

命令	ニーモニック		オペコード		アドレッシングモード(数値はステート数)									フラグ 変化	説明
	命令	オペランド	OP	Rd Rx	Drct	Index	Imm	FP Rlt	Reg	Imm4	InDr	B InDr	Othr		
No Operation	NO		00h	0h 0h	--	--	--	--	--	--	--	--	3	×	何もしない
Load	LD	Rd,EA	08h	Rd EA	7	7	5	5	4	4	5	5	--	×	Rd ← [EA]
Store	ST	Rd,EA	10h	Rd EA	7	7	--	5	--	--	5	5	--	×	[EA] ← Rd
Add	ADD	Rd,EA	18h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd + [EA]
Subtract	SUB	Rd,EA	20h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd - [EA]
Compare	CMP	Rd,EA	28h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd - [EA]
Logical And	AND	Rd,EA	30h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd and [EA]
Logical Or	OR	Rd,EA	38h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd or [EA]
Logical Xor	XOR	Rd,EA	40h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd xor [EA]
Add with Scale	ADDS	Rd,EA	48h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd + [EA]*2
Multiply	MUL	Rd,EA	50h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd × [EA]
Divide	DIV	Rd,EA	58h	Rd EA	23	23	21	21	20	20	21	21	--	○	Rd ← Rd / [EA]
Modulo	MOD	Rd,EA	60h	Rd EA	23	23	21	21	20	20	21	21	--	○	Rd ← Rd % [EA]
Shift Left Arithmetic	SHLA	Rd,EA	80h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd << [EA]
Shift Left Logical	SHLL	Rd,EA	88h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd <<< [EA]
Shift Right Arithmetic	SHRA	Rd,EA	90h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd >> [EA]
Shift Right Logical	SHRL	Rd,EA	98h	Rd EA	7	7	5	5	4	4	5	5	--	○	Rd ← Rd >>> [EA]
Jump on Zero	JZ	EA	A0h	0h EA	5	5	--	--	--	--	--	--	--	×	If (Z) PC ← EA
Jump on Carry	JC	EA	A0h	1h EA	5	5	--	--	--	--	--	--	--	×	If (C) PC ← EA
Jump on Minus	JM	EA	A0h	2h EA	5	5	--	--	--	--	--	--	--	×	If (S) PC ← EA
Jump on Overflow	JO	EA	A0h	3h EA	5	5	--	--	--	--	--	--	--	×	if (V) PC ← EA
Jump on greater than	JGT	EA	A0h	4h EA	5	5	--	--	--	--	--	--	--	×	If (not (Z or (S xor V))) PC ← EA
Jump on greater or equal	JGE	EA	A0h	5h EA	5	5	--	--	--	--	--	--	--	×	if (not (S xor V)) PC ← EA
Jump on less or equal	JLE	EA	A0h	6h EA	5	5	--	--	--	--	--	--	--	×	If (Z or (S xor V)) PC ← EA
Jump on less than	JLT	EA	A0h	7h EA	5	5	--	--	--	--	--	--	--	×	If (S xor V) PC ← EA
Jump on Non Zero	JNZ	EA	A0h	8h EA	5	5	--	--	--	--	--	--	--	×	If (not Z) PC ← EA
Jump on Non Carry	JNC	EA	A0h	9h EA	5	5	--	--	--	--	--	--	--	×	If (not C) PC ← EA
Jump on Non Minus	JNM	EA	A0h	Ah EA	5	5	--	--	--	--	--	--	--	×	If (not S) PC ← EA
Jump on Non Overflow	JNO	EA	A0h	Bh EA	5	5	--	--	--	--	--	--	--	×	If (not V) PC ← EA
Jump on higher	JHI	EA	A0h	Ch EA	5	5	--	--	--	--	--	--	--	×	If (not (Z or C)) PC ← EA
Jump on lower or same	JLS	EA	A0h	Eh EA	5	5	--	--	--	--	--	--	--	×	If (Z or C) PC ← EA
Jump	JMP	EA	A0h	Fh EA	5	5	--	--	--	--	--	--	--	×	PC ← EA
Call subroutine	CALL	EA	A8h	0h EA	7	7	--	--	--	--	--	--	--	×	[--SP] ← PC, PC ← EA
Input	IN	Rd,EA	B0h	Rd EA	6	--	--	--	--	--	4	4	--	×	Rd ← IO[EA]
Output	OUT	Rd,EA	B8h	Rd EA	5	--	--	--	--	--	3	3	--	×	IO[EA] ← Rd
Push Register	PUSH	Rd	C0h	Rd 0h	--	--	--	--	--	--	--	--	5	×	[--SP] ← Rd
Pop Register	POP	Rd	C4h	Rd 0h	--	--	--	--	--	--	--	--	5	×	Rd ← [SP++]
Return from Subroutine	RET		D0h	0h 0h	--	--	--	--	--	--	--	--	6	×	PC ← [SP++]
Return from Interrupt	RETI		D4h	Fh 0h	--	--	--	--	--	--	--	--	9	×	FLAG ← [SP++], PC ← [SP++]
Supervisor Call	SVC		F0h	0h 0h	--	--	--	--	--	--	--	--	14	×	システムコール
Halt	HALT		FFh	0h 0h	--	--	--	--	--	--	--	--	3	×	CPU停止

アドレッシングモード(上の表中EAの詳細)に付いて

アドレッシングモード	略記	ニーモニック (EA部分の標記方法)	命令フォーマット		EA(実効アドレス)の決め方	
			第1ワード	第2ワード	略記	解説
Direct	Drct	OP Rd, <u>Dsp</u>	OP+0 Rd0h	Dsp	[Dsp]	Dsp番地
Indexed	Index	OP Rd, <u>Dsp, Rx</u>	OP+1 RdRx	Dsp	[Dsp+Rx]	(Dsp+Rxレジスタの内容)番地
Immediate	Imm	OP Rd, <u>#Imm</u>	OP+2 Rd0h	Imm	Imm	Immそのもの
FP Relative	FP Rlt	OP Rd, <u>Dsp4, FP</u>	OP+3 RdD4	--	[Dsp4+FP]	(D4を符号拡張した値×2 + FPレジスタの内容)番地(D4=Dsp4/2)
Register	Reg	OP Rd, <u>Rs</u>	OP+4 RdRs	--	Rs	Rsレジスタの内容
4bit Signed Immediate	Imm4	OP Rd, <u>#Imm4</u>	OP+5 RdI4	--	Imm4	I4を符号拡張した値そのもの
Register Indirect	InDr	OP Rd, <u>@Rx</u>	OP+6 RdRx	--	[Rx]	Rxレジスタの内容番地
Byte Register Indirect	B InDr	OP Rd, <u>@Rx</u>	OP+7 RdRx	--	[Rx]	Rxレジスタの内容番地(但し番地の内容は8bitデータ)
Other	Othr	OP Rd	OP Rd0h	--	--	なし
		OP	OP 0h0h	--	--	なし

注4

※アセンブリ言語でDspとDsp4、ImmとImm4の標記は同じ(値によりアセンブラが自動判定)

※FP相対で、Dsp4は-16～+14の偶数

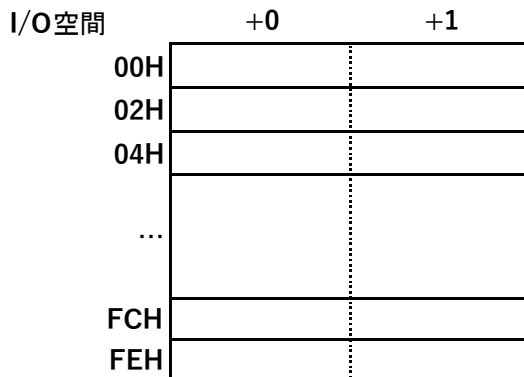
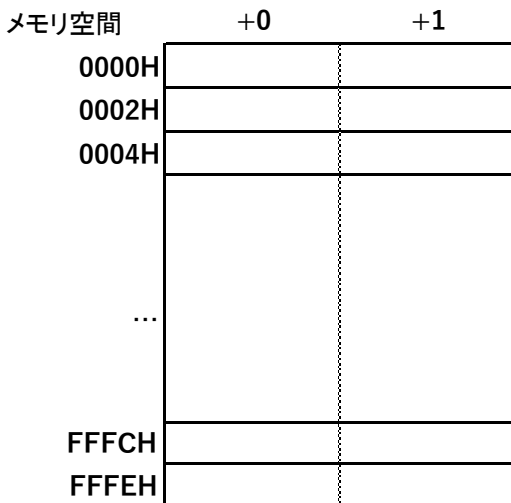
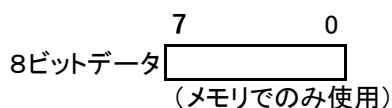
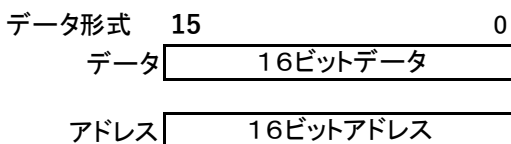
色付きのセルは特権命令 特権違反が発生時は、スタックに違反を起こす前のPCが保存される

注1: RETI命令は特権モードでのみEPIフラグを変化させる

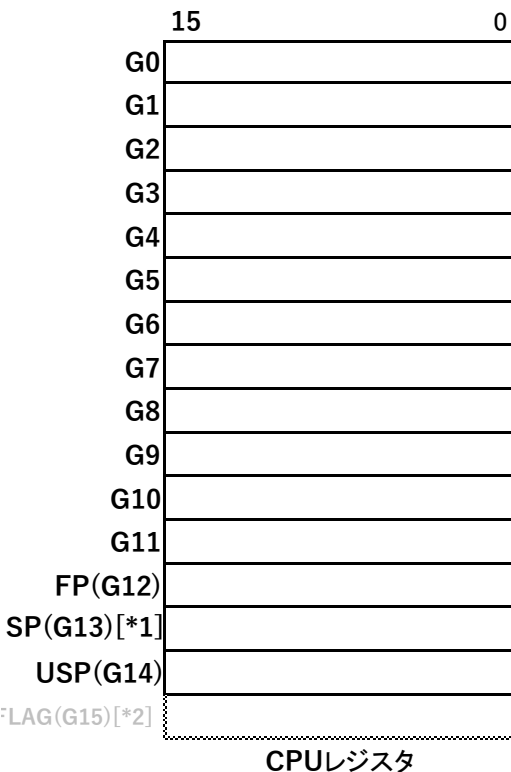
注2: D4はDsp4(4bitディスポースメント)の1/2の値

注3: I4はImm4 (4bit即値)のこと

注4: アドレッシングモードによりOPの値が変化する



## レジスタ構成



\*1:SPはカーネルモードではSSP, ユーザモードではUSP

\*2:FLAGはCPUレジスタ(G15)として扱うこともできる

\*3:Uフラグは単なる1ビットのレジスタ

\*4:VフラグはADD, SUB, CMPで有効

\*5:CフラグはADD, SUB, CMPで有効, また, SHXXでは1ビットシフトのときだけ有効

ダイレクト(\*0)

OP	Rd	0H	Dsp
----	----	----	-----

ショートイミディエイト(\*5)

OP	Rd	Imm4
----	----	------

インデクスト(\*1)

OP	Rd	Rx	Dsp
----	----	----	-----

レジスタインダイレクト(\*6)

OP	Rd	Rx
----	----	----

イミディエイト(\*2)

OP	Rd	0H	Imm
----	----	----	-----

バイト・レジスタインダイレクト(\*7)

OP	Rd	Rx
----	----	----

FP相対(\*3)

OP	Rd	Dsp4
----	----	------

レジスタ(\*8)

OP	Rd	0H
----	----	----

レジスタレジスタ(\*4)

OP	Rd	Rs
----	----	----

オペランドなし(\*9)

OP	00H
----	-----

		OP 下位3ビット							
		0	1	2	3	4	5	6	7
OP 上位5ビット	00000	NO(*9)							
	00001	LD(*0)	LD(*1)	LD(*2)	LD(*3)	LD(*4)	LD(*5)	LD(*6)	LD(*7)
	00010	ST(*0)	ST(*1)		ST(*3)			ST(*6)	ST(*7)
	00011	ADD(*0)	ADD(*1)	ADD(*2)	ADD(*3)	ADD(*4)	ADD(*5)	ADD(*6)	ADD(*7)
	00100	SUB(*0)	SUB(*1)	SUB(*2)	SUB(*3)	SUB(*4)	SUB(*5)	SUB(*6)	SUB(*7)
	00101	CMP(*0)	CMP(*1)	CMP(*2)	CMP(*3)	CMP(*4)	CMP(*5)	CMP(*6)	CMP(*7)
	00110	AND(*0)	AND(*1)	AND(*2)	AND(*3)	AND(*4)	AND(*5)	AND(*6)	AND(*7)
	00111	OR(*0)	OR(*1)	OR(*2)	OR(*3)	OR(*4)	OR(*5)	OR(*6)	OR(*7)
	01000	XOR(*0)	XOR(*1)	XOR(*2)	XOR(*3)	XOR(*4)	XOR(*5)	XOR(*6)	XOR(*7)
	01001	ADDS(*0)	ADDS(*1)	ADDS(*2)	ADDS(*3)	ADDS(*4)	ADDS(*5)	ADDS(*6)	ADDS(*7)
	01010	MUL(*0)	MUL(*1)	MUL(*2)	MUL(*3)	MUL(*4)	MUL(*5)	MUL(*6)	MUL(*7)
	01011	DIV(*0)	DIV(*1)	DIV(*2)	DIV(*3)	DIV(*4)	DIV(*5)	DIV(*6)	DIV(*7)
	01100	MOD(*0)	MOD(*1)	MOD(*2)	MOD(*3)	MOD(*4)	MOD(*5)	MOD(*6)	MOD(*7)
	01101								
	01110								
	01111								
	10000	SHLA(*0)	SHLA(*1)	SHLA(*2)	SHLA(*3)	SHLA(*4)	SHLA(*5)	SHLA(*6)	SHLA(*7)
	10001	SHLL(*0)	SHLL(*1)	SHLL(*2)	SHLL(*3)	SHLL(*4)	SHLL(*5)	SHLL(*6)	SHLL(*7)
	10010	SHRA(*0)	SHRA(*1)	SHRA(*2)	SHRA(*3)	SHRA(*4)	SHRA(*5)	SHRA(*6)	SHRA(*7)
	10011	SHRL(*0)	SHRL(*1)	SHRL(*2)	SHRL(*3)	SHRL(*4)	SHRL(*5)	SHRL(*6)	SHRL(*7)
	10100	JMP(*0)	JMP(*1)						
	10101	CALL(*0)	CALL(*1)						
	10110	IN(*0)						IN(*6)	
	10111	OUT(*0)						OUT(*6)	
	11000	PUSH(*8)				POP(*8)			
	11001								
	11010	RET(*9)				RETI(*8)※			
	11011								
	11100								
	11101								
	11110	SVC(*9)							
	11111								HALT(*9)

特権命令

※：RETIのRdはFLAGを表すFh

	>	>=	=	!=	<=	<
符号あり	JGT	JGE	JZ	JNZ	JLE	JLT
符号無し	JHI	JNC	JZ	JNZ	JLS	JC

FLAGのビット割り  
(00000000EPIUVCSZ)

Rd/Rs/Rx	
値	意味
0	G0
1	G1
2	G2
3	G3
4	G4
5	G5
6	G6
7	G7
8	G8
9	G9
A	G10
B	G11
C	G12(FP)
D	SP(SSP/USP)
E	USP
F	FLAG

SPの意味はPフラグで変化  
(P=1:SSP、P=0:USP)

JMP命令のRd	
値	意味
0	JZ
1	JC
2	JM
3	JO
4	JGT
5	JGE
6	JLE
7	JLT
8	JNZ
9	JNC
A	JNM
B	JNO
C	JHI
D	
E	JLS
F	JMP

メモリマップ			I/Oマップ					
+0番地		+1番地	+0番地		+1番地			
0000h	RAM(56KiB)	RAM	00h	Timer0(In:現在値/Out:周期)		タイマー		
0002h			02h	Timer0(In:フラグ/Out:コントロール)				
0004h			04h	Timer1(In:現在値/Out:周期)				
			06h	Timer1(In:フラグ/Out:コントロール)				
			08h	00H	FT232RL-Data		FT232RL	
			0Ah	00H	FT232RL-Stat/Ctrl			
			0Ch	00H	TeC-Data		TeC	
			0Eh	00H	TeC-Stat/Ctrl			
			10h	00H	uSD-Stat/Ctrl		マイクロSD	
			12h	uSD-MemAddr				
			14h	uSD-BlkAddrH				
			16h	uSD-BlkAddrL				
			18h	00H	I/Oポート(In/Out)		入出力ポート	
			1Ah	00H	ADC参照電圧(Out)			
			1Ch	00H	出力ポートHi(Out)			
			1Eh	00H	モード(In)			
			20h	00H	SPI-Data(In/Out)			
			22h	00H	SPI-Stat/Sclk			
			24h	00H	PIO-Mask			
			26h	00H	PIO-Xor			
			28h	00H	RN4020-Data		RN4020	
			2Ah	00H	RN4020-Stat/Ctrl			
			2Ch	00H	RN4020-Cmd			
			2Eh	00H	RN4020-RAM			
			30h	00H	TeC(In:DLed)			
			32h	00H	TeC(Out:DSw)			
			34h	00H	Tec(Out:Fnc)			
			36h	00H	TeC(Ctl)			
DFFEh			38h	00H	00H			
E000h	RAM(8160B)	RAM	...	...				
...								
FFDEh								
FFE0h			Timer0		80h	TLB[0]上位8bit		MMU
FFE2h			Timer1		82h	TLB[0]下位16bit		
FFE4h			RN4020 受信		84h	TLB[1]上位8bit		
FFE6h			RN4020 送信		86h	TLB[1]下位16bit		
FFE8h			FT232RL 受信		...	...		
FFEAh			FT232RL 送信		9Ch	TLB[7]上位8bit		
FFEC			TeC 受信		9Eh	TLB[7]下位16bit		
FFEEh			TeC 送信		A0h	b0=IPL切離し(OUT)		
FFF0h	マイクロSD		A2h	b0=MMU有効(OUT)/違反アドレス(IN)				
FFF2h	PIO		A4h	b1=badAddr,b0=memVio(IN)				
FFF4h	TLB miss (※1)		A6h	ページ番号(IN)				
FFF6h	メモリ保護違反 (※1)		...	空き				
FFF8h	ゼロ除算 (※1)	割り込みベクタ	F8h	データレジスタ(Out)/データSW(IN)		コンソール		
FFFAh	特権違反 (※1)		FAh	アドレスレジスタ (IN)				
FFFCh	未定義命令 (※1)		FCh	00H	ロータリーSW(IN)			
FFFEh	SVC (※1)		FEh	00H	機能レジスタ(IN)			

※ 1：例外（割込み禁止の影響を受けない）

出力ポートHi（M000 VVVV）  
M（0：入力，1：出力），VVVV（I7～I4に出力）  
RN4020-RAM：リセットの影響を受けない8bitレジスタ

I/Oポート詳細			
番地	I/Oポート	ビット	意味
	*-Ctrl(OUT)	TR00 0000	T=Enable Transmitter Interrupt, R=Enable Reciver Interrupt
	*-Stat(IN)	TR00 0000	T=Transmitter Ready, R=Reciver Ready
02h	Timer0 コントロール	I000 ... 000S	I=Enable Interrupt, S=Start
06h	Timer1 コントロール	I000 ... 000S	I=Enable Interrupt, S=Start
11h	uSD-Ctrl	E000 0IRW	E=INT_ENA, I=INIT, R=READ, W=WRITE
11h	uSD-Stat	IE00 000C	I=IDLE, E=ERROR,C=Card Detection(Active=0)
1Fh	モード	0000 0MMM	MMM：000=TeC,001=TaC,010=DEMO1,011=DEMO2,111=RN4020FactoryReset
2Dh	RN4020-Cmd	0000 FHCS	RN4020(F=Flow Control, H=Hw Pin, C=Cmd Pin, S=Sww Pin (初期値=0001))
30h-	TeCコンソール	-	詳細は「I/Oマップ詳細」シートに掲載
D0h-	MMU	-	詳細は「I/Oマップ詳細」シートに掲載
FDh	ロータリーSW(IN)	000S SSSS	0=G0,1=G1,...11=G11,12=FP,13=SP,14=PC,15=FLAG,16=MD,17=MA
FFh	機能レジスタ(IN)	0000 FFFF	0=ReadReg, 1=WriteReg, 13=ReadMem, 14=WriteMem

コンソール制御のI/Oポート解説

TeCコンソールI/Oアドレス			
		Read	Write
データLED	(30h)	データランプ	空き
データSW	(32h)	00H	データスイッチ
機能SW	(34h)	00H	ABCD EFGH
制御と機能SW	(36h)	---- --RS	I--- -JKL

TeCコンソールの操作ビット					
A	BREAK-SW	B	STEP-SW	C	RUN-SW
D	STOP-SW	E	SETA-SW	F	INCA-SW
G	DECA-SW	H	WRITE-SW	I	ENABLE
J	RESET-SW	K	LEFT-SW	L	RIGHT-SW

TeCコンソールの状態確認ビット					
R	RESET	S	SETA-SW		

80hから9Fhに配置されるTLBエントリー解説

TLBエントリーの構成								
上位 8 ビット(偶数アドレス)			下位 1 6 ビット(奇数アドレス)					
23-16	15	14	13	12	11	10-8	7-0	
PAGE	V	*	*	R	D	R/W/X	FRAME	

PAGE:ページ番号 V:Valid    \*:未定義    R:Reference    R/W/X:Read/Write/eXecute

D:Dirty    FRAME:フレーム番号