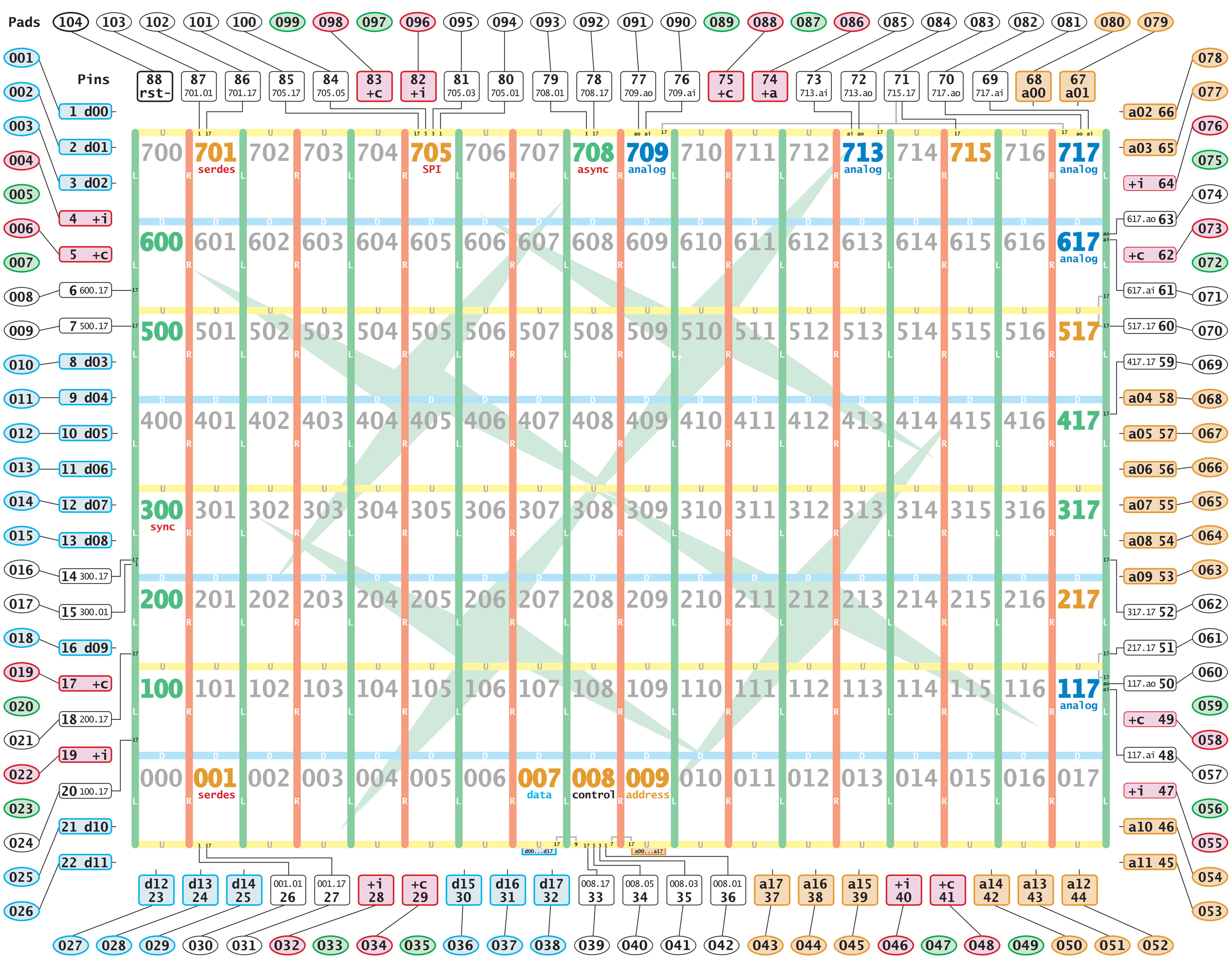

IO	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Read	PIN	Rr_	Rw	Dr_	Dw	Lr_	Lw	Ur_	Uw				PIN		PIN		PIN	
Write	PIN state					DB	WD		pin		pin		PIN state		PIN state		PIN state	
	SR				VCO						9-bit DAC output level				(155 xor)			

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		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**	**. +**
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**	**7-bit auto-increment**	**dup**	**18**	
@+	09	fetch via a	7-bit auto-increment	pop	19	
@b	0a	fetch via b		over	1a	
@	0b	fetch via a		a	1b	fetch from register a
!p	**0c**		**7-bit auto-increment**	**.**	**1c**	**nop**
!+	0d	store via a	7-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

Poster by Bill Muench

