CDP 1802	N															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
I 0	IDL	D = M(R[N])														
I 1	R[N]++															
I 2	R[N]--															
I 3	ALWAYS low(R[P]) = M(R[P])	if Q low(R[P]) = M(R[P])	if D==0 low(R[P]) = M(R[P])	if DF low(R[P]) = M(R[P])	if EF1 low(R[P]) = M(R[P])	if EF2 low(R[P]) = M(R[P])	if EF3 low(R[P]) = M(R[P])	if EF4 low(R[P]) = M(R[P])	R[P]++	if !Q low(R[P]) = M(R[P])	if D!=0 low(R[P]) = M(R[P])	if !DF low(R[P]) = M(R[P])	if !EF1 low(R[P]) = M(R[P])	if !EF2 low(R[P]) = M(R[P])	if !EF3 low(R[P]) = M(R[P])	if !EF4 low(R[P]) = M(R[P])
I 4	D = M(R[N]) R[N]++															
I 5	M(R[N]) = D															
I 6	R[X]++	OUT(M(R[X]), N) R[X]++							EXT	M(R[X]) = D = INPUT(N)						
I 7	(X,P)=M(R[X]) R[X]++ IE=1	IE=0	D= M(R[X])	M(R[X]) =D R[X]++	ADC	SDB	SHRC	SMB	M(R[X]) =T	MARK	Q=0	Q=1	ADCI	SDBI	SHLC	SMBI
I 8	D = low(R[N])															
I 9	D = high(R[N])															
I A	low(R[N]) = D															
I B	high(R[N]) = D															
I C	ALWAYS high(R[P]) = M(R[P]) low(R[P]) = M(R[P]+1)	if Q high(R[P]) = M(R[P]) low(R[P]) = M(R[P]+1)	if D==0 high(R[P]) = M(R[P]) low(R[P]) = M(R[P]+1)	if DF high(R[P]) = M(R[P]) low(R[P]) = M(R[P]+1)	NOP	if !Q R[P]+=2	if D!=0 R[P]+=2	if !DF R[P]+=2	ALWAYS R[P]+=2	if !Q R[P]+=2	if D!=0 R[P]+=2	if !DF R[P]+=2	if !IE R[P]+=2	if Q R[P]+=2	if D==0 R[P]+=2	if DF R[P]+=2
I D	P = N															
I E	X = N															
I F	D= M(R[X])	D= M(R[X]) | D	D= M(R[X]) & D	D= M(R[X]) ^ D	ADD	SD	SHR	SM	LDI	ORI	ANI	XRI	ADI	SDI	SHL	SMI

ADC	D,DF = M(R[X])+D+DF
ADCI	D,DF = M(R[P])+D+DF; R[P]++
ADD	D,DF = M(R[X])+D
ADI	D,DF = M(R[P])+D; R[P]++
ANI	M(R[P]) & D ->D; R[P]++
IDL	
LDI	D = M(R[P]); R[P]++
MARK	T=(X,P); M(R(2))=T; X=P; R[2]--
ORI	D = M(R[P]) | D; R[P]++
SD	D,DF = M(R[X])-D
SDB	D,DF = FM(R[X])-D-(NOT DF)

SDBI	D,DF = M(R[P])-D-(Not DF); R[P]++
SDI	D,DF = M(R[P])-D; R[P]++
SHL	0 => (D<<1) =>
SHLC	DF <= (D<<1) <= DF
SMI	D,DF = D - M(R[P]); R[P]++
SHR	0 => (D>>1) =>
SHRC	DF => (D>>1) => DF
SM	D,DF = D-M(R[X])
SMB	D,DF = D-M(R[X])-(NOT DF)
SMBI	D,DF = D-M(R[P])-(NOT DF); R[P]++
XRI	D = M(R[P]) ^ D; R[P]++

R[]	16 Bits 1 of 16 Scratchpad Registers
D	8 Bits Data Register (Accumulator)
T	8 Bits Holds X and P during interrupt, X is high nybble, not directly accessable
P	4 Bits Designates which register is Program Counter
X	4 Bits Designates which register is Data Pointer

I	4 Bits Holds High-Order Instruction Digit, not directly accessable
N	4 Bits Low nybble of instruction byte, not directly accessable
DF	1-Bit Data Flag (ALU Carry/borrow)
IE	1-Bit Interrupt Enable
Q	1-Bit Output Flip/Flop

EF1 to EF4	(4 Flags) These inputs enable the I/O controllers to transfer status information to the processor. The levels can be tested by the conditional branch instructions.