La commande Break, mnémonique BRK, code #00

par André C.

L'excellent retour de Yann L. sur le 6502 m'a rebranché sur certains aspects mystérieux du langage machine, par exemple à propos de la commande Break (CEO-mag n°324 page20, avril 2017).

Etat de la question

En première approximation, l'instruction BRK (code #00) est censée faciliter le débogage des programmes langage machine (BRK signifie arrêt forcé). En simplifiant outrageusement, on pourrait dire que le processeur s'arrête, quand il rencontre ce BRK, cet arrêt permettant alors d'étudier l'état des registres, des variables, etc.

Yann indique que c'est loin d'être aussi simple et décrit brièvement ce qui se passe lors d'un BRK et les conséquences possibles. Son sujet n'étant par le BRK, mais la revue générale du 6502, il ne pouvait évidemment pas approfondir, sous peine d'écrire un livre, car il n'y a pas que le BRK dans le 6502 !

Dans ce qui suit, je traite du BRK de l'Oric (nom générique pour désigner l'Oric-1 et l'Atmos dans la configuration de base, sans lecteur de disquette). Je ne traite qu'accessoirement des autres configurations. Par souci de simplification, je ne donne donc que les adresses mémoire de l'Atmos.

Petit rappel préalable

Le Status Register (SR ou Indicateur d'état, parfois noté P) du processeur 6502, comporte 7 drapeaux, dont un seul nous intéresse ici, le drapeau B (parfois noté b) nommé "Break command", qui est mis à 1 lorsque le processeur rencontre un BRK en exécutant un programme. Il est utilisé pour déterminer si une interruption a été causée par un BRK ou par une séquence d'interruption normale (IRQ). Il n'existe aucune instruction pour mettre le drapeau B à 0 ou à 1 (pas de BRK SET ni de BRK CLEAR). Ce drapeau n'a de signification que lors de l'analyse d'une séquence d'interruption normale.

Que se passe-t-il donc lorsque le microprocesseur rencontre un BRK?

Ce n'est pas si simple à déterminer, les différentes sources présentant de légères différences. Cela est principalement dû à une confusion entre ce que fait le microprocesseur et ce que fait ensuite la machine hôte. Coup de chapeau à Yann pour son bref mais précis résumé, cité plus haut. Voyons ce que fait le processeur :

- Il met à 1 le drapeau B (commande Break) du SR (Status register), ceci pour signaler qu'il ne s'agit pas d'une interruption normale IRQ, mais d'un BRK (les 2 utilisent la même routine de gestion).
- Empile SR (toujours le Status register).
- Empile PC (Program Counter). La valeur empilée est l'adresse de BRK+2, comme si la commande BRK occupait 2 octets. Voir plus loin les conséquences sur l'utilisation de la commande BRK. Ceci fait, le processeur est paré pour s'y retrouver à la sortie de la gestion du BRK.
- Décrémente 3 fois SP (Stack Pointer ou pointeur de pile, aussi noté S).
- Initialise PC avec l'adresse #FFFE et enfin il saute à l'adresse indiquée par ce vecteur.

Et c'est tout ! La suite concerne la machine hôte. Dans le cas de l'Atmos, l'adresse indiquée en #FFFE est #0244 (en Ram), qui elle-même renvoie en #EE22 (en Rom, routine de gestion des IRQ/BRK). Cette routine teste le drapeau B du SR (Status Register). S'il s'agit d'un BRK, elle renvoie en #024A (encore en Ram) pour exécuter un simple RTI (ReTurn from Interrupt). Ce RTI remet en place le SR et le PC qui avaient été empilés au début de la procédure et reprend donc le programme en exécutant (ou en tentant d'exécuter) le 2e octet après BRK, sautant ainsi l'octet qui suit BRK. Cet octet "sauté" peut être considéré comme un pseudo argument, éventuellement utilisable si la routine de gestion IRQ/BRK est modifiée pour le traiter (voir plus loin le cas du Telestrat). Mais comme le souligne Yann la reprise d'exécution au 2e octet après le BRK peut être scabreuse si on ne prend pas soin d'ajouter un NOP (No OPeration) après le BRK.

Le BRK de l'Oric

Dans ce tout ce que vous venez de lire, la procédure décrite est propre au 6502, sauf bien sûr les deux passages en Ram en #0244 et #024A, ainsi que ce qu'on trouve à ces adresses (dans notre cas, appel à la routine en Rom en #EE22, test du drapeau B et simple retour avec RTI), qui sont propres à l'Oric.

Vous l'avez déjà deviné : le comportement d'un BRK varie d'une machine à l'autre, selon la configuration prévue par les concepteurs de la machine.

Dans le cas de l'Atmos, c'est finalement simple : il saute le BRK et l'octet qui suit et poursuit bêtement l'exécution du programme en langage machine, comme si de rien n'était. C'est très fâcheux si on voulait justement faire un arrêt pour

déboguer le programme ! Encore plus fâcheux, comme le signale Yann, l'insertion du BRK a toutes les chances de faire planter le programme. Yann conseille donc d'insérer un BRK suivit d'un NOP (No OPeration) plutôt que d'insérer un BRK tout seul ! Ce n'est pas tout à fait aussi simple et ce problème de plantage sera ultérieurement évoqué dans un article "Trucs et Astuces Programmation". Note : L'oric-1 et l'Atmos utilisent la même routine de gestion IRQ/BRK et se comportent donc de manière identique (j'ai testé).

Tous les ordinateurs ne sont pas configurés comme l'Oric. La plupart des machines retournent simplement au "prompt" du Basic ou du DOS. Certains DOS, notamment XL-DOS et Sedoric (dont l'auteur est Fabrice Broche comme pour le Telemon, voir ci-après) utilisent également le BRK de manière spécifique.

Le cas de Sedoric se doit d'être souligné : Après un BRK, le programme interrompu retourne gentiment au prompt du DOS. Pas besoin de NOP. La seule condition est de placer le BRK au bon endroit, c'est-à-dire pas là où il serait pris pour l'argument d'une autre commande. Fabrice Broche a clairement approfondi le cas du BRK, car il a laissé sa marque sur tous les systèmes qu'il a touché.

Travaux pratiques avec Atmos + Sedoric

Faisons un petit essai avec notre programme fétiche (voir Trucs et Astuces Programmation 2), que j'avais sauvegardé sur le nom "SALUT".

Voici l'explication du code :

```
9801 53 61 6C 75 74 20 6C 65 73 20 67 61 72 73 20 21 00 "Salut les gars !"
9812 A0 00    LDY #00    mettre la valeur zéro dans le registre Y
9814 B9 01 98 LDA 9801,Y lire l'octet présent à l'adresse 9801+Y
9817 F0 06    BEQ 981F   si c'est 0, fini, on termine en 981F
9819 99 82 BB STA BB82,Y sinon on copie l'octet en BB82+Y
981C C8       INY Y=Y+1  pour indexer l'octet suivant
981D D0 F5    BNE 9814   reboucle tant que Y ne repasse pas à zéro
981F 60       RTS        retourne au point d'appel
```

En #9819, remplaçons #99 (STA) par #00 (BRK). Lançons le programme, ainsi modifié, par un CALL#9812. On constate qu'il ne se passe rien, sinon que l'écran affiche "BREAK ON BYTE #9819" preuve que ça a fonctionné (Figure 1). Mais le programme a été arrêté avant d'afficher l'octet #53 sur la ligne service. Contrairement à ce qui se serait passé avec le même programme modifié et avec un Atmos sans Microdisc, la machine n'est pas plantée. Elle a simplement rendu la main. Recommençons en plaçant notre BRK un peu plus loin, à la place du #D0 (BNE). Cette fois, on voit apparaître un "S" sur la ligne service (Figure 2). C'est le 1er caractère de notre message. Mais le BRK a arrêté l'exécution du programme avant qu'il puisse afficher la suite. La machine a rendu la main sans planter. Merci monsieur Broche !

Fig 1

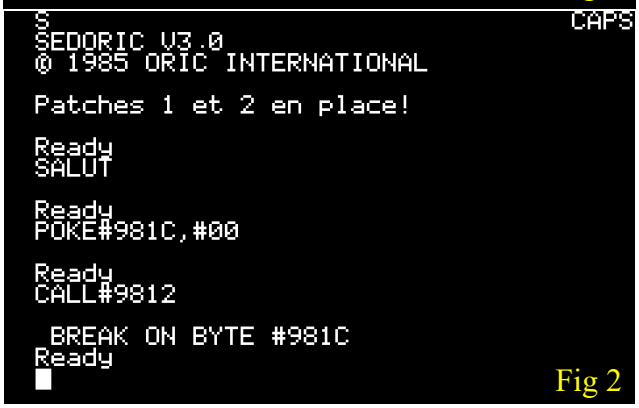

Fig 2

Un cas intéressant est celui du Telestrat.

La plupart des articles traitant du BRK dans le CEO-mag sont consacrés au BRK du Telestrat. Dans le Telemon, l'instruction BRK a une utilisation spéciale (voir l'excellent article de Jérôme D. dans le CEO-mag n°323, pages 10 et 11). Elle permet d'appeler une routine système repérée par un code d'un octet. Il suffit de programmer BRK+code (code restant "dans

l'ombre" de BRK, car le processeur saute par-dessus). La routine de gestion du BRK récupère ensuite ce code (en fait tout simplement le numéro d'une routine système) et déclenche un saut à cette routine. En somme il s'agit d'une sorte de "JSR n°_de_routine" au lieu du classique "JSR adresse".

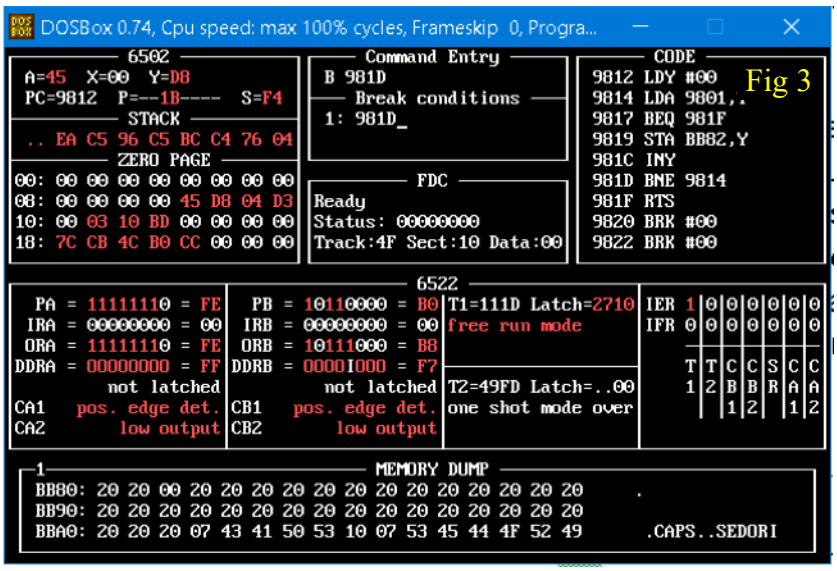

Fig 3

Le BRK du débogueur intégré à Euphoric

Placer un BRK en absence de moniteur reste de peu d'intérêt pour explorer la situation quand on cherche à déboguer un programme. Heureusement pour nous, Fabrice F. a concocté une prothèse de choix en ajoutant un débogueur intégré à Euphoric (touche F11). En mode débogueur, il est possible de poser un BRK là où on a besoin d'arrêter l'exécution et d'examiner tout à loisir l'état des lieux. Pour éviter les écueils, Fabrice n'utilise pas pour de vrai la commande BRK du processeur. Euphoric se contente de noter l'adresse à laquelle il doit arrêter l'exécution de la routine langage machine et affiche alors l'état de tous les registres, de la mémoire etc. Ça fonctionne net et précis, nickel chrome!

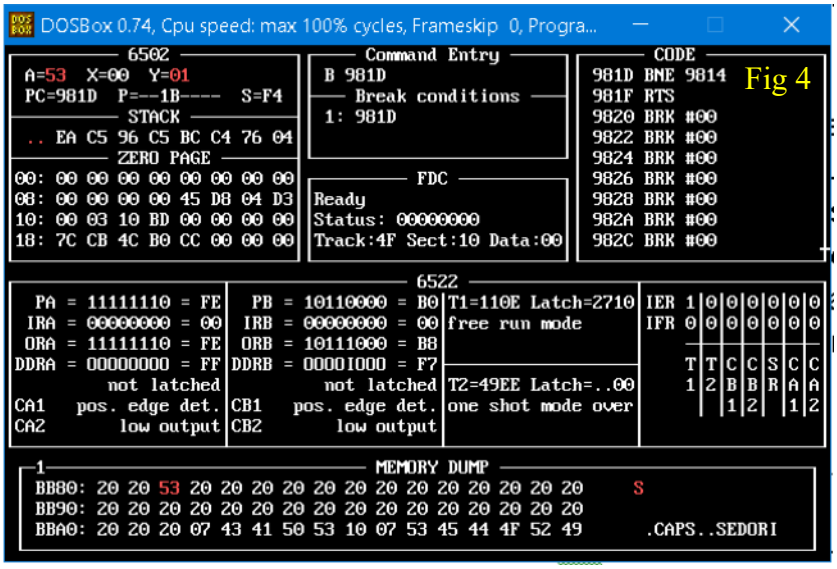

Fig 4

Note : Il est possible de placer un vrai BRK (un code #00 dans le programme) et d'examiner ce qui se passe avec le débogueur d'Euphoric. Ça marche aussi et c'est intéressant pour comprendre comment une configuration donnée gère la commande BRK.

Travaux pratiques avec le débogueur intégré à Euphoric

Faisons un petit essai avec le programme fétiche utilisé ci-dessus. Voyons ce qu'Euphoric fera si on lui dit de mettre un BRK après #981C. Le mode d'emploi du BRK du débogueur (CEO-mag n°281, pages 40-48) traite justement de cet exemple. Tapez "PC 9811" pour placer le processeur au début du programme, "D BB80" pour visualiser la zone mémoire de l'écran, "B 981D" pour mettre un BRK après le INY et enfin F5 pour lancer l'exécution jusqu'au BRK (figure 3).

Effectivement celle-ci s'arrête sur l'instruction BNE 9814, comme l'indique le panneau "CODE". On peut vérifier que l'octet #53 a bien été chargé dans les data (panneau "6502"), que la lettre "S" correspondante a été affichée dans l'écran (panneau "MEMORY DUMP") et finalement que Y a bien été incrémenté à 1 (panneau "6502") (figure 4). On peut vérifier en listant le programme avec un autre désassembleur, que le débogueur d'Euphoric n'a pas remplacé "pour de vrai" l'octet #D0 présent en #981D (BNE) par un octet #00 (BRK). Il a donc seulement noté qu'il fallait arrêter l'exécution quand le PC atteindrait #981D. Donc pas d'embrouille avec ni avec le 6502, ni avec la gestion du BRK liée à la configuration de la machine. Au poil ! Merci Fabrice !

La suite…

Je prépare un second article consacré au BRK, dans la rubrique "Trucs et Astuces Programmation", dans lequel nous verrons comment planter l'Oric en plaçant un BRK n'importe comment et surtout comment profiter du passage en Ram pour modifier la gestion du BRK selon nos besoins.

A bientôt donc…