Initiation à l'Assembleur (3)

par André C. et tous ceux qui voudront bien y participer...

Il y a bien longtemps déjà que les deux premiers articles de cette série sont parus (Ceo-Mag 159-160 pages 15 à 20 et 161 pages 14 à 16 en 2003 !). Que ceux qui m'avaient demandé une suite me pardonnent. J'y pense souvent, mais c'est toujours le même problème...

Ayant depuis longtemps constaté que d'une part ce sont toujours les mêmes instructions qui sont 'délicates à utiliser' et que d'autre part une poignée seulement d'instructions reviennent tout le temps, j'avais écrit (pour mon propre usage) un petit mémento pour les quelques instructions qui m'obligeaient sans arrêt à consulter la documentation. C'est ce petit mémento, tout juste un peu finalisé, que je vous propose. Garder-en une copie à portée de main et vous constaterez que, finalement, le langage machine est très simple à utiliser.

Remarque préliminaire : Un octet est formé de 8 bits, qui sont numérotés de droite à gauche de b0 à b7 où b0 est le bit de poids le plus faible et b7 le bit de poids le plus fort. Autre convention, les valeurs sur deux octets sont souvent notées LLHH, c'est à dire l'octet de poids faible (Low) en premier, suivit de l'octet de poids fort (High).

LES 8 BRANCHEMENTS

La principale difficulté lorsqu'on essaie de comprendre un listing de désassemblage est due aux 8 ordres de branchement en fonction de l'état des drapeaux Z, N, C, et V.

BNE	(**B**ranch if **N**ot **E**qual)	c'est à dire branche si Z = 0
BEQ	(**B**ranch if **EQ**ual)	c'est à dire branche si Z = 1
BPL	(**B**ranch if **PL**us)	c'est à dire branche si N = 0
BMI	(**B**ranch if **MI**nus)	c'est à dire branche si N = 1
BCC	(**B**ranch if **C**arry **C**lear)	c'est à dire branche si C = 0
BCS	(**B**ranch if **C**arry **S**et)	c'est à dire branche si C = 1
BVC	(**B**ranch if o**V**erflow **C**lear)	c'est à dire branche si V = 0
BVS	(**B**ranch if o**V**erflow **S**et)	c'est à dire branche si V = 1

En fait la difficulté vient plutôt du fait que, souvent, l'on ne connaît pas de façon précise la position des fameux drapeaux Z, N, C et V, notamment après les instructions CMP, CPX et CPY qui sont très utilisées et que nous verrons bientôt.

RAPPELS SUR LES DRAPEAUX N, Z, C et V

Z, N, C et N sont mis à 0 ou à 1 selon les instructions 6502 effectuées.

Drapeau Z : Zéro est mis à 1 si le résultat d'une instruction est nul: le plus facile des drapeaux.

Drapeau N : Négatif, recopie le b7 du résultat d'une instruction, (nombre signé ou non).

On appelle nombre signé, un nombre dont le bit de poids le plus fort sert d'indicateur de signe : 0 pour les valeurs positives et 1 pour les valeurs négatives. Les valeurs négatives sont exprimées sous forme de complément à deux (voir plus loin). Sur 8 bits, on peut écrire des nombres non signés de 0 à 255 et des nombres signés de -128 à +127

Pour les nombres signés (de -128 à +127) le b7 (et donc N) sert de signe. S'il est à 0, il s'agit d'un nombre positif (de 0 à 127, c'est à dire de 0000 0000 à 0111 1111 ou #00 à #7F). S'il est à 1, il s'agit d'un nombre négatif (de -128 à -1, c'est à dire de 1000 0000 à 1111 1111 ou #80 à #FF). Notez bien que -128 correspond réellement à 1000 0000 en binaire (#80 en hexadécimal) et que -1 correspond réellement à 1111 1111 (#FF).

Pour les nombres non signés (de 0 à 255) le b7 sert seulement à savoir s'il est > 127 (c'est à dire > 0111 1111 ou > #7F) (utilisé par exemple pour l'affichage en vidéo inverse ou pour reconnaître un token BASIC).

Drapeau C : Explication de base : Lors d'une opération arithmétique, si le résultat dépasse #FF (capacité limitée à 255 sur 8 bits), la retenue est mise dans **C** (carry = retenue en anglais). Par exemple #FF + #02 = #01 et C = 1, car le résultat est #101. D'une manière plus générale on dit que Carry retient la retenue qui 'sort' du bit n°7 à la suite d'une instruction 6502. Nous verrons plus loin par exemple que lors d'un décalage des bits vers la gauche (ASL), le b7 qui 'sort' de l'octet est mis dans **C**.

Attention, lors d'une soustraction la signification de Carry est inversée (à l'entrée et à la sortie). Dans ce cas, Carry est en fait le complément à 1 de la retenue. En général, à l'entrée il n'y a pas de retenue : il faut poser C = 1. Si à la sortie on a C = 1, cela signifie qu'il n'y a pas de retenue. Par exemple : #F6 - #18 = #DE et C=1 soit 246 - 24 = 222).

La soustraction SBC effectue en fait A = A - M - (1 - C). Lorsqu'il n'y a pas de retenue, C = 1 et A = A - M. Lorsqu'il y a une retenue, C = 0 et A = A - M - (1 - 0) soit A = A - (M + 1) (voir plus loin). C'est une des rares notions un peu 'délicates' à mémoriser.

Drapeau V : o**V**erflow c'est l'indicateur le plus complexe et le moins utilisé. Il sert à savoir s'il y a eu dépassement de capacité lors d'une opération arithmétique. Lorsque les nombres non signés (de 0 à 255), codés sur 8 bits dépassent 255, une retenue passe de b7 à 'b8'. Cette retenue est gardée dans Carry. Vous imaginerez facilement que la chose est plus compliquée avec les nombres signés. Les nombres signés (de -128 à +127) étant en fait codés sur 7 bits, il y a overflow lorsqu'ils dépassent +127 ou lorsqu'ils sont < -128. Mais à cause des soustractions, il n'était pas facile de trouver un indicateur qui marche dans tous les cas, c'est pourquoi il est très compliqué : L'indicateur V est obtenu en effectuant un OU exclusif (XOR) entre la retenue lors du passage de b6 à b7 et la retenue lors du passage de b7 à 'b8' (carry). Ainsi si l'une de ces 2 retenues (mais pas les 2) est à 1, alors V est à 1.

RAPPELS SUR LES INSTRUCTIONS QUI AFFECTENT LES DRAPEAUX

Lorsque vous comprendrez bien comment sont affectés les drapeaux, alors la programmation en langage machine sera un jeu d'enfant pour vous. En effet, les instructions de base, peu nombreuses, sont en fait utilisées de manière répétitive. Ce sont surtout les instructions peu utilisées qui vont feront hésiter.

INSTRUCTIONS N'AFFECTANT PAS LES DRAPEAUX Z, N, C ET V

Je les aime bien celles-là. Ils s'agit des instructions "Store" (STA, STX et STY), "Jump" (JMP, JSR et RTS, mais attention RTI affecte tous les drapeaux), "Branch" (BNE, BEQ, BPL, BMI, BCC, BCS, BVC et BVS), "PusH" (PHA et PHP) ainsi que NOP et TXS.

INSTRUCTIONS AFFECTANT LES DRAPEAUX Z ET N

(autres mouvements, incrémentation, décrémentation et opérations logiques)

Tous les "Load" (LDA, LDX et LDY), mais aucun "Store" (STA, STX, STY).
Tous les "Transfer" (TAX, TXA, TAY, TYA et TSX) mais, attention pas TXS.
Tous les "PulL" (PLA et PLP), mais pas les "PusH" (ni PHA, ni PHP).
Tous les "Increase" (INC, INX et INY) et les "Decrease" (DEC, DEX et DEY).
Toutes les opérations logiques (AND, ORA, EOR et BIT) (attention pour BIT en plus b6 -> V).

INSTRUCTIONS AFFECTANT LES DRAPEAUX Z, N ET C

Toutes les autres opérations "bit à bit" (ASL, LSR, ROL, ROR) que nous verrons en détail plus loin.
Tous les "Compare" (CMP, CPX et CPY) que nous verrons également en détail, car très utilisées.

OPERATIONS ARITHMETIQUES

ADC et SBC positionnent non seulement N et Z mais aussi C et V. Nous consacrerons un gros morceau à ces opérations mathématiques, car elles sont la source de bien des soucis.

INSTRUCTIONS SPECIALES

PLP et RTI affectent tous les drapeaux (Z, N, C, V, D et I).
BRK, CLI et SEI affectent le drapeau I et ce n'est pas surprenant !
CLC et SEC affectent C par définition.
CLD et SED affectent D aussi par définition.
CLV affecte le drapeau V et toujours par définition.

à suivre...