってから信号送出の間隔が終了したことを示します。RWUDNIRQ がアサートされると、SPI マスタは SIGRWU = 0 をセットして信号送出をオフにする必要があります。10 ミリ秒の信号送出の間隔において SPI マスタが SIGRWU = 0 をセットした場合、10 ミリ秒が経過した後もなお RWU 信号はオフのままです。RWUDNIRQ 割込みがアサートされたときに SPI マスタが SIGRWU ビットをクリアしない場合、SIE は、5 ミリ秒のフロート状態(J)、10 ミリ秒の K 状態で、連続してバスを駆動します。

**参考:** USB 周辺機器は、以下の 2 つの前提条件が満たされる場合にのみ、リモートウェイクアップ信号を送出します:

1. エnumレーションの間に、デバイスのファームウェアによって、デバイスがリモートウェイクアップ信号を送出できることをホストに通知します。特に、構成記述子の bmAttributes フィールドは、ビット 5 を使用して、構成がリモートウェイクアップをサポートしているかどうかを示します(1 = yes、0 = no)。
2. ホストは、機能選択フィールドを「Device\_Remote\_Wakeup」に設定した状態で、Set\_Feature 要求を発行します。ホストは、最初の条件であるリモートウェイクアップ機能が誤りであることを発見した場合、この要求を発行することはありません。
- 3.

ホストは、選択フィールドを「Device\_Remote\_Wakeup」に設定した状態で Clear\_Feature 要求を発行することによって、後でリモートウェイクアップ信号の送出を禁止することができます。

<b>意味:</b>	エンドポイント 0 への IN 要求に回答して STALL ハンドシェイクを返す
<b>位置:</b>	EPSTALLS.0
<b>セット:</b>	CPU は、このビットをセットして、エンドポイント 0 宛ての IN 要求に対して STALL ハンドシェイクを返すように SIE に指示します。
<b>クリア:</b>	SIE は、SETUP トークンが到着するたびにこのビットをクリアします。
<b>パワーオンリセット:</b>	STLEP0IN = 0
<b>チップリセット:</b>	STLEP0IN = 0
<b>バスリセット:</b>	STLEP0IN = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

CPU は STALL ハンドシェイクを送信して、エンドポイント 0 への不正な要求または未知の要求であることを示します。

エンドポイント 0 には、以下に示す 3 つのストールビットがあります。これらは、ステータスステージ、およびオプションの IN と OUT のデータステージに対応するためのものであり、転送で使用することが可能です:

- STLSTAT(64 ページ)
- STLEP0IN
- STLEP0OUT(60 ページ)

CONTROL 転送の機能を停止する場合には、データステージ(データステージがある場合)とステータスステージの両方が STALL ハンドシェイクを受信する必要があります。次の SETUP トークンが到着すると 3 つのストールビットはすべてクリアされるため、あらゆる CONTROL 転送の機能を停止する最善の方法は、エンドポイント 0 の 3 つのストールビットをすべてセットすることです。これによって、ホストが送信可能なあらゆる CONTROL 転送の全ステージの機能が正しく停止されます。

# STLEP0OUT

---

<b>意味:</b>	エンドポイント 0 への OUT 要求に応答して STALL ハンドシェイクを返す
<b>位置:</b>	EPSTALLS.1
<b>セット:</b>	CPU は、このビットをセットして、エンドポイント 0 宛ての OUT 要求に対して STALL ハンドシェイクを返すように SIE に指示します。
<b>クリア:</b>	SIE は、SETUP トークンが到着するたびに、このビットをクリアします。
<b>パワーオンリセット:</b>	STLEP0OUT = 0
<b>チップリセット:</b>	STLEP0OUT = 0
<b>バスリセット:</b>	STLEP0OUT = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

CPU は STALL ハンドシェイクを送信して、エンドポイント 0 への不正な要求または未知の要求であることを示します。

エンドポイント 0 には、以下に示す 3 つのストールビットがあります。これらは、ステータスステージ、およびオプションの IN と OUT のデータステージに対応するためのものであり、転送で使用することが可能です。

- STLSTAT(64 ページ)
- STLEP0IN(59 ページ)
- STLEP0OUT

CONTROL 転送の機能を停止する場合には、データステージ(データステージがある場合)とステータスステージの両方が STALL ハンドシェイクを受信する必要があります。次の SETUP トークンが到着すると 3 つのストールビットはすべてクリアされるため、あらゆる CONTROL 転送の機能を停止する最善の方法は、エンドポイント 0 の 3 つのストールビットをすべてセットすることです。これによって、ホストが送信可能なあらゆる CONTROL 転送の全ステージの機能が正しく停止されます。

# STLEP1OUT

---

**意味:** EP1-OUT の機能を停止する(エンドポイント 1 への OUT 要求に応答して STALL ハンドシェイクを返す)

**位置:** EPSTALLS.2

**セット:** CPU は、このビットをセットして、エンドポイント 1 宛での OUT 要求に対して STALL ハンドシェイクを返すように SIE に指示します。

**クリア:** CPU は、このビットをクリアして、EP1-OUT を通常の ACK-NAK の動作に戻します。

**パワーオンリセット:** STLEP1OUT = 0

**チップリセット:** STLEP1OUT = 0

**バスリセット:** STLEP1OUT = 0

**パワーダウン:** 読出しのみ

## プログラミング上の注意:

CPU は、EP1-OUT 宛での Set\_Feature(HALT)を受信すると、このビットをセットします。

CPU は、EP1-OUT 宛での Clear\_Feature(HALT)を受信すると、このビットをクリアします。

# STLEP2IN

---

**意味:** EP2-IN の機能を停止する(エンドポイント 2 への IN 要求に回答して STALL ハンドシェイクを返す)

**位置:** EPSTALLS.3

**セット:** CPU は、このビットをセットして、エンドポイント 2 宛ての IN 要求に対して STALL ハンドシェイクを返すように SIE に指示します。

**クリア:** CPU は、このビットをクリアして、EP2-IN を通常の ACK-NAK の動作に戻します。

**パワーオンリセット:** STLEP2IN = 0

**チップリセット:** STLEP2IN = 0

**バスリセット:** STLEP2IN = 0

**パワーダウン:** 読出しのみ

## プログラミング上の注意:

CPU は、EP2-IN 宛ての Set\_Feature(HALT)を受信すると、このビットをセットします。

CPU は、EP2-IN 宛ての Clear\_Feature(HALT)を受信すると、このビットをクリアします。

- 意味:** エンドポイント 3 への IN 要求に応答して STALL ハンドシェイクを返す
- 位置:** EPSTALLS.4
- セット:** CPU は、このビットをセットして、エンドポイント 3 宛ての IN 要求に対して STALL ハンドシェイクを返すように SIE に指示します。
- クリア:** CPU は、EP3-IN 宛ての Clear\_Feature(Halt)要求を受信すると、このビットをクリアします。
- パワーオンリセット:** STLEP3IN = 0
- チップリセット:** STLEP3IN = 0
- バスリセット:** STLEP3IN = 0
- パワーダウン:** 読出しのみ

**プログラミング上の注意:**

CPU は、EP3-IN 宛ての Set\_Feature(HALT)を受信すると、このビットをセットします。

CPU は、EP3-IN 宛ての Clear\_Feature(HALT)を受信すると、このビットをクリアします。

# STLSTAT

---

- 意味:** CONTROL 転送の STATUS ステージに回答して STALL ハンドシェイクを返す
- 位置:** EPSTALLS.5
- セット:** CPU は、このビットをセットして、CONTROL 転送のステータスステージへの応答として STALL ハンドシェイクを送信します。CPU が転送に肯定応答するまで(ACKSTAT ビット = 1)、あるいは転送の機能を停止するまで(STLSTAT = 1)、SIE は、CONTROL 転送のステータスステージに NAK ハンドシェイクで応答します。
- クリア:** SIE は、SETUP トークンが到着するたびにこのビットをクリアします。
- パワーオンリセット:** STLSTAT = 0
- チップリセット:** STLSTAT = 0
- バスリセット:** STLSTAT = 0
- パワーダウン:** 読出しのみ

## プログラミング上の注意:

エンドポイント 0 には、以下に示す 3 つのストールビットがあります。これらは、ステータスステージ、およびオプションの IN と OUT のデータステージに対応するためのものであり、転送で使用することが可能です:

- STLSTAT
- STLEP0IN(59 ページ)
- STLEP0OUT(60 ページ)

CONTROL 転送の機能を停止する場合には、データステージ(データステージがある場合)とステータスステージの両方が STALL ハンドシェイクを受信する必要があります。次の SETUP トークンが到着すると 3 つのストールビットはすべてクリアされるため、あらゆる CONTROL 転送の機能を停止する最善の方法は、エンドポイント 0 の 3 つのストールビットをすべてセットすることです。これによって、ホストが送信可能なあらゆる CONTROL 転送の全ステージの機能が正しく停止されます。

- 意味:** SETUP データ利用可能の割込みイネーブル
- 位置:** EPIEN.5
- セット:** CPU は、このビットをセットして、SUDAV 割込み要求(66 ページ)をイネーブルにします。
- クリア:** SIE は、このビットをクリアして、SUDAV 割込み要求をディセーブルにします。

- パワーオンリセット:** SUDAVIE = 0
- チップリセット:** SUDAVIE = 0
- バスリセット:** SUDAVIE = 0
- パワーダウン:** 読出しのみ

#### プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルをセットアップする初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

# SUDAVIRQ

---

**意味:** SETUP データ利用可能の割込み要求

**位置:** EPIRQ.5

**セット:** SIE は、CONTROL 転送でエラーフリーのセットアップデータ 8 バイトを受信した後に、このビットをセットします。

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

**パワーオンリセット:** SUDAVIRQ = 0

**チップリセット:** SUDAVIRQ = 0

**バスリセット:** SUDAVIRQ = 0

**パワーダウン:** 読出しのみ

## プログラミング上の注意:

この IRQ は、SUDFIFO(67 ページ)から 8 バイトのセットアップデータを読み出してホストの要求を解釈するように CPU に通知します。

# SUDFIFO

---

**意味:** セットアップデータの FIFO

**位置:** SUDFIFO[7:0]

**書込み:** SIE は、ホストから CONTROL 転送を受信すると、8 バイトの SETUP データをこの FIFO にコピーします。このバイトがエラーフリーであることが確認されると、SIE はセットアップデータ利用可能の割込み要求(66 ページ)をアサートします。

**読出し:** CPU は、このレジスタを 8 回読み出して、8 バイトの SETUP を取り出します。

**パワーオンリセット:** SUDFIFO = 0

**チップリセット:** SUDFIFO = 0

**バスリセット:** SUDFIFO = 0

**パワーダウン:** 読出しも書込みもなし

## プログラミング上の注意:

MAX3420E は、EP0FIFO(10 ページ)でデータとコマンドのバイトが混合されないようにするために、この FIFO を個別に用意しています。これによって、FIFO 内でデータのタイプを見つけ出すというプログラマの作業が軽減されます。すなわち、EP0FIFO(10 ページ)には常にデータが格納され、SUDFIFO には常にコマンドが格納されます。

# SUSPIE

---

<b>意味:</b>	SUSPEND の割込みイネーブル
<b>位置:</b>	USBIEN.4
<b>セット:</b>	CPU は、このビットをセットして、SUSPEND の割込み要求(69 ページ)をイネーブルにします。
<b>クリア:</b>	CPU は、このビットをクリアして、SUSPEND の割込み要求をディセーブルにします。
<b>パワーオンリセット:</b>	SUSPIE = 0
<b>チップリセット:</b>	SUSPIE = 0
<b>バスリセット:</b>	SUSPIE = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルビットを立てる初期化ルーチンを、USB バスリセットの処理の一環として呼び出す必要があります。

<b>意味:</b>	SUSPEND の割込み要求
<b>位置:</b>	USBIRQ.4
<b>セット:</b>	SIE は、USB のサスペンドイベント(3 ミリ秒間バストラフィックがない)を検出すると、このビットをセットします。
<b>クリア:</b>	CPU は、1 を書き込むことによってこのビットをクリアします。
<b>パワーオンリセット:</b>	SUSPIRQ = 0
<b>チップリセット:</b>	SUSPIRQ = 0
<b>バスリセット:</b>	SUSPIRQ = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

CPU は、システムの周辺機器をシャットダウンすることによってこの割込みに応答した後、MAX3420E を低電力状態(PWRDOWN = 1。55 ページ)にします。

# URES DNIE

---

- 意味:** USB バスリセット処理の割込みイネーブル
- 位置:** USBIEN.7
- セット:** CPU は、このビットをセットして、URES DN IRQ(71 ページ)をイネーブルにします。
- クリア:** CPU は、このビットをクリアして、URES DN IRQ をディセーブルにします。
- パワーオンリセット:** URES DNIE = 0
- チップリセット:** URES DNIE = 0
- バスリセット:** 変化なし
- パワーダウン:** 読出しのみ

## プログラミング上の注意:

これは、USB バスリセットの間にクリアされない 2 つのビットのうちの 1 つです。もう 1 つのビットは URES IE(72 ページ)です。これらのビットはどちらも、通常、バスリセットの間に CPU によって使用されます。

# URES DNIRQ

---

- 意味:** USB バスリセット処理の割込み要求
- 位置:** USBIRQ.7
- セット:** SIE は、USB バスリセットの終了を検出したときにこのビットをセットします。
- クリア:** CPU は、1 を書き込むことによってこのビットをクリアします。
- パワーオンリセット:** URES DNIRQ = 0
- チップリセット:** URES DNIRQ = 0
- バスリセット:** 変化なし
- パワーダウン:** 読出しのみ

# URESIE

---

意味:	USB リセットの割込みイネーブル
位置:	USBIEN.3
セット:	CPU は、このビットをセットして、USB リセットの割込み要求(73 ページ)をイネーブルにします。
クリア:	CPU は、このビットをクリアして、USB リセットの割込み要求をディセーブルにします。
パワーオンリセット:	URESIE = 0
チップリセット:	URESIE = 0
バスリセット:	変化なし
パワーダウン:	読出しのみ

## プログラミング上の注意:

これは、USB バスリセットの間にクリアされない 2 つのビットのうちの 1 つです。もう 1 つのビットは URESDNIE (70 ページ)です。これらのビットはどちらも、通常、バスリセットの間に CPU によって使用されます。

# URESIRQ

---

<b>意味:</b>	USB バスリセットの割込み要求
<b>位置:</b>	USBIRQ.3
<b>セット:</b>	SIE は、USB バスリセット(少なくとも 2.5 $\mu$ sec の SE0 バス状態)を検出すると、このビットをセットします。
<b>クリア:</b>	CPU は、1 を書き込むことによってこのビットをクリアします。
<b>パワーオンリセット:</b>	URESIRQ = 0
<b>チップリセット:</b>	URESIRQ = 0
<b>バスリセット:</b>	URESIRQ = 1
<b>パワーダウン:</b>	読出しのみ

# VBUSIE

---

**意味:** Vbus 検出の割込みイネーブル

**位置:** USBIEN.6

**セット:** CPU は、このビットをセットして、VBUS 割込み要求(75 ページ)をイネーブルにします。

**クリア:** CPU は、このビットをクリアして、VBUS 割込み要求をディセーブルにします。

**パワーオンリセット:** VBUSIE = 0

**チップリセット:** VBUSIE = 0

**バスリセット:** VBUSIE = 0

**パワーダウン:** 読出しのみ

## プログラミング上の注意:

ほとんどの割込みイネーブルビットは USB バスリセットの間にクリアされるため、割込みイネーブルビットを立てる初期化ルーチンは、USB バスリセットの処理の一環として呼び出す必要があります。

# VBUSIRQ

---

<b>意味:</b>	VBUS 提示の割込み要求
<b>位置:</b>	USBIRQ.6
<b>セット:</b>	SIE は、Vbus コンパレータによって 0 から 1 への遷移が確立すると、このビットをセットしません。
<b>クリア:</b>	CPU は、1 を書き込むことによってこのビットをクリアします。
<b>パワーオンリセット:</b>	VBUSIRQ = 0
<b>チップリセット:</b>	VBUSIRQ = 0
<b>バスリセット:</b>	VBUSIRQ = 0
<b>パワーダウン:</b>	読出しのみ

## プログラミング上の注意:

この IRQ ビットを使用すると、MAX3420E によって組み込まれた USB 周辺機器が USB ホストに接続されたことを、自己給電デバイスが容易に確認することができます。

# VBGATE

---

- 意味:** Vbus ゲート
- 位置:** USBCTL.6
- セット:** CPU は、このビットをセットして、Vbus ピン上に Vbus が存在するかどうかを条件として CONNECT ビット(5 ページ)が動作するようにします。
- クリア:** CPU は、このビットをクリアして、Vbus が有効であるかどうかとは無関係に CONNECT ビットが動作するようにします。
- パワーオンリセット:** VBGATE = 0
- 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:

USB 仕様には、フルスピードの USB デバイスが、Vbus のない状態で DPLUS プルアップ抵抗器に電力を供給してはならないと規定されています。これは、専用の電源を有する自己給電デバイスで起こる可能性があり、自己給電デバイスは Vbus ピンからじかにプルアップ抵抗器に電力を供給しません。ホストが Vbus をオンにすることによってダウンストリームポートをアクティブにした場合にのみ、周辺機器は、プルアップ抵抗器に電力を供給し、バスに接続されていることを示すことができます。

VBGATE ビットは、以下の表に示すように、Vbus が存在する状態で CONNECT ビットの動作をゲート制御することによって、この要件を自動的に実行します:

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

VBGATE = 1 で CONNECT = 1 の場合、有効な Vbus 電圧の存在を Vbus 検出器が示すまで、プルアップ抵抗器は接続されません。

---

MAX3420E のレジスタにアクセスする方法 .....	3
ACKSTAT .....	1
BUSACTIE .....	2
BUSACTIRQ.....	3
CHIPRES.....	4
CONNECT .....	5
CTGEP1OUT .....	6
CTGEP2IN .....	7
CTGEP3IN .....	8
EP0BC.....	9
EP0FIFO.....	10
EP0INAK.....	11
EP1DISAB .....	12
EP1OUTBC .....	13
EP1OUTFIFO .....	14
EP2DISAB .....	15
EP2INAK.....	16
EP2INBC .....	17
EP2INFIFO .....	18
EP3DISAB .....	19
EP3INAK.....	20
EP3INBC .....	21
EP3INFIFO .....	22
FDUPSPI.....	23
FNADDR.....	26
GPIN0 .....	27
GPIN1 .....	28
GPIN2 .....	29
GPIN3 .....	30
GPOUT0 .....	31
GPOUT1 .....	32
GPOUT2 .....	33
GPOUT3 .....	34
GPXA .....	35
GPXB .....	36
HOSCSTEN.....	37
IE .....	38
IN0BAVIE .....	39
IN0BAVIRQ.....	40
IN2BAVIE .....	41
IN2BAVIRQ.....	42
IN3BAVIE .....	43
IN3BAVIRQ.....	44
INTLEVEL.....	45
NOVBUSIE.....	46
NOVBUSIRQ .....	47

<b>OSCOKIE</b> .....	<b>48</b>
<b>OSCOKIRQ</b> .....	<b>49</b>
<b>OUT0DAVIE</b> .....	<b>50</b>
<b>OUT0DAVIRQ</b> .....	<b>51</b>
<b>OUT1DAVIE</b> .....	<b>52</b>
<b>OUT1DAVIRQ</b> .....	<b>53</b>
<b>POSINT</b> .....	<b>54</b>
<b>PWRDOWN</b> .....	<b>55</b>
<b>RWUDNIE</b> .....	<b>56</b>
<b>RWUDNIRQ</b> .....	<b>57</b>
<b>SIGRWU</b> .....	<b>58</b>
<b>STLEP0IN</b> .....	<b>59</b>
<b>STLEP0OUT</b> .....	<b>60</b>
<b>STLEP1OUT</b> .....	<b>61</b>
<b>STLEP2IN</b> .....	<b>62</b>
<b>STLEP3IN</b> .....	<b>63</b>
<b>STLSTAT</b> .....	<b>64</b>
<b>SUDAVIE</b> .....	<b>65</b>
<b>SUDAVIRQ</b> .....	<b>66</b>
<b>SUDFIFO</b> .....	<b>67</b>
<b>SUSPIE</b> .....	<b>68</b>
<b>SUSPIRQ</b> .....	<b>69</b>
<b>URESDNIE</b> .....	<b>70</b>
<b>URESDNIRQ</b> .....	<b>71</b>
<b>URESIE</b> .....	<b>72</b>
<b>URESIRQ</b> .....	<b>73</b>
<b>VBUSIE</b> .....	<b>74</b>
<b>VBUSIRQ</b> .....	<b>75</b>
<b>VBGATE</b> .....	<b>76</b>
<b>索引</b> .....	<b>77</b>

2005年9月28日 変更:

レジスタ R12 ビット 5 を SUDAVIE に変更

SCK を SCLK に置き換え(MAX3420E のデータシートの信号名と一致させるため)