

## 9/4.5.3

# Les instructions Basic PGCD, PPCM et FRAC

---

Votre AMSTRAD ne connaît pas encore ces instructions, mais il est possible de lui apprendre ces nouveaux mots, ainsi que la façon de les traiter.

Nous allons donc créer ce qui s'appelle des RSX (*ReSident eXtension*), dont vous avez le détail de la programmation en Partie 4, Chapitre 1.6.2, pages 1 à 35.

Cette création passe, hélas, par une phase de programmation en langage machine, mais nous avons pensé aux utilisateur de Basic, en leur proposant un chargeur Basic des codes.

Si nous savons maintenant résoudre le PGCD, le PPCM et simplifier les fractions par des algorithmes très simples, qui utilisent des opérations simples (addition, soustraction, multiplication et division). Mais il faut savoir que le Z80 n'est pas capable d'effectuer des opérations telles que la multiplication et la division.

Soit, nous allons lui apprendre.

### LA MULTIPLICATION EN LANGAGE MACHINE

Nous allons ici nous intéresser uniquement à la multiplication de deux nombres entiers. Le Z80 étant un microprocesseur 8 bits, nous pousserons tout de même notre effort, pour effectuer une multiplication sur 16 bits grâce aux registres doubles **BC**, **DE**, **HL**.

Etudions d'abord la procédure manuelle utilisée pour effectuer une multiplication binaire (le binaire étant le code de base utilisé par le composant microprocesseur).

Nous pouvons observer en page suivante que l'on additionne en fait le multiplicande décalé d'une position vers la gauche pour chaque chiffre 1, selon sa position. Si l'on a un 0 (zéro), on effectue en fait un décalage et on ajoute 0.

1 0 0 1 1 0 1 1	← MULTIPLICANDE
*           1 1 0 1	← MULTIPLICATEUR
+       1 0 0 1 1 0 1 1	← 10011011 * 1
+       0 0 0 0 0 0 0 0 .	← 10011011 * 00
+   1 0 0 1 1 0 1 1 . .	← 10011011 * 100
+ 1 0 0 1 1 0 1 1 . . .	← 10011011 * 1000
= 1 1 1 1 1 0 1 1 1 1 1	← RESULTAT

Déroulement d'une multiplication binaire.

L'étude de cette figure, nous permettra de décrire l'ordinogramme ci-après.

	dividande		diviseur
	1 0 0 1 1 0 1 1 .		1 1
a	- 0 0		0 1 1 0 0 1 1
	1 0 0 1 1 0 1 1		a b c d e f g
b	- 1 1		quotient
	0 0 1 1 1 0 1 1		
c	- 1 1		
	0 0 1 0 1 1		
d	- 0 0		
	0 1 0 1 1		
e	- 0 0		
	1 0 1 1		
f	- 1 1		
	1 0 1		
g	- 1 1		
	reste 1 0		

Déroulement d'une division binaire.

La multiplication réalisée est effectuée sur les registres **DE** et **BC**, **DE** contenant le multiplicande, **BC** le multiplicateur ; le résultat final sera rendu dans le registre **HL**.

Chaque test sur les bits du multiplicateur est effectué par une rotation des registres **C** et **A** (on a transféré **C** dans **A** et **B** dans **C**) par l'intermédiaire de l'indicateur Carry.

Selon le résultat du test, on ajoute le multiplicande ou non à **HL**.

Cette opération doit être effectuée 16 fois (le nombre de bits traités), qui est comptée grâce au registre **B**.

Nous n'avons pas effectué le test de débordement du résultat (nombre supérieur à 23767), que vous pourrez effectuer pour, éventuellement, afficher un message d'erreur.

Il existe une autre façon, très rapide, de réaliser une multiplication : il s'agit de

l'algorithme de la multiplication à la russe, dont nous vous fournissons ci-dessous l'algorithme que vous pourrez étudier, et éventuellement appliquer.

— DEBUT

— SI  $X > Y$

— ALORS

— Mettre le contenu de X dans Y

— Mettre le contenu de Y dans X

— Mettre le contenu de Z dans Y

— FINSI

— Initialiser une variable RESULTAT à zéro

— TANT QUE  $X > 0$

— Mettre dans la variable MOITIE le résultat de la division entière du contenu de X / 2

— Affecter à RESTE le résultat de l'opération  $X - 2 * DEMI$

— SI  $RESTE = 0$

— ALORS

— Affecter à X la valeur de X / 2

— Affecter à Y la valeur de Y \* 2

— SINON

— Affecter à RESULT la valeur de RESULT + 1

— Donner à X la valeur de X - 1

— FIN SI

— FINTANTQUE

— CO

Le résultat se trouve maintenant dans RESULT

— FINCO

— FIN

Vous remarquerez que l'on utilise dans cet algorithme la multiplication et la division. Cela ne pose pas de problème, car ce ne sont qu'une multiplication par 2 (on décale alors le multiplicande une fois à gauche), et une division par 2 (on décale le multiplicande une fois à droite).

### LA DIVISION EN LANGAGE MACHINE

Tout mathématicien vous dira que la division est l'opération inverse de la multiplication, aussi, pour obtenir le résultat, au lieu de décaler vers la gauche, nous décalerons vers la droite l'opération à effectuer, opération qui était précédemment une addition, et qui se transformera ici en une soustraction.

En fait, avant de réaliser une soustraction, nous essayons cette soustraction le plus à gauche possible. Si cela n'est pas possible, on retire zéro (donc rien), et on se décale à droite sous le dividende. Cela pour les 16 bits à traiter.

La division que nous avons réalisée, et que vous trouverez dans le programme, est effectuée sous les registres **DE** et **BC**, **DE** contenant le diviseur, et **BC** le dividende. Le résultat final se trouvant dans **BC**. Le registre **HL** est utilisé pour effectuer les soustractions sur 16 bits, et permet ainsi de comptabiliser les différents restes après chaque opération.

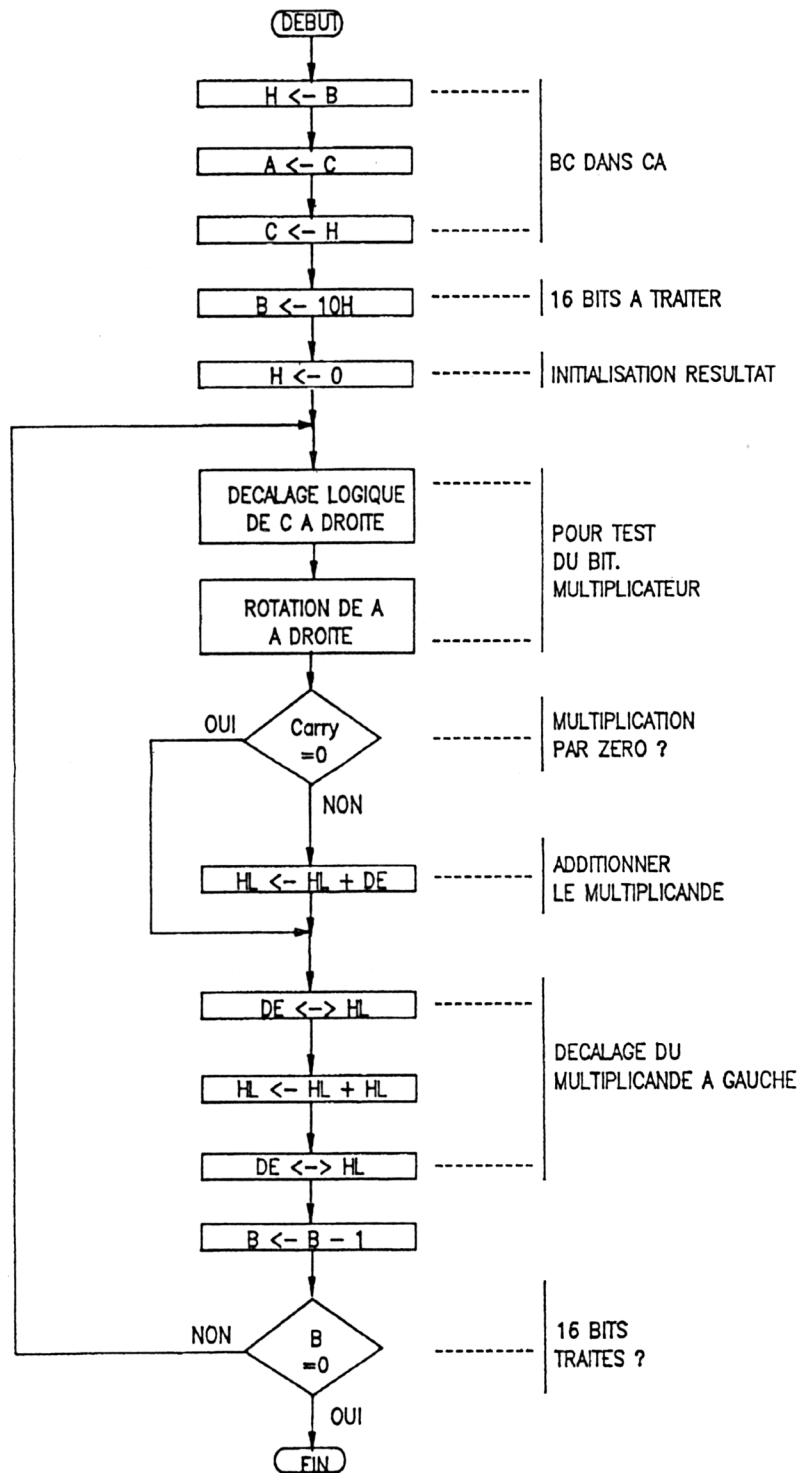
### LE PROGRAMME ASSEMBLEUR

A partir des explications précédentes, nous pouvons écrire les sous-programmes Assembleur correspondant aux traitements des multiplications et additions.

Il faut ensuite ajouter le programme principal qui réalise le calcul du PGCD, celui du PPCM, et celui de la réduction de FRACTIONS.

Nous vous proposons un programme largement commenté ce afin que vous remarquiez que nous avons été obligés de réaliser un sous-programme, permettant de tester et reclasser dans l'ordre correct le dividende et le diviseur lors du calcul du PGCD.

Nous pouvons rapporter l'algorithme ci-après pour observer le mécanisme de la division.



```

1      ;*****
2      ;**
3      ;**  CREATION D'INSTRUCTIONS  **
4      ;**  MATHEMATIQUES SOUS BASIC **
5      ;**  GRACE A L'AJOUT DE RSXs  **
6      ;**
7      ;** | PGCD,@X,@A,@B -> X=PGCD(A,B)**
8      ;** | PPCM,@X,@A,@B -> X=PPCM(A,B)**
9      ;** | FRAC,@A,@B -> FRACTION  **
10     ;**          SIMPLIFIEE DE A/B**
11     ;*****
12     ;
13     ;
14     ;*** TABLE DES EQUIVALENCES ***
15     ;
16     KLOGTX:      EQU  0BCD1H
17     ;
18     ;*****
19     ;
20     ;
21     ;*** ORIGINE D'ASSEMBLAGE ***
22     ;*****
23     ;
24     ;          ORG  0A000H
25     ;
26     ;
27     ;*** ORIGINE DE CHARGEMENT ***
28     ;
29     ;          LOAD 0A000H
30     ;
31     ;*****
32     ;
33     ;
34     ;*****
35     ;***  INSTALLATION DES RSXs  **
36     ;*****
37     ;
38     DEPART:      EQU  $
39     A000 2127A0   LD    HL,KERNEL          ;RESERVATION POUR
40     ;CHAINAGE
41     A003 010FA0   LD    BC,VECTEU        ;POINTE VECTEUR DES
42     ;INSTRUCTIONS
43     A006 CDD1BC   CALL KLOGTX            ;INSERE LES
44     ;INSTRUCTIONS
45     A009 3EC9     LD    A,0C9H            ;RET DANS A
46     A00B 3200A0   LD    (DEPART),A        ;POUR EVITER
47     ;UN NOUVEL APPEL
48     A00E C9      RET                    ;FIN
49     ;
50     ;
51     VECTEU:      EQU  $
52     A00F 1AA0     DEFW TABLE          ;VECTEUR
53     ;POINTANT LA TABLE DES
54     ;INSTRUCTIONS
55     ;
56     A011 C32BA0   JP    GCD              ;EXECUTION PGCD
57     A014 C340A0   JP    PCM              ;EXECUTION PPCM
58     A017 C36BA0   JP    FRA              ;EXECUTION FRAC
59     ;
60     ;

```

```

61          TABLE:      EQU  $
62          ;TABLE DES NOMS
63 A01A 504743          DEFB "PGC"
64 A01D C4              DEFB "D"+080H
65 A01E 505043          DEFB "PPC"
66 A021 CD              DEFB "M"+080H
67 A022 465241          DEFB "FRA"
68 A025 C3              DEFB "C"+080H
69 A026 00              DEFB 0
70          ;
71          ;
72          ;RESERVE POUR LIER INSTRUCTIONS
73          KERNEL:     DEFS 4
74          ;
75          ;*****
76          ;
77          ;
78          ;*****
79          ;* EXECUTION POUR CALCUL *
80          ;*      DU PGCD      *
81          ;*****
82          ;
83          GCD:        EQU  $
84 A02B FE03            CP 3                                ; 3 PARAMETRES?
85 A02D C0              RET NZ                            ; SINON FIN
86 A02E CDB2A0          CALL RECUPE                       ; ALLER RECHERCHER
87          ;LES PARAMETRES
88 A031 E5              PUSH HL                            ; SAUVER ADRESSE
89          ;DE LA VARIABLE RESULTAT
90 A032 CDD9A0          CALL COMPAR                       ; COMPARER ET
91          ;ORDONNER LES NOMBRES EN
92          ;FONCTION DE LEUR GRANDEUR
93 A035 CD06A1          CALL CHARGE                        ; CHARGER DANS
94          ;LES REGISTRES ADEQUATES
95 A038 CD13A1          CALL PGCD                         ; EFFECTUER LE
96          ;PLUS GRAND DIVISEUR COMMUN
97 A03B E1              POP HL                            ; RECUPERER ADRESSE
98          ;DE LA VARIABLE RESULTAT
99 A03C CD27A1          CALL RANGE                        ; RANGER DANS
100         ;LA VARIABLE RESULTAT
101 A03F C9              RET                               ; FIN
102         ;
103         ;*****
104         ;
105         ;
106         ;*****
107         ;* EXECUTION POUR CALCUL *
108         ;*      DU PPCM      *
109         ;*****
110         ;
111         PCM:         EQU  $
112 A040 FE03            CP 3                                ; TROIS PARAMETRES?
113 A042 C0              RET NZ                            ; NON ALORS RETOUR
114 A043 CDB2A0          CALL RECUPE                       ; RECUPERER LES
115         ;PARAMETRES TRANSMIS
116 A046 E5              PUSH HL                            ; SAUVER ADRESSE
117         ;DE LA VARIABLE RESULTAT
118 A047 CDD9A0          CALL COMPAR                       ; COMPARER ET
119         ;ORDONNER LES NOMBRES
120 A04A CD06A1          CALL CHARGE                        ; CHARGER LES

```

```

121 ;REGISTRES POUR CALCUL
122 A04D CD13A1 CALL PGCD ;CALCUL PGCD
123 A050 C5 PUSH BC ;SAUVER
124 A051 D1 POP DE ;ET RECUPERER
125 ;LE PGCD DANS DE
126 A052 2168A1 LD HL,NB1 ;PREMIER NOMBRE
127 A055 4E LD C,(HL) ;A RECHERCHER
128 A056 23 INC HL ;ET PLACER
129 A057 46 LD B,(HL) ;DANS BC
130 A058 CD41A1 CALL DIVISI ;EFFECTUER LA
131 ;DIVISION a/PGCD(a,b)
132 ;RESULTAT DANS BC
133 A05B 216AA1 LD HL,NB2 ;DEUXIEME NOMBRE
134 A05E 5E LD E,(HL) ;A RECHERCHER
135 A05F 23 INC HL ;ET PLACER
136 A060 56 LD D,(HL) ;DANS DE
137 A061 CD2DA1 CALL MULTIP ;EFFECTUER LA
138 ;MULTIPLICATION (a/PGCD)*b
139 A064 E5 PUSH HL ;SAUVER RESULTAT
140 A065 C1 POP BC ;POUR REPENDRE
141 ;DANS REGISTRE BC
142 A066 E1 POP HL ;RECUPERER ADRESSE
143 ;DE LA VARIABLE RESULTAT
144 A067 CD27A1 CALL RANGE ;ET LE RANGER
145 A06A C9 RET ;FIN
146 ;
147 ;*****
148 ;
149 ;
150 ;*****
151 ;* EXECUTION POUR *
152 ;* REDUCTION DE *
153 ;* FRACTIONS ENTIERES *
154 ;*****
155 ;
156 FRA: EQU $
157 A06B FE02 CP 2 ;DEUX PARAMETRES?
158 A06D C0 RET NZ ;NON ALORS FIN
159 A06E CDB2A0 CALL RECUPE ;RECUPERER LES
160 ;DEUX PARAMETRES
161 A071 CDD9A0 CALL COMPAR ;COMPARER POUR
162 ;LES ORDONNER
163 A074 CD06A1 CALL CHARGE ;CHARGER LES
164 ;PARAMETRES
165 A077 CD13A1 CALL PGCD ;ET EN
166 ;EFFECTUER LE PGCD
167 A07A C5 PUSH BC ;SAUVEGARDER
168 A07B C5 PUSH BC ;DEUX FOIS, CAR
169 ;DEUX NOMBRES A TRAITER
170 A07C D1 POP DE ;RECUPERER UNE
171 ;FOIS DANS DE
172 A07D 2168A1 LD HL,NB1 ;PREMIER
173 A080 4E LD C,(HL) ;NOMBRE
174 A081 23 INC HL ;A
175 A082 46 LD B,(HL) ;RECUPERER
176 A083 CD41A1 CALL DIVISI ;ET DIVISER
177 ;PAR LE PGCD
178 A086 3A70A1 LD A,(FLAG) ;TEST
179 A089 FE00 CP 00 ;SI INVERSION
180 A08B 200E JR NZ,FRA1 ;SINON SAUT

```



```

181 A08D 2A6CA1      LD   HL,(POINT1)      ;SI OUI
182 A090 E5          PUSH HL               ;INVERSER
183 A091 2A6EA1      LD   HL,(POINT2)      ;LES
184 A094 226CA1      LD   (POINT1),HL      ;ADRESSES
185 A097 E1          POP  HL               ;DES
186 A098 226EA1      LD   (POINT2),HL      ;POINTEURS
187                                     ;
188 A09B 2A6CA1      FRA1: LD HL,(POINT1)      ;RANGER
189 A09E 71          LD   (HL),C           ;DANS
190 A09F 23          INC  HL               ;LE PREMIER
191 A0A0 70          LD   (HL),B           ;PARAMETRE
192 A0A1 D1          POP  DE               ;REPRENDRE PGCD.
193 A0A2 216AA1      LD   HL,NB2           ;DEUXIEME
194 A0A5 4E          LD   C,(HL)           ;NOMBRE
195 A0A6 23          INC  HL               ;A
196 A0A7 46          LD   B,(HL)           ;RECUPERER
197 A0A8 CD41A1      CALL DIVISI           ;ET DIVISER.
198 A0AB 2A6EA1      LD   HL,(POINT2)      ;RANGER
199 A0AE 71          LD   (HL),C           ;DANS
200 A0AF 23          INC  HL               ;LE DEUXIEME
201 A0B0 70          LD   (HL),B           ;PARAMETRE.
202 A0B1 C9          RET                   ;FIN
203                                     ;
204                                     ;*****
205                                     ;
206                                     ;
207                                     ;*****
208 *      SOUS-PROGRAMME DE*      *
209 *      RECUPERATION DES      *
210 *      PARAMETRES TRANSMIS  *
211 *      DANS LA PILE POINTEE *
212 *      PAR LE REGISTRE IX   *
213 *      *****
214                                     ;
215 RECUPE:      EQU  $
216                                     ;
217 ** TROISIEME PARAMETRE **
218 A0B2 DD6E00      LD   L,(IX+00H)       ;DERNIER
219 A0B5 DD6601      LD   H,(IX+01H)       ;PARAMETRE
220 A0B8 226CA1      LD   (POINT1),HL     ;DANS POINT1
221 A0BB 5E          LD   E,(HL)           ;ET
222 A0BC 23          INC  HL               ;RECUPERATION
223 A0BD 56          LD   D,(HL)           ;DE SA VALEUR
224 A0BE EB          EX   DE,HL           ;A SAUVEGARDER
225 A0BF 226AA1      LD   (NB2),HL        ;DANS NB1
226                                     ;
227 ** DEUXIEME PARAMETRE **
228 A0C2 DD6E02      LD   L,(IX+02H)       ;DEUXIEME
229 A0C5 DD6603      LD   H,(IX+03H)       ;PARAMETRE
230 A0C8 226EA1      LD   (POINT2),HL     ;DANS POINT2
231 A0CB 5E          LD   E,(HL)           ;ET
232 A0CC 23          INC  HL               ;RECUPERATION
233 A0CD 56          LD   D,(HL)           ;DE SA VALEUR
234 A0CE EB          EX   DE,HL           ;A SAUVEGARDER
235 A0CF 2268A1      LD   (NB1),HL        ;DANS NB2
236                                     ;
237 ** PREMIER PARAMETRE **
238 A0D2 DD6E04      LD   L,(IX+04H)       ;RECUPERATION
239 A0D5 DD6605      LD   H,(IX+05H)       ;PREMIER PARAMETRE
240 ; QUI EST LE POINTEUR

```

```

241 ;DE LA VARIABLE DU
242 ;RESULTAT POUR
243 ;|PGCD ET |PPCM
244 ;MAIS N'A AUCUNE VALEUR
245 ;POUR |FRAC -> TRAITEMENT
246 ;ULTERIEUR
247 A0D8 C9 RET ;FIN
248 ;
249 ;*****
250 ;
251 ;
252 ;*****
253 ;* SOUS-PROGRAMME DE *
254 ;* COMPARAISON DES DEUX *
255 ;* NOMBRES POUR LES *
256 ;* RANGER DANS L'ORDRE *
257 ;* POUR LA DIVISION *
258 ;*****
259 ;
260 COMPAR: EQU $
261 A0D9 AF XOR A ;ANNULATION
262 A0DA 3270A1 LD (FLAG),A ;DU DRAPEAU
263 A0DD 2168A1 LD HL,NB1 ;OCTET HAUT
264 A0E0 23 INC HL ;DE NB1
265 A0E1 46 LD B,(HL) ;DANS
266 A0E2 78 LD A,B ;REGISTRE A
267 A0E3 216AA1 LD HL,NB2 ;POINTER
268 A0E6 23 INC HL ;OCTET HAUT DE NB2
269 A0E7 BE CP (HL) ;COMPARER LES DEUX
270 ;OCTETS MSB
271 A0E8 2007 JR NZ,COMPA1 ;SI NON EGAUX
272 ;ALLER VOIR LE PLUS GRAND
273 A0EA 3A68A1 LD A,(NB1) ;SI OUI
274 A0ED 216AA1 LD HL,NB2 ;PROCEDER DE
275 A0F0 BE CP (HL) ;LA MEME FACON AVEC
276 ;LES OCTETS BAS
277 COMPA1: EQU $
278 A0F1 D0 RET NC ;POUR VOIR
279 ;SI NB1 PLUS GRAND,
280 ;ALORS SORTIR
281 ;SINON INVERSER CI DESSOUS
282 INVERS: EQU $
283 A0F2 3EFF LD A,0FFH ;SIGNALER INVERSION
284 A0F4 3270A1 LD (FLAG),A ;DANS FLAG
285 A0F7 2A68A1 LD HL,(NB1) ;ET
286 A0FA ED5B6AA1 LD DE,(NB2) ;EFFECTUER
287 A0FE ED5368A1 LD (NB1),DE ;INVERSION DES
288 A102 226AA1 LD (NB2),HL ;DEUX NOMBRES
289 A105 C9 RET ;FIN
290 ;
291 ;*****
292 ;
293 ;
294 ;*****
295 ;* SOUS-PROGRAMME DE *
296 ;* CHARGEMENT DES NOMBRES *
297 ;* DANS LES REGISTRES DU *
298 ;* Z-80 POUR EFFECTUER *
299 ;* LES CALCULS DE PGCD *
300 ;*****

```

```

301
302
303 A106 2168A1
304 A109 4E
305 A10A 23
306 A10B 46
307 A10C 216AA1
308 A10F 5E
309 A110 23
310 A111 56
311 A112 C9
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327 A113 D5
328 A114 C5
329 A115 CD41A1
330
331 A118 CD2DA1
332
333 A11B EB
334 A11C E1
335 A11D AF
336 A11E ED52
337
338 A120 EB
339
340
341
342 A121 C1
343
344 A122 7A
345 A123 B3
346 A124 20ED
347 A126 C9
348
349
350
351
352
353
354
355
356
357
358
359
360 A127 79

CHARGE: EQU $
LD HL,NB1 ;NB1
LD C,(HL) ;DANS BC
INC HL
LD B,(HL)
LD HL,NB2 ;NB2
LD E,(HL) ;DANS DE
INC HL
LD D,(HL)
RET ;FIN

;
;*****
;
;
;*****
;* SOUS-PROGRAMME DE *
;* CALCUL DU PGCD DE DEUX *
;* NOMBRE DE HUIT BITS *
;* LE PLUS GRAND DANS BC *
;* LE PLUS PETIT DANS DE *
;* LE RESULTAT SERA RANGE *
;* DANS LE REGISTRE BC *
;*****
;
PGCD: EQU $
PUSH DE ;SAUVEGARDE
PUSH BC ;DES REGISTRES
CALL DIVISI ;DIVISION ENTIERE
; POUR AVOIR LE QUOTIENT
CALL MULTIP ;MULTIPLICATION
; DU QUOTIENT ET DIVISEUR
EX DE,HL ;DANS DE
POP HL ;RECUPERER DIVIDANDE
XOR A ;ANNULE INDICATEURS
SBC HL,DE ;CALCUL RESTE
; PAR SOUSTRACTION
EX DE,HL ;ECHANGE RESTE
; ET q*b POUR OBTENIR LE
; RESTE DANS DE POUR LA
; PROCHAINE OPERATION
POP BC ;REPLACE a
; PAR LE DIVISEUR
LD A,D ;RECHERCHE
OR E ;SI RESTE NUL
JR NZ,PGCD ;SINON RECOMMENCER
RET ;SI OUI FIN

;
;*****
;
;
;*****
;* SOUS PROGRAMME D'AFFECTION *
;* DE LA VALEUR CONTENUE DANS *
;* BC A LA VARIABLE DE SORTIE *
;* DE L'INSTRUCTION BASIC *
;*****
;
RANGE: EQU $
LD A,C ;RANGE

```

```

361 A128 77          LD   (HL),A          ;OCTET BAS
362 A129 78          LD   A,B           ;RANGE
363 A12A 23          INC  HL            ;OCTET
364 A12B 77          LD   (HL),A          ;HAUT
365 A12C C9          RET                ;FIN
366
367                ;*****
368                ;
369                ;
370                ;*****
371                ;*   SOUS PROGRAMME DE   *
372                ;*   MULTIPLICATION   *
373                ;* DE DEUX NOMBRES 16 BITS *
374                ;* CONTENUS DANS DE ET BC *
375                ;* LE RESULTAT SE TROUVANT *
376                ;* DANS LE REGISTRE HL   *
377                ;*****
378                ;
379 MULTIP:          EQU  $
380 A12D 60          LD   H,B           ;OCTET HAUT
381                ;ET
382 A12E 79          LD   A,C           ;OCTET BAS
383                ;DU MULTIPLICATEUR
384 A12F 4C          LD   C,H           ;A INVERSER
385 A130 0610        LD   B,010H        ;COMPTEUR 16 BITS
386 A132 210000      LD   HL,0         ;VALEUR INITIALE
387                ;
388 MULT11:          EQU  $
389 A135 CB39        SRL  C             ;DECALAGE
390 A137 1F          RRA                ;DECALAGE+CARRY
391 A138 3001        JR   NC,PASADD     ;SI NUL
392                ;ON N'ADDITIONNERA PAS
393 A13A 19          ADD  HL,DE         ;SINON ADDITION
394                ;DU MULTIPLICANDE
395 PASADD:          EQU  $
396 A13B EB          EX   DE,HL         ;DECALAGE
397 A13C 29          ADD  HL,HL         ;DU MULTIPLICANDE
398 A13D EB          EX   DE,HL         ;ET RANGE
399 A13E 10F5        DJNZ MULT11       ;SI NON NUL
400                ;ON RECOMMANCE
401 A140 C9          RET                ;SINON FIN
402                ;
403                ;*****
404                ;
405                ;
406                ;*****
407                ;*   SOUS-PROGRAMME DE   *
408                ;*   DIVISION ENTIERE DE DEUX *
409                ;*   NOMBRES 16 BITS   *
410                ;* CONTENUS DANS DE ET BC *
411                ;* DIVIDANDE DANS BC   *
412                ;* DIVISEUR DANS DE   *
413                ;* RESULTAT DANS BC   *
414                ;*****
415                ;
416 DIVISI:          EQU  $
417 A141 78          LD   A,B           ;OCTET HAUT
418 A142 210000      LD   HL,0         ;QUOTIENT A 0
419 A145 0610        LD   B,16        ;COMPTEUR 16 BITS
420

```

```

421          DIVIS1:      EQU   $
422 A147 CB11          RL    C           ;ROTATION
423 A149 17            RLA                   ;AVEC A POUR
424          ;BIT SUPERIEUR
425 A14A ED6A          ADC   HL,HL         ;A AJOUTER A
426          ;HL+HL(DECALAGE DE HL)
427 A14C ED52          SBC   HL,DE         ;SOUTRAIT DIVISEUR
428          DIVIS2:      EQU   $
429 A14E 3F            CCF                   ;INVERSE FLAG
430 A14F 3010          JR    NC,DIVIS5      ;SOUSTRACTION
431          ;POSSIBLE?
432          DIVIS3:      EQU   $
433 A151 10F4          DJNZ  DIVIS1         ;NON RECOMMENCER
434          ;SI 16 BITS NON TRAITES
435 A153 180E          JR    DIVIS6         ;SI NUL
436          ;ALORS SAUTER
437          DIVIS4:      EQU   $
438 A155 CB11          RL    C           ;ROTATION
439 A157 17            RLA                   ;DIVIDANDE
440 A158 ED6A          ADC   HL,HL         ;ET DECALER LE
441          ;RESTE POUR POSER LE
442          ;PROCHAIN BIT
443          AND    A           ;ANNULE CARRY
444 A15B ED5A          ADC   HL,DE         ;POUR ADDITION
445 A15D 38F2          JR    C,DIVIS3      ;TESTER
446 A15F 28ED          JR    Z,DIVIS2      ;A NOUVEAU
447          DIVIS5:      EQU   $
448 A161 10F2          DJNZ  DIVIS4         ;SI NON NUL
449          DIVIS6:      EQU   $
450 A163 CB11          RL    C           ;SI FIN ALORS
451 A165 17            RLA                   ;DERNIERE ROTATION
452 A166 47            LD    B,A          ;PUIS RESTAURE B
453 A167 C9            RET                   ;FIN
454          ;
455          ;*****
456          ;
457          ;
458          ;*****
459          ;* AFFECTATIONS MEMOIRE *
460          ;* POUR PARAMETRES DE *
461          ;* SAUVEGARDE UTILISES *
462          ;*****
463          ;
464          ;* PREMIER NOMBRE *
465          ;* =TROISIEME PARAMETRE *
466 A168 0000          NB1:    DEFW 0000
467          ;
468          ;* DEUXIEME NOMBRE *
469          ;* =DEUXIEME PARAMETRE *
470 A16A 0000          NB2:    DEFW 0000
471          ;
472          ;* POINTEURS INTERMEDIAIRES *
473 A16C 0000          POINT1:  DEFW 0000
474 A16E 0000          POINT2:  DEFW 0000
475          ;
476          ;* DRAPEAU POUR INVERSION *
477 A170 00            FLAG:    DEFB 00H
478          ;
479          ;*****
480          ;
481          ;
482          END

```

On remarquera, en tout début du programme, la définition des RSXs grâce au célèbre vecteur KL-LOG-EXTENSION placé à l'adresse &BCD1.

### LE CHARGEUR BASIC

Bien sûr, nous n'oublions pas le programme Basic qui n'a pas encore eu le temps de se procurer de logiciel d'assemblage, et nous lui proposons ci-dessous le programme Basic permettant de charger les codes hexadécimaux sous l'interpréteur.

```

10 REM *****
20 REM * INSTALLATION D'UNE ROUTINE *
30 REM * CREANT DES INSTRUCTIONS *
40 REM * MATHEMATIQUES NOUVELLES *
50 REM * |PGCD,@R%,@A%,@B% *
60 REM * |PPCM,@R%,@A%,@B% *
70 REM * |FRAC,@A%,@B% *
80 REM *****
90 REM
100 REM *** CHARGEUR BASIC ***
110 REM
120 ADR = &A000
130 I = 0
140 SOMME = 0
150 RESTORE 420
160 READ AS
170 IF AS = "XX" THEN 250
180 BS = "&" + AS
190 B = VAL(BS)
200 SOMME = SOMME + B
210 POKE ADR,B
220 ADR = ADR + 1
230 I = I + 1
240 GOTO 160
250 IF SOMME <> 42942 THEN CLS:PRINT CHR
$(7);"ERREUR DANS LES DATAS":FOR I = 1 T
O 1000:NEXT I:LIST
260 REM *** SAUVEGARDE ***
270 MODE 1
280 REM
290 PRINT"POUR SAUVEGARDER LA ROUTINE EN
BINAIRE"
300 PRINT"SAVE ";CHR$(34);"ROUTINE.BIN";
CHR$(34);".B,&A000,";RIGHT$(STR$(I),LEN(
STR$(I))-1)
310 PRINT
320 PRINT
330 PRINT "CHARGEMENT PAR"
340 PRINT "MEMORY &9FFF"
350 PRINT"LOAD ";CHR$(34);"ROUTINE.BIN";
CHR$(34);",&A000"

```

```
360 PRINT
370 PRINT
380 PRINT"INITIALISATION PAR CALL &A000
390 PRINT
400 STOP
410 REM
420 REM *** ROUTINE L.M. ***
430 REM
440 REM *** INSTALLATION RSX ***
450 DATA 21,27,A0,01,0F,A0,CD,D1
460 DATA BC,3E,C9,32,00,A0,C9
470 REM
480 REM *** VECTEURS ET TABLE ***
490 DATA 1A
500 DATA A0,C3,2B,A0,C3,40,A0,C3
510 DATA 6B,A0,50,47,43,C4,50,50
520 DATA 43,CD,46,52,41,C3,00
530 REM *** KERNEL ***
540 DATA 00,00,00,00
550 REM
560 REM *** TRAITEMENT PGCD ***
570 DATA FE,03,C0,CD,B2
580 DATA A0,E5,CD,D9,A0,CD,06,A1
590 DATA CD,13,A1,E1,CD,27,A1,C9
600 REM
610 REM *** TRAITEMENT PPCM ***
620 DATA FE,03,C0,CD,B2,A0,E5,CD
630 DATA D9,A0,CD,06,A1,CD,13,A1
640 DATA C5,D1,21,68,A1,4E,23,46
650 DATA CD,41,A1,21,6A,A1,5E,23
660 DATA 56,CD,2D,A1,E5,C1,E1,CD
670 DATA 27,A1,C9
680 REM
690 REM *** TRAITEMENT FRACTION ***
700 DATA FE,02,C0,CD,B2
710 DATA A0,CD,D9,A0,CD,06,A1,CD
720 DATA 13,A1,C5,C5,D1,21,68,A1
730 DATA 4E,23,46,CD,41,A1,3A,70
740 DATA A1,FE,00,20,0E,2A,6C,A1
750 DATA E5,2A,6E,A1,22,6C,A1,E1
760 DATA 22,6E,A1,2A,6C,A1,71,23
770 DATA 70,D1,21,6A,A1,4E,23,46
780 DATA CD,41,A1,2A,6E,A1,71,23
790 DATA 70,C9
800 REM
810 REM *** RECUPERE PARAMETRES ***
820 DATA DD,6E,00,DD,66,01
830 DATA 22,6C,A1,5E,23,56,EB,22
840 DATA 6A,A1,DD,6E,02,DD,66,03
850 DATA 22,6E,A1,5E,23,56,EB,22
860 DATA 68,A1,DD,6E,04,DD,66,05
```

```
870 DATA C9
880 REM
890 REM *** COMPARAISON ET INVERSION ***
900 DATA AF,32,70,A1,21,68,A1
910 DATA 23,46,78,21,6A,A1,23,BE
920 DATA 20,07,3A,68,A1,21,6A,A1
930 DATA BE,D0,3E,FF,32,70,A1,2A
940 DATA 68,A1,ED,5B,6A,A1,ED,53
950 DATA 68,A1,22,6A,A1,C9
960 REM
970 REM *** CHARGE ***
980 DATA 21,68
990 DATA A1,4E,23,46,21,6A,A1,5E
1000 DATA 23,56,C9
1010 REM
1020 REM *** CALCUL EFFECTIF PGCD ***
1030 DATA D5,C5,CD,41,A1
1040 DATA CD,2D,A1,EB,E1,AF,ED,52
1050 DATA EB,C1,7A,B3,20,ED,C9
1060 REM
1070 REM *** RANGE LES VALEURS ***
1080 REM *** DANS LES REGISTRES **
1090 REM *** APPROPRIES ***
1100 DATA 79,77,78,23,77,C9
1110 DATA 60,79,4C
1120 REM
1130 *** EFFECTUE LA MULTIPLICATION ***
1140 DATA 06,10,21,00,00,CB,39,1F
1150 DATA 30,01,19,EB,29,EB,10,F5
1160 DATA C9
1170 REM
1180 *** EFFECTUE LA DIVISION ***
1190 DATA 78,21,00,00,06,10,CB
1200 DATA 11,17,ED,6A,ED,52,3F,30
1210 DATA 10,10,F4,18,0E,CB,11,17
1220 DATA ED,6A,A7,ED,5A,38,F2,28
1230 DATA ED,10,F2,CB,11,17,47,C9
1240 REM
1250 REM *** NB1,NB2 ***
1260 DATA 00,00,00,00
1270 REM
1280 REM *** POINT1,POINT 2 ***
1290 DATA 00,00,00,00
1300 REM
1310 REM *** FLAG ***
1320 DATA 00
1330 REM
1340 *** FIN DE DATAS ***
1350 DATA XX
1360 END
```