smaky info

D.Bug

Introduction

D.Bug est un déverminer interactif développé à l'usage des programmeurs travaillant en assembleur.

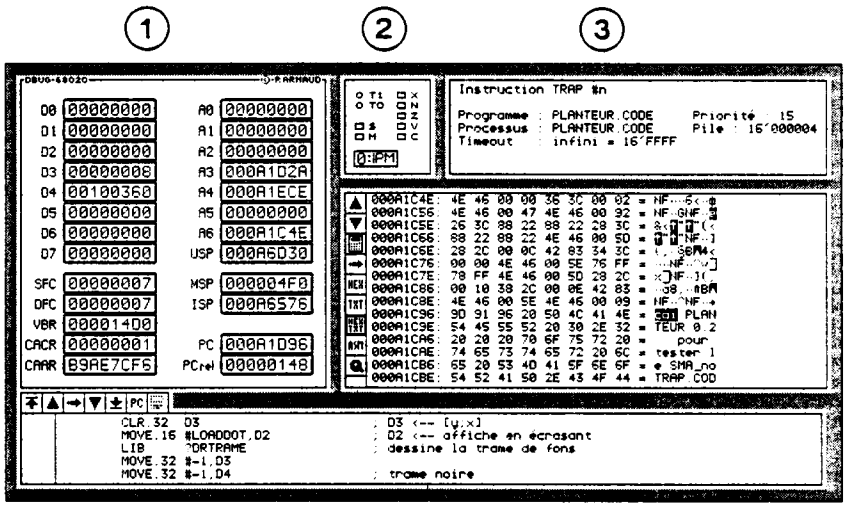
Pour déverminer un programme, il suffit de lancer **D.Bug** dans une fenêtre et d'exécuter le programme dans une autre. Dès qu'une exception survient (*TRAP, DIV #0*, erreur de bus, etc.), **D.Bug** reprend le contrôle de la situation.

Pour exécuter un programme **pas à pas** depuis la première instruction, il faut donner sur la ligne de commande :

DBUG monprog.code/X

Le "/X" signale qu'il faut exécuter le programme juste après le chargement de **D.Bug**.

Il est aussi possible de demander le déverminage de programmes par la suite en pressant DEFINE.



Fenêtres de D.Bug

L'écran de **D.Bug** est séparé en 5 fenêtres :

1. **Le contenu des registres.** Les registres sont éditables. La valeur contenue dans un registre peut être copiée dans le bloc-notes par un COPY clic.

2. **Les fanions de SF.** Un clic permet de changer l'état de ces bits. Les bits *superviseurs* sont aussi modifiables. On peut ainsi passer en mode trace dans un appel système (clic sur le bit **T1**).

3. **Le nom de l'exception et l'état du processus.** Lorsque D.Bug affiche "...", cela signifie que des détails peuvent être obtenus par HELP.

4. **Le contenu de la mémoire.** La barre d'icones à gauche de cette fenêtre sera étudiée plus loin.

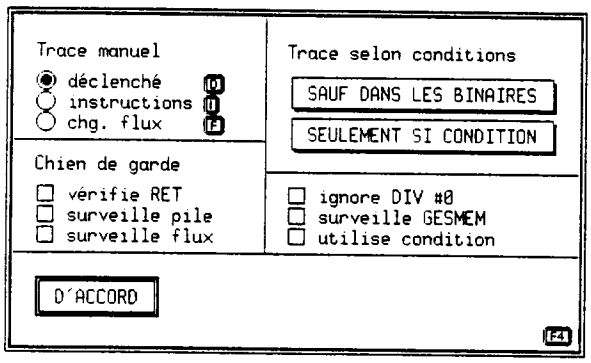
5. **Le fichier de listage.** Cette fenêtre sera étudiée en détail plus loin.

Les fonctions de D.Bug

D.Bug utilise abondamment les touches fonction (voir *fig.1*), bien que toutes les commandes soient aussi accessibles par le menu du bouton du milieu.

Le bloc F1...F3 affiche le nom de l'exception et permet de continuer dans le mode choisi (normal, pas à pas, etc).

La touche F4 accède au dialogue des définitions du mode de trace :



Le *trace manuel* en mode *instructions* avance pas à pas alors qu'en mode *chg. flux*, il avance jusqu'à la prochaine instruction du type **JUMP**, **CALL**, **RET**, **TRAP**, etc.

Le *chien de garde* permet de vérifier si le **RET** retourne bien sur l'instruction suivant un **CALL**, si la pile reste dans les marges imposées et si le **PC** ne part pas dans les choux.

Il est possible d'ignorer le trace manuel dans certains binaires ou au contraire, de ne le signaler que lorsqu'une condition est remplie (il ne faut alors pas oublier d'allumer la croix *utilise condition*).

Le mode *ignore DIV #0* est pratique si l'on désire sauter les divisions par zéro.

Le mode *surveille GESMEM* nécessite quelques explications :

Il permet de **signaler les débordements de mémoire avant que ceux-ci n'aient physiquement lieu**. Cela est possible grâce à l'émulation des appels **GESMEM** et à la simulation individuelle par **D.Bug** de chaque accès mémoire.

Cela ralentit considérablement l'exécution du programme à déverminer, car chaque accès mémoire fait alors un **LOCK** et un **UNLOCK** !

La touche F6 permet de définir quels fichiers de listage conserver en mémoire et lequel afficher.

Les fonctions *Step over* F7 et *Finish* F8 permettent respectivement de traiter un **CALL** ou **LINEA** comme une instruction en mode trace et de terminer la routine en cours.

Les touches *Auto. cont.* F10, *Discret* F11, *Swap Key* F12 et *Montre list.* F13 présentent un comportement à bascule. Lorsqu'elles sont actives, elles ont les significations suivantes :

- continue automatiquement si trace en cours.
- le chien de garde affiche un message discret.
- commute de fenêtre entre le logiciel à déverminer et **D.Bug**.
- montre le fichier de listage associé.

Finalement, *Abort proc.* F15 permet de quitter brutalement un programme en cours de déverminage.

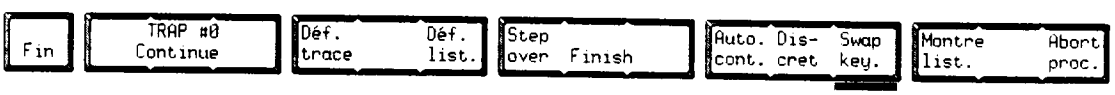
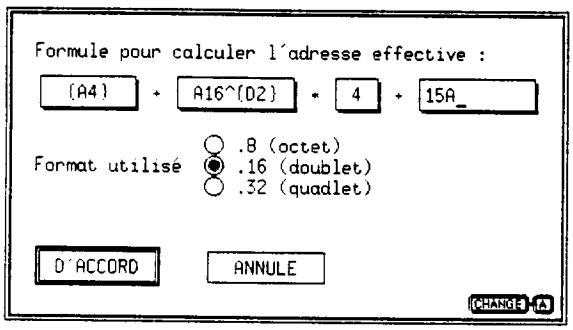


Fig.1 : Touches de fonction de D.Bug lorsqu'une exception est survenue.

Visualisation de la mémoire

La visualisation de la mémoire est possible de tout temps. Un *clic* sur la calculatrice [▦] permet de définir l'adresse mémoire où commencer l'affichage :



La syntaxe utilisée pour définir l'adresse effective est la même qu'en **CALM**.

Le *format utilisé* permet de déterminer la façon dont sera affichée la mémoire.

Les boutons hexa. [HEX], texte [TXT], hexa/texte [HEX TXT] et désassemblé [ASM] permettent de choisir le format de l'affichage.

Les flèches [▲][▼] permettent de se déplacer dans la mémoire et la petite loupe [🔍] indique de quel binaire il s'agit.

Enfin, le rafraîchisseur [⇻] permet de rafraîchir l'affichage. Des définitions sont accessibles par un CHANGE clic.

Le fichier de listage

Si un fichier de listage assembleur au format *CALM*, *CALM 2* ou *CALM 3* existe, il est possible de l'afficher en pressant sur Montre list..

Il faut prendre garde au fait que le source doit au moins comporter un **.TITLE** et un **.END**

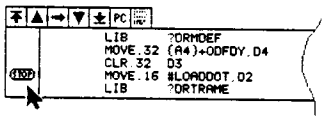
La taille du fichier de listage est comparée à la taille du binaire et s'il y a désaccord, le fichier de listage ne peut pas être utilisé.

Le fichier de listage est cherché dans le dossier courant et dans l'unité *#DBUGLIST:*. En cas d'échec, un dialogue permet de charger manuellement le fichier de listage.

La barre d'icones audessus de la fenêtre affichant le fichier de listage permet de se déplacer (flèches haut/bas), de montrer la ligne où se trouve le PC courant [⇻], de cherche un PC relatif donné [PC] et de chercher une fonction [▤].

Points d'arrêt

D.Bug permet de poser des points d'arrêt dans les fichiers de listage chargés. Il suffit de cliquer dans la marge en face de la ligne où poser le *break-point* :



D.Bug insère en fait un **EXIT** à cet endroit et émulera l'instruction qui s'y trouvait.

D.Bug et NoTrap

La version de **NoTrap** distribuée avec le système 8 est compatible avec **D.Bug**.

Après que **NoTrap** a affiché son dialogue habituel, il est possible de charger **D.Bug**, puis de revenir dans le dialogue de **NoTrap**. Il suffit alors de presser les touches PROGRA SHOW X.

D.Bug reprend alors l'erreur qui avait été interceptée par **NoTrap**.

Les conditions

La définition d'une condition de trace se fait par le dialogue présenté à la *figure 2.*

La valeur **X** est une adresse effective avec la notation *CALM.*

La condition porte sur la comparaison des valeurs **A** (calculée à partir de **X**) et **B** (définie constante).

La valeur **A** peut être soit l'adresse **X**, soit le contenu mémoire pointé par **X**.

Si plusieurs conditions sont définies, c'est leur **union** qui sera utilisée. Si l'option *trace sauf si..* est utilisée en même temps que d'autres conditions, cela permet d'exclure le trace dans le cas déterminé.

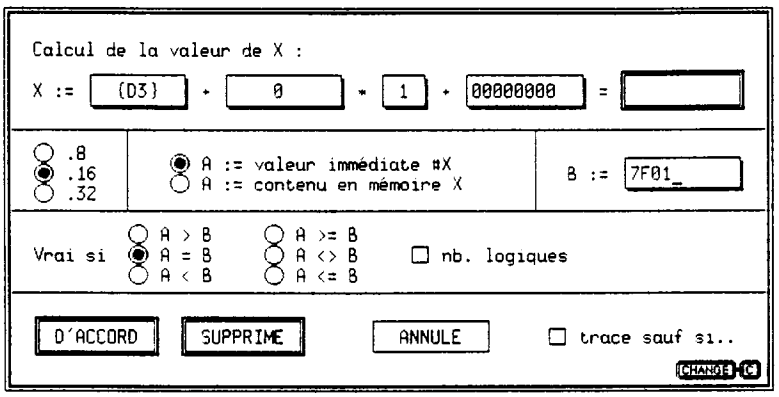


Fig. 2 : Dialogue de définition d'une condition.

Trucs et astuces

Il n'est normalement pas possible de passer en mode trace dans un appel système (*NTREL, FOS, LIB*), mais cela est possible en allumant soi-même le bit **T1** ou **T0** de **SF** (par un clic souris).

Il est possible d'attendre qu'une instruction donnée débute en tirant parti des conditions de trace. Pour ne tracer que les appels LIB, il suffit de créer une condition telle que :

X := {PC} et mode .16

A := #X, B := 4E46

Vrai si A = B

Lors d'un débordement de mémoire, il est possible de trouver le **PC** où a été fait le **GETMEM** correspondant à la tranche mémoire. Il suffit de presser [HELP] lorsque le message *débordement de mémoire* est affiché.

Remarques

Le contenu des registres et des fanions de **SF** reflètent l'état du processeur **après** l'exécution de l'instruction signalée. Dans le cas d'appels système, le **TRAP** a déjà été exécuté au moment où il est signalé, mais la routine système n'a pas encore été exécutée.

Si une erreur fatale survient dans une zone d'exclusion du pilote écran, **D.Bug** est impuissant.

Une fonction spéciale accessible par [PROGRA] [1] permet d'inspecter l'état des processus en cours de déverminage. N'utilisez les fonctions [DEBLOQUE] et [TRACE] qu'avec prudence !

Et surtout, n'hésitez pas à faire part de vos remarques et de vos suggestions.