Date: Jun. 30, 2016

RENESAS TECHNICAL UPDATE

TOYOSU FORESIA, 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan
Renesas Electronics Corporation

Product Category	MPU/MCU		Document No.	TN-RL*-A014B/E	Rev.	2.00
Title	RL78/G1C Direction of use		Information Category	Technical Notification		
Applicable Product	RL78/G1C Group	Lot No. All lots	Reference Document	RL78/G1C User's Manual: Hardware Rev.1.10 R01UH0348EJ0110 (Nov. 2013)		

A restriction on directions of transitions of the CPU clock state has been added for products of the RL78/G1C group.

Restriction reported in this document

Section	Restriction	Target Products	Page Nos. in This Document
1.1	Restriction on transitions of the CPU clock state	All products of the RL78/G1C group	Pages 2 to 5

List of restrictions which have already been reported

Section	Restriction	Target Products	Page Nos. in This Document
2.1	Operating precaution for data flash read access	All products of the RL78/G1C group R5F10JxxA, R5F10JxxG, R5F10KxxA, R5F10KxxG Please see Appendix1.	Pages 6 to 10

Revision history

Revision history of technical updates on restrictions of the RL78/G1C

Document Number	Issued Date	Description
TN-RL*-A014A/E	Aug. 19, 2013	First edition Section 2.1 shown in the table under "List of restrictions which have already been reported"
TN-RL*-A014B/E	Jun. 30, 2016	Second revision Section 1.1 added in the table under "Restriction reported in this document" (this document)

1. Restriction added in this document

1.1 Restriction on transitions of the CPU clock state

1.1.1 Restriction

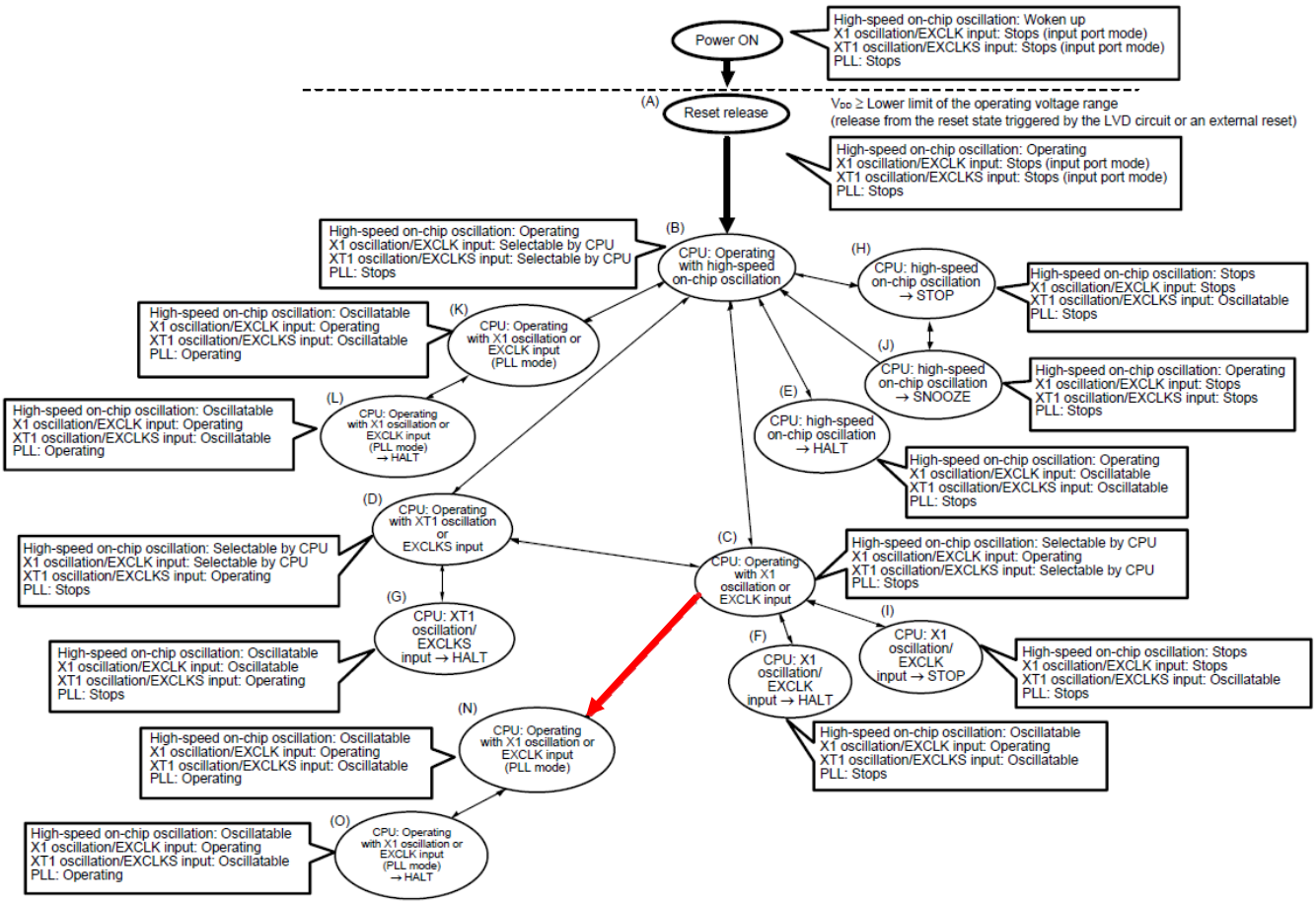
[Applicable Usage]

The restriction is applicable when usage satisfies both conditions (1) and (2) below.

(1) The PLL clock signal is selected as the operating clock for the USB 2.0 host/function module (USB).

(2) In resuming from the suspended state, X1 oscillation or EXCLK input is selected as the source of the CPU clock to return from the STOP or HALT mode (transitions of states between (N) and (O) from (C) shown in figure 1).

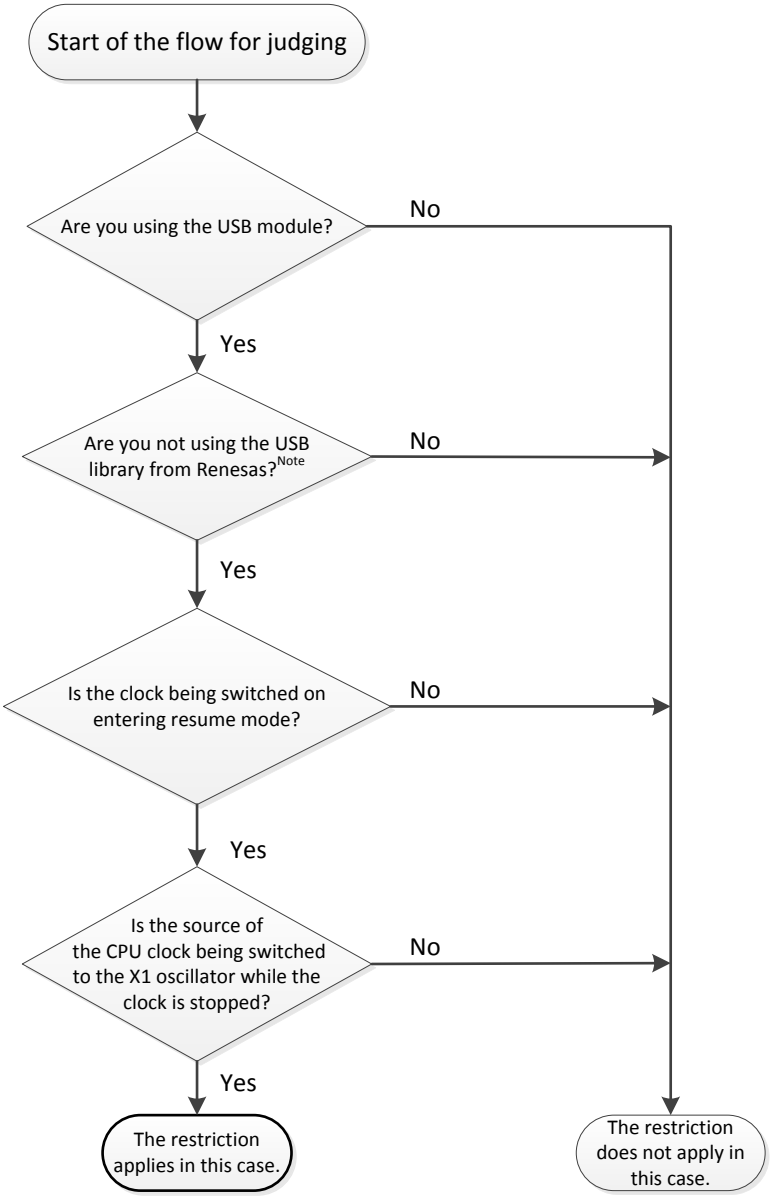
Figure 1 CPU Clock Status Transition Diagram



[Flowchart for judging whether the problem will affect your own usage of the USB module]

A flowchart for judging whether the problem will affect your own usage of the USB module is given in figure 2.

Figure 2 Flowchart for judging if your software is affected



Note The restriction is never applicable if you are using the “USB Host and Peripheral Basic Mini Firmware” USB library without change.

Note that the USB library is for reference. Hence, we do not guarantee its operation.

1.1.2. Details of the Restriction

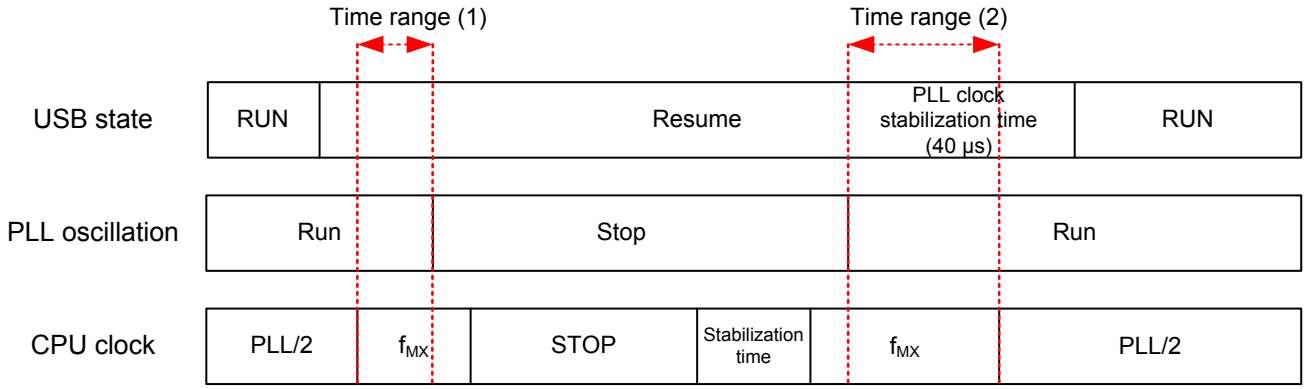
In switching from the X1 oscillator to the PLL oscillator in a way that satisfies the conditions on page 2, registers of the USB module may become inaccessible if the following conditions are also met.

Condition 1: f_{MX} is selected as the CPU clock and the PLL clock is running.

Condition 2: The USB registers are accessed.

Specifically, in release from the STOP of HALT mode, attempted access to the USB registers (see table 14-3 in the RL78/G1C User's Manual: Hardware) at times (1) and (2), where the clock is being changed, may lead to reading or writing not proceeding correctly.

Figure 3 Timing Chart of the Processing that Gives Rise to the Restriction



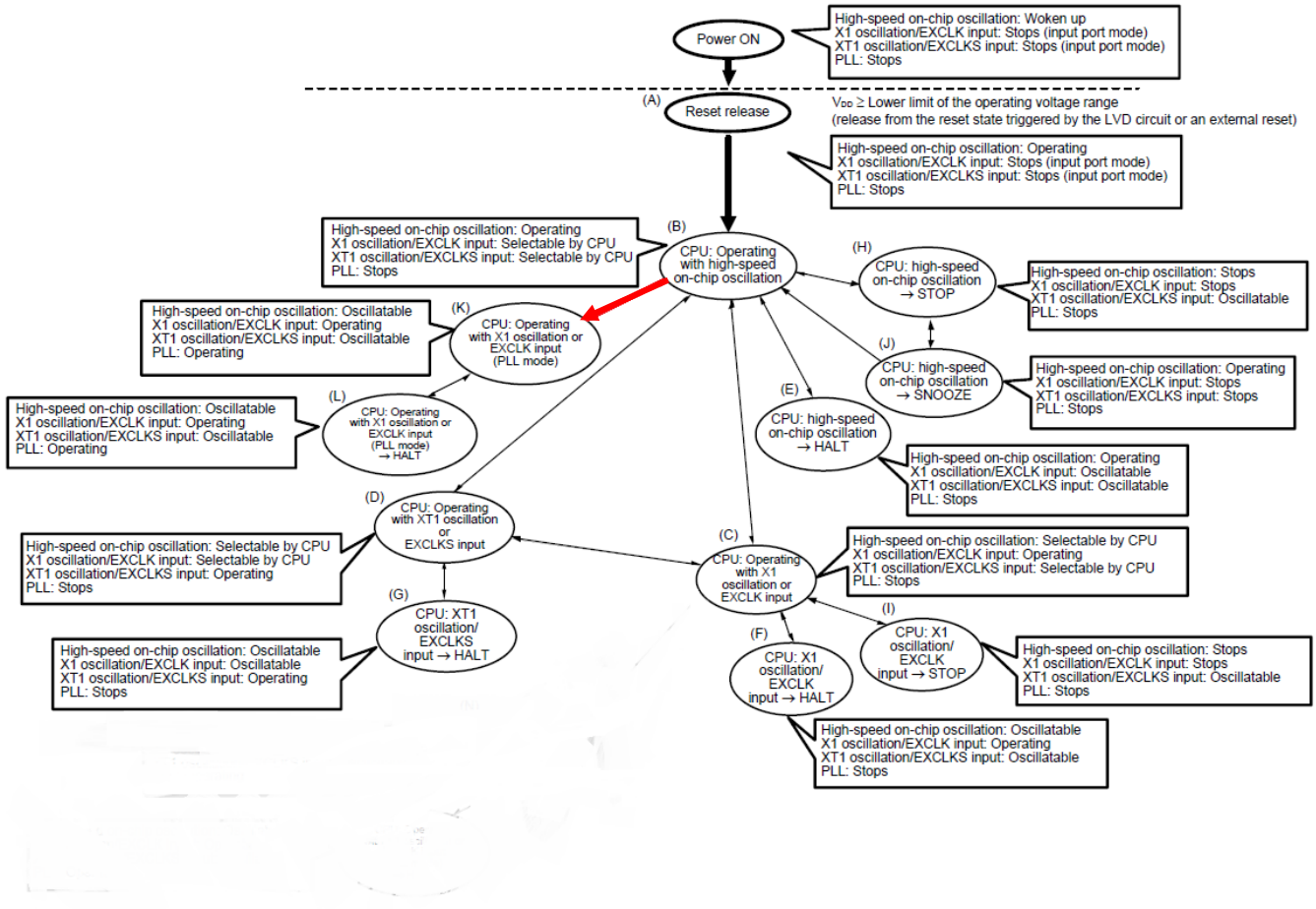
1.1.3. Workaround

The workaround is to avoid transitions of the CPU clock state from (N) to (O).

To avoid the problem, switch the clock source by using transitions between (B) and (K) as shown in figure 4, CPU Clock Status Transition Diagram (New).

When having the USB operate with the PLL clock signal and suspend or resume its operation while the device is in the suspended state, set the high-speed on-chip oscillator as the source of the CPU clock.

Figure 4 CPU Clock Status Transition Diagram (New)



1.1.4. Modification schedule

We are handling this countermeasure as a restriction on usage.

This matter is added to “CPU Clock Status Transition Diagram” of CHAPTER 5 CLOCK GENERATOR in the user’s manual by the next revision.

2. Direction notified in Rev.1

2.1 Operating Precaution for Data Flash read access

2.1.1 Restriction

【Applicable Usage】

The usage which meets to all of (1), (2), and (3) is applicable to the restriction.

(1) Using both DMA and Data Flash.

(2) DMA is operating when Data Flash Note1 read occurs.

(3) Data Flash is read using flash-related libraries Renesas Electronics is offering, which are listed below. Otherwise instead of using those libraries, the combination of CPU Related instructions Note2 are used for reading related memory Note3 and Data Flash.

- The EEL(EEPROM emulation library) Note1 Pack01 V1.12 or earlier version.
- The FDL(Data Flash library) Type01 V1.11 or earlier version.
- The FDL Type02 V1.00 or earlier version.
- The FDL Type04 V1.04 or earlier version.

Note1. When EEL is used, commands other than READ command are also related: they also make read access to the Data Flash.

Note2. See Appendix2 about the combination of the related instructions1 and 2.

Note3. Related memory is RAM(Include general purpose register area),SFR,2nd SFR(No wait), ES,CS,PSW,SP

【Detail of Restriction】

In the case that DMA transfer is operated and it is immediately followed by read access to the target memory (Related instructions 1) which access is also immediately followed in sequence, by read access to Data Flash (Related instructions 2), because of the conflict on the internal bus between read access to the target memory and to the Data Flash, the read out result from the target memory may be wrongly changed.

Example for an instruction sequence causing this issue:

DMA transfer trigger

DMA transfer

```
MOVW    HL,!addr16    ; read data from RAM (Related instruction 1)
MOV     A,[DE]        ; read data from Data Flash (Related instructions 2)
```

When DMA transfer occurs as mentioned above timing, a wrong data is loaded into HL register.

2.1.2 Workaround

If you have any possibility that read access to the Data Flash and the DMA transfer could operate in the same time, please apply the following procedures according to the way to read out the Data Flash.

Case 1:

Data Flash Read access via the 'Data Flash Access Library' (FDL) and/or EEPROM Emulation Library (EEL). Both libraries are developed under the responsibility of Renesas.

Workaround for Case 1:

There are currently one type of EEL supported and three type of FDL supported and all of them will be updated to cover the aforementioned workaround.

Library version (Not installer version)

EEL (Pack01) version V1.13 Note or later

FDL (Type01) version V1.12 Note or later

FDL (Type02) version V1.01 Note or later

FDL (Type04) version V1.05 Note or later

Case 2:

Data Flash Read access directly executed in the user software without library.

Workaround for Case 2:

Please apply either of the following procedures.

(A) Holding DMA or forcing termination DMA

In case, the user software has to perform a direct Data Flash Read access without using the FDL read commend, any possible DMA transfer must be stopped before the Data Flash read access is executed. To stop any DMA transfer, please follow the procedure given in the User Manual.
Furthermore, please make sure to wait at least 3 clocks(f_{CLK}) after setting DWAITn bit to "1" before the Data Flash read instruction is executed. Restart any DMA transfer (by clearing DWAITn bit to "0") after The Data Flash read access have been finished.

(B) Reading Data Flash by using library

When access Data Flash, please use latest Data Flash library of case 1.

(C) Inserting a NOP instruction

Such kind of conflict can be avoided by inserting a NOP instruction immediately prior to any Data Flash Read access.

Example to avoid this issue: operand

```
MOVW    HL,   !addr16     ; Read data from RAM
NOP                       ; Insert a NOP prior to the DF read access
MOV     A, [DE]           ; Read data from Data Flash
```

In case the application software will use the DMA feature, Renesas strongly recommend not to perform a direct Data Flash Read access in the user software, because in case of a high level language (e.g. C-Language) it cannot be avoided that the C-compiler may generate a code sequence as described before. Therefore, Renesas strongly recommend to perform the Data Flash Read access ONLY via the corresponding FDL read command.

Note. The modified version of EEL(EEPROM Emulation library) and FDL(Data Flash library) will be released in sequence after July 2013.

Remark. f_{CLK}: CPU/peripheral hardware clock frequency

2.1.3 Modification schedule

This matter is added to “Procedure for accessing data flash memory” of CHAPTER 26 FLASH MEMORY in the user’s manual by the next revision.

【Target products' name list】

RL78/G1C

32pin LQFP 7x7mm	R5F10JBCAFP, R5F10KBCAFP R5F10JBCGFP, R5F10KBCGFP
32pin HWQFN 5x5mm	R5F10JBCANA, R5F10KBCANA R5F10JBCGNA, R5F10KBCGNA
48pin LFQFP 7x7mm	R5F10JGCAFB, R5F10KGCAFB R5F10JGCGFB, R5F10KGCGFB
48pin HWQFN 7x7mm	R5F10JGCANA, R5F10KGCANA R5F10JGCGNA, R5F10KGCGNA

【Appendix2-1】

【Related instructions list】

In case that the Data Flash is read out by "Related instructions 2" immediately after the target memory is read out by "Related instructions 1", this is within the restriction; however, particular combinations of related instructions shown in Appendix2-2 are excepted.

Related instructions 1: Read instructions of RAM(Include general purpose register area), SFR,2nd SFR(No wait),ES,CS, PSW,SP

Note: Read instructions of 2nd SFR with wait, mirror area and Data Flash are not related.

	Operand
MOV	A, saddr
	A, sfr
	A, !addr16
	A, PSW
	A, ES
	A, CS
	A, [DE]
	A, [DE+byte]
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
	A, word[B]
	A, word[C]
	A, word[BC]
	A, [SP+byte]
MOVW	AX, saddrp
	AX, sfrp
	AX, !addr16
	AX, [DE]
	AX, [DE+byte]
	AX, [HL]
	AX, [HL+byte]
	AX, word[B]
	AX, word[C]
	AX, word[BC]
	AX, [SP+byte]
ADD	A, saddr
	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]

	Operand
ADDC	A, saddr
	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
SUB	A, saddr
	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
SUBC	A, saddr
	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
AND	A, saddr
	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
OR	A, saddr
	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]

	Operand
XOR	A, saddr
	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
CMP	A, saddr
	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
ADDW	AX, saddrp
	AX, !addr16
	AX, [HL+byte]
SUBW	AX, saddrp
	AX, !addr16
	AX, [HL+byte]
CMPW	AX, saddrp
	AX, !addr16
	AX, [HL+byte]
MOVW	AX, SP

	Operand
MOV	ES, saddr
	B, saddr
	B, !addr16
	C, saddr
	C, !addr16
	X, saddr
	X, !addr16
MOVW	BC, saddrp
	BC, !addr16
	DE, saddrp
	DE, !addr16
	HL, saddrp
	HL, !addr16
	BC, SP
	DE, SP
	HL, SP
CMP	saddr, #byte
	!addr16, #byte
CMP0	saddr
	!addr16
CMPS	X, [HL+byte]

	Operand
MOV1	CY, saddr.bit
	CY, sfr.bit
	CY, PSW.bit
	CY, [HL].bit
AND1	CY, saddr.bit
	CY, sfr.bit
	CY, PSW.bit
	CY, [HL].bit
OR1	CY, saddr.bit
	CY, sfr.bit
	CY, PSW.bit
	CY, [HL].bit
XOR1	CY, saddr.bit
	CY, sfr.bit
	CY, PSW.bit
	CY, [HL].bit
POP	rp

Related instructions 2: Read instructions of Data Flash

	Operand
MOV	A, !addr16
	A, [DE]
	A, [DE+byte]
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
	A, word[B]
	A, word[C]
	A, word[BC]

	Operand
ADD	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
ADDC	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
SUB	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
SUBC	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]

	Operand
AND	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
OR	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
XOR	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]
CMP	A, !addr16
	A, [HL]
	A, [HL+byte]
	A, [HL+B]
	A, [HL+C]

	Operand
MOV	B, !addr16
	C, !addr16
	X, !addr16
CMP	!addr16, #byte
CMP0	!addr16
CMPS	X, [HL+byte]

【Appendix2-2】

Safe combinations of related instructions1 and 2 <1>

Related instruction 1		Related instruction 2	
	Operand		Operand
MOVW	DE, saddrp	MOV	A, [DE]
	DE, !addr16		A, [DE+byte]
	DE, SP		
POP	DE		

Safe combinations of related instructions1 and 2 <2>

Related instruction 1		Related instruction 2					
	Operand		Operand		Operand		Operand
MOVW	HL, saddrp	MOV	A, [HL]	ADD	A, [HL]	AND	A, [HL]
	HL, !addr16		A, [HL+byte]		A, [HL+byte]		A, [HL+byte]
	HL, SP		A, [HL+B]		A, [HL+B]		A, [HL+B]
POP	HL		A, [HL+C]		A, [HL+C]		A, [HL+C]
				ADDC	A, [HL]	OR	A, [HL]
			Operand		A, [HL+byte]		A, [HL+byte]
		CMPS	X, [HL+byte]		A, [HL+B]		A, [HL+B]
					A, [HL+C]		A, [HL+C]
				SUB	A, [HL]	XOR	A, [HL]
					A, [HL+byte]		A, [HL+byte]
					A, [HL+B]		A, [HL+B]
					A, [HL+C]		A, [HL+C]
				SUBC	A, [HL]	CMP	A, [HL]
					A, [HL+byte]		A, [HL+byte]
					A, [HL+B]		A, [HL+B]
					A, [HL+C]		A, [HL+C]

Safe combinations of related instructions1 and 2 <3>

Related instruction 1		Related instruction 2					
	Operand		Operand		Operand		Operand
MOV	B, saddr	MOV	A, [HL+B]	ADD	A, [HL+B]	AND	A, [HL+B]
	B, !addr16		A, word[B]	ADDC	A, [HL+B]	OR	A, [HL+B]
MOVW	BC, saddrp			SUB	A, [HL+B]	XOR	A, [HL+B]
	BC, !addr16			SUBC	A, [HL+B]	CMP	A, [HL+B]
	BC, SP						
POP	BC						

Safe combinations of related instructions1 and 2 <4>

Related instruction 1		Related instruction 2					
	Operand		Operand		Operand		Operand
MOV	C, saddr	MOV	A, [HL+C]	ADD	A, [HL+C]	AND	A, [HL+C]
	C, !addr16		A, word[C]	ADDC	A, [HL+C]	OR	A, [HL+C]
MOVW	BC, saddrp			SUB	A, [HL+C]	XOR	A, [HL+C]
	BC, !addr16			SUBC	A, [HL+C]	CMP	A, [HL+C]
	BC, SP						
POP	BC						

Safe combinations of related instructions1 and 2 <5>

Related instruction 1		Related instruction 2	
	Operand		Operand
MOV	B, saddr	MOV	A, word[BC]
	B, !addr16		
	C, saddr		
	C, !addr16		
MOVW	BC, saddrp		
	BC, !addr16		
	BC, SP		
POP	BC		