

# RENESAS TECHNICAL UPDATE

〒211-8668 神奈川県川崎市中原区下沼部 1753

ルネサス エレクトロニクス株式会社

問合せ窓口 <http://japan.renesas.com/contact/>E-mail: [csc@renesas.com](mailto:csc@renesas.com)

製品分類	MPU & MCU	発行番号	TN-RL*-A035A/J	Rev.	第1版
題名	RL78/L1C 制限事項について		情報分類	技術情報	
適用製品	RL78/L1C R5F110xx,R5F111xx	対象ロット等	関連資料	RL78/L1C ユーザーズマニュアル ハードウェア編 Rev.2.00 R01UH0409JJ0200 (Jan. 2014)	
		全ロット			

上記適用製品において、除算命令(DIVHU、DIVWU)に制限事項がございます。

## 今回通知する制限事項一覧

項目	制限事項	本通知での 該当ページ
1	除算命令(DIVHU、DIVWU)の制限事項	p.2-p.6

## 発行文書履歴

RL78/L1C 制限事項 発行文書履歴

文書番号	発行日	記事
TN-RL*-A035A/J	2014年10月14日	初版発行 制限事項一覧の項目1(本通知です。)

## 1. 制限事項について

### 【対象の使用方法】

使用方法が下記の①～④の**全てにあてはまる**場合、本制限の対象となります。

- ①割り込み処理内で“対象命令 1”となる除算命令(DIVHU, DIVWU)を実行している。
- ②除算(DIVHU, DIVWU)を実行する割り込み処理で、多重割り込みを許可にしている。
- ③除算命令(DIVHU, DIVWU)を実行する割り込み処理中に、優先順位の異なる2つ以上の割り込みが発生する。  
対象となる割り込みの優先順位に関しては、2. 制限事項の内容 の 表 1 をご確認ください。
- ④除算(DIVHU, DIVWU)の次の命令に“対象命令 2”を使用している。  
“対象命令 2”に関しては、5. 対象命令 2 一覧 をご確認ください。  
ただし、除算命令(DIVHU, DIVWU)を RAM で実行している場合は、全ての命令が“対象命令 2”になります。

## 2. 制限事項の内容

下記の①～③が全て実行されると、割り込みAから割り込みCへ分岐時、もしくは、割り込みCから割り込みAへ分岐時に不正な動作をする場合があります。

- ①多重割り込みを許可している割り込み A 処理内で対象命令 1(DIVHU, DIVWU)と対象命令 2 が連続している。
- ②割り込み A の実行中に割り込み B が発生し、割り込み B が保留状態となる。
- ③除算命令が実行完了される直前 2 クロックの期間(DIVHU 命令:8, 9 クロック, DIVWU 命令:16, 17 クロック)に割り込み C が発生する。

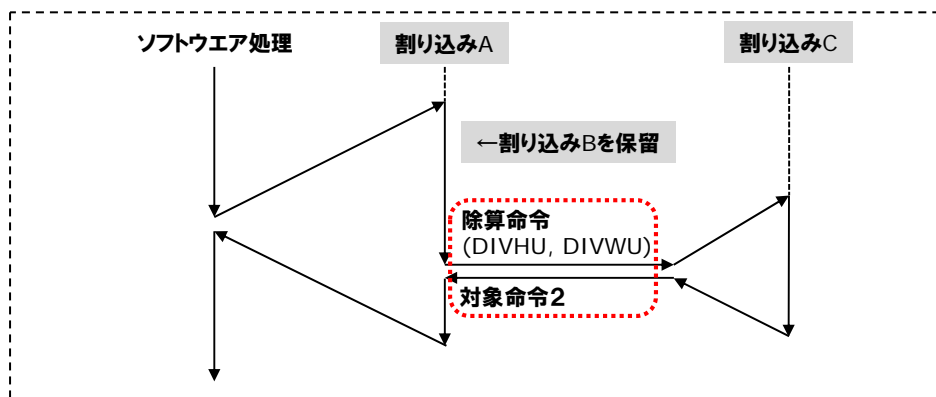


図 1. 制限事項の動作

注意 1 “対象命令 2”に関しては 5. 対象命令 2 一覧をご確認ください。

注意 2 割り込み発生/保留の関係は割り込み優先順位レベル(0~3)の組み合わせに依存します。

対象になる割り込み優先順位レベルの組み合わせは表 1 の通りです。

表 1. 制限事項が発生する割り込み優先順位の組み合わせ

割り込み A の割り込み優先順位	割り込み B の割り込み優先順位	割り込み C の割り込み優先順位	対象/対象外
レベル 0	レベル 1/ レベル 2/ レベル 3	レベル 0	対象
レベル 1	レベル 1/ レベル 2/ レベル 3	レベル 0	
レベル 2	レベル 2/ レベル 3	レベル 0/ レベル 1	
レベル 3	レベル 3	レベル 0/ レベル 1/ レベル 2	
上記以外			対象外

### 3. ソフトウェアによる対策

多重割り込みを許可する割り込み処理内で除算 / 剰余算命令を実行する場合は、下記の対策のいずれかを行ってください。

(A) 除算 / 剰余算演算中の割り込みを禁止にしてください。

対策適用例:

```
__asm("push PSW");
DI();
C 言語での除算 / 剰余算演算
__asm("pop PSW");
```

(B) 除算命令(DIVHU, DIVWU)の直後に NOP 命令を挿入してください。

除算命令(DIVHU, DIVWU)を RAM で実行している場合は、実行場所をコード・フラッシュへ変更し、下記の対策を行ってください。

対策適用例 :

```
DIVWU ; 除算命令
NOP ; NOP 命令を挿入
RET ; 対象命令 2
```

但し、C 言語などの高級言語を使用している場合、1コードに対してコンパイラが制限事項の対象となる 2 命令を生成する場合があります。この場合、NOP 命令を挿入する対策を実施することは非常に困難なため、対策 (A)を実施してください。

注意 ルネサス製コンパイラ CA78K0 の場合、DI(); の使用には #pragma di の記述が必要です。

#### 4. 恒久対策

恒久対策として、ルネサス製コンパイラ CA78K0R に今回の問題の対策を行います。

対策内容:

ビルド時に DIVWU 命令、および DIVHU 命令が出力される場合、これらの命令の直後に必ず NOP 命令を挿入します。この対策により「3. ソフトウェアによる対策」を行う必要がなくなります。<sup>注</sup>

対策後、CA78K0R のバージョンは V1.71 となります。

対策版コンパイラのリリース予定時期:11月18日

注. 除算命令(DIVHU、DIVWU)を RAM で実行している場合は対策が必要です。

### 5. 対象命令2一覧

対象命令1(DIVHU、DIVWU)の次に“対象命令2”を使用している場合、制限事項の対象となります。  
条件1～3のうち、1条件でも当てはまる場合は対象命令2となります。

条件 1. 命令の実行に2クロック以上必要な命令。

命令	オペランド	命令	オペランド	命令	オペランド
XCH	A, saddr	INC	saddr	CALL	全て
	A, sfr	INC	laddr16	CALLT	全て
	A, !addr16	INC	[HL+byte]	BRK	-
	A, [DE]	DEC	saddr	RET	-
	A, [DE+byte]	DEC	laddr16	RETI	-
	A, [HL]	DEC	[HL+byte]	RETB	-
	A, [HL+byte]	INCW	saddrp	BR	全て
	A, [HL+B]	INCW	laddr16	BC	全て
	A, [HL+C]	INCW	[HL+byte]	BNC	全て
ADD	saddr, #byte	DECW	saddrp	BZ	全て
ADD C	saddr, #byte	DECW	laddr16	BNZ	全て
SUB	saddr, #byte	DECW	[HL+byte]	BH	全て
SUB C	saddr, #byte	MOV1	saddr.bit, CY	BNH	全て
AND	saddr, #byte	MOV1	sfr.bit, CY	BT	全て
OR	saddr, #byte	MOV1	[HL].bit, CY	BF	全て
XOR	saddr, #byte	SET1	saddr.bit	BTCLR	全て
		SET1	sfr.bit	HALT	-
		SET1	laddr16.bit	STOP	-
		SET1	[HL].bit		
		CLR1	saddr.bit		
		CLR1	sfr.bit		
		CLR1	laddr16.bit		
		CLR1	[HL].bit		

条件 2. コード・フラッシュ・メモリの読出しを行う命令。

下記の命令を使用し、Mirror 領域、コード・フラッシュ・メモリ領域を読み出す場合は対象になります。

命令	オペランド	命令	オペランド	命令	オペランド	命令	オペランド
MOV	A, !addr16	MOVW	AX, !addr16	ADD ADDC SUB SUBC AND OR XOR	A, !addr16	ADDW	AX, !addr16
	A, [DE]		AX, [DE]		A, [HL]		AX, [HL+byte]
	A, [DE+byte]		AX, [DE+byte]		A, [HL+byte]		AX, ES:!addr16
	A, [HL]		AX, [HL]		A, [HL+B]		AX, [HL+byte]
	A, [HL+byte]		AX, [HL+byte]		A, [HL+C]		AX, ES:!addr16
	A, [HL+B]		AX, word[B]		A, ES:!addr16		AX, [HL+byte]
	A, [HL+C]		AX, word[C]		A, ES:[HL]		AX, ES:!addr16
	A, word[B]		AX, word[BC]		A, ES:[HL+byte]		AX, ES:[HL+byte]
	A, word[C]		BC, !addr16		A, ES:[HL+B]		AX, !addr16
	A, word[BC]		DE, !addr16		A, ES:[HL+C]		AX, [HL+byte]
	B, !addr16		HL, !addr16	A, !addr16	AX, ES:!addr16		
	C, !addr16		AX, ES:!addr16	A, [HL]	AX, ES:[HL+byte]		
	X, !addr16		AX, ES:[DE]	A, [HL+byte]	AX, ES:[HL+byte]		
	A, ES:!addr16		AX, ES:[DE]	A, [HL+B]	CY, [HL].bit		
	A, ES:[DE]		AX, ES:[DE+byte]	A, [HL+C]	CY, ES:[HL].bit		
	A, ES:[DE+byte]		AX, ES:[HL]	laddr16, #byte	CY, [HL].bit		
	A, ES:[HL]		AX, ES:[HL]	A, ES:!addr16	CY, ES:[HL].bit		
	A, ES:[HL+byte]		AX, ES:[HL+byte]	A, ES:[HL]	CY, [HL].bit		
	A, ES:[HL+B]		AX, ES:word[B]	A, ES:[HL+byte]	CY, ES:[HL].bit		
	A, ES:[HL+C]		AX, ES:word[C]	A, ES:[HL+B]	CY, [HL].bit		
A, ES:word[B]	AX, ES:word[BC]	A, ES:[HL+C]	CY, ES:[HL].bit				
A, ES:word[C]	AX, ES:word[BC]	ES:!addr16, #byte	CY, ES:[HL].bit				
A, ES:word[BC]	BC, ES:!addr16	laddr16	BT				
B, ES:!addr16	DE, ES:!addr16	CMP0	ES:[HL].bit, \$saddr20				
C, ES:!addr16	HL, ES:!addr16	CMP0	ES:[HL].bit, \$saddr20				
X, ES:!addr16		CMPS					
			X, [HL+byte]				
			X, ES:[HL+byte]				

条件3. 割り込み要求の保留命令。

下記の命令は割り込み要求を保留するため対象となります。

命令	オペランド
MOV	PSW, #byte
MOV	PSW, A
MOV1	PSW.bit, CY
SET1	PSW.bit
CLR1	PSW.bit
RETB	-
RETI	-
POP	PSW
BTCLR	PSW.bit, \$addr20
EI	-
DI	-
SKC	-
SKNC	-
SKZ	-
SKNZ	-
SKH	-
SKNH	-

また、下記レジスタに対する書き込み命令も割り込み要求を保留するため対象となります。

下記のレジスタが配置されているアドレスを指定しての書き込みも対象です。

- ・ 割り込み要求フラグ・レジスタ  
IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L
- ・ 割り込みマスク・フラグ・レジスタ  
MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L
- ・ 優先順位指定フラグ・レジスタ  
PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L  
PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L

上記のレジスタに対する書き込みが可能な命令は下記通りです。

命令	オペランド	命令	オペランド	命令	オペランド
MOV	sfr, #byte	XCH	A, sfr	INC	!addr16
MOV	!addr16, #byte	XCH	A, !addr16	INC	[HL+byte]
MOV	sfr, A	XCH	A, [DE]	DEC	!addr16
MOV	!addr16, A	XCH	A, [DE+byte]	DEC	[HL+byte]
MOV	[DE], A	XCH	A, [HL]	INCW	!addr16
MOV	[DE+byte], #byte	XCH	A, [HL+byte]	INCW	[HL+byte]
MOV	[DE+byte], A	XCH	A, [HL+B]	DECW	!addr16
MOV	[HL], A	XCH	A, [HL+C]	DECW	[HL+byte]
MOV	[HL+byte], #byte	ONEB	!addr16	MOV1	sfr.bit, CY
MOV	[HL+byte], A	CLRB	!addr16	MOV1	[HL].bit, CY
MOV	[HL+B], A	MOVW	[HL+byte], X	SET1	sfr.bit
MOV	[HL+C], A	MOVW	sfrp, #word	SET1	!addr16.bit
MOV	word[B], #byte	MOVW	sfrp, AX	SET1	[HL].bit
MOV	word[B], A	MOVW	!addr16, AX	CLR1	sfr.bit
MOV	word[C], #byte	MOVW	[DE], AX	CLR1	!addr16.bit
MOV	word[C], A	MOVW	[DE+byte], AX	CLR1	[HL].bit
MOV	word[BC], #byte	MOVW	[HL], AX	BTCLR	sfr.bit, \$addr20
MOV	word[BC], A	MOVW	[HL+byte], AX	BTCLR	[HL].bit, \$addr20
		MOVW	word[B], AX		
		MOVW	word[C], AX		
		MOVW	word[BC], AX		

以上