



IO	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Read	PIN	Rr_	Rw	Dr_	Dw			Ur_	Uw				PIN		PIN		PIN	
Write	PIN	state											PIN	state	PIN	state	PIN	state
Write2					VCO				T				9-bit D/A output level (155 xor)					
P									CY				8-bit address (6-bit auto-increment)					
PIN state	01=WEAK PULLDOWN			00=tristate			10=Vss(0)			11=Vdd(1)			- Pin State					
VCO	10=OFF			00=D/A			01=Vdd (calibrate)			11=A/D			- Voltage Controlled Oscillator					
T	0=ENABLED			1=disabled									- no-Terminator on D/A output pin					
CY	0=ADD			1=ADD with carry									- p register, Extended Arithmetic Mode					

Port	Address	Description	Port	Address	Description
-d-u	0fd	Down, Up	rd-u	0bd	Right, Down, Up
-d--	0f5	Down	rd--	0b5	Right, Down
-dlu	0ed	Down, Left, Up	rdlu	0ad	Right, Down, Left, Up
-dl-	0e5	Down, Left	rdl-	0a5	Right, Down, Left
---u	0dd	Up	r--u	09d	Right, Up
io	0d1	I/O Control/Status Register	r---	095	Right
--lu	0cd	Left, Up	r-lu	08d	Right, Left, Up
--l-	0c5	Left	r-l-	085	Right, Left

p	10-bit program register, 6-bit auto-increment	r	18-bit 1+8 return stack
a	18-bit general, address, 6-bit auto-increment	t, s	18-bit 2+8 data stack
b	8-bit address (write only)	io	18-bit I/O Control and Status Register

Opcode	Hex	Notes	-- ADDRESS opcodes	Opcode	Hex	Notes	-- ALU opcodes
;	00	return		+	10	. +	
ex	01	execute via r (swap p and r)		2*	11	left shift	
name ;	02	jump to a red word, name		2/	12	right shift (signed)	
name	03	call to a red word, name		-	13	invert (3ffff xor)	
unext	04	jump r≠0 decrement r		+	14	. + (or add with carry)	
next	05	jump r≠0 decrement r		and	15		
if	06	jump t=0		or	16	exclusive or (xor)	
-if	07	jump t17=0		drop	17		
@p	08	literal 6-bit auto-increment		dup	18		
@+	09	fetch via a 6-bit auto-increment		pop	19		
@b	0a	fetch via b		over	1a		
@	0b	fetch via a		a	1b	fetch from register a	
!p	0c	6-bit auto-increment		.	1c	nop	
!+	0d	store via a 6-bit auto-increment		push	1d		
!b	0e	store via b		b!	1e	store into register b	
!	0f	store via a		a!	1f	store into register a	