0 IDL D = M(R[N])  1 R[N]++  2 R[N]  3 ALWAYS if Q if D==0 if DF if EF1 if EF2 if EF3 if EF4   If	.802		N														
R[N]++		0	1	2	3	4	5	6	7	8	9	Α	В	С	D	Е	F
R[N]	0	IDL							D =	M(R[	N])						
3 ALMAYS & LEQ & LED DOWN (R[P]) = M(R[P])  1 OW (R[P]) = M(R[P])  2 D = M(R[N]) R[N] ++  5 M(R[N]) = D  6 R[X] ++  7 (X, P) = M(R[X])   D = M(R[X])   D = M(R[X])   D = INPUT (N)  R[X] ++  7 (X, P) = M(R[X])   D = M(R[X])   D = M(R[X])   D = INPUT (N)  8 D = Low (R[N])  9 D = high (R[N])  A Low (R[N]) = D  C ALMAYS & LEQ & LED DOWN (R[X])   NOP	1		R[N]++														
3 ALMAYS & LEQ & LED DOWN (R[P]) = M(R[P])  1 OW (R[P]) = M(R[P])  2 D = M(R[N]) R[N] ++  5 M(R[N]) = D  6 R[X] ++  7 (X, P) = M(R[X])   D = M(R[X])   D = M(R[X])   D = INPUT (N)  R[X] ++  7 (X, P) = M(R[X])   D = M(R[X])   D = M(R[X])   D = INPUT (N)  8 D = Low (R[N])  9 D = high (R[N])  A Low (R[N]) = D  C ALMAYS & LEQ & LED DOWN (R[X])   NOP	2		R[N]														
D = M(R[N]) = D  D = M(R[N])   D		ALWAYS if Q if D==0 if DF if EF1 if EF2 if EF3 if EF4 if !Q if D!=0 if !DF if !EF1 if !EF2 if !EF3 if !E															if !EF
R[N]++	3			low(F	R[P])	= M(	R[P])		_ v			101	w(R[P	)]) =	M(R[	P])	
OUT (M(R[X]), N)	4																
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	5		M(R[N]) = D														
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	6	R[X]++	R[X]++							EXT	M(R[X]) = D = INPUT(N)						
D = low(R[N])  D = high(R[N])  E	7				M(R[X])			SHRC	SMB		MARK	Q=0	Q=1	ADCI	SDBI	SHLC	SMBI
D = high(R[N])  D = D  D = D  R[P]+=2  D P = N  E  X = N  F   D = D = D = D = D = D = D = D = D = D			IE=0	R[X	<]++			מ	= 10	v (R [N	1)						
A																	
B	9		D = high(R[N])														
C   Always   if Q   if D=0   if DF   NOP   if !Q   if D!=0   if !DF   Always   if !Q   if D!=0   if !DF   if !IE   if Q   if D==0   if DE   high(R[P]) = M(R[P])   NOP   R[P]+=2   high(R[P]) = M(R[P])   NOP   R[P]+=2   NOP   NOP   R[P]+=2   NOP   NOP   NOP   R[P]+=2   NOP   NOP	Α		low(R[N]) = D														
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	В		high(R[N]) = D														
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	С	hi	$high(R[P]) = M(R[P]) \qquad NOP \qquad R[D]$								high(R[P]) = M(R[P]) $R[D] += 2$						
F D= D= M(R[X]) M(R[X]) M(R[X]) M(R[X]) M(R[X]) ADD SD SHR SM LDI ORI ANI XRI ADI SDI SHL SM:  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  SHL O => (D<<1) =>	D		low(R[P]) = M(R[P]+1) $R[P]+=2$ $low(R[P]) = M(R[P]+1)$ $R[P]+=2$														
F D= D= M(R[X]) M(R[X]) M(R[X]) M(R[X]) M(R[X]) ADD SD SHR SM LDI ORI ANI XRI ADI SDI SHL SM:  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  SHL O => (D<<1) =>	E																
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  SHL 0 => (D<<1) =>		D=				ADD	CD.	CHD	CM	TDT	ODT	ANIT	VDT	ADT	CDT	СИТ	смт
ADCI D, DF = M(R[P]) +D+DF; R[P]++  ADD D, DF = M(R[X]) +D  SDI D, DF = M(R[P]) -D; R[P]++  SHL 0 => (D<<1) =>	<u> </u>	M(R[X])				ADD	מפ	SHK	SM	пот	ORI	ANI	AKI	ADI	SDI	SUL	SMI
ADD D, DF = M(R[X])+D SHL 0 => (D<<1) =>										SDBI	D,DF	= M(R[	P])-D-(	Not DF)	; R[P]+	+	_
		ADCI								SDI	D,DF	= M(R[	P])-D;	R[P]++			
ADI D, DF = M(R[P])+D; R[P]++ SHLC DF <= (D<<1) <= DF		ADD								SHL	0 =>	(D<<1)	=>				
		ADI								SHLC							
ANI M(R[P]) & D ->D; R[P]++ SMI D, DF = D - M(R[P]); R[P]++		ANI								SMI				; R[P]+	+		
IDL SHR 0 => (D>>1) =>		IDL								SHR	·						
LDI D = M(R[P]); R[P]++ SHRC DF => (D>>1) => DF		LDI	<b>LDI</b> D = M(R[P]); R[P]++							SHRC	DF => (D>>1) => DF						
<b>MARK</b> $T = (X, P); M(R(2)) = T; X = P; R[2] - $ <b>SM</b> D, DF = D-M(R[X])		MARK								SM							
ORI $D = M(R[P]) \mid D; R[P]++$ SMB $D, DF = D-M(R[X]) - (NOT DF)$		ORI								SMB	D,DF	= D-M(1	R[X])-(	NOT DF)			
		SD								SMBI							
SDB         D, DF = FM(R[X]) - D- (NOT DF)         XRI         D = M(R[P]) ^ D; R[P]++		SDB D, DF = FM(R[X])-D-(NOT DF)								XRI	$D = M(R[P]) ^ D; R[P]++$						
R[] 16 Bits 1 of 16 Scratchpad Registers I 4 Bits Holds High-Order Instruction Digit, not directly accessable		R[] 16 Bits 1 of 16 Scratchpad Registers								I							
R Bits Data Perister (Accumulator)		D								N	4 Bits Low nybble of instruction byte, not						
8 Bits Holds X and P during interrupt, X is		т	8 Bits Holds X and P during interrupt, X is								directly accessable  1-Bit Data Flag (ALU Carry/borrow)						
4 Bits Designates which register is Program  TE 1-Bit Interrunt Enable			4 Bits Designates which register is Program														1
Counter  4 Bits Designates which register is Data  1-Bit Output Flin/Flon		Counter  4 Bits Designates which register is Data															
Pointer		v			ucco						1-Bit Output Flip/Flop						
EF1 to EF4 (4 Flags) These inputs enable the 1/0 controllers to transfer status information to the processor. The levels can be tested by the conditional branch instructions.  www.diegocueva.c		X	Point	er					controll	Q					nrocoss	or Tho	-