

ニーモニック	命令	第 1 バイト		第 2 バイト	フラグ変化	ステート数※	動作
		OP	GRXR				
NO	No Opration	0000	00 00	————	×	3	何もしない
LD	Load	0001	GR XR	aaaa aaaa	×	5/7/7	GR <- [EA]
ST	Store	0010	GR XR	aaaa aaaa	×	-/7/7	[EA] <- GR
ADD	Add	0011	GR XR	aaaa aaaa	○	5/7/7	GR <- GR + [EA]
SUB	Subtract	0100	GR XR	aaaa aaaa	○	5/7/7	GR <- GR - [EA]
CMP	Compare	0101	GR XR	aaaa aaaa	○	5/7/7	GR - [EA]
AND	Logical And	0110	GR XR	aaaa aaaa	○	5/7/7	GR <- GR & [EA]
OR	Logical Or	0111	GR XR	aaaa aaaa	○	5/7/7	GR <- GR [EA]
XOR	Logical Xor	1000	GR XR	aaaa aaaa	○	5/7/7	GR <- GR ^ [EA]
SHLA	Shift Left Arithmetic	1001	GR 00	————	○	4	GR <- GR << 1
SHLL	Shift Left Logical	1001	GR 01	————	○	4	GR <- GR << 1
SHRA	Shift Right Arithmetic	1001	GR 10	————	○	4	GR <- GR >> 1
SHRL	Shift Right Logical	1001	GR 11	————	○	4	GR <- GR >>> 1
JMP	Jump	1010	00 XR	aaaa aaaa	×	-/5/6	PC <- EA
JZ	Jump on Zero	1010	01 XR	aaaa aaaa	×	4/5/6	if Zero PC <- EA
JC	Jump on Carry	1010	10 XR	aaaa aaaa	×	4/5/6	if Carry PC <- EA
JM	Jump on Minus	1010	11 XR	aaaa aaaa	×	4/5/6	if Sign PC <- EA
CALL	Call subroutine	1011	11 XR	aaaa aaaa	×	-/6/7	[--SP]<-PC, PC<-EA
IN	Input	1100	GR 00	0000 pppp	×	8	GR <- IO[P]
OUT	Output	1100	GR 11	0000 pppp	×	8	IO[P] <- GR
PUSH	Push Register	1101	GR 00	————	×	6	[--SP] <- GR
PUSHF	Push Flag	1101	11 01	————	×	6	[--SP] <- FLAG
POP	Pop Register	1101	GR 10	————	×	6	GR <- [SP++]
POPF	Pop Flag	1101	11 11	————	○	6	FLAG <- [SP++]
EI	Enable Interrupt	1110	00 00	————	×	4	割り込み許可
DI	Disable Interrupt	1110	00 11	————	×	4	割り込み禁止
RET	Return from subroutine	1110	11 00	————	×	6	PC <- [SP++]
RETI	Return from Interrupt	1110	11 11	————	×	6	PC <- [SP++], EI
HALT	Halt	1111	11 11	————	×	4	停止

GR	意味	メモリマップ		INT1 : SIO 受信割り込み INT2 : SIO 送信割り込み	
00	G0	Addr	内容	I/Oマップ	
01	G1	00 DB	RAM	Addr	Read/Write
10	G2			0	Data-Sw/b0:Beep
11	SP	DC	INT0 割り込みベクタ	1	Data-Sw/b0:Speaker
XR	意味	DD	INT1 割り込みベクタ	2	SIO-Data/SIO-Data
00	ダイレクトモード	DE	INT2 割り込みベクタ	3	b7:Tx Ready / b7:Tx STI b6:Rx Ready / b6:Rx STI
01	G1インデクスドモード	DF	INT3 割り込みベクタ	4	空き/空き
10	G2インデクスドモード	E0 FF	ROM(IPL)
11	イミディエイトモード			F	空き/空き