

1802 Membership Card Summary

©2019 by TMSI, 814 8th Ave N, Sartell MN 56377 <leeahart@earthlink.net>

Register Summary

D	8 bit	D register (Accumulator)	DF	1 bit	Data flag (ALU carry)
Rn	16 bit	1 of 16 registers (n=0-F)	P	4 bit	Program register select
X	4 bit	Data register select	T	8 bit	Saved X,P after interrupt
IE	1 bit	Interrupt Enable	Q	1 bit	Output flip-flop

Interrupt: Finish executing current instruction, then save X and P in T, then set P=1, X=2 and IE=0 (Inhibit). Execution continues at M(R1).
DMA: Finish executing current instruction, then use R0 as memory address for data transfer. DMA-IN writes data from bus to memory, or DMA-OUT reads data from memory to Bus. Then increment R0. If simultaneous, DMA-IN has highest priority, then DMA-OUT, then Interrupt.
EF1-4: External 1-bit flags that can be tested by Branch instructions.
Q: External 1-bit output that can be set, reset, and tested by Branch and Skip instructions.

<http://www.sunrise-ev.com/1802.htm>

Register Operations

1n .. INC Rn	Rn=Rn+1	0n .. LDN Rn	D=M(Rn) for n not 0
2n .. DEC Rn	Rn=Rn-1	4n .. LDA Rn	D=M(Rn), then inc Rn
6n .. IRX Rx	Rx=Rx+1	F0 .. LDX	D=M(Rx)
8n .. GLO Rn	D=Rn(low8)	72 .. LDXA	D=M(Rx), then inc Rx
9n .. GHI Rn	D=Rn(high8)	F8 nn .. LDI nn	D=M(Rp)
An .. PLO Rn	Rn(low8)=D	5n .. STR Rn	M(Rn)=D
Bn .. PHI Rn	Rn(high8)=D	73 .. STXD	M(Rx)=D, then dec Rx

Memory Load / Store

38 nn .. NBR nn	no branch (skip nn)
39 nn .. BNQ nn	branch if Q=0
3A nn .. BNZ nn	branch if D not 0
3B nn .. BNF nn	branch if DF=0, or BM nn branch if minus, or BL nn branch if less than
3C nn .. BN1 nn	branch if EF1=0
3D nn .. BN2 nn	branch if EF2=0
3E nn .. BN3 nn	branch if EF3=0
3F nn .. BN4 nn	branch if EF4=0

Short Branch Instructions (nn is low byte of address on same page)

30 nn .. BR nn	branch to nn
31 nn .. BQ nn	branch if Q=1
32 nn .. BZ nn	branch if D=0
33 nn .. BDF nn	branch if DF=1, or BPZ nn br. if positive or 0, or BGE nn br. if equal or greater
34 nn .. B1 nn	branch if EF1=1
35 nn .. B2 nn	branch if EF2=1
36 nn .. B3 nn	branch if EF3=1
37 nn .. B4 nn	branch if EF4=1

name dots=bus cycles opcode 2nd byte description

Long Branch & Skip (hhll=address high byte, low byte)

C0 hhll .. LBR hhll	branch to hhll	C8 hhll .. NLBR hhll	no br. (skip hhll)
C1 hhll .. LBQ hhll	branch if Q=1	C9 hhll .. LBNQ hhll	branch if Q=0
C2 hhll .. LBZ hhll	branch if D=0	CA hhll .. LBNZ hhll	branch if D not 0
C3 hhll .. LBDF hhll	branch if DF=1	CB hhll .. LBNF hhll	branch if DF=0
C4 .. NOP	no operation	CC .. LSIE	skip 2 if IE=1
C5 .. LSNQ	skip 2 if Q=0	CD .. LSQ	skip 2 if Q=1
C6 .. LSNZ	skip 2 if Dnot0	CE .. LSZ	skip 2 if D=0
C7 .. LSNF	skip 2 if DF=0	CF .. LSDF	skip 2 if DF=1

Logic, Arithmetic, and Shift (result in D, carry/borrow/shift in DF)

F1 .. OR	D=D or M(Rx)	F9 nn .. ORI nn	D=D or nn
F2 .. AND	D=D and M(Rx)	FA nn .. ANI nn	D=D and nn
F3 .. XOR	D=D xor M(Rx)	FB nn .. XRI nn	D=D xor nn
F4 .. ADD	D=D+M(Rx)	FC nn .. ADI nn	D=D+nn
F7 .. ADC	D=D+M(Rx)+DF	7C nn .. ADCI nn	D=D+nn+DF
F5 .. SD	D=M(Rx)-D	FD nn .. SDI nn	D=D-nn-D
F6 .. SDB	D=M(Rx)-D-not DF	7D nn .. SDBI nn	D=D-nn-D-not DF
F7 .. SM	D=D-M(Rx)	FF nn .. SMI nn	D=D-nn
F7 .. SMB	D=D-M(Rx)-not DF	7F nn .. SMBI nn	D=D-nn-not DF

only add, subtract, and shift instructions change DF

Logic, Arithmetic, and Shift Instructions (continued)

F6 .. SHR	shift D right;	76 .. SHRC	shift D right with carry;
FE .. SHL	shift D left;	7E .. SHLC	shift D left with carry;
	D(lsb)=0, DF=D(msb)		D(msb)=DF, DF=D(lsb)
			or RSHR D(lsb)=DF, DF=D(msb)

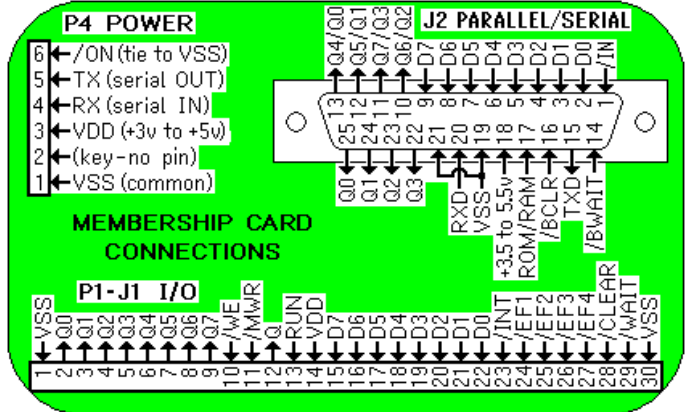
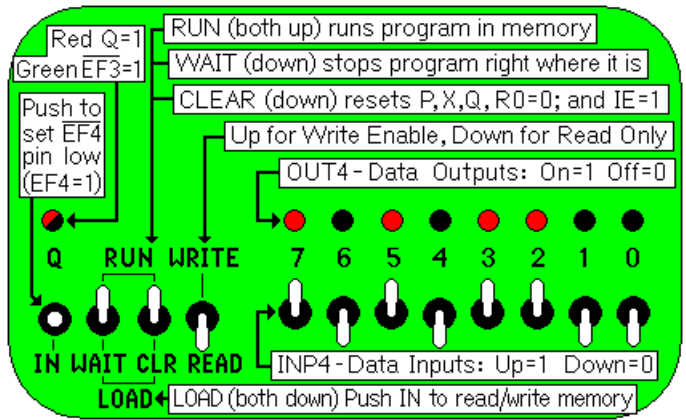
Control Instructions

00 .. IDL	wait for INT or DMA	78 .. SAV	M(Rx)=T
68 ..	reserved for 1804/5/6	7A .. REQ	reset Q=0
70 .. RET	X,P=M(Rx); inc Rx; IE=1	7B .. SEQ	set Q=1
71 .. DIS	X,P=M(Rx); inc Rx; IE=0	Dn .. SEP n	set P=n
79 .. MARK	T=X,P; M(R2)=X,P; X=P; dec R2	En .. SEX n	set X=n

Input/Output Instructions (with N output on pins N0, N1, and N2)

61 .. OUT1	bus=M(Rx), N=1; inc Rx	69 .. INP1	M(Rx)=D=bus, N=1
62 .. OUT2	bus=M(Rx), N=2; inc Rx	6A .. INP2	M(Rx)=D=bus, N=2
63 .. OUT3	bus=M(Rx), N=3; inc Rx	6B .. INP3	M(Rx)=D=bus, N=3
64 .. OUT4	bus=M(Rx), N=4; inc Rx	6C .. INP4	M(Rx)=D=bus, N=4
65 .. OUT5	bus=M(Rx), N=5; inc Rx	6D .. INP5	M(Rx)=D=bus, N=5
66 .. OUT6	bus=M(Rx), N=6; inc Rx	6E .. INP6	M(Rx)=D=bus, N=6
67 .. OUT7	bus=M(Rx), N=7; inc Rx	6F .. INP7	M(Rx)=D=bus, N=7

Add: DF=1 if carry. Subtract: DF=0 if borrow (negative).



Hex	Binary	Dec	MSD							
			0x	1x	2x	3x	4x	5x	6x	7x
0	0000	0	x0 0000	NUL DLE (sp)	0	@	P			
1	0001	1	x1 0001	SOH DC1	!	1	A	Q	a	q
2	0010	2	x2 0010	STX DC2	"	2	B	R	b	r
3	0011	3	x3 0011	ETX DC3	#	3	C	S	c	s
4	0100	4	x4 0100	EOT DC4	\$	4	D	T	d	t
5	0101	5	x5 0101	ENQ NAK	%	5	E	U	e	u
6	0110	6	x6 0110	ACK SYN	&	6	F	V	f	v
7	0111	7	x7 0111	BEL ETB	'	7	G	W	g	w
8	1000	8	x8 1000	BS CAN	(8	H	X	h	x
9	1001	9	x9 1001	HT EM)	9	I	Y	i	y
A	1010	10	xA 1010	LF SUB	*		J	Z	j	z
B	1011	11	xB 1011	VT ESC	+		K	[k	{
C	1100	12	xC 1100	FF FS	<		L	\	l	
D	1101	13	xD 1101	CR GS	=		M]	m	}
E	1110	14	xE 1110	SO RS	>		N	^	n	~
F	1111	15	xF 1111	SI US	/		O	_	o	DEL

