

NOM	NB_ARGS	OPCODE	ARGUMENTS	BINAIRE	HEXA	CYCLES	DESCRIPTION	OCP	CARRY	Precision	NOM
live	1	1	DIR	00000001	0x01	10	Dire "Je suis en vie"	NON	NON	Parametre sur 4 octets	live
ld	2	2	DIR   IND, REG	00000010	0x02	5	Charger la valeur du premier param dans le registre	OUI	OUI		ld
st	2	3	REG, IND   REG	00000011	0x03	5	Stocker valeur registre dans second parametre	OUI	NON		st
add	3	4	REG, REG, REG	00000100	0x04	10	Additionne les 2 premier et met le resultat dans le 3eme	OUI	OUI		add
sub	3	5	REG, REG, REG	00000101	0x05	10	Soustrait les 2 premier et met le resultat dans le 3eme	OUI	OUI		sub
and	3	6	REG   IND   DIR, REG   IND   DIR, REG	00000110	0x06	6	Fait un & logique binaire sur les 2 premier et met le resultat dans le 3eme	OUI	OUI		and
or	3	7	REG   IND   DIR, REG   IND   DIR, REG	00000111	0x07	7	Fait un   (OU) logique binaire sur les 2 premier et met le resultat dans le 3eme	OUI	OUI		or
xor	3	8	REG   IND   DIR, REG   IND   DIR, REG	00001000	0x08	8	Fait un OU EXCLUSIF logique binaire sur les 2 premier et met le resultat dans le 3eme	OUI	OUI		xor
zjmp	1	9	DIR	00001001	0x09	20	Prend une adresse et fait un saut vers cette adresse	NON	NON	Fait un saut si le carry est a 1	zjmp
ldi	3	10	REG   IND   DIR, REG   DIR, REG	00001010	0x0A	25	Additionne les 2 premier, traite ça comme une adresse, y lit une valeur de la taille d'une registre et la met dans la 3eme	OUI	NON		ldi
sti	3	11	REG, REG   DIR   IND, DIR   REG	00001011	0x0B	25	Additionne les deux dernier, utilise cette somme comme une adresse ou sera copié la valeur du premier parametre	OUI	NON		sti
fork	1	12	DIR	00001100	0x0C	800	Crée un nouveau processus qui herite des etats du pere sauf le PC qui est mis a PC + (1er param % IDX_MOD)	NON	NON		fork
lfd	2	13	DIR   IND, REG	00001101	0x0D	10	Crée un nouveau processus qui herite des etats du pere sauf le PC qui est mis a PC + 1er param	OUI	OUI		lfd
lfdi	3	14	REG   IND   DIR, REG   DIR, REG	00001110	0x0E	50	comme ldi mais n'applique aucun modulo aux adresses	OUI	OUI		lfdi
lfork	1	15	DIR	00001111	0x0F	1000	Comme fork sans modulo	NON	NON		lfork
aff	1	16	REG	00010000	0x10	2	le contenu du registre est interprete comme la valeur ASCII d'un char a afficher sur la sortie standard. Ce code est modulo 256	OUI	NON		add

VOCABULAIRE DESCRIPTION

PC	Registre qui contient l'adresse de la prochaine instruction a decoder et executer
OCP	Octet de Codage de parametres. Permet a la VM de savoir comment charger les params d'une instruction
OPCODE	C'est le code (l'ld) de la commande
CARRY	C'est un flag qui vaut 1 si la demiere operation est reussi ET si la commande modifie le Carry

LABELS_CHARS	abcd_[]xyz_0123456789
LABEL_CHAR	:
SEPARATOR_CHAR	,
DIRECT_CHAR	%

PARAMETRES DIRECT CODÉ SUR 4 OCTETS :

live	1
ld	2
and	6
or	7
xor	8
lfd	13

CODE DE L'OCP

0	0	Absent
0	1	Registre
1	0	Direct
1	1	Indirect
Exemple : ld avec : DIR et REG		1001 0000 = 0x00 REG DIR DIR 0110 1000

Formatage instruction : Un label (Optionnel), un opcode, et les parametres separes par SEPARATOR\_CHAR  
Formatage label : chaîne de caractere parmi les LABEL\_CHARS, suivi par LABEL\_CHAR

PRECISION SUR LE FORMATAGE DES ARGUMENTS

REGISTRE :	Apres 'r', accepte maximum 2 chiffres et rien d'autre ! de 00 a 99 MAXIMUM 16 registres par asm code
DIRECT :	% 'r' avec des LABEL_CHARS apres Un nombre positif ou negatif STOCKÉ DANS UN "long long int", (9223372036854775807), fait une boucle continuellement et marque FF FF au dessus Au dessus de ce nombre, tout les octets sont a FF.
INDIRECT :	'r' avec des LABEL_CHARS apres Un nombre positif ou negatif STOCKÉ DANS UN "long long int", (9223372036854775807) au dessus de 65535, FF FF Au dessus de ce nombre, tout les octets sont a FF.