Visitons la Rom Monitoring

(6e partie) par André Chéramy et Claude Sittler

f) Deuxième table située de FE78 à FE85 pour la commande DES. Selon le mode d'adressage, cette fournit les informations sur ce qu'il faut afficher (les 6 bits de poids le plus fort), ainsi que le nombre d'octets de données (les 2 bits de poids le plus faible) qui suivent l'octet d'instruction. Cette table est lue en E9AF en utilisant comme index un code de #0 à #D (14 entrées correspondant aux 14 modes d'adressage). La valeur lue est stockée en 50, puis on en extrait le nombre d'octets de donnés suivant l'octet à décoder qui est stocké en 51.

adr	valeur	mode	exemple	nbre octets de données
FE78-	00 = 0000 0000	[donnée]		
FE79-	21 = 0010 0001	Immédiat	LDA #$00	1
FE7A-	81 = 1000 0001	Page zéro	LDA #00	1
FE7B-	82 = 1000 0010	Absolu	LDA #9800	2
FE7C-	00 = 0000 0000	Implicite	INY	0
FE7D-	00 = 0000 0000	Accumulateur	ASL A	0
FE7E-	59 = 0101 1001	(Indirect,X)	LDA (#00,X)	1
FE7F-	4D = 0100 1101	(Indirect),Y	LDA (#00),Y	1
FE80-	91 = 1001 0001	Page zéro,X	LDY #00,X	1
FE81-	92 = 1001 0010	Absolu,X	LDA #9800,X	2
FE82-	86 = 1000 0110	Absolu,Y	LDA #9800,Y	2
FE83-	4A = 0100 1010	Indirect	JMP (#9800)	2
FE84-	85 = 1000 0101	Page zéro,Y	LDX #00,Y	1
FE85-	9D = 1001 1101	Relatif	BNE #9800	1

g) Table FE85-FE8A contenant 6 caractères à afficher pour la commande DES.

FE86- 2C = ‘,’
FE87- 29 = ‘)’
FE88- 2C = ‘,’
FE89- 23 = ‘#’
FE8A- 28 = ‘(‘
FE8B- 23 = ‘#’

h) Table FE8B-FE90 contenant 6 caractères à afficher pour la commande DES.

FE8C- 59 = ‘Y’
FE8D- 00 = rien
FE8E- 58 = ‘X’
FE8F- 24 = ‘$’
FE90- 23 = ‘#’
FE91- 00 = rien

i) Tableaux des mnémoniques pour la commande DES de FE92-FED1combiné à FED2-FF11.

Exemple de combinaison des bits de FE92 & FED2 : 00011 10011 01100 0 qui correspond à : 03+3F=42 soit ‘B’ puis 13+3F=52 soit ‘R’ puis 0C+3F=4B soit ‘K’qui donne BRK et 0

```
FE92 & FED2 1CD8 = 00011 10011 01100 0 soit BRK et 0
FE93 & FED3 8A62 = 10001 01001 10001 0 soit PHP et 0
FE94 & FED4 1C5A = 00011 10001 01101 0 soit BPL et 0
FE95 & FED5 2348 = 00100 01101 00100 0 soit CLC et 0
FE96 & FED6 5D26 = 01011 10100 10011 0 soit JSR et 0
FE97 & FED7 8B62 = 10001 01101 10001 0 soit PLP et 0
FE98 & FED8 1B94 = 00011 01110 01010 0 soit BMI et 0
FE99 & FED9 A188 = 10100 00110 00100 0 soit SEC et 0
FE9A & FEDA 9D54 = 10011 10101 01010 0 soit RTI et 0
FE9B & FEDB 8A44 = 10001 01001 00010 0 soit PHA et 0
FE9C & FEDC 1DC8 = 00011 10111 00100 0 soit BVC et 0
FE9D & FEDD 2354 = 00100 01101 01010 0 soit CLI et 0
FE9E & FEDE 9D68 = 10011 10101 10100 0 soit RTS et 0
FE9F & FEDF 8B44 = 10001 01101 00010 0 soit PLA et 0
FEA0 & FEE0 1DE8 = 00011 10111 10100 0 soit BVS et 0
FEA1 & FEE1 A194 = 10100 00110 01010 0 soit SEI et 0
FEA2 & FEE2 0000 = 00000 00000 00000 0 soit ??? et 0
FEA3 & FEE3 29B4 = 00101 00110 11010 0 soit DEY et 0
FEA4 & FEE4 1908 = 00011 00100 00100 0 soit BCC et 0
FEA5 & FEE5 AE84 = 10101 11010 00010 0 soit TYA et 0
FEA6 & FEE6 6974 = 01101 00101 11010 0 soit LDY et 0
FEA7 & FEE7 A8B4 = 10101 00010 11010 0 soit TAY et 0
FEA8 & FEE8 1928 = 00011 00100 10100 0 soit BCS et 0
FEA9 & FEE9 236E = 00100 01101 10111 0 soit CLV et 0
FEAA & FEEA 2474 = 00100 10001 11010 0 soit CPY et 0
FEAB & FEEB 53F4 = 01010 01111 11010 0 soit INY et 0
FEAC & FEEC 1BCC = 00011 01111 00110 0 soit BNE et 0
FEAD & FEED 234A = 00100 01101 00101 0 soit CLD et 0
FEAE & FEEE 2472 = 00100 10001 11001 0 soit CPX et 0
FEAF & FEEF 53F2 = 01010 01111 11001 0 soit INX et 0
FEB0 & FEF0 19A4 = 00011 00110 10010 0 soit BEQ et 0
FEB1 & FEF1 A18A = 10100 00110 00101 0 soit SED et 0
FEB2 & FEF2 0000 = 00000 00000 00000 0 soit ??? et 0
FEB3 & FEF3 1AAA = 00011 01010 10101 0 soit BIT et 0
FEB4 & FEF4 5BA2 = 01011 01110 10001 0 soit JMP et 0
FEB5 & FEF5 5BA2 = 01011 01110 10001 0 soit JMP et 0
FEB6 & FEF6 A574 = 10100 10101 11010 0 soit STY et 0
FEB7 & FEF7 6974 = 01101 00101 11010 0 soit LDY et 0
FEB8 & FEF8 2474 = 00100 10001 11010 0 soit CPY et 0
FEB9 & FEF9 2472 = 00100 10001 11001 0 soit CPX et 0
FEBA & FEFA AE44 = 10101 11001 00010 0 soit TXA et 0
FEBB & FEFB AE68 = 10101 11001 10100 0 soit TXS et 0
FEBC & FEFC A8B2 = 10101 00010 11001 0 soit TAX et 0
FEBD & FEFD AD32 = 10101 10100 11001 0 soit TSX et 0
FEBE & FEFE 29B2 = 00101 00110 11001 0 soit DEX et 0
FEBF & FEFF 0000 = 00000 00000 00000 0 soit ??? et 0
FEC0 & FF00 7C22 = 01111 10000 10001 0 soit NOP et 0
FEC1 & FF01 0000 = 00000 00000 00000 0 soit ??? et 0
FEC2 & FF02 151B = 00010 10100 01101 1 soit ASL et 1
FEC3 & FF03 9C1B = 10011 10000 01101 1 soit ROL et 1
FEC4 & FF04 6D27 = 01101 10100 10011 1 soit LSR et 1
FEC5 & FF05 9C27 = 10011 10000 10011 1 soit ROR et 1
FEC6 & FF06 A572 = 10100 10101 11001 0 soit STX et 0
FEC7 & FF07 6972 = 01101 00101 11001 0 soit LDX et 0
FEC8 & FF08 2988 = 00101 00110 00100 0 soit DEC et 0
```

```
FEC9 & FF09 53C8 = 01010 01111 00100 0 soit INC et 0
FECA & FF0A 84C4 = 10000 10011 00010 0 soit ORA et 0
FECB & FF0B 13CA = 00010 01111 00101 0 soit AND et 0
FECC & FF0C 3426 = 00110 10000 10011 0 soit EOR et 0
FECD & FF0D 1148 = 00010 00101 00100 0 soit ADC et 0
FECE & FF0E A544 = 10100 10101 00010 0 soit STA et 0
FECF & FF0F 6944 = 01101 00101 00010 0 soit LDA et 0
FED0 & FF10 23A2 = 00100 01110 10001 0 soit CMP et 0
FED1 & FF11 A0C8 = 10100 00011 00100 0 soit SBC et 0
```

j) Table FF12-FF2D (non identifiée)

```
FF12- 2E 1A 7B 80 16 40 2D 80 B0 DE 4C C0 00 08
FF20- 9B 44 64 80 16 40 10 6A 6F 2E 7E 9C 48 56
```

Ceci ressemble bien à un résidu de mise au point.

k) Nouvelle routine «Affiche en hexadécimal le nombre AY». Cette routine est appelée par les commandes DES, DUMP et MOD, via les sous-routines F85B et F892.

FF2E-	48	PHA	sauve 2e digit LL du nombre
FF2F-	98	TYA	prend 1er digit HH du nombre
FF30-	20 34 FF	JSR **FF34**	affiche le 1er digit en hexa
FF33-	68	PLA	récupère le LL du nombre (2e digit) et l'affiche

l) Nouvelle routine «Affiche en hexadécimal le contenu de A»

FF34-	48	PHA	préserve le contenu de A
FF35-	29 F0	AND #F0	élimine 4 bits de poids faible
FF37-	4A	LSR	décale les 4 bits de poids fort
FF38-	4A	LSR	vers la droite
FF39-	4A	LSR	ils prennent donc la place
FF3A-	4A	LSR	des 4 bits de poids faible
FF3B-	20 3F FF	JSR **FF3F**	les convertit en caractère de 0 à 9 ou de A à F et affiche
FF3E-	68	PLA	restaure contenu initial de A

Sous-routine «Convertit les 4 bits de poids faible en caractère de 0 à 9 ou de A à F et affiche»

FF3F-	29 0F	AND #0F	élimine les 4 bits de poids fort, reste 0000 xxxx
FF41-	09 30	ORA #30	masque 0011 0000, reste 0011 xxxx
FF43-	C9 3A	CMP #3A	est-ce < #3A ? (c'est à dire un nombre de #30 à #39)
FF45-	90 02	BCC **FF49**	si oui, on l'affiche (c'est à dire Ascii de 0 à 9)
FF47-	69 06	ADC #06	sinon, A=A+6+C (ex. #3A devient #41soit la lettre A etc.)
FF49-	4C D9 CC	JMP CCD9	affiche le caractère Ascii contenu dans A et retourne

m) Nouvelle routine «Affiche le caractère Ascii contenu dans A ou un espace si code CTRL». Tous les codes CTRL quel que soit leur bit b7 (c'est à dire de #00 à #1F et de #80 à #9F) seront affichés comme un espace normal (bloc de couleur papier, ici noir). Tous les caractères normaux (c'est à dire de #20 à #7F) seront affichés en vidéo normale (ici encre jaune sur papier noir). Tous les caractères inverses (c'est à dire de #A0 à #FF) seront affichés en vidéo inverse (ici encre bleue sur papier blanc). Avec cette Rom Monitoring, dans les conditions normales (pas de jeu de caractères chargés après coup), seuls les caractères de #20 (espace) à #5A (lettre Z) ont été définis et sont lisibles. Pour les autres (caractères semi-graphiques, minuscules etc.) la Ram est restée avec ses #55 inscrits par le test de la mémoire lors du boot, ce qui se traduit par un petit bloc rayé verticalement jaune sur noir ou bleu sur blanc selon le bit b7. Pour juger de ce que ça donne essayez le mini programme suivant : FOR I=0 TO 255:AD=#1000+I:POKEAD,I:NEXT puis lancez DUMP#1000

FF4C-	48	PHA	sauve l'octet contenu dans A
FF4D-	29 7F	AND #7F	élimine le bit de poids fort b7, il reste 0xxx xxxx
FF4F-	38	SEC	prépare une soustraction pour tester si c'est un code CTRL
FF50-	E9 20	SBC #20	l'octet à afficher est-il inférieur à #20 ?
FF52-	B0 04	BCS FF58	si pas un code CTRL, on saute les 3 instructions suivantes
FF54-	68	PLA	dépile l'octet initial à afficher
FF55-	A9 20	LDA #20	et met #20 (code de 'espace')
FF57-	48	PHA	à la place sur la pile
FF58-	68	PLA	récupère l'octet d'origine ou le code d'espace
FF59-	C9 FF	CMP #FF	est-ce le code #FF ? (?)
FF5B-	D0 02	BNE FF5F	si pas #FF, saute l'instruction suivante
FF5D-	A9 FF	LDA #FF	si #FF, reprend #FF (?)
FF5F-	AC 69 02	LDY 0269	contient le n° de colonne du curseur 'Text' (de 0 à 39)
FF62-	91 12	STA (12),Y	12-13 contient l'adresse de la ligne du curseur 'Text' : l'octet est donc affiché au curseur
FF64-	EE 69 02	INC 0269	envoie le curseur sur la colonne suivante, la routine d'affichage du curseur change de ligne si besoin
FF67-	60	RTS	et retourne

n) Entrée de la nouvelle commande HDK.

Syntaxe HDK#xx ou HDK#xxxx

Equivalent à ?HEX$(PEEK(xx)) ou à ?HEX$(DEEK(xxxx))

FF68-	20 53 E8	JSR E853	évalue nombre sur 2 octets
FF6B-	A0 00	LDY #00	index pour deux lectures
FF6D-	B1 33	LDA(33),Y	lit LL en 33
FF6F-	AA	TAX	le garde dans X
FF70-	C8	INY	octet suivant
FF71-	B1 33	LDA(33),Y	lit HH en 34
FF73-	A8	TAY	le garde dans Y
FF74-	8A	TXA	le nombre est dans AY
FF75-	4C 92 F8	JMP **F892**	va la ligne puis affiche en hexadécimal le nombre AY

à suivre...