

Les instructions chip8

Nicolas Monmarché

25 novembre 2024

Introduction

Ce document présente les instructions chip8 (aussi appelées opcodes).

Ces instructions, sont codées sur 2 octets (16 bits). Les 4 premiers bits déterminent l'instruction, les 2×4 bits suivants servent souvent de paramètres (indice de registre...) et les 4 derniers bits servent à préciser l'instruction ou bien contiennent un paramètre.

Par convention, les adresses sont notées *nnn* (sur $3 \times 4 = 12$ bits, $2^{12} = 4096$ valeurs) et les indices de registre *x* et *y* (4 bits chacun).

Comme les entiers codés sur 4 bits vont de 0 à $2^4 - 1 = 15$, on peut représenter la valeur d'un paquet de 4 bits par un symbole en notation hexadécimale : 0, 1 . . . , 9, *A, B, C, D, E, F*.

Une instruction et ses paramètres est donc représentable par 4 symboles en notation hexadécimale.

Le nom (code de 3 lettres) choisi pour l'instruction est arbitraire (mémotechnique).

Chapitre 1

Présentation des instructions

1.1 Sauts

Notation	Mémotechnique	Description	Code
SYS nnn	instruction qui n'est plus utilisée	saut à l'adresse nnn	0nnn
RET	return	retour d'appel de routine, l'adresse de retour est dépilée de la pile	00EE
JMP nnn	jump	saut (inconditionnel) à l'adresse nnn	1nnn
CAL nnn	call	saut d'appel à un sous-programme à l'adresse nnn	2nnn
JAI nnn	jump to address with index	saute à l'adresse nnn+V0	Bnnn
SEQ Vx, kk	skip if equal	saute la prochaine instruction si le registre Vx contient la valeur kk	3xkk
SNE Vx, kk	skip if not equal	saute la prochaine instruction si le registre Vx ne contient pas la valeur kk	4xkk
SRE Vx, Vy	skip if registers equal	saute la prochaine instruction si le registre Vx contient la même valeur que le registre Vy	5xy0
SRN Vx, Vy	skip if registers not equal	saute la prochaine instruction si Vx≠Vy	9xy0

1.2 Manipulation des registres

Notation	Mémotechnique	Description	Code
LVR Vx, kk	load v alue in r egister	le registre Vx prend la valeur kk	6xkk
LRR Vx, Vy	load r egister in r egister	le registre Vx prend la valeur du registre Vy	8xy0
LAR nnn	load a ddress in r egister	le registre I prend la valeur d'adresse nnn	Annn
ARI Vx	a dd r egister to I	ajoute la valeur du registre Vx au registre I	Fx1E
CRD Vx	convert r egister in d ecimal	converti la valeur de Vx en décimal : le chiffre des centaines à l'adresse I, les dizaine à I+1 et les unités à I+2	Fx33
LRI Vx	load r egisters at loca- tion I	les registres V0 à Vx sont copiés à l'adresse I	Fx55
LIR Vx	load from I to r egister	les registres V0 à Vx prennent les valeurs à partir de l'adresse I	Fx65

1.3 Opérations

Notation	Mémotechnique	Description	Code
ADV Vx, kk	add value	le registre Vx prend la valeur $Vx+kk$	7xkk
ADD Vx, Vy	add	le registre Vx prend la valeur de $Vx + Vy$ ($Vf=1$ si le résultat est > 255)	8xy4
SUB Vx, Vn	subtract	le registre Vx prend la valeur de $Vx - Vy$ ($Vf=1$ si $Vx > Vy$)	8xy5
SBU Vx, Vy	subtract (inv)	le registre Vx prend la valeur de $Vy - Vx$ ($Vf=1$ si $Vy > Vx$)	8xy7
OR Vx, Vy	or	le registre Vx prend la valeur de Vx or Vy (opération bit à bit)	8xy1
AND Vx, Vy	and	le registre Vx prend la valeur de Vx and Vy (opération bit à bit)	8xy2
XOR Vx, Vy	xor	le registre Vx prend la valeur de Vx xor Vy (opération bit à bit)	8xy3
SHR Vx	shift right	décalage du registre Vx à droite , Vf prend la valeur du bit perdu	8xy6 (y?)
SHL Vx	shift left	décalage du registre Vx à gauche , Vf prend la valeur du bit perdu	8xyE (y?)

1.4 Affichage

Notation	Mémotechnique	Description	Code
CLS	clear screen	effacement de l'écran	00E0
DRW Vx, Vy, n	draw	lit n octets en mémoire à partir de l'adresse I et affiche (xor) le sprite lu à la position Vx, Vy . Si des pixels sont écrasés, $Vf=1$	Dxyn
LSI Vx	load symbol sprite to I	le registre I prend la valeur de l'adresse du sprite représentant les symboles 0..F correspondant à la valeur du registre Vx	Fx29

1.5 Périphériques

Notation	Mémotechnique	Description	Code
SKP Vx	skip if key p ressed	saute la prochaine instruction si la touche Vx est pressée	Ex9E
SKN Vx	skip if key n ot pressed	saute la prochaine instruction si la touche Vx n'est pas pressée	ExA1
WKP Vx	wait for key p ressed	attend la prochaine touche pressée. Le numéro de la touche sera placé dans Vx	Fx0A
LRS Vx	load register in sound timer	Le timer de son prend la valeur du registre Vx	Fx18

1.6 Divers

Notation	Mémotechnique	Description	Code
RND Vx, kk	random	place dans un entier aléatoire dans Vx et fait un et logique bit à bit avec kk	Cxkk
LTR Vx	load t imer in r egister	le registre Vx prend la valeur du registre de timer DT	Fx07
LRT Vx	load r egister in t imer	le registre de timer DT prend la valeur du registre Vx	Fx15