Chip 8 instruction set

(all values in Hexadecimal unless stated)

The Chip-8 instruction set runs in 4k of memory (addresses 000 - FFF). Programs start at 200, memory before that containing the chip-8 interpreter on a real 1802 based machine. The screen is 64 x 32 (128 x 64 on superchip) and is monochrome. Their is a sound buzzer

There are 16 primary registers, called V0 - VF. VF is used for carries and borrows and shouldn't really be used as a general purpose register. There is a 12 bit index register called I. There is a program counter and stack pointer, but neither of these are accessible from program code.

There are 2 counters, the sound timer and the delay timer. Both count down at about 60Hz (on Chip8 they count down in threes using the PC's 18.2Hz Clock). When the sound timer is non-zero the buzzer sounds.

This is the Chip-8 Instruction set as I understand it....

NNN is an address
KK is an 8 bit constant
X and Y are two 4 bit constants

Code	Assembler	Description	Notes
00CN	scdown N	scroll the screen down N lines	SuperChip only
00E0	cls	clear the screen	
00EE	rts	return from subroutine call	
00FB	scright	scroll screen 4 pixels right	SuperChip only
00FC	scleft	scroll screen 4 pixels left	SuperChip only
00FE	low	disable extended screen mode	SuperChip only
00FF	high	enable extended screen mode (128 x 64)	SuperChip only
1NNN	jmp nnn	jump to address NNN	
2NNN	jsr nnn	jump to subroutine at address NNN	16 levels maximum

3XRR	skeq vx,rr	skip next istruction if register VX == constant RR	
4XRR	skne vx,rr	skip next intruction if register VX != constant RR	
5XY0	skeq vx,vy	skip next instruction if register VX == register VY	
6XRR	mov vx,rr	move constant RR to register VX	
7XRR	add vx,rr	add constant RR to register VX	No carry generated
8XY0	mov vx,vy	move register VY into VX	
8XY1	or vx,vy	or register VY with register VX, store result into register VX	
8XY2	and vx,vy	and register VY with register VX, store result into register VX	
8XY3	xor vx,vy	exclusive or register VY with register VX, store result into register VX	
8XY4	add vx,vy	add register VY to VX, store result in register VX, carry stored in register VF	
8XY5	sub vx,vy	subtract register VY from VX, borrow stored in register VF	register VF set to 1 if borrows
8X06	shr vx	shift register VX right, bit 0 goes into register VF	
8XY7	rsb vx,vy	subtract register VX from register VY, result stored in	register F set to 1 if borrows

		register VX	
8X0E	shl vx	shift register VX left, bit 7 stored into register VF	
9XY0	skne vx,vy	skip next instruction if register VX != register VY	
ANNN	mvi nnn	Load index register (I) with constant NNN	
BNNN	jmi nnn	Jump to address NNN + register V0	
CXKK	rand vx,kk	register VX = random number AND KK	
DXYN	sprite vx,vy,n	Draw sprite at screen location (register VX,register VY) height N	Sprites stored in memory at location in index register (I), maximum 8bits wide. Wraps around the screen. If when drawn, clears a pixel, register VF is set to 1 otherwise it is zero. All drawing is XOR drawing (e.g. it toggles the screen pixels)
dry0	xsprite rx,ry	Draws extended sprite at screen location rx,ry	Superchip only: As above,but sprite is always 16 x 16.
ek9e	skpr k	skip if key (register rk) pressed	The key is a key number, see the chip-8 documentation
eka1	skup k	skip if key (register rk) not pressed	
fr07	gdelay vr	get delay timer into vr	
fr0a	key vr	wait for for keypress,put key in register vr	
fr15	sdelay vr	set the delay timer to vr	
fr18	ssound vr	set the sound timer to vr	
fr1e	adi vr	add register vr to the index register	

fr29	font vr	point I to the sprite for hexadecimal character in vr	Sprite is 5 bytes high
fr30	xfont vr	point I to the sprite for hexadecimal character in vr	Superchip only: Sprite is 10 bytes high.
fr33	bcd vr	store the bcd representation of register vr at location I,I+1,I+2	Doesn't change I
fr55	str v0-vr	store registers v0-vr at location I onwards	I is incremented to point to the next location on. e.g. I = I + r + 1
fx65	ldr v0-vr	load registers v0-vr from location I onwards	as above.

Back To: Chip8 Main Page