

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

MESC TECHNICAL NEWS

No. M16C-60-0009

M30240 グループ USB 制御レジスタ に関する注意事項

1. 対象品種

M30240 グループ

2. 現 象

表1に示すUSB制御レジスタにデータを書き込む際にリードモディファイライト命令を使用すると、タイミングによってはごくまれに誤ったデータが書かれる可能性があります。

2.1 詳細説明

表 1 に示す USB 制御レジスタは、同一レジスタ内にハードウェアとソフトウェアの両方で変化可能なステータスフラグ / 制御ビットを含みます。これらのレジスタに対してリードモディファイライト命令を実行すると、CPUがレジスタ値を読み出した後にハードウェアによってビットステータスが変化した場合、その変化したビットに対してCPUは前に読み出した時の値を上書きします。2.2に例を示します。

表 1. USB エンドポイント x 制御 / ステータスレジスタ

シンボル	レジスタ名	番 地
EP0CS	USB エンドポイント 0 制御 / ステータスレジスタ	031116
EP1ICS	USB エンドポイント 1 IN 制御 / ステータスレジスタ	031916
EP1OCS	USB エンドポイント 1 OUT 制御 / ステータスレジスタ	031A16
EP2ICS	USB エンドポイント 2 IN 制御 / ステータスレジスタ	032116
EP2OCS	USB エンドポイント 2 OUT 制御 / ステータスレジスタ	032216
EP3ICS	USB エンドポイント 3 IN 制御 / ステータスレジスタ	032916
EP3OCS	USB エンドポイント 3 OUT 制御 / ステータスレジスタ	032A16
EP4ICS	USB エンドポイント 4 IN 制御 / ステータスレジスタ	033116
EP4OCS	USB エンドポイント 4 OUT 制御 / ステータスレジスタ	033216

表 2. リードモディファイライト命令一覧

機 能	ニ ー モ ニ ッ ク
ビット処理	BCLR, BNOT, BSET, BTSTC, BTSTS
シフト	ROLC, RORC, ROT, SHA, SHL
算術	ABS, ADC, ADCF, ADD, DEC, EXTS, INC, MUL, MULU, NEG, SBB, SUB
論理	AND, NOT, OR, XOR
ジャンプ	ADJNZ, SBJNZ

2.2 例 (図1参照)

EP1ICSのIN_PKT_RDY ビットを“1”にセットした状態で、BSET命令を使用してISO/TOGGLE_INIT ビットを“1”にセットする場合を例に説明します。

- (1) EP1ICS レジスタの内容が BSET 命令のリードステージで読み出される。(このとき IN_PKT_RDY ビットは“1”)
- (2) 命令実行中にデータ転送が完了した場合、USB ファンクション制御ユニット (FCU) によって IN_PKT_RDY ビットが“0”にクリアされる。
- (3) BSET 命令のモディファイステージで ISO ビットが“0”にクリアされた EP1ICS レジスタの値が、ライトステージで EP1ICS レジスタに書かれる。このとき IN_PKT_RDY ビットには前の値である“1”が上書きされる。
- (4) IN_PKT_RDY ビットが“0”から“1”に変化したため、MCU はホストに対して余分なパケットを送信してしまう。

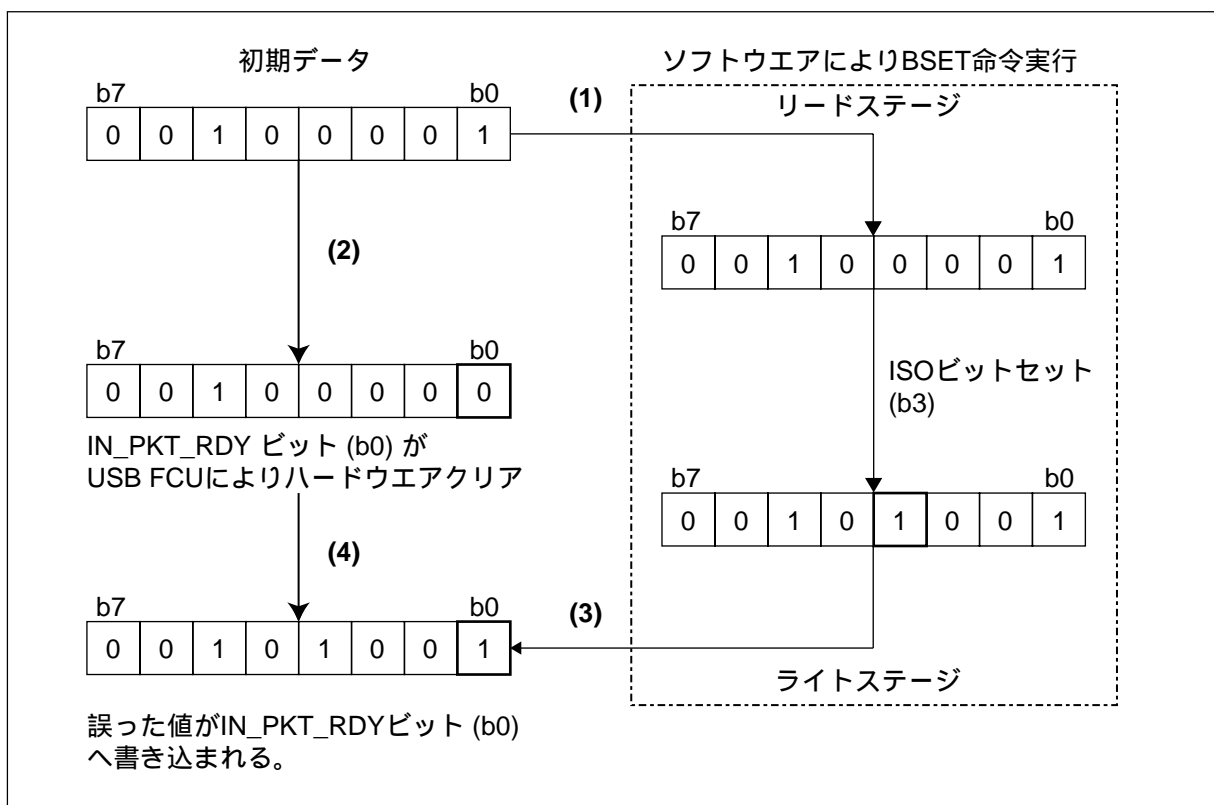


図1. リードモディファイライト命令によるエラー例

3. 対策

表1のUSB制御レジスタに対して、表2に示すリードモディファイライト命令を使用しないでください。

これらのレジスタ内のビットをソフトウェアによって変更する際は、転送命令を使用し、次の手順で変更してください。

- (1) レジスタの内容を変数またはデータレジスタに退避。
- (2) 変数またはデータレジスタ上で、目的のビットを変更。同時に、ソフトウェア書き込みによるステータス変化を避けるべきビット値をマスクする (表3参照)。
- (3) MOV 命令などの転送命令を使用して、変数またはデータレジスタからレジスタへ値を書き込む。

表3に、ソフトウェア書き込みによってステータスが変化する可能性のあるビットと、その書き込み回避値を示します。表3の値はソフトウェアによる書き込みができない値であり、この値を書き込んでもビットステータスが変化することはありません。

図2に、EP1ICSを変更する場合のC言語プログラミング例を示します。

この例では、ソフトウェア書き込みによるビットステータス変化を避けるためIN_PKT_RDYビットに“0”を、UNDER_RUNビットに“1”を書き込んでいます。

表3. ソフトウェア書き込みによりステータス変化の可能性のあるビット

レジスタ名	ビット名	書き込みによる影響がない値
EP0CS	IN_PKT_RDY (b1)	“ 0 ”
	DATA_END (b3)	“ 0 ”
	FORCE_STALL (b4)	“ 1 ”
EPxICS (x = 1 ~ 4)	IN_PKT_RDY (b0)	“ 0 ”
	UNDER_RUN (b1)	“ 1 ”
EPxOCS (x = 1 ~ 4)	OUT_PKT_RDY (b0)	“ 1 ”
	OVER_RUN (b1)	“ 1 ”
	FORCE_STALL (b4)	“ 1 ”
	DATA_ERR (b5)	“ 1 ”

C 言語記述例

```
#pragma ADDRESS EP1ICS 0319h
char near Variable;

void Func (void)
{
    :
    :
    Variable = EP1ICS;          /* EP1ICS レジスタ退避 */
    Variable |= 0x0A;          /* ISO ビットセット。UNDER_RUN ビットに “ 1 ” 書き込み */
    Variable &= 0xFE;         /* IN_PKT_RDY ビットに “ 0 ” 書き込み */
    EP1ICS = Variable;        /* EP1ICS レジスタに再書き込み */
    :
    :
}
```



アセンブラ言語、展開イメージ

```
:
:
mov.b EP1ICS,Variable      ; EP1ICS レジスタ退避
or.b #0Ah,Variable        ; ISO ビットセット。UNDER_RUN ビットに “ 1 ” 書き込み
and.b #0FEh,Variable      ; IN_PKT_RDY ビットに “ 0 ” 書き込み
mov.b Variable,EP1ICS     ; EP1ICS レジスタに再書き込み
:
:
```

図2. C 言語プログラミング例

以上