



IO	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Read	PIN	Rr_	Rw	Dr_	Dw	Lr_	Lw	Ur_	Uw		pin	pin	PIN	PIN	PIN	PIN	PIN	
Write	PIN	state				DB	WD		pin		pin		PIN	state	state	state	state	
	SR			VCO									9-bit DAC output level (155 xor)					

PIN state 01=WEAK PULLDOWN 00=tristate 10=Vss(0) 11=Vdd(1) -Pin State
WD 0=NORMAL 1=inverted -Wake Direction
DB 1=OUTPUT 0=tristate -Data Bus direction
SR 0=RECEIVE 1=send -Serializer/Deserializer
VCO 10=OFF 00=input 01=Vdd calibrate 11=Vss calibrate -Voltage Controlled Oscillator

Port	Address	Description
-d-u	105	Down, Up
-d--	115	Down
-dlu	125	Down, Left, Up
-dl-	135	Down, Left
data	141	Up, no handshake
---u	145	Up
io	15d	18-bit I/O Control/Status
--lu	165	Left, Up
--l-	175	Left

Port	Address	Description
rd-u	185	Right, Down, Up
rd--	195	Right, Down
rdlu	1a5	Right, Down, Left, Up
rdl-	1b5	Right, Down, Left
r--u	1c5	Right, Up
r---	1d5	Right
r-lu	1e5	Right, Left, Up
r-l-	1f5	Right, Left

Note: ALWAYS refer to port names, addresses may change

address 18-bit external address bus
 data 18-bit external data bus
 a 18-bit general, address, 7-bit auto-increment
 b 9-bit address (write only)

p 10-bit program register, 7-bit auto-increment
 r 18-bit 1+8 return stack
 t, s 18-bit 2+8 data stack
 io 18-bit I/O Control and Status Register

Opcode	Hex	Notes	ADDRESS opcodes
;	00	return	
ex	01	execute via r (swap p and r)	
name ;	02	jump to a red word, name	
name	03	call to a red word, name	
unext	04	jump r≠0 decrement r	
next	05	jump r≠0 decrement r	
if	06	jump t=0	
-if	07	jump t17=0	
@p	08	literal 7-bit auto-increment	
@+	09	fetch via a 7-bit auto-increment	
@b	0a	fetch via b	
@	0b	fetch via a	
!p	0c	7-bit auto-increment	
!+	0d	store via a 7-bit auto-increment	
!b	0e	store via b	
!	0f	store via a	

Opcode	Hex	Notes	ALU opcodes
++	10	. ++	
2*	11	left shift	
2/	12	right shift (signed)	
-	13	invert (3ffff xor)	
+	14	. +	
and	15		
or	16	exclusive or (xor)	
drop	17		
dup	18		
pop	19		
over	1a		
a	1b	fetch from register a	
.	1c	nop	
push	1d		
b!	1e	store into register b	
a!	1f	store into register a	