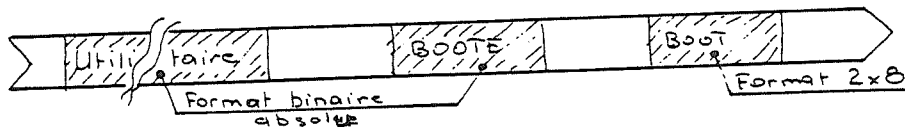


Le Bootstrapp se trouve en tête de chaque bande d'utilitaire.



Fonctionnement

- L'action sur LOAD charge les 40 mots de BOOT
- L'action sur RUN lance l'exécution de BOOT
- BOOT charge BOOTE et lui donne la main
- BOOTE calcule la capacité mémoire, initialise les mémoires et registres suivants:
 - ASTF, ESTF et RSTF
 - NS = 0
 - SLØ = 0
 - SLE = 'OFFF (64K maxi possible)
 - MAXMEM = capacité mémoire : Nb de Kmots - 1 en Format SLØ
 - FIRSTM = '38
 - Y = Ø d'implantation de l'utilitaire.
- puis rend la main à BOOT
- BOOT charge l'utilitaire et lui donne la main.
- BOOT s'implante de '0038 à '005E, BOOTE s'implante à la suite de '005F à '00B5.

Utilisation: chargement d'un utilitaire par un TTY.

- Positionner la bande sur le lecteur du TTY
- Appuyer sur le bouton "MOTEUR" du TTY pour mettre le moteur sous tension
- Faire INI, puis LOAD, puis RUN sur le PUC
- Faire START sur le levier de lecteur TTY
- Si on ne peut pas laisser le bouton "MARCHE" du TTY enclenché, il faut appuyer de temps en temps (toutes les 30 s) sur ce bouton pour réarmer le monostable de mise sous-tension.
- Après la frappe de "OK?" sur le TTY arrêter manuellement le lecteur.

Bootstrap papier



SPS 5

N° Document

71-F7 31MS

Date

547

Page

0. 2.1



Le Boot implante les utilitaires en fond de mémoire limité à 32K.

Si on désire charger l'utilitaire avant le fond de la mémoire ; il faut :

- Faire LOAD
- Puis INI
- Charger par le POP la mémoire ADIMP (0039) par l'adresse désirée (par module de 1K).
- Faire RUN.

1- FORMAT BINAIRE GENERAL

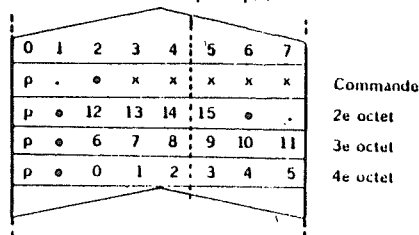
Une bande binaire ne contient que des caractères pairs, elle peut contenir les caractères **PERFO ON** ou **PERFO OFF** ; l'information utile se présente sous forme de quadruplets :

p = bit de parité de l'octet

o = bit à 1

. = bit à 0

0, 1, 2, ... 15-bits information



Dans le cas de la commande «commentaire» ou «nom de section» les 3 derniers caractères du quadruplet sont 3 caractères du commentaire ou du nom de la section.

2- LISTE DES COMMANDES

* A0	charger en
* 24	charger à la suite sans translater
* 28	adresse de lancement
* AC	repeat load
* 30	fin de bande(checksum)
* B4	commentaire
* 22	suite de commande
* A6	charger à la suite en translatant
* AA	remonter une chaîne d'adresses
* 2E	ajouter un déplacement sur adresse
* B2	PST
* 36	nom de section
* 3A	définition de référence externe
* BE	définition de référence avant relative
* A3	module translatable

3- FORMAT BOOTSTRAPABLE

Une bande bootstrapable ne contient ni **PERFO ON** ni **PERFO OFF** ; elle doit avoir la consigne suivante :

- le quadruplet : «charger en » adresse début
- quadruplets suivants : «charger à la suite »
- avant dernier quadruplet : «adresse de lancement»
- dernier quadruplet : «checksum»

4- FORMAT BINAIRE ABSOLU

Une telle bande peut contenir des **PERFO ON** ou **PERFO OFF** ; elle comporte, par rapport au format bootstrapable, les commandes supplémentaires suivantes :

- «repeat load» : répéter n fois la commande «charger à la suite » qui suit
- «commentaire»

L'adresse de chargement du programme est précisée dans le premier quadruplet.

5- FORMAT BINAIRE TRANSLATABLE

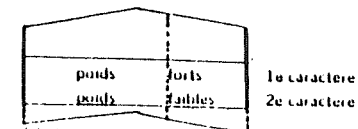
Une bande binaire translatable peut contenir des **PERFO ON** et **PERFO OFF** et reconnaît toutes les commandes de la liste du paragraphe 2 vue précédemment.

Une telle bande commence nécessairement par une commande «module translatable»

6- BOOTSTRAP

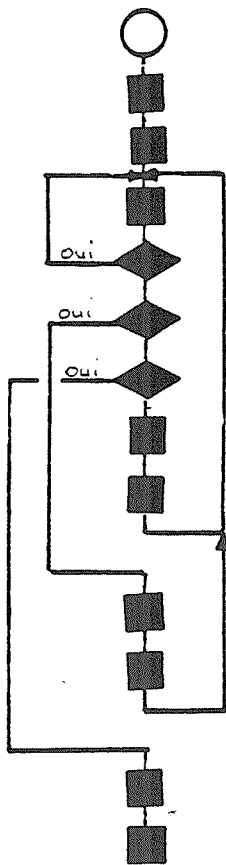
Il est lu en format 2 x 8 et implanté à l'adresse hexadécimale '38.

Format



La mémoire 'B (FIRSTM) est initialisée à l'adresse de la première mémoire banalisée ('38) et gérée par les chaînes qui y mémorisent l'adresse de fin du plus haut chargement

Organigramme de BOOT



Depart en '38 après action sur RUN.

Lecture du sélecteur de BOOTSTRAPP

Si lecteur de bande : modifier de l'adresse périph.

Entrée d'un quadruplet (4 octets) (A):= com^{de}, (B):= info.

Est-ce "check-sum"?

Est-ce "Charger à la suite"?

Est-ce "à de lancement"?

C'est donc "charger en"

Init B, L et W par \bar{a} d'implantation.

On range B en C par rapport à W

+1 \rightarrow W.

On range B (\bar{a} de lancement) dans INI ('0008)

Branchement au contenu de INI.

↳ Branchement à BOOTE

↳ " " " " à l'utilitaire chargé.

Organigramme de BOOTE



Initialisation des mémoires : PSTHA, NS
des registres : SLØ et SLE
des Files ASTF, ESTF et RSTF du scheduler.

Calcul de la capacité mémoire.

MAXMEM ('000A) = nb de K mots de mémoire en format SLØ

FIRSTM ('000B) = '0038

(Y) = \bar{a} d'implantation de l'utilitaire.

Branchement à BOOT pour rentrer l'utilitaire.

Bull



SPS 5

Bootstrap papier

N° Document

71 F7 31MS

Date

547

Page

0. 2.3

```

COMMON CBOOT
0000 NS: EQU $
0000 DZS 2
0002 PSTHA: EQU $
0002 DZS 4
0008 INI: EQU $
0008 DZS 2
000A MAXMEM: EQU $
000A DZS 1
000B FIRSTM: EQU $
000B DZS '15
0020 ASTF: EQU $
0020 DZS 8
0028 ESTF: EQU $
0028 DZS 8
0030 RSTF: EQU $
0030 DZS 8
003B 2004 WORD '2004 < JMP START
9 0000 ADIMP: WORD 0
A 17FB ENTPER: WORD '17FB
B 17FA ETAPER: WORD '17FA
003C BASEC: EQU $
PROG PBOOT
USE C:BASEC
003C 2FE4 START: LRP
D 1E19 RDSI
E 13E0 ANDI 'E0 <EST-CE LA TTY?
F 2605 JAE LEC-1 <OUI
0040 1008 LAI 8 <NON C'EST LE LR
1 4IFE STA ENTPER <ON CORRIGE L'ADRESSE
2 0802 ADRI 2:A
3 4IFF STA ETAPER
4 1600 LYI 0
5 1700 LEC: LBI 0 <LECTURE QUADRUPLET
6 47FE SIO ENTPER <RELANCE
7 47FF SIO ETAPER
0048 28CE TBT 14 <TEST BIT VAL
9 01FE JNC $-2
A 47FE SIO ENTPER <ACQUISITION
B 1512 CPI '12
C 06FA JE LEC+1
D 2A44 SCRD 6
E 21FB JAGE LEC+1 <OCTET SUIVANT
F 2908 SLRS 8 <A:COMMANDE B:INFO
0050 15B0 CPI 'B0 <ELIMINATION
1 06F4 JE LEC <DU CHECKSUM
2 15A4 CPI 'A4 <COMMANDE
3 0606 JE SUITE
4 0308 JG LANC <'2B:ADR LANCEMENT
5 2C19 ADR Y,B <A2:ADR IMPLANTATION
6 28CE LR B+W
7 28CD LR B+L
005B 20ED JMP LEC
9 CC00 SUITE: STB 0+W <CHARGER A LA SUITE
A 0E01 ADRI 1+W
B 20EA JMP LEC <QUADRUPLET SUIVANT
C 2C19 LANC: ADR Y,B <B=B+Y
D 4CCC STB INI
E 45CC BR INI <ADR DE LANCEMENT EN 'B
005F 0000 SLO: WORD 0
0060 0FFF SLE: WORD '0FFF
1 0FFF NE0FFF:WORD 'FFF <64 K-1 FORMAT SLO
2 03FF NE03FF:WORD '3FF <1 K-1
3 ABAB CSTE: WORD 'ABAB
4 0400 UNK: WORD '400 <1 K
5 F000 NEF000:WORD 'F000
6 1000 NB1000:WORD '1000 <64K FORMAT SLO
7 FFFF NBFFFF:WORD 'FFFF <64K -1
0068 07FF NE07FF:WORD '7FF <32K FORMAT SLO
9 003C TABLE: WORD '3C <PSTH 0
A 0000 WORD 0
B X00AD WORD SORTIE
C 8000 WORD '8000
D X0095 WORD SORMEM
E 0045 ADLEC: WORD '45
FROG FROOTE
USE C:BASEC
USE L:SLO
006F 9D0A DE00TE: LAD TABLE <ADR TABLE EN 2
0070 4DC6 STA PSTHA
1 30FF LAI -1
2 4DF4 STA RSTF
3 4DEC STA ESTF
4 4DE4 STA ASTF
5 5ED4 STZ NS
6 9000 LA SLO
7 9701 LR SLE
0078 1E1E INITSS:WDE <ECRITURE SLO SLE
9 9603 LY NRO3FF <Y=1K-1
A 1E1A AJOU: LAR <SAUVEGARDE MEMOIRE
B 28C1 LR A+B <DANS B
C 9004 LA CSTE
D 1E1D STAR <ESSAI D'ECRITURE
E 1E1A LAR <LECTURE
F 9504 CP CSTE <VERIFICATION
0080 0215 JNE SORMEM <SI DIF ALLER A SORMEM
1 28C8 LR B+A <RESTITUTION
2 1E1D STAR <MEMOIRE
3 28D8 LR Y+A
4 9508 CP NBFFFF
5 0604 JE MODSS <SI Y>64K-1 ALLER A MODC
6 8905 AD UNK <Y=1K
7 28C3 LR A+Y
0088 20F2 JMP AJOU
9 9001 MODSS: LA SLE
A 8907 AD NB1000 <SLE=SLE+64K
B 8D01 STA SLE
C 28C1 LR A+B
D 9000 LA SLO
E 9506 CP NEF000
F 0203 JNE $+3
0090 9008 LA NBFFFF
1 200D JMP MISMAL
2 8907 AD NB1000 <SLO=SLO+64K
3 8D00 STA SLO
4 20E4 JMP INITSS
5 28D8 SORMEM:LR Y+A <ON VIEN ICI APRES DEWORD
6 8805 SB UNK <Y=Y-1K
7 9508 CP NBFFFF
009F 204 JNE $+4
9 Y900 DC SLO
A 9000 LA SLO
B 2003 JMP MISMAL
C 2904 SLRS 4
D 8900 AD SLO
E 4DCE MISMAL:STA MAXMEM
F 50FD LA ADIMP <ADR DONNEE PAR L'UTILISATEUR
00A0 2207 JANE FONDE
1 50CE LA MAXMEM
2 2503 JAL SUP32K
3 9509 CP NE07FF
4 0702 JLE $+2
5 9007 SUP32K:LA NE07FF
6 2984 SLLS 4
7 290A FONDE: SLRS 10
00AB 298A SLLS 10
9 28C3 LR A+Y <Y=NB DE K-1
A 1038 LAI '38
B 4ICF STA FIRSTM <INIT FIRSTM
C 850F BR ADLEC <RETONNE LA MAIN A BOOT
D 970E SORTIE:LR TABLE+4<ADR SUI DANS B
E 8C0C STB TABLE+2<MODIF ADR RETOUR
F 1700 LBI 0 <RESTITUTION MODE MAITRE
00B0 2850 SBT 16
1 8C0D STB TABLE+3
2 37FF LBI -1
3 2810 RBT 16
4 2F49 XIMR B
5 1E00 ACC
END

```

Bootstrap papier



N° Document

Date

Page

71 F7 31MS

547

0. 2.4