

Les adresses du graphique

Jean-Francois Duvivier

Pour ceux d'entre vous qui désirent explorer dans le détail les entrailles de leur Apple, nous avons rassemblé dans cet article un certain nombre d'adresses utiles relatives au mode graphique. Le tableau 1 vous montre comment, sous moniteur ou en BASIC, passer en mode graphique, aller d'une page à l'autre, ...

Le second tableau indique les zones mémoire utilisées par les divers modes de fonctionnement graphique. Ces zones utilisées par les pages 1 et 2 ne sont pas totalement exploitées par le système. Afin de rendre l'adressage plus simple et plus rapide, le système a laissé quelques trous dont vous pouvez faire ce que vous voulez.

Ces adresses libres sont fournies, pour les graphiques en basse résolution, dans le tableau 3. On peut constater que les huit octets situés après le dernier octet de chaque ligne sont ainsi libres et utilisables à loisir. Une aubaine pour ceux qui sont en permanence à la recherche de place !

Le quatrième tableau montre, pour chaque ligne en basse résolution de la page 1, la localisation de l'octet représentant la première, la 21ème et la dernière colonne de chaque ligne.

Ces adresses apparaissent en hexadécimal dans la partie gauche et en décimal dans la partie droite. L'analyse des adresses permet de constater que l'écran est découpé en trois parties horizontales, et que les adresses cyclent sur ces trois parties. A l'écran, il est d'ailleurs visible que l'apparition d'un graphique se réalise en parallèle, ou presque, sur trois tiers d'écran.

Le tableau 5(A) donne les adresses décimales du début et de la fin de chaque ligne de la page 1 en graphique haute résolution. Comme l'indique le découpage en traits gras, nous retrouvons le regroupement en trois tiers d'écran. A l'intérieur de chaque tiers, ainsi que l'illustrent les traits fins, on remarque des blocs de huit lignes, cette taille ayant bien entendu été choisie pour des raisons d'efficacité d'adressage.

Le lecteur pourra par lui-même identifier les zones mémoire laissées libres par le graphique haute résolution, telles que 8312-8319, ...

Pour la basse comme pour la haute résolution, une simple translation permet de passer des adresses de la page 1 à celles de la page 2.

TABLEAU 1

SOUS MONITEUR	SOUS BASIC	FONCTION
* C050	Poke - 16304,0	Mode graphique
* C051	Poke - 16303,0	Mode Texte
* C052	Poke - 16302,0	Graphique total
* C053	Poke - 16301,0	Graphique Mixte
* C054	Poke - 16300,0	Page 1
* C055	Poke - 16299,0	Page 2
* C056	Poke - 16298,0	Graphique Basse Résolution
* C057	Poke - 16297,0	Graphique Haute Résolution

TABLEAU 2

ADRESSES HEXA	ADRESSES DECIMALES	FONCTION
400 à 7FF	1024 à 2047	Page 1 texte ou graphique B R
800 à BFF	2048 à 3071	Page 2 " "
2000 à 3FFF	8192 à 16383	Page 1 Graphique haute résolution
4000 à 5FFF	16384 à 24575	Page 2 " "

TABLEAU 3

ADRESSES HEXADECIMALES	ADRESSES DECIMALES	
478 - 47F	1144 - 1151	Adresses libres en page 1
4F8 - 4FF	1272 - 1279	
578 - 57F	1400 - 1407	
5F8 - 5FF	1528 - 1535	
678 - 67F	1656 - 1663	
6F8 - 6FF	1784 - 1791	
778 - 77F	1912 - 1919	
7F8 - 7FF	2040 - 2047	
878 - 87F	2168 - 2175	Adresses libres en page 2
8F8 - 8FF	2296 - 2303	
978 - 97F	2424 - 2431	
9F8 - 9FF	2552 - 2559	
A78 - A7F	2680 - 2687	
AF8 - AFF	2808 - 2815	
B78 - B7F	2936 - 2943	
BF8 - BFF	3064 - 3071	

Tableau 4	COLONNE 1		COLONNE 21		COLONNE 40	
	LIGNES	ADR.HEXA	ADR.DEC	ADR.HEXA	ADR.DEC	ADR.HEXA
0-1	400	1024	414	1044	427	1063
2-3	480	1152	494	1172	4A7	1191
4-5	500	1280	514	1300	527	1319
6-7	580	1408	594	1428	5A7	1447
8-9	600	1536	614	1556	627	1575
10-11	680	1664	694	1684	6A7	1703
12-13	700	1792	714	1812	727	1831
14-15	780	1920	794	1940	7A7	1959
16-17	428	1064	43C	1084	44F	1103
18-19	4A8	1192	48C	1212	4CF	1231
20-21	528	1320	53C	1340	54F	1359
22-23	5A8	1448	58C	1468	5CF	1487
24-25	628	1576	63C	1596	64F	1615
26-27	6A8	1704	68C	1724	6CF	1743
28-29	728	1832	73C	1852	74F	1871
30-31	7A8	1960	78C	1980	7CF	1999
32-33	450	1104	464	1124	477	1143
34-35	400	1232	4E4	1252	4F7	1271
36-37	550	1360	564	1380	577	1399
38-39	500	1488	5E4	1508	5F7	1527
40-41	650	1616	664	1636	677	1655
42-43	600	1744	6E4	1764	6F7	1783
44-45	750	1872	764	1892	777	1911
46-47	700	2000	7E4	2020	7F7	2039

●●●/●●●

Des boucles à s'attacher les cheveux !

Alexandre Duback

Essayez de faire fonctionner le programme suivant; vous constaterez qu'il fonctionne parfaitement bien, puisqu'il ne fait rien de mal et qu'on ne lui en demande pas plus.

```

10 FOR I=1 TO 3 : IF I=2 THEN 30
20 NEXT I
30 REM DEBUT DES BOUCLES DE TEST
40 FOR I=1 TO 3 : FOR J=1 TO 2 : NEXT J : NEXT I
50 FOR J=1 TO 2 : FOR I=1 TO 3 : NEXT I : NEXT J

```

Otez maintenant l'instruction 40, et relancez le programme. Vous vous apercevrez alors avec stupeur que votre Apple favori vous envoie une injure à la figure :
NEXT WITHOUT FOR ERROR IN 50

Cela provient d'un bug rencontré dans la plupart des interpréteurs BASIC de Microsoft: cette erreur survient quand une sortie de boucle sur un indice (ici, I=2) est suivie d'une boucle imbriquée dans laquelle cet indice est interne.

Conclusion : si vous aimez les sorties de boucles, modifiez-les de la façon dont nous le faisons ci-dessous pour la ligne 10 :

```

10 FOR I=1 TO 3 : IF I=2 THEN I=3 : NEXT I : GOTO 30

```

TABLEAU 5 (A)

ADRESSES DECIMALES

0 8192- 8231	48 8960- 8999	96 8744- 8783	144 8528- 8567
1 9216- 9255	49 9984-10023	97 9768- 9807	145 9552- 9591
2 10240-10279	50 11008-11047	98 10792-10831	146 10576-10615
3 11264-11303	51 12032-12071	99 11816-11855	147 11600-11639
4 12288-12327	52 13056-13095	100 12840-12879	148 12624-12663
5 13312-13351	53 14080-14119	101 13864-13903	149 13648-13687
6 14336-14375	54 15104-15143	102 14888-14927	150 14672-14711
7 15360-15399	55 16128-16167	103 15912-15951	151 15696-15735
8 8320- 8359	56 9088- 9127	104 8872- 8911	152 8656- 8695
9 9344- 9383	57 10112-10151	105 9896- 9935	153 9680- 9719
10 10368-10407	58 11136-11175	106 10920-10959	154 10704-10743
11 11392-11431	59 12160-12199	107 11944-11983	155 11728-11767
12 12416-12455	60 13184-13223	108 12968-13007	156 12752-12791
13 13440-13479	61 14208-14247	109 13992-14031	157 13776-13815
14 14464-14503	62 15232-15271	110 15016-15055	158 14800-14839
15 15488-15527	63 16256-16295	111 16040-16079	159 15824-15863
16 8448- 8487	64 8232- 8271	112 9000- 9039	160 8784- 8823
17 9472- 9511	65 9256- 9295	113 10024-10063	161 9808- 9847
18 10496-10535	66 10280-10319	114 11048-11087	162 10832-10871
19 11520-11559	67 11304-11343	115 12072-12111	163 11856-11895
20 12544-12583	68 12328-12367	116 13096-13135	164 12880-12919
21 13568-13607	69 13352-13391	117 14120-14159	165 13904-13943
22 14592-14631	70 14376-14415	118 15144-15183	166 14928-14967
23 15616-15655	71 15400-15439	119 16168-16207	167 15952-15991
24 8576- 8615	72 8360- 8399	120 9128- 9167	168 8912- 8951
25 9600- 9639	73 9384- 9423	121 10152-10191	169 9936- 9975
26 10624-10663	74 10408-10447	122 11176-11215	170 10960-10999
27 11648-11687	75 11432-11471	123 12200-12239	171 11984-12023
28 12672-12711	76 12456-12495	124 13224-13263	172 13008-13047
29 13696-13735	77 13480-13519	125 14248-14287	173 14032-14071
30 14720-14759	78 14504-14543	126 15272-15311	174 15056-15095
31 15744-15783	79 15528-15567	127 16296-16335	175 16080-16119
32 8704- 8743	80 8488- 8527	128 8272- 8311	176 9040- 9079
33 9728- 9767	81 9512- 9551	129 9296- 9335	177 10064-10103
34 10752-10791	82 10536-10575	130 10320-10359	178 11088-11127
35 11776-11815	83 11560-11599	131 11344-11383	179 12112-12151
36 12800-12839	84 12584-12623	132 12368-12407	180 13136-13175
37 13824-13863	85 13608-13647	133 13392-13431	181 14160-14199
38 14848-14887	86 14632-14671	134 14416-14455	182 15184-15223
39 15872-15911	87 15656-15695	135 15440-15479	183 16208-16247
40 8652- 8691	88 8616- 8655	136 8400- 8439	184 9168- 9207
41 9856- 9895	89 9640- 9679	137 9424- 9463	185 10192-10231
42 10880-10919	90 10664-10703	138 10448-10487	186 11216-11255
43 11904-11943	91 11688-11727	139 11472-11511	187 12240-12279
44 12928-12967	92 12712-12751	140 12496-12535	188 13264-13303
45 13952-13991	93 13736-13775	141 13520-13559	189 14288-14327
46 14976-15015	94 14760-14799	142 14544-14583	190 15312-15351
47 16000-16039	95 15784-15823	143 15568-15607	191 16336-16375

TABLEAU 5 (E)

ADRESSES HEXADÉCIMALES			
0	2000-2027	48	2300-2327
1	2400-2427	49	2700-2727
2	2800-2827	50	2B00-2B27
3	2C00-2C27	51	2F00-2F27
4	3000-3027	52	3300-3327
5	3400-3427	53	3700-3727
6	3800-3827	54	3B00-3B27
7	3C00-3C27	55	3F00-3F27
8	2080-20A7	56	2380-23A7
9	2480-24A7	57	2780-27A7
10	2880-28A7	58	2B80-2BA7
11	2C80-2CA7	59	2F80-2FA7
12	3080-30A7	60	3380-33A7
13	3480-34A7	61	3780-37A7
14	3880-38A7	62	3B80-3BA7
15	3C80-3CA7	63	3F80-3FA7
16	2100-2127	64	2028-204F
17	2500-2527	65	2428-244F
18	2900-2927	66	2828-284F
19	2D00-2D27	67	2C28-2C4F
20	3100-3127	68	3028-304F
21	3500-3527	69	3428-344F
22	3900-3927	70	3828-384F
23	3D00-3D27	71	3C28-3C4F
24	2180-21A7	72	20A8-20CF
25	2580-25A7	73	24A8-24CF
26	2980-29A7	74	28A8-28CF
27	2D80-2DA7	75	2CA8-2CCF
28	3180-31A7	76	30A8-30CF
29	3580-35A7	77	34A8-34CF
30	3980-39A7	78	38A8-38CF
31	3D80-3DA7	79	3CA8-3CCF
32	2200-2227	80	2128-214F
33	2600-2627	81	2528-254F
34	2A00-2A27	82	2928-294F
35	2E00-2E27	83	2D28-2D4F
36	3200-3227	84	3128-314F
37	3600-3627	85	3528-354F
38	3A00-3A27	86	3928-394F
39	3E00-3E27	87	3D28-3D4F
40	2280-22A7	88	21A8-21CF
41	2680-26A7	89	25A8-25CF
42	2A80-2AA7	90	29A8-29CF
43	2E80-2EA7	91	2DA8-2DCF
44	3280-32A7	92	31A8-31CF
45	3680-36A7	93	35A8-35CF
46	3A80-3AA7	94	39A8-39CF
47	3E80-3EA7	95	3DA8-3DCF
		96	2228-224F
		97	2628-264F
		98	2A28-2A4F
		99	2E28-2E4F
		100	3228-324F
		101	3628-364F
		102	3A28-3A4F
		103	3E28-3E4F
		104	22A8-22CF
		105	26A8-26CF
		106	2AA8-2ACF
		107	2EA8-2ECF
		108	32A8-32CF
		109	36A8-36CF
		110	3AA8-3ACF
		111	3EA8-3ECF
		112	2328-234F
		113	2728-274F
		114	2B28-2B4F
		115	2F28-2F4F
		116	3328-334F
		117	3728-374F
		118	3B28-3B4F
		119	3F28-3F4F
		120	23A8-23CF
		121	27A8-27CF
		122	2BA8-2BCF
		123	2FA8-2FCF
		124	33A8-33CF
		125	37A8-37CF
		126	3BA8-3BCF
		127	3FA8-3FCF
		128	2050-2077
		129	2450-2477
		130	2850-2877
		131	2C50-2C77
		132	3050-3077
		133	3450-3477
		134	3850-3877
		135	3C50-3C77
		136	20D0-20F7
		137	24D0-24F7
		138	28D0-28F7
		139	2CD0-2CF7
		140	30D0-30F7
		141	34D0-34F7
		142	38D0-38F7
		143	3CD0-3CF7
		144	2150-2177
		145	2550-2577
		146	2950-2977
		147	2D50-2D77
		148	3150-3177
		149	3550-3577
		150	3950-3977
		151	3D50-3D77
		152	21D0-21F7
		153	25D0-25F7
		154	29D0-29F7
		155	2DD0-2DF7
		156	31D0-31F7
		157	35D0-35F7
		158	39D0-39F7
		159	3DD0-3DF7
		160	2250-2277
		161	2650-2677
		162	2A50-2A77
		163	2E50-2E77
		164	3250-3277
		165	3650-3677
		166	3A50-3A77
		167	3E50-3E77
		168	22D0-22F7
		169	26D0-26F7
		170	2AD0-2AF7
		171	2ED0-2EF7
		172	32D0-32F7
		173	36D0-36F7
		174	3AD0-3AF7
		175	3ED0-3EF7
		176	2350-2377
		177	2750-2777
		178	2B50-2B77
		179	2F50-2F77
		180	3350-3377
		181	3750-3777
		182	3B50-3B77
		183	3F50-3F77
		184	23D0-23F7
		185	27D0-27F7
		186	2BD0-2BF7
		187	2FD0-2FF7
		188	33D0-33F7
		189	37D0-37F7
		190	3BD0-3BF7
		191	3FD0-3FF7

Les Codes ASCII épluchés

Jean-François Duvivier

Certains de nos lecteurs nous ont reproché de copier des articles déjà parus dans des revues étrangères. La critique est aisée, mais l'art est difficile ! De par leur nombre, leur périodicité et leur antériorité, les revues américaines ont accumulé un nombre impressionnant d'articles. Il serait assurément possible à un collectionneur assidu des Softalk, Call-Apple, Nibble, Compute et autres revues de trouver pour la majorité des articles parus dans Pom's un article déjà publié traitant du même sujet ou ayant des ressemblances.

Devrions-nous nous abstenir de publier des articles sur le DOS, des comparaisons entre les systèmes de traitements de texte, une initiation à l'assembleur, etc... sous prétexte que des choses semblables ont été vues dans des revues étrangères ? Outre que nos lecteurs ne lisent pas tous l'anglais couramment, on n'a pas toujours le loisir d'avalier (et de digérer ...) une dizaine de revues par mois. Nous tenons à vous offrir une revue intéressante, vivante et utile, avec le maximum d'articles originaux. Mais, lorsque nous trouvons des choses intéressantes chez les autres, pourquoi ne pas vous en faire profiter ? C'est le cas ici, puisque les tableaux 1 à 4 nous ont été inspirés par un article de la revue Micro, the 6502 and 6809 Journal, du mois d'octobre 1981.

Nous espérons que ces tableaux vous aideront et qu'ils seront en bonne place dans votre aide-mémoire. Afin que leur utilisation vous soit plus facile, voici ci-dessous les explications concernant chaque colonne :

- * HEXA : contient tous les nombres hexadécimaux de \$00 à \$FF.
- * DECIMAL : traduction décimale du nombre hexadécimal.
- * BINAIRE : octet de 8 bits représentatif du nombre en hexa.
- * DEC*256 : valeur décimale de l'octet haut d'un nombre ou d'une adresse 16 bits.
- * ASCII : signe ASCII attribué au nombre hexa correspondant. Ce code étant exprimé en 7 bits, les signes sur les valeurs de \$00 à \$7F sont identiques à ceux des valeurs de \$80 à \$FF.
- * ECRAN : le manuel de référence donne en page 15 la correspondance exacte entre la valeur hexa et sa représentation sur l'écran. Les lettres I, F et N représentent les modes

Inverse, Flash et Normal. Les valeurs non affichées à l'écran sont remplacées par des tirets.

* TOUCHE : indique la touche à appuyer afin d'obtenir la valeur correspondante. A noter que les valeurs \$00 à \$7F ne peuvent être obtenues à partir du clavier. Les lettres C, S et CS représentent les touches Control, Shift et la combinaison des deux.

* OPCODES : représente en mnémoniques 6502 la signification des différentes valeurs hexa, dans le cas du langage machine. Les notations suivantes ont été utilisées :

A	absolu	Im	immédiat
A,X	absolu indexé par X	(I)	indirect
A,Y	absolu indexé par Y	ZP	page zéro
(I,X)	indirect indexé par X		
(I,Y)	indirect indexé par Y		
ZP,X	page zéro indexé par X		
ZP,Y	page zéro indexé par Y		

* TOKEN : le BASIC Integer et l'Applesoft utilisent, afin de diminuer l'encombrement mémoire, un codage des mots réservés du BASIC. Les codes \$00 à \$7F sont réservés à l'Integer, ceux entre \$80 et \$FF à l'Applesoft. Dans le cas de l'Integer, où un même signe peut avoir plusieurs codes selon le contexte, un court exemple est donné à droite de la colonne.

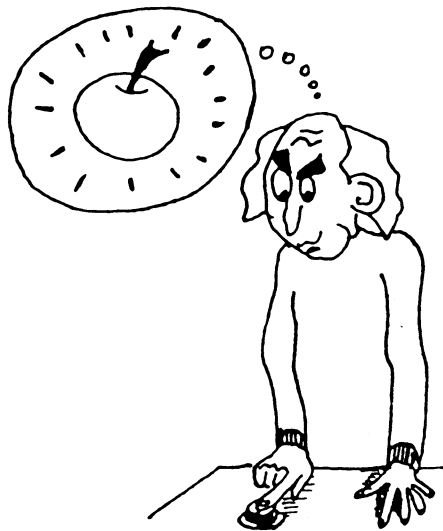


TABLEAU 1 (\$00 a \$3F)

HEXA	DEC	BINAIRE	DEC\$256	ASCII	ECRAN	TOUCH	OPCODE-6502	TOKEN INTEGER
00	0	00000000	0	nul	I:␣	-	BRK	Debut de ligne
01	1	00000001	256	soh	I:A	-	ORA-(I,X)	Fin de ligne
02	2	00000010	512	stx	I:B	-	---	Non utilise
03	3	00000011	768	etx	I:C	-	---	: separateur
04	4	00000100	1024	eot	I:D	-	---	LOAD K7
05	5	00000101	1280	enq	I:E	-	ORA-ZP	SAVE K7
06	6	00000110	1536	ack	I:F	-	ASL-ZP	CON
07	7	00000111	1792	bel	I:G	-	---	RUN ligne n
08	8	00001000	2048	bs	I:H	-	PHP	RUN debut
09	9	00001001	2304	ht	I:I	-	ORA-Im	DEL
0A	10	00001010	2560	lf	I:J	-	ASL	, DEL 0,5
0B	11	00001011	2816	vt	I:K	-	---	NEW
0C	12	00001100	3072	ff	I:L	-	---	CLR
0D	13	00001101	3328	cr	I:M	-	ORA-A	AUTO
0E	14	00001110	3584	so	I:N	-	ASL-A	, AUTO 0,5
0F	15	00001111	3840	si	I:O	-	---	MAN
10	16	00010000	4096	dle	I:P	-	BPL	HIMEM:
11	17	00010001	4352	dc1	I:Q	-	ORA-(I),Y	LOMEM:
12	18	00010010	4608	dc2	I:R	-	---	+ operateurs
13	19	00010011	4864	dc3	I:S	-	---	- numeriques
14	20	00010100	5120	dc4	I:T	-	---	* par exemple:
15	21	00010101	5376	nak	I:U	-	ORA-ZP,X	/ A=14*(27+15)
16	22	00010110	5632	syn	I:V	-	ASL-ZP,X	=
17	23	00010111	5888	etb	I:W	-	---	* operateurs
18	24	00011000	6144	can	I:X	-	CLC	>= logiques
19	25	00011001	6400	em	I:Y	-	ORA-A,Y	> pour
1A	26	00011010	6656	sub	I:Z	-	---	<= variables
1B	27	00011011	6912	esc	I:[-	---	<> numeriques
1C	28	00011100	7168	fs	I:\	-	---	< par exemple!
1D	29	00011101	7424	gs	I:]	-	ORA-A,X	AND IF X>=12
1E	30	00011110	7680	rs	I:^	-	ASL-A,X	OR THEN...
1F	31	00011111	7936	us	I:_	-	---	MOD
20	32	00100000	8192		I:~	-	JSR	^
21	33	00100001	8448	!	I:!	-	AND-(I,X)	Non utilise
22	34	00100010	8704	"	I:"	-	---	(DIM A\$(3)
23	35	00100011	8960	#	I:#	-	---	, A\$(3,3)
24	36	00100100	9216	\$	I:\$	-	BIT-ZP	THEN IFX=3THEN10!
25	37	00100101	9472	%	I:%	-	AND-ZP	THEN IFX=3THENA=2!
26	38	00100110	9728	&	I:&	-	ROL-ZP	, INPUT"ST",A\$!
27	39	00100111	9984	'	I:'	-	---	, INPUT"ST",A!
28	40	00101000	10240	(I:(-	PLP	" D(but!
29	41	00101001	10496)	I:)	-	AND-Im	" Fin!
2A	42	00101010	10752	*	I:*	-	ROL	(A\$(3)
2B	43	00101011	11008	+	I:+	-	---	Non utilise
2C	44	00101100	11264	,	I:,	-	BIT-A	Non utilise
2D	45	00101101	11520	-	I:-	-	AND-A	(A(3)
2E	46	00101110	11776	.	I:.	-	ROL-A	PEEK
2F	47	00101111	12032	/	I:/	-	---	RND
30	48	00110000	12288	0	I:0	-	BMI	SGN
31	49	00110001	12544	1	I:1	-	AND-(I),Y	ABS
32	50	00110010	12800	2	I:2	-	---	PDL
33	51	00110011	13056	3	I:3	-	---	Non utilise
34	52	00110100	13312	4	I:4	-	---	(DIM A(3)
35	53	00110101	13568	5	I:5	-	AND-ZP,X	+ A=+3!
36	54	00110110	13824	6	I:6	-	ROL-ZP,X	- A=-3!
37	55	00110111	14080	7	I:7	-	---	NOT
38	56	00111000	14336	8	I:8	-	SEC	(
39	57	00111001	14592	9	I:9	-	AND-A,Y	= IFA\$="C"THEN...
3A	58	00111010	14848	:	I>:	-	---	* IFA\$#"C"THEN...
3B	59	00111011	15104	;	I:;	-	---	LEN(
3C	60	00111100	15360	<	I:<	-	---	ASC(
3D	61	00111101	15616	=	I:=	-	AND-A,X	SCRN(
3E	62	00111110	15872	>	I:>	-	ROL-A,X	, SCRN(3,5)
3F	63	00111111	16128	?	I:?	-	---	(

TABLEAU 2 (\$40 a \$7F)

HEXA	DEC	BINAIRE	DEC*256	ASCII	ECRAN	TOUCH	OPCODE-6502	TOKEN INTEGER
40	64	01000000	16384	␣	F:␣	-	RTI	\$ A\$
41	65	01000001	16640	A	F:A	-	EOR-(I,X)	Non utilise
42	66	01000010	16896	B	F:B	-	---	(
43	67	01000011	17152	C	F:C	-	---	,
44	68	01000100	17408	D	F:D	-	---	;
45	69	01000101	17664	E	F:E	-	EOR-ZP	;
46	70	01000110	17920	F	F:F	-	LSR-ZP	;
47	71	01000111	18176	G	F:G	-	---	;
48	72	01001000	18432	H	F:H	-	PHA	,
49	73	01001001	18688	I	F:I	-	EOR-Im	,
4A	74	01001010	18944	J	F:J	-	LSR	,
4B	75	01001011	19200	K	F:K	-	---	TEXT
4C	76	01001100	19456	L	F:L	-	JMP-A	GR
4D	77	01001101	19712	M	F:M	-	EOR-A	CALL
4E	78	01001110	19968	N	F:N	-	LSR-A	DIM DIM A\$(3)
4F	79	01001111	20224	O	F:O	-	---	DIM DIM A(3)
50	80	01010000	20480	P	F:P	-	BVC	TAB
51	81	01010001	20736	Q	F:Q	-	EOR-(I),Y	END
52	82	01010010	20992	R	F:R	-	---	INPUT INPUT A\$
53	83	01010011	21248	S	F:S	-	---	INPUT INPUT*T",A\$
54	84	01010100	21504	T	F:T	-	---	INPUT INPUT A
55	85	01010101	21760	U	F:U	-	EOR-ZP,X	FOR
56	86	01010110	22016	V	F:V	-	LSR-ZP,X	= FOR I=1 TO 3
57	87	01010111	22272	W	F:W	-	---	TO
58	88	01011000	22528	X	F:X	-	CLI	STEP
59	89	01011001	22784	Y	F:Y	-	EOR-A,Y	NEXT
5A	90	01011010	23040	Z	F:Z	-	---	, NEXT I,J
5B	91	01011011	23296	[F:[-	---	RETURN
5C	92	01011100	23552	\	F:\	-	---	GOSUB
5D	93	01011101	23808]	F:]	-	EOR-A,X	REM
5E	94	01011110	24064	^	F:^	-	LSR-A,X	LET
5F	95	01011111	24320	-	F:-	-	---	GOTO
60	96	01100000	24576		F:	-	RTS	IF
61	97	01100001	24832	a	F:a	-	ADC-(I,X)	PRINT PRINT A\$
62	98	01100010	25088	b	F:b	-	---	PRINT PRINT A
63	99	01100011	25344	c	F:c	-	---	PRINT PRINT
64	100	01100100	25600	d	F:d	-	---	POKE
65	101	01100101	25856	e	F:e	-	ADC-ZP	, POKE 3,3
66	102	01100110	26112	f	F:f	-	ROR-ZP	COLOR=
67	103	01100111	26368	g	F:g	-	---	PLOT
68	104	01101000	26624	h	F:h	-	PLA	, PLOT 3,3
69	105	01101001	26880	i	F:i	-	ADC-Im	HLIN
6A	106	01101010	27136	j	F:j	-	ROR	, HLIN 3,5 AT
6B	107	01101011	27392	k	F:k	-	---	AT HLIN 3,5 AT
6C	108	01101100	27648	l	F:l	-	JMP-(I)	VLIN
6D	109	01101101	27904	m	F:m	-	ADC-A	, VLIN 3,5 AT
6E	110	01101110	28160	n	F:n	-	ROR-A	AT VLIN 3,5 AT
6F	111	01101111	28416	o	F:o	-	---	VTAB
70	112	01110000	28672	p	F:p	-	BVS	= A\$="HELLO"
71	113	01110001	28928	q	F:q	-	ADC-(I),Y	= A=3
72	114	01110010	29184	r	F:r	-	---)
73	115	01110011	29440	s	F:s	-	---	Non utilise
74	116	01110100	29696	t	F:t	-	---	LIST LIST 3,5
75	117	01110101	29952	u	F:u	-	ADC-ZP,X	, LIST 3,5
76	118	01110110	30208	v	F:v	-	ROR-ZP,X	LIST LIST
77	119	01110111	30464	w	F:w	-	---	POP
78	120	01111000	30720	x	F:x	-	SEI	NODSP NODSP A\$
79	121	01111001	30976	y	F:y	-	ADC-A,Y	NODSP NODSP A
7A	122	01111010	31232	z	F:z	-	---	NOTRACE
7B	123	01111011	31488	{	F:{	-	---	DSP DSP A\$
7C	124	01111100	31744	}	F:}	-	---	DSP DSP A
7D	125	01111101	32000	~	F:~	-	ADC-A,X	TRACE
7E	126	01111110	32256	rub	F:rub	-	ROR-A,X	PR#
7F	127	01111111	32512	rub	F:rub	-	---	IN#

TABLEAU 3 (\$80 a \$BF)

HEXA	DEC	BINAIRE	DEC*256	ASCII	ECRAN	TOUCH	OPCODE-6502	TOKEN APPLESOFT
80	128	10000000	32768	nul	N:␣	---	---	END
81	129	10000001	33024	soh	N:A	CA	STA-(I,X)	FOR
82	130	10000010	33280	stx	N:B	CB	---	NEXT
83	131	10000011	33536	etx	N:C	CC	---	DATA
84	132	10000100	33792	eot	N:D	CD	STY-ZP	INPUT
85	133	10000101	34048	enq	N:E	CE	STA-ZP	DEL
86	134	10000110	34304	ack	N:F	CF	STY-ZP	DIM
87	135	10000111	34560	bel	N:G	CG	---	READ
88	136	10001000	34816	bs	N:H	CH	DEY	GR
89	137	10001001	35072	ht	N:I	CI	---	TEXT
8A	138	10001010	35328	lf	N:J	CJ	TXA	PR#
8B	139	10001011	35584	vt	N:K	CK	---	IN#
8C	140	10001100	35840	ff	N:L	CL	STY-A	CALL
8D	141	10001101	36096	cr	N:M	CM	STA-A	PLOT
8E	142	10001110	36352	so	N:N	CN	STX-A	HLIN
8F	143	10001111	36608	si	N:O	CO	---	VLIN
90	144	10010000	36864	dle	N:P	CP	BCC	HGR2
91	145	10010001	37120	dc1	N:Q	CQ	STA-(I),Y	HGR
92	146	10010010	37376	dc2	N:R	CR	---	HCOLOR=
93	147	10010011	37632	dc3	N:S	CS	---	HPLLOT
94	148	10010100	37888	dc4	N:T	CT	STY-ZP,X	DRAW
95	149	10010101	38144	nak	N:U	CU	STA-ZP,X	XDRAW
96	150	10010110	38400	syn	N:V	CV	STX-ZP,Y	HTAB
97	151	10010111	38656	etb	N:W	CW	---	HOME
98	152	10011000	38912	can	N:X	CX	TYA	ROT=
99	153	10011001	39168	em	N:Y	CY	STA-A,Y	SCALE=
9A	154	10011010	39424	sub	N:Z	CZ	TXS	SHLOAD
9B	155	10011011	39680	esc	N:[esc	---	TRACE
9C	156	10011100	39936	fs	N:\	---	---	NOTRACE
9D	157	10011101	40192	gs	N:]	CSM	STA-A,X	NORMAL
9E	158	10011110	40448	rs	N:^	CSN	---	INVERSE
9F	159	10011111	40704	us	N:_	---	---	FLASH
A0	160	10100000	40960	!	N:!	esp	LDY-Im	COLOR=
A1	161	10100001	41216	"	N:"	S1	LDA-(I,X)	POP
A2	162	10100010	41472	#	N:#	S2	LDX-Im	VTAB
A3	163	10100011	41728	\$	N:\$	S3	---	HIMEM:
A4	164	10100100	41984	%	N:%	S4	LDY-ZP	LOMEM:
A5	165	10100101	42240	&	N:&	S5	LDA-ZP	ONERR
A6	166	10100110	42496	'	N:'	S6	LDX-ZP	RESUME
A7	167	10100111	42752	(N:(S7	---	RECALL
A8	168	10101000	43008)	N:)	S8	TAY	STORE
A9	169	10101001	43264	*	N:*	S9	LDA-Im	SPEED=
AA	170	10101010	43520	+	N:+	S:	TAX	LET
AB	171	10101011	43776	,	N:,	S;	---	GOTO
AC	172	10101100	44032	-	N:-	,	LDY-A	RUN
AD	173	10101101	44288	.	N:.	-	LDA-A	IF
AE	174	10101110	44544	/	N:/	.	LDX-A	RESTORE
AF	175	10101111	44800	0	N:0	/	---	&
B0	176	10110000	45056	1	N:1	0	BCS	GOSUB
B1	177	10110001	45312	2	N:2	1	LDA-(I),Y	RETURN
B2	178	10110010	45568	3	N:3	2	---	REM
B3	179	10110011	45824	4	N:4	3	---	STOP
B4	180	10110100	46080	5	N:5	4	LDY-ZP,X	ON
B5	181	10110101	46336	6	N:6	5	LDA-ZP,X	WAIT
B6	182	10110110	46592	7	N:7	6	LDX-ZP,Y	LOAD
B7	183	10110111	46848	8	N:8	7	---	SAVE
B8	184	10111000	47104	9	N:9	8	CLV	DEF
B9	185	10111001	47360	:	N>:	9	LDA-A,Y	POKE
BA	186	10111010	47616	;	N:;	:	TSX	PRINT
BB	187	10111011	47872	<	N:<	;	---	CONT
BC	188	10111100	48128	=	N:=	<	LDY-A,X	LIST
BD	189	10111101	48384	>	N:>	=	LDA-A,X	CLEAR
BE	190	10111110	48640	?	N:?	>	LDX-A,Y	GET
BF	191	10111111	48896			?	---	NEW

TABLEAU 4 (\$C0 a \$FF)

HEXA	DEC	BINAIRE	DEC*256	ASCII	ECRAN	TOUCH	OPCODE-6502	TOKEN APPLESOFT
C0	192	11000000	49152	␣	N:␣	SP	CPY-Im	TAB(
C1	193	11000001	49408	A	N:A	A	CMP-(I,X)	TO
C2	194	11000010	49664	B	N:B	B	---	FN
C3	195	11000011	49920	C	N:C	C	---	SPC(
C4	196	11000100	50176	D	N:D	D	CPY-ZP	THEN
C5	197	11000101	50432	E	N:E	E	CMP-ZP	AT
C6	198	11000110	50688	F	N:F	F	DEC-ZP	NOT
C7	199	11000111	50944	G	N:G	G	---	STEP
C8	200	11001000	51200	H	N:H	H	INY	+
C9	201	11001001	51456	I	N:I	I	CMP-Im	-
CA	202	11001010	51712	J	N:J	J	DEX	*
CB	203	11001011	51968	K	N:K	K	---	/
CC	204	11001100	5224	L	N:L	L	CPY-A	^
CD	205	11001101	52480	M	N:M	M	CMP-A	AND
CE	206	11001110	52736	N	N:N	N	DEC-A	OR
CF	207	11001111	52992	O	N:O	O	---	>
DO	208	11010000	53248	P	N:P	P	BNE	=
D1	209	11010001	53504	Q	N:Q	Q	CMP-(I),Y	<
D2	210	11010010	53760	R	N:R	R	---	SGN
D3	211	11010011	54016	S	N:S	S	---	INT
D4	212	11010100	54272	T	N:T	T	---	ABS
D5	213	11010101	54528	U	N:U	U	CMP-ZP,X	USR
D6	214	11010110	54784	V	N:V	V	DEC-ZP,X	FRE
D7	215	11010111	55040	W	N:W	W	---	SCRN(
D8	216	11011000	55296	X	N:X	X	CLD	PDL
D9	217	11011001	55552	Y	N:Y	Y	CMP-A,Y	POS
DA	218	11011010	55808	Z	N:Z	Z	---	SGR
DB	219	11011011	56064	[N:[---	---	RND
DC	220	11011100	56320	\	N:\	---	---	LOG
DD	221	11011101	56576]	N:]	SM	CMP-A,X	EXP
DE	222	11011110	56832	^	N:^	SN	DEC-A,X	COS
DF	223	11011111	57088	-	N:-	---	---	SIN
E0	224	11100000	57344	-	N:-	---	CPX-Im	TAN
E1	225	11100001	57600	a	N:a	---	SBC-(I,X)	ATN
E2	226	11100010	57856	b	N:b	---	---	PEEK
E3	227	11100011	58112	c	N:c	---	---	LEN
E4	228	11100100	58368	d	N:d	---	CPX-ZP	STR\$
E5	229	11100101	58624	e	N:e	---	SBC-ZP	VAL
E6	230	11100110	58880	f	N:f	---	INC-ZP	ASC
E7	231	11100111	59136	g	N:g	---	---	CHR\$
E8	232	11101000	59392	h	N:h	---	INX	LEFT\$
E9	233	11101001	59648	i	N:i	---	SBC-Im	RIGHT\$
EA	234	11101010	59904	j	N:j	---	NOP	MID\$
EB	235	11101011	60160	k	N:k	---	---	---
EC	236	11101100	60416	l	N:l	---	CPX-A	---
ED	237	11101101	60672	m	N:m	---	SBC-A	---
EE	238	11101110	60928	n	N:n	---	INC-A	---
EF	239	11101111	61184	o	N:o	---	---	---
F0	240	11110000	61440	p	N:p	---	BEQ	---
F1	241	11110001	61696	q	N:q	---	SBX-(I),Y	---
F2	242	11110010	61952	r	N:r	---	---	---
F3	243	11110011	62208	s	N:s	---	---	---
F4	244	11110100	62464	t	N:t	---	---	---
F5	245	11110101	62720	u	N:u	---	SBC-ZP,X	---
F6	246	11110110	62976	v	N:v	---	INC-ZP,X	---
F7	247	11110111	63232	w	N:w	---	---	---
F8	248	11111000	63488	x	N:x	---	SED	---
F9	249	11111001	63744	y	N:y	---	SBC-A,Y	---
FA	250	11111010	64000	z	N:z	---	---	---
FB	251	11111011	64256	{	N:{	---	---	---
FC	252	11111100	64512	:	N::	---	---	---
FD	253	11111101	64768	}	N:}	---	SBC-A,X	---
FE	254	11111110	65024	~	N:~	---	INC-A,X	---
FF	255	11111111	65280	rub	N:?	---	---	---