

UTILISATION DES NOUVEAUX GENERATEURS DE FONCTION

H. Van der Beken

Cette note décrit les possibilités de manipulation des nouveaux générateurs de fonction. Le dernier paragraphe est destiné aux personnes intéressées aux détails de la structure du système. Une connaissance préliminaire des terminaux utilisés facilite la compréhension de certains points. Pour cela veuillez consulter les références ^{1,2}).

1. ACCES AUX PROGRAMMES "ON-LINE"

Tous les programmes peuvent être appelés depuis le superset ISAAC-FG. La syntaxe utilisée est équivalente à celle des autres supersets. Certains programmes nécessitent de nombreux accès aux disques et peuvent être plus ou moins lents si d'autres utilisateurs emploient aussi le disque No. 2.

2. DESCRIPTION DES DIFFERENTS APPELS

Il y a actuellement 63 fonctions et 63 terminaux définis dans le système. En plus, une fonction spéciale est appelée "OFLIN" et un terminal est appelé "DISPL" (display).

Une fonction peut être utilisée par plus d'un terminal, mais ne doit pas être utilisée par plus de 8 terminaux. Si cette règle n'est pas respectée, les modifications "on line" ne seront effectives que sur 8 terminaux.

2.1 CALL LIST (NAME, X)

Cet appel permet d'obtenir des informations sur un terminal ou sur une fonction.

Si NAME est un nom de fonction, il faut que X = F (Fonction). Dans ce cas on affiche sur l'écran du PDS1 les caractéristiques de la fonction spécifiée et la liste (au maximum 8) des terminaux qui l'utilisent. Les caractéristiques de la fonction peuvent être changées par le programme UPDAT. Un exemple est donné en Annexe 1.

Si NAME est un nom de terminal, il faut que X = T (Terminal). Dans ce cas on affiche sur le PDS-1 les caractéristiques du terminal (voir Annexe II), ainsi que le nom de la fonction utilisée par ce terminal et ses caractéristiques.

Si NAME = ALL, X est une lettre quelconque, CALL LIST (ALL, B) par exemple. On imprime alors sur le line printer la liste complète des terminaux, des fonctions qu'ils utilisent et leurs caractéristiques, ainsi que la liste complète des fonctions, de leurs caractéristiques et des terminaux auxquels elles sont envoyées.

2.2 CALL UPDAT (NAME F, ONLIN/OFLIN)

En général on se borne en opération à modifier une fonction déjà existante, cependant, spécialement au début de l'exploitation, on peut vouloir créer une fonction ou changer complètement ses caractéristiques. UPDAT permet de le faire soit OFF-LINE (OFLIN) soit ON-LINE (ONLIN). La différence entre ces deux modes est la suivante:

- en mode ONLIN à la fin du programme UPDAT, la fonction créée ou modifiée est envoyée à tous les terminaux qui utilisaient NAMEF (8 maximum), et les données sont mises en mémoire sur le disque

dans les "Current Control Values" de la fonction NAMEF (cf AUTO, BAS, BUFF);

- en mode OFLIN à la fin du programme, la fonction est mise en mémoire sous le nom de "OFLIN" (cf RENAM).

Dans les deux modes les résultats sont envoyés au fur et à mesure dans le terminal "DISPL" qui permet de visualiser la fonction sur un oscilloscope. Il faut que le nom de la fonction existe déjà (cf programmes auxiliaires).

2.2.1 Spécifications générales

La première phase du programme UPDAT permet de spécifier les caractéristiques générales de la fonction par la technique CHECK-LIST. Ces caractéristiques sont N MAX = Nombre de vecteurs maximum utilisables pour cette fonction; ce nombre doit être inférieur ou égal à 31. Chaque fonction peut avoir en moyenne 17 vecteurs; si trop de fonctions utilisent plus de 17 vecteurs, il se peut que l'on ne puisse augmenter N MAX. Dans ce cas il faut appeler CALL LIST (ALL, X) et faire le ménage.

Si ce nombre NMAX est changé, le programme doit réorganiser les données sur le disque et ceci peut être assez long spécialement si d'autres utilisateurs emploient le disque No. 2. Lorsque cette phase est terminée, le programme réaffiche les nouvelles caractéristiques.

UPDAT affiche aussi (voir Annexe III) la correspondance entre les "bits" et les unités physiques, par exemple = 1 BIT = 5 : MV. Ceci permet à l'utilisateur de travailler par la suite en unités physiques. Il peut changer par CHECK-LIST les spécifications ci-dessus en = 2 BIT = 7 : MA.

UPDAT permet aussi de changer l'amplitude initiale de la fonction qui est l'amplitude en sortie du générateur entre la fin du dernier vecteur jusqu'au START de la fonction. Cette amplitude est à spécifier en unité physique REST AMPLITUDE = 0 par exemple.

L'utilisateur peut aussi spécifier s'il désire que sa fonction soit Bipolaire ou Unipolaire (cf. 1,2)).

BIPOLAR: Y par exemple.

On peut enfin spécifier l'horloge utilisée en mettant 0, 1, 2 ou 3 à la question

USER TIMING = 3 par exemple.

0	signifie	train B
1	"	train 10 μ s
2	"	train 2 μ s
3	"	train 100 μ s

Ayant modifié, ou non, les caractéristiques générales, la touche XMIT les envoie à l'ordinateur, qui après vérification les envoie au terminal DISPLAY et l'on passe à la phase suivante.

2.2.2 Spécification des vecteurs

Si la phase précédente s'est bien déroulée, on peut spécifier les caractéristiques de chacun des vecteurs. Pour faciliter ce travail le vecteur dont les caractéristiques sont affichées sur l'écran est encadré sur l'oscilloscope par deux points surintensifiés. (Voir annexe IV comme exemple de spécification).

On peut spécifier si l'on désire créer un vecteur ou non

CREAT : Y par exemple

dans ce cas le vecteur créé est inséré juste avant le vecteur intensifié. Les caractéristiques de ces vecteurs sont spécifiées comme pour une modification.

On peut aussi décider de supprimer le vecteur indiqué

ERASE : Y.

Si l'on a CREATE : N, ERASE : N, le vecteur indiqué peut être modifié.

Avant d'entrer dans les détails de la modification, il est bon d'indiquer l'utilisation de

JUMP AT VECTOR = X.

X est normalement le N⁰ du vecteur suivant si l'on met X = 0 on re-

vient à la phase 1, et toutes les spécifications précédentes sont oubliées. Si l'on veut, on peut sauter directement à un vecteur situé plus loin dans la fonction, mais pour l'instant on ne peut pas revenir en arrière. X doit toujours être plus grand que le N du vecteur ou égal à 0.

On peut pour chaque vecteur spécifier son mode "A"mplitude ou "V"ector (cf 1,2)
MODE : V par exemple.

On peut spécifier un arrêt sur le vecteur; il faut alors que le générateur reçoive une autre impulsion de START pour continuer si l'on spécifie
INTERNAL STOP : N
la fonction continue d'elle même.

On peut spécifier la longueur qui est le nombre d'impulsions de l'horloge choisie
LENGTH = 720 par exemple.

La longueur maximum est 16383, on peut spécifier si ce vecteur est le dernier de la fonction. Dans ce cas on doit être en mode amplitude

LAST VECTOR : N par exemple.

On peut par les vecteurs en mode amplitude décider de diviser l'horloge interne ou externe par 10 en indiquant

CLOCK DIVIDER BY 10 : Y.

Pour les vecteurs en mode amplitude il est possible pour certains générateurs de changer d'horloge en cours de fonction par la spécification du nouveau timing. Si initialement on avait le train B, on peut passer au train 0,1 ms en spécifiant

TIMING = 3.

Enfin on peut spécifier l'amplitude qui est exprimée en unités physiques. Cette amplitude est la valeur obtenue à la fin du vecteur en mode vecteur. En mode amplitude c'est la valeur obtenue pendant toute la durée du vecteur.

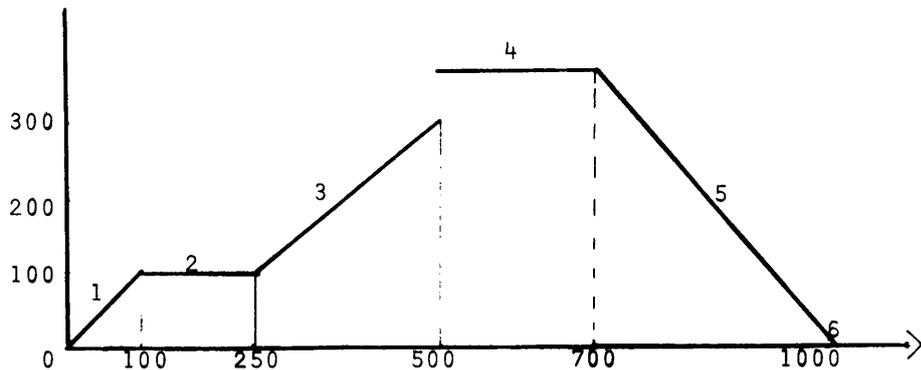
En cas d'erreur, les nouvelles spécifications sont oubliées et l'on réaffiche les anciennes. Les messages d'erreurs apparaissent en bas de l'écran (voir aussi Annexe V).

Le programme se termine lorsque l'on spécifie le dernier vecteur et effectue les opérations indiquées plus haut suivant le mode choisi.

CALL UPDAT (OFLIN, OFLIN) ou CALL UPDAT (OFLIN, ONLIN) est parfaitement légal.

On peut aussi sortir en utilisant "Control A", mais ceci fait (pour l'instant) perdre le contrôle du clavier à ISAAC et il faut faire un deuxième "Control A" pour sortir d'ISAAC et l'appeler à nouveau.

Supposons que l'on veuille créer la fonction ci-dessous



on spécifiera les vecteurs de la façon suivante:

VECTEUR 1 CREATE:N ERASE:N JUMP AT VECTOR = 2
MODE: V INTERNAL STOP:N LENGTH = 100 LAST VECTOR: N
CLOCK DIVIDED BY 10: TIMING = AMPLITUDE = 100 MV

VECTEUR 2 CREATE:N ERASE:N JUMP AT VECTOR = 3
MODE:V INTERNAL STOP:N LENGTH = 150 LAST VECTOR: N
CLOCK DIVIDED BY 10: TIMING = AMPLITUDE = 100 mV

VECTEUR 3 CREATE:N ERASE:N JUMP AT VECTOR = 4
MODE:V INTERNAL STOP:N LENGTH = 250 LAST VECTOR: N
CLOCK DIVIDED BY 10: TIMING = AMPLITUDE = 300

VECTEUR 4 CREATE:N ERASE:N JUMP AT VECTOR = 5
MODE:A INTERNAL STOP:N LENGTH = 200 LAST VECTOR:N
CLOCK DIVIDED BY 10:N TIMING =3 AMPLITUDE = 350

VECTEUR 5 CREATE:N ERASE:N JUMP AT VECTOR = 6
MODE:V INTERNAL STOP:N LENGTH = 300 LAST VECTOR: N
CLOCK DIVIDED BY 10: TIMING = AMPLITUDE = 0.

VECTEUR 6 CREATE:N ERASE:N JUMP AT VECTOR = 7
MODE:A INTERNAL STOP:N LENGTH = 10 LAST VECTOR:Y
CLOCK DIVIDED BY 10:N TIMING = 3 AMPLITUDE = 0.

Le dernier vecteur, ici le vecteur 6, n'est en fait pas utilisé, il sert seulement à indiquer la fin de la fonction.

Si la fonction initiale avait moins de 6 vecteurs, il aurait fallu faire des créations, mais il est nécessaire que NMAX soit au moins égal à 6.

2.3 CALL RENAM (NAME 1, NAME 2)

Cet appel est principalement destiné à être utilisé après CALL UPDAT (NAMEF, OFLIN) si l'on est satisfait des résultats obtenus et que l'on veut utiliser la fonction modifiée. Dans ce cas on appelle CALL RENAM (OFLIN, NAMEF).

Les données qui avaient été mises en mémoire dans la fonction spéciale OFLIN sont écrites dans la fonction NAMEF et envoyées à tous les terminaux utilisant NAMEF.

Il est parfaitement possible de faire cette opération pour n'importe quelle fonction mais en sachant que les "Current Control Values" de NAME2 sont détruites et remplacées par les C.C.V. de NAME1.

Cette opération peut être impossible si le nombre de vecteurs maximum prévu pour NAME2 est plus petit que le nombre de vecteurs utilisé par NAME1. Dans ce cas il faut changer N2MAX avec UPDAT.

2.4 CALL SENDF (NAME F, NAME T)

Cet ordre envoie la fonction NAMEF au terminal NAMET et spécifie que NAMET utilisera dorénavant NAMEF. L'ancienne fraction utilisée par

NAMET est oubliée.

2.5 CALL INIT (NAME)

Si NAME est le nom d'une fonction, les C.C.V. de cette fonction sont envoyées à tous les terminaux l'utilisant. C'est en quelque sorte une Initialisation qui peut être fort utile après une coupure de réseau.

Si NAME est ALL, l'action décrite plus haut est effectuée pour tous les terminaux, ce qui évidemment prend un certain temps, les contrôles étant faits à la fin du cycle.

2.6 REPEAT MODIF (NAMEF, ONLIN/OFLIN)

Cet ordre permet de modifier 2 vecteurs consécutifs de la fonction NAMEF.

Restriction: les 2 vecteurs doivent être en mode vecteur.

Si l'on a spécifié ONLIN, les modifications sont envoyées à tous les cycles à tous les terminaux utilisant NAMEF et au DISPLAY. En sortant par U MODIF, les résultats sont mis en mémoire dans les C.C.V. de NAMEF.

Si l'on spécifie OFLIN, les modifications ne sont envoyées qu'au DISPL et en sortant par U MODIF, les résultats sont mis en mémoire dans les C.C.V. de OFLIN. On peut utiliser ensuite RENAM.

CALL MODIF (NAMEF, ONLIN/OFLIN) ne doit pas être utilisé normalement.

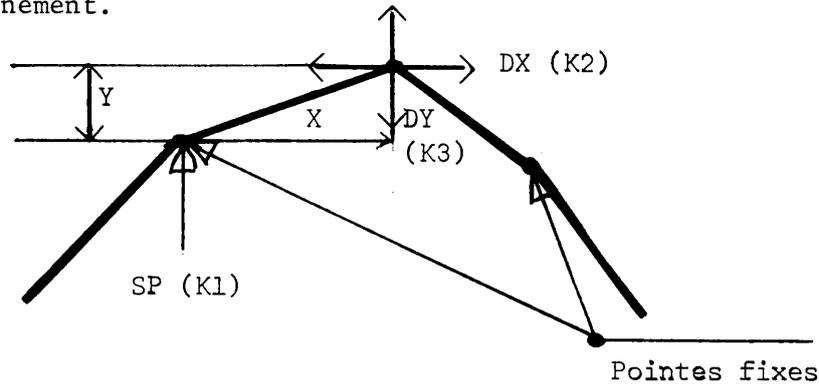
Dès que le programme est actif, les shaft encoders sont automatiquement utilisés (attendre les affichages).

K1 choisit SP (Starting Point) qui est le premier point fixe et qui est surintensifié.

Le deuxième point fixe est automatiquement fixé à SP + 2 et intensifié.

K2 permet de modifier de DX la longueur X du premier vecteur. DX et X sont affichés. L'amplitude ne change pas.

K3 permet de modifier de DY la variation d'amplitude Y, créée par le premier vecteur. DY et Y sont affichés. Le schéma ci-dessous explique le fonctionnement.



Le point situé entre les deux points fixes est intensifié et ses coordonnées sont liées à K2 et K3.

Si X devient trop grand et si le nombre maximum de vecteur le permet, un vecteur est créé automatiquement en coupant le premier vecteur en deux.

Si la longueur du premier ou du deuxième vecteur devient trop petite, un message d'erreur apparaît.

Il est recommandé d'utiliser K1 sur la vitesse "NORMAL".

SP, DY, DX étant des variables software, elles peuvent aussi être manipulées par des ordres du genre

```
SET SP = 3          ou
SET DY UP 10
```

K1 doit être tel que $0 < SP < N-3$.

2.7 REPEAT MULTP (NAMEF, ONLIN/OFLIN)

Cet ordre permet de faire une homothétie en amplitude de la fonction NAMEF sur une longueur choisie.

Restriction : Le premier et le dernier vecteur modifiés doivent être en mode vecteur.

Les spécifications OFLIN ou ONLIN ont le même effet que pour R MODIF. Normalement il ne faut pas utiliser CALL MULTP.

Dès que le programme est actif, les shaft encoders sont automatiquement utilisés (attendre l'affichage).

K1 choisit SP (Starting Point) premier point fixe intensifié.

K4 choisit LG (Longueur), le dernier point intensifié est fixe et est déterminé par

$$SP + 2 + LG.$$

Il est recommandé d'utiliser K1 et K4 sur la vitesse "NORMAL" et l'on doit choisir K1 et K4 tels que

$$\phi < SP < N-3$$

$$SP + LG < N-3$$

$$\phi < LG.$$

Les deux points intensifiés sont des points fixes et K3 choisit en millième le coefficient d'homothétie DY qui est affiché, ainsi que SP et LG.

Ainsi si l'on choisit DY = - 800, la partie de la courbe comprise entre les 2 points fixes est multipliée par 0,2 ou diminuée de 80%. Le coefficient est calculé ainsi

$$K = \frac{1000 + DY}{1000}.$$

Les valeurs SP, LG, DY sont des variables software qui peuvent être manipulées par les ordres ordinaires.

On sort du programme par U MULTP et les résultats sont mis en mémoire dans les C.C.V. de NAMEF ou de OFLIN suivant le choix de ONLIN ou OFLIN respectivement. Si l'option était OFLIN, on peut ensuite utiliser RENAM.

Remarque : Pendant l'exécution de MODIF ou MULTP, certains appels ne sont pas possibles (voir J.P. Potier).

2.8 CALL INVER (NAMEF)

Cet ordre inverse la polarité de la fonction NAMEF et envoie la fonction inversée à tous les terminaux utilisant NAMEF, puis met les résultats en mémoire dans les C.C.V. de NAMEF.

2.9 CALL AUTO (NAMEF/ALL)

Cet ordre transfère les C.C.V. de NAMEF (ou de toutes les fonctions si l'on spécifie ALL) dans les "Reference Values".

2.10 CALL BAS (NAMEF/ALL)

Transfère les C.C.V. de NAMEF (ou de toutes les fonctions) dans le Buffer et utilise les valeurs de Reference comme nouvelles valeurs de contrôle.

2.11 CALL BUFF (NAMEF/ALL)

Utilise les valeurs du Buffer comme nouvelles valeurs de contrôle pour la fonction NAMEF ou pour toutes les fonctions si ALL était spécifié.

3. PROGRAMMES AUXILIAIRES

3.1 Modifications spéciales

Un programme Off-Line permet de changer les noms des fonctions, les noms des terminaux, les adresses de contrôle ou d'acquisition des terminaux. Ce programme sera décrit dans une note à paraître de Mme L. Merard.

3.2 Transfert sur ou depuis des cartes

Un programme Off-Line permet de transférer sur cartes certaines fonctions ou de remettre en mémoire certaines fonctions depuis des cartes. Ce programme sera décrit dans une note à paraître de E. Ratcliff.

4. ORGANISATION DES FICHIERS ET REPERTOIRES

L'ensemble des données relatives aux générateurs de fonctions est divisé en 2 répertoires et 3 fichiers.

Les 3 fichiers contiennent respectivement les valeurs courantes de contrôle (C.C.V.), les valeurs de référence et les valeurs de "Buffer". Leur structure est identique si la Fonction F a un nombre maximum de vecteurs de NMAX. La longueur réservée dans chacun des 3 fichiers est de $2*NMAX + 1$, car chaque vecteur nécessite 2 mots et une fonction a besoin d'un mot supplémentaire qui définit l'amplitude au repos, le fait qu'elle soit bipolaire ou non, le fait qu'elle soit répétitive ou non et l'horloge utilisée.

Un répertoire contient toutes les indications relatives aux terminaux (Terminal Reference File, T.R.F.). Il est actuellement constitué de 64 blocs de 7 mots (voir contenu sur schéma).

L'autre répertoire contient les indications relatives aux fonctions (Function Status File, F.S.F.) et comporte actuellement 64 blocs de 8 mots (voir contenu sur schéma).

L'ensemble des répertoires et fichiers occupe actuellement 3 cylindres sur le disque no.2.

La plupart des indications utiles est décrite dans un "MACRO" nommé "EQUFG".

L'organisation des fichiers est schématisée sur un schéma donné en Annexe VI.

Une description abrégée des ordres est donnée en Annexe VII.

Références

- 1) Propositions pour la réalisation d'un générateur de fonctions banalisé en terminal STAR, MPS/CO/Note 73-8.
- 2) Générateur de fonctions banalisé en terminal STAR, MPS/CO/Note 73-18.

Distribution

MPS Group Leaders

PS Programmers

Members CUC

MST

PSS + EiC

CCI - computer section

MCR

Floor operators

E. Asséo

G. Daems

R. Debordes

B. Kuiper

G.L. Munday

F. Rohner

****PSI PDS1 HARD COPY****

74/ 7/ 16 14H 50' 20''

ISAAC SUPERSET 5 - F.S.

FUNCTION FFFF1 1 BIT = 5 MV POINTER 0 N = 10 NMAX= 20
USED BY TERMINAL TTT:0 DUMMY DUMMY DUMMY DUMMY DUMMY DUMMY DUMMY

C LIST(FFFF1,F)
C COPY

RESTART
RESTART
IBM NO RESP. 2

Annex I

****PSI PDS1 HARD COPY****

74/ 7/ 16 14H 51' 32''

ISAAC SUPERSET 5 - F.G.

TERMINAL TTTT1 HAS STARC /2531 STARA /0000 GROUPE 0 AND IS USING
FUNCTION FTST3 1 BIT = 5 MV POINTER 811 N = 9 NMAX=

C LIST(TTTT1,T)
C COPY

RESTART
RESTART
RE-TYPE *REPEAT* INSTRUCTION (IF ANY)

Annex II

****PSI PDS1 HARD COPY****

74/ 7/ 16 15H 9' 25''

ISAAC SUPERSET 5 - F.S.

FUNCTION FTST3 USES 9 VECTORS HAS NMAX = 21 = 1 BIT = 6 . 7

REST AMPLITUDE = 0 BIPOLAR . Y USER TIMING = 3
TIMING 0-R, 1-10US, 2-2US, 3-1MS

IF YOU CHANGE NMAX A REPACKING WILL OCCUR AND IT TAKES 2 MINUTES

C UPDAT(FTST3,ONLIN)

RESTART
RESTART
RESTART

Annex III

PSI PDS1 HARD COPY

74/ 7/ 16 15H 10' 56''

ISAAC SUPERSET 5 - F.G.

VECTOR 4 CREATE .N ERASE .N JUMP AT VECTOR = 5
MODE .V INTERNAL STOP .N LENGTH = 300 LAST VECTOR .
CLOCK DIVIDED BY 10 . TIMING = AMPLITUDE = 5540 MV

IF YOU CHANGE NMAX A REPACKING WILL OCCUR AND IT TAKES 2 MINUTES

C UPDAT(FTST3,ONLIN)

RESTART
RESTART
RESTART

Annex IV

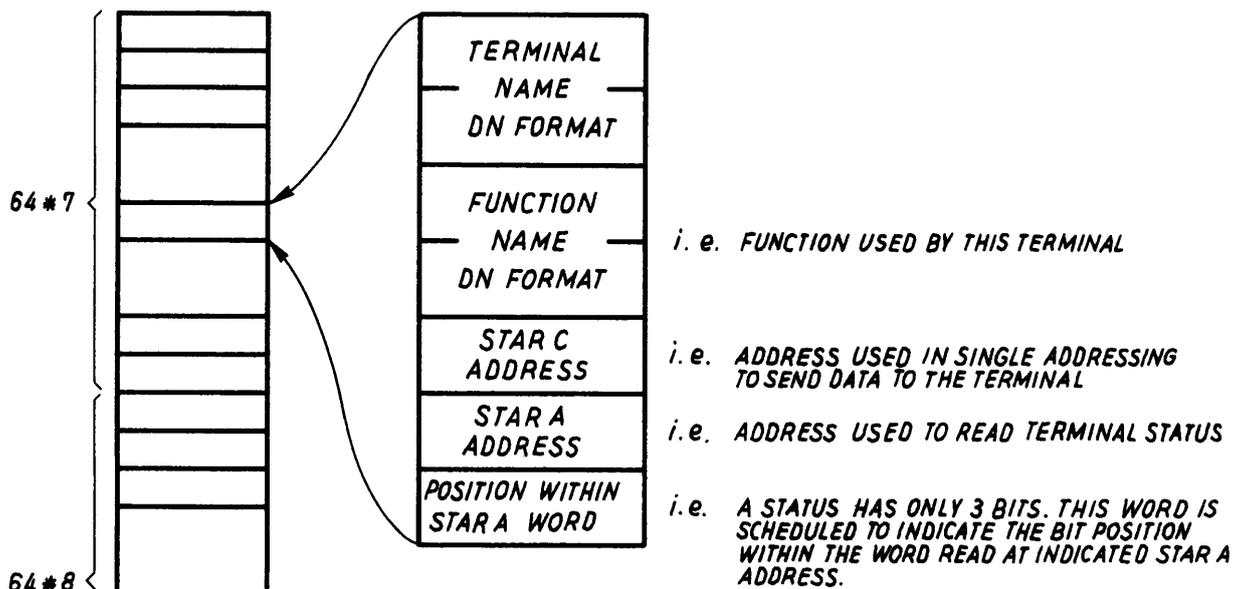
ANNEXE V

Code d'erreur pour Build

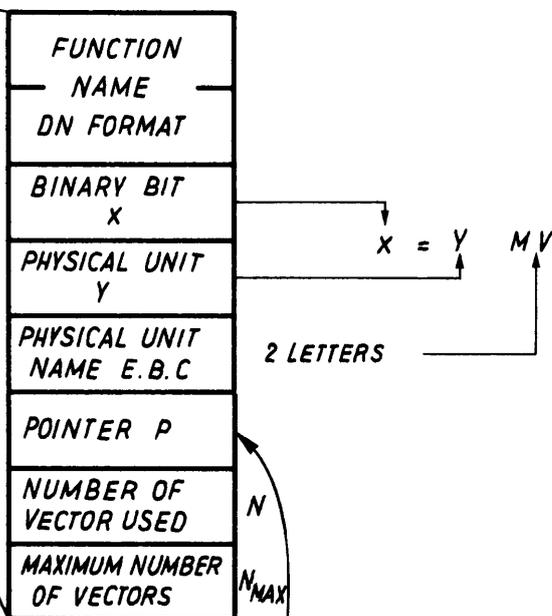
- 1 ni amplitude ni vecteur mode
- 2 internal stop ni Y ni N
- 3 longueur mauvaise
- 4 last vecteur ni Y ni N
- 5 clock divided ni Y ni N
- 6 mauvais choix du timing
- 7 amplitude ou trop grande, ou ne pouvant être atteinte dans le temps spécifié.

FILES AND DIRECTORIES LAYOUT ON DISK DRIVE # 2

T.R.F. FILE 177



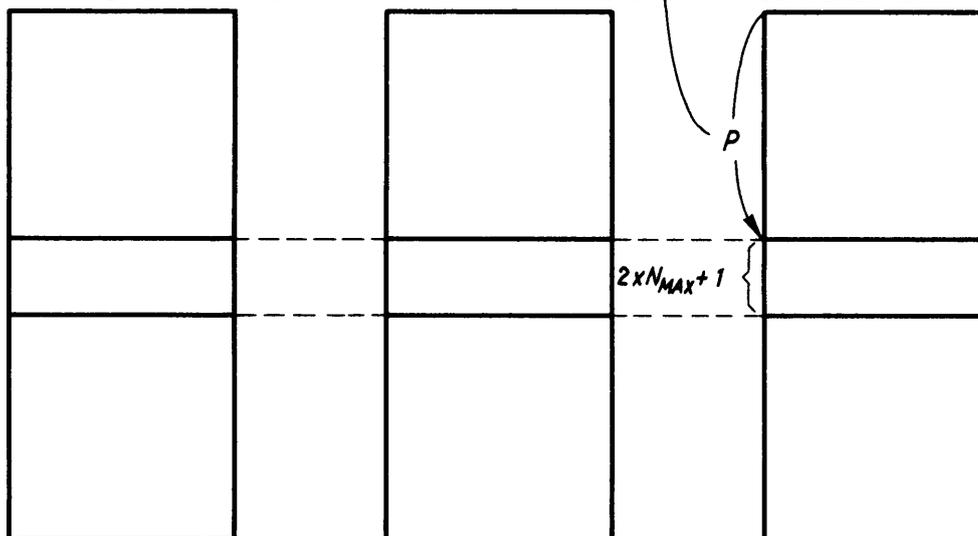
F.S.F.



REFERENCE DATA FILE 186

C.C.V FILE 187

BUFFERED DATA FILE 194



ANNEXE VII AIDE MEMOIRE

1. LIST

CALL LIST (NAMEF, F)

Affiche caractéristiques fonction NAMEF

CALL LIST (NAMET, T)

Affiche caractéristiques terminal NAMET

CALL LIST (ALL, X)

Imprime sur L.P. toutes les informations.

2. UPDAT

CALL UPDAT (NAMEF, OFLIN)

Permet de modifier entièrement ou de créer une fonction qui est ensuite mise en mémoire sous le nom "OFLIN" (cf RENAM).

CALL UPDAT (NAMEF, ONLIN)

Permet de modifier entièrement ou de créer une fonction qui est ensuite envoyée à tous les terminaux qui utilisaient NAMEF et mise en mémoire dans les "current control values".

3. RENAM

En général

CALL RENAM (OFLIN, NAMEF)

après résultats satisfaisants de

CALL UPDAT (NAMEF, OFLIN)

il est possible de faire

CALL RENAM (NAME1, NAME2).

Les C.C.V. de la fonction NAME1 sont écrites dans les C.C.V. de NAME2 et envoyées aux terminaux utilisant NAME2.

4. SENDF

CALL SENDF (NAMEF, NAMET)

envoie fonction NAMEF au terminal NAMET.

5. INIT

CALL INIT (NAMEF)

initialise tous les terminaux utilisant NAMEF (envoie des C.C.V.)

CALL INIT (ALL)

initialise toutes les fonctions.

6. MODIF

REPEAT MODIF (NAMEF, ONLIN)

Modifie deux vecteurs consécutifs (choisis par K1) en amplitude (K3) et durée (K2).

Les modifications sont envoyées au DISPLAY et aux terminaux.

U MODIF termine le programme et met en mémoire les résultats dans les C.C.V.

REPEAT MODIF (NAMEF, OFLIN)

Même action mais les modifications ne sont envoyées qu'au DISPLAY et les résultats mis en mémoire dans les C.C.V. de OFLIN (on peut ensuite utiliser RENAM).

7. MULTP

REPEAT MULTP (NAMEF, ONLIN)

Multiplie l'amplitude de la courbe à partir d'un point fixe SP (choisi par K1) et sur une longueur LG (choisie par K4). Le coefficient est contrôlé par $K3 = DY$ et vaut $K = \frac{1000 + DY}{1000}$.

Les résultats sont envoyés au DISPLAY et aux terminaux utilisant NAMEF. En sortant par U MULTP les résultats sont mis en mémoire dans les C.C.V. de NAMEF.

REPEAT MULTP (NAMEF, OFLIN)

Permet les mêmes manipulations, mais les modifications n'apparaissent que sur le DISPLAY et les résultats vont (en sortant par U MODIF) dans les C.C.V. de OFLIN. RENAM peut alors être utilisé.

8. INVER

CALL INVER (NAMEF)

Inverse la polarité de NAMEF dans tous les terminaux et dans les C.C.V. de NAMEF.

9. AUTO

CALL AUTO (NAMEF)

Transfere les C.C.V. de NAMEF en valeurs de Référence.

CALL AUTO (ALL)

Transfere toutes les C.C.V. en valeurs de Référence.

10. BAS

CALL BAS (NAMEF)

Transfere les C.C.V. de NAMEF dans le Buffer et utilise les valeurs de référence comme nouvelles valeurs de contrôle.

CALL BAS (ALL)

Action identique à CALL BAS (NAMEF), mais pour toutes les fonctions.

11. BUFF

CALL BUFF (NAMEF)

Reprend les valeurs du Buffer comme nouvelles valeurs de contrôle pour la fonction NAMEF.

CALL BUFF (ALL)

Reprend les valeurs du Buffer comme nouvelles valeurs de contrôle pour toutes les fonctions.