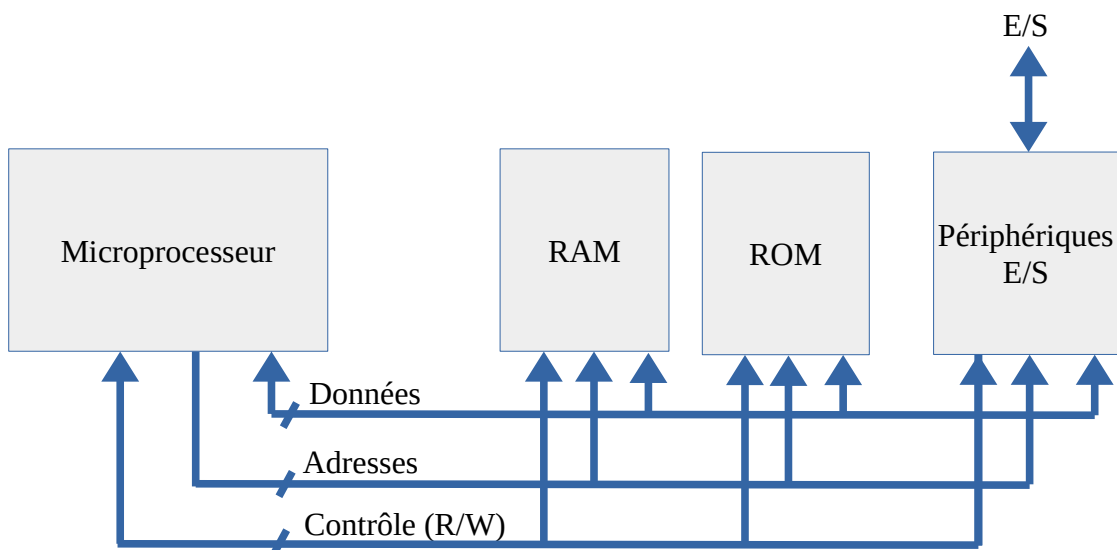


# Microprocesseur et programmation

## Le système minimum



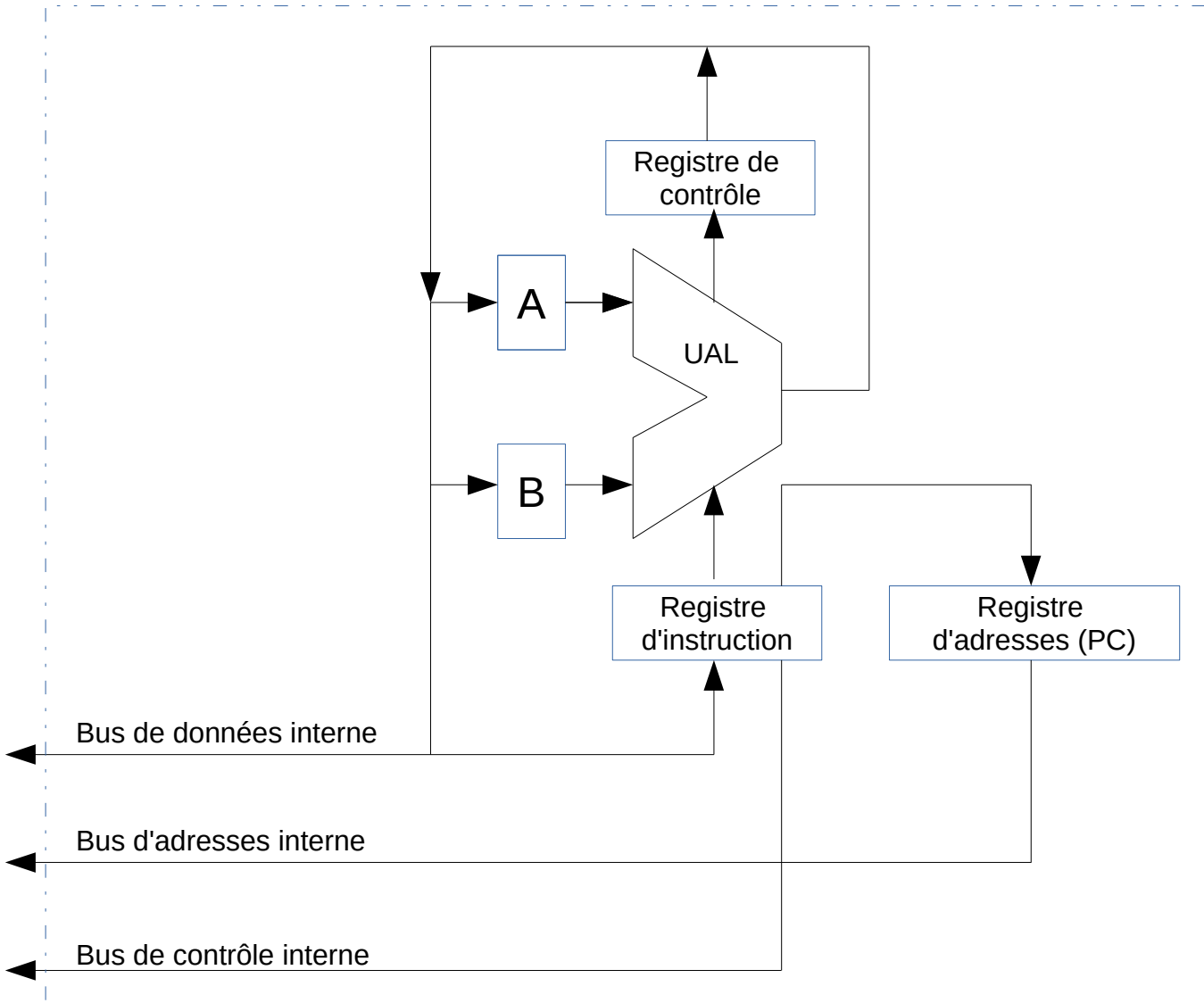
### Les composants :

- ROM : mémoire non volatile (données et programme)
- RAM : mémoire vive, volatile (données et programme)
- Microprocesseur : commande les circuits, exécute le programme

### Les bus :

- Données : bus bidirectionnel
- Adresses : produit par le microprocesseur. Définit la case mémoire (RAM, ROM ou périphérique) avec laquelle communique le microprocesseur.
- Contrôle : contient entre autres le fil de R/W qui définit le sens de transfert des données
  - R/W = 1 : lecture, transfert des données vers le processeur ;
  - R/W = 0 : écriture, transfert des données vers la RAM, la ROM ou les périphériques

## Le processeur



Description :

- Le registre d'instruction
  - mémorise et décode l'instruction ;
  - commande les autres éléments lors de l'exécution de l'instruction.
- Registre d'adresse
  - mémorise l'adresse de la prochaine instruction à exécuter (PC, Program Counter)
- A et B
  - Accumulateur ou registre temporaire
- l'UAL (Unité arithmétique et logique ou ALU pour Arithmetic Logic Unit)
  - c'est ce qui permet au microprocesseur de réaliser des calculs
- Registre de contrôle
  - Donne des informations sur le résultat du calcul ;
  - Bit C pour mémoriser la retenue ;
  - Bit Z pour tester si le résultat est nul ;
  - Bit N pour tester si le résultat est négatif ;
  - Bit V pour tester le dépassement de format.

# Les opérations de l'UAL

L'UAL de base réalise des opérations élémentaires sur des formats limités :

- mathématique
  - addition
  - soustraction
- logique
  - "ET"
  - "OU"
  - "XOR"
  - "NOT" complément à 1
- les décalages (multiplication et division par 2) ;
- ...

On parle de processeur 8 bits lorsque l'UAL réalise des opérations élémentaires sur 8 bits. Il en est de même pour les processeurs 16, 32 ou 64 bits.

Certains microprocesseurs réalisent d'autres opérations : la multiplication, la division, le calcul avec des nombres à virgule.

Les opérations plus complexes se réalisent avec un programme.

## Les instructions d'un microprocesseur

Les instructions d'un microprocesseur peuvent réaliser :

- Une lecture
  - lecture de la RAM, la ROM et des périphériques
  - écriture dans les registres internes au microprocesseur
  - le bus de contrôle place le signal R/W à 1
- Une écriture
  - écriture dans la RAM, la ROM et des périphériques
  - lecture des registres internes au microprocesseur
  - le bus de contrôle place le signal R/W à 0
- Un calcul
  - les calculs sont réalisés dans l'UAL (les opérations de base sont donc limitées)
- Un test
  - par lecture du registre de contrôle de l'opération
  - Bit C pour mémoriser la retenue ;
  - Bit Z pour tester si le résultat est nul ;
  - Bit N pour tester si le résultat est négatif ;
  - Bit V pour tester le dépassement de format.
- Un saut ou branchement à une partie du programme

Les structures de test et de boucle (si alors sinon, pour, tant que ...) utilisent les instructions de test, de saut et de branchement.

# Programmations bas niveau et haut niveau

Les programmations assembleur ou machine sont dites programmation bas niveau. A l'opposé, les programmations de type python et en langage C sont dites de haut niveau.

Les programmes ci-dessous réalisent le même algorithme.

Langage machine	Langage assembleur	Langage python	Langage C
<pre>;\$0000 db \$00 ;\$0001 db \$05  NbN EQU \$0000 NbB EQU \$0001  prgpal: FC00 7F 0000 FC03 F6 0001  boucle: FC06 B6 1000 FC09 8B 02 FC0B B7 1000 FC0E 5A FC0F 26 F5</pre>	<pre>;\$0000 db \$00 ;\$0001 db \$05  NbN EQU \$0000 NbB EQU \$0001  prgpal: clr NbN ldb NbB  boucle: lda NbN adda #\$02 sta NbN decb bne boucle end</pre>	<pre>N = 0 B = 5 while B != 0 :     N = N + 2     B = B - 1</pre>	<pre>int N = 0 int B = 5  while (B != 0) {     N = N + 2 ;     B = B - 1 ; }</pre>

Le microprocesseur n'exécute que le programme en langage machine.

Les autres programmes doivent donc être traduits en langage machine (assemblage, compilation, interprétation).

Les codes machines sont consultables dans une table donnée par le constructeur.

## Exemple

Dans la table ci-dessous on peut retrouver les codes :

- B6 pour lda
- 8B pour adda
- 5A pour decb
- 26 pour bne
- ...

# Jeu d'instruction du 6809

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES

OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	
00	NEG	Direct ↑	6	2	30	LEAX	Indexed ↓	4+	2+	60	NEG	Indexed ↑	6+	2+	
01	*		02	*	31	LEAY		4+	2+	61	*				
03	COM		6	2	32	LEAS		4+	2+	62	*				
04	LSR		6	2	33	LEAU		Indexed	4+	2+	63		COM	6+	2+
05	*		6	2	34	PSHS		Immed	5+	2	64		LSR	6+	2+
06	ROR		6	2	35	PULS		Immed	5+	2	65		*	6+	2+
07	ASR		6	2	36	PSHU		Immed	5+	2	66		ROR	6+	2+
08	ASL, LSL		6	2	37	PULU		Immed	5+	2	67		ASR	6+	2+
09	ROL		6	2	38	*		—	—	—	68		ASL, LSL	6+	2+
0A	DEC		6	2	39	RTS		Inherent	5	1	69		ROL	6+	2+
0B	*		6	2	3A	ABX		↑	3	1	6A		DEC	6+	2+
0C	INC		6	2	3B	RTI		↓	6/15	1	6B		*	6+	2+
0D	TST		6	2	3C	CWAI		≥ 20	2	—	6C		INC	6+	2+
0E	JMP		3	2	3D	MUL		Inherent	11	1	6D		TST	6+	2+
0F	CLR		Direct ↓	6	2	3E		*	—	—	6E		JMP	3+	2+
						3F		SWI	Inherent	19	1		6F	CLR	Indexed ↓
10	Page 2	—	—	—	40	NEGA	Inherent ↑	2	1	70	NEG	Extended ↑	7	3	
11	Page 3	—	—	—	41	*		2	1	71	*				
12	NOP	Inherent	2	1	42	*		2	1	72	*		7	3	
13	SYNC	Inherent	≥ 4	1	43	COMA		2	1	73	COM		7	3	
14	*	—	—	—	44	LSRA		2	1	74	LSR		7	3	
15	*	—	—	—	45	*		2	1	75	*		7	3	
16	LBRA	Relative	5	3	46	RORA		2	1	76	ROR		7	3	
17	LBSR	Relative	9	3	47	ASRA		2	1	77	ASR		7	3	
18	*	—	—	—	48	ASLA, LSLA		2	1	78	ASL, LSL		7	3	
19	DAA	Inherent	2	1	49	ROLA		2	1	79	ROL		7	3	
1A	ORCC	Immed	3	2	4A	DECA		2	1	7A	DEC		7	3	
1B	*	—	—	—	4B	*		2	1	7B	*		7	3	
1C	ANDCC	Immed	3	2	4C	INCA		2	1	7C	INC		7	3	
1D	SEX	Inherent	2	1	4D	TSTA		2	1	7D	TST		7	3	
1E	EXG	Immed	8	2	4E	*		2	1	7E	JMP		4	3	
1F	TFR	Immed	6	2	4F	CLRA		Inherent	2	1	7F		CLR	Extended ↓	7
20	BRA	Relative ↑	3	2	50	NEGB	Inherent ↑	2	1	80	SUBA	Immed ↑	2	2	
21	BRN		3	2	51	*		2	1	81	CMPA		2	2	
22	BHI		3	2	52	*		2	1	82	SBCA		2	2	
23	BLS		3	2	53	COMB		2	1	83	SUBD		4	3	
24	BHS, BCC		3	2	54	LSRB		2	1	84	ANDA		2	2	
25	BLO, BCS		3	2	55	*		2	1	85	BITA		2	2	
26	BNE		3	2	56	RORB		2	1	86	LDA		2	2	
27	BEQ		3	2	57	ASRB		2	1	87	*		2	2	
28	BVC		3	2	58	ASLB, LSLB		2	1	88	EORA		2	2	
29	BVS		3	2	59	ROLB		2	1	89	ADCA		2	2	
2A	BPL		3	2	5A	DECB		2	1	8A	ORA		2	2	
2B	BMI		3	2	5B	*		2	1	8B	ADDA		2	2	
2C	BGE		3	2	5C	INCB		2	1	8C	CMPX		Immed ↓	4	3
2D	BLT		3	2	5D	TSTB		2	1	8D	BSR		Relative	7	2
2E	BGT		3	2	5E	*		2	1	8E	LDX		Immed	3	3
2F	BLE		Relative ↓	3	2	5F		CLRB	Inherent	2	1		8F	*	3

TABLE 9 – HEXADECIMAL VALUES OF MACHINE CODES (CONTINUED)

OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	
90	SUBA	Direct ↑ ↓	4	2	C0	SUBB	Immed ↑ ↓	2	2	Page 2 and 3 Machine Codes					
91	CMPA		4	2	C1	CMPB		2	2						
92	SBCA		4	2	C2	SBCB		2	2	1021	LBRN	Relative ↑ ↓	5	4	
93	SUBD		6	2	C3	ADDD		4	3	1022	LBHI		5(6)	4	
94	ANDA		4	2	C4	ANDB		2	2	1023	LBLS		5(6)	4	
95	BITA		4	2	C5	BITB		2	2	1024	LBHS, LBCC		5(6)	4	
96	LDA		4	2	C6	LDB		2	2	1025	LBCS, LBLO		5(6)	4	
97	STA		4	2	C7	*		2	2	1026	LBNE		5(6)	4	
98	EORA		4	2	C8	EORB		2	2	1027	LBEO		5(6)	4	
99	ADCA		4	2	C9	ADCB		2	2	1028	LBVC		5(6)	4	
9A	ORA	4	2	CA	ORB	2	2	1029	LBVS	5(6)	4				
9B	ADDA	4	2	CB	ADDB	2	2	102A	LBPL	5(6)	4				
9C	CMPX	6	2	CC	LDD	3	3	102B	LBMi	5(6)	4				
9D	JSR	7	2	CD	*	3	3	102C	LBGE	5(6)	4				
9E	LDX	5	2	CE	LDU	3	3	102D	LBTL	5(6)	4				
9F	STX	5	2	CF	*			102E	LBGT	5(6)	4				
A0	SUBA	Indexed ↑ ↓	4+	2+	D0	SUBB	Direct ↑ ↓	4	2	102F	LBLE	Relative ↑ ↓	5(6)	4	
A1	CMPA		4+	2+	D1	CMPB		4	2	103F	SWI2		Inherent	20	2
A2	SBCA		4+	2+	D2	SBCB		4	2	1083	CMPD		Immed	5	4
A3	SUBD		6+	2+	D3	ADDD		6	2	108C	CMPY		Immed	5	4
A4	ANDA		4+	2+	D4	ANDB		4	2	108E	LDY		Immed	4	4
A5	BITA		4+	2+	D5	BITB		4	2	1093	CMPD		Direct	7	3
A6	LDA		4+	2+	D6	LDB		4	2	109C	CMPY		Direct	7	3
A7	STA		4+	2+	D7	STB		4	2	109E	LDY		Direct	6	3
A8	EORA		4+	2+	D8	EORB		4	2	109F	STY		Direct	6	3
A9	ADCA		4+	2+	D9	ADCB		4	2	10A3	CMPD		Indexed	7+	3+
AA	ORA		4+	2+	DA	ORB		4	2	10AC	CMPY		Indexed	7+	3+
AB	ADDA		4+	2+	DB	ADDB		4	2	10AE	LDY		Indexed	6+	3+
AC	CMPX		6+	2+	DC	LDD		5	2	10AF	STY		Indexed	6+	3+
AD	JSR		7+	2+	DD	STD		5	2	10B3	CMPD		Extended	8	4
AE	LDX		5+	2+	DE	LDU		5	2	10BC	CMPY		Extended	8	4
AF	STX		5+	2+	DF	STU		5	2	10BE	LDY		Extended	7	4
B0	SUBA	Extended ↑ ↓	5	3	E0	SUBB	Indexed ↑ ↓	4+	2+	10BF	STY	Extended ↑ ↓	7	4	
B1	CMPA		5	3	E1	CMPB		4+	2+	10CE	LDS		Immed	4	3
B2	SBCA		5	3	E2	SBCB		4+	2+	10DE	LDS		Direct	6	3
B3	SUBD		7	3	E3	ADDD		6+	2+	10DF	STS		Direct	6	3
B4	ANDA		5	3	E4	ANDB		4+	2+	10EE	LDS		Indexed	6+	3+
B5	BITA		5	3	E5	BITB		4+	2+	10EF	STS		Indexed	6+	3+
B6	LDA		5	3	E6	LDB		4+	2+	10FE	LDS		Extended	7	4
B7	STA		5	3	E7	STB		4+	2+	10FF	STS		Extended	7	4
B8	EORA		5	3	E8	EORB		4+	2+	113F	SWI3		Inherent	20	2
B9	ADCA		5	3	E9	ADCB		4+	2+	1183	CMPU		Immed	5	4
BA	ORA		5	3	EA	ORB		4+	2+	118C	CMPS		Immed	5	4
BB	ADDA		5	3	EB	ADDB		4+	2+	1193	CMPU		Direct	7	3
BC	CMPX		7	3	EC	LDD		5+	2+	119C	CMPS		Direct	7	3
BD	JSR		8	3	ED	STD		5+	2+	11A3	CMPU		Indexed	7+	3+
BE	LDX		6	3	EE	LDU		5+	2+	11AC	CMPS		Indexed	7+	3+
BF	STX		6	3	EF	STU		5+	2+	11B3	CMPU		Extended	8	4
										11BC	CMPS	Extended	8	4	
					F0	SUBB	Extended	5	3						
					F1	CMPB	↑	5	3						
					F2	SBCB	↑	5	3						
					F3	ADDD	↑	7	3						
					F4	ANDB	↑	5	3						
					F5	BITB	↑	5	3						
					F6	LDB	↑	5	3						
					F7	STB	↑	5	3						
					F8	EORB	↑	5	3						
					F9	ADCB	↑	5	3						
					FA	ORB	↓	5	3						
					FB	ADDB	↓	5	3						
					FC	LDD	↓	6	3						
					FD	STD	↓	6	3						
					FE	LDU	↓	6	3						
					FF	STU	↓	6	3						

NOTE: All unused opcodes are both undefined and illegal

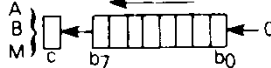
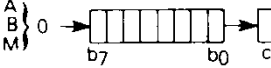
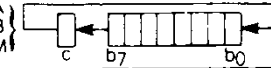
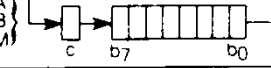
FIGURE 19 — PROGRAMMING AID

Instruction	Forms	Addressing Modes															Description	5	3	2	1	0						
		Immediate			Direct			Indexed			Extended			Inherent				H	N	Z	V	C						
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#												
ABX																3A	3	1	B + X → X (Unsigned)					•	•	•	•	•
ADC	ADCA ADCB	89 C9	2 2	2 2	99 D9	4 4	2 2	A9 E9	4+ 4+	2+ 2+	B9 F9	5 5	3 3				A + M + C → A B + M + C → B					1	1	1	1	1		
ADD	ADDA ADDB ADDD	8B CB C3	2 2 4	2 2 3	9B DB D3	4 4 6	2 2 2	AB EB E3	4+ 4+ 6+	2+ 2+ 2+	BB FB F3	5 5 7	3 3 3				A + M → A B + M → B D + M: M + 1 → D					1	1	1	1	1		
AND	ANDA ANDB ANDCC	84 C4 1C	2 2 3	2 2 2	94 D4	4 4	2 2	A4 E4	4+ 4+	2+ 2+	B4 F4	5 5	3 3				A ∧ M → A B ∧ M → B CC ∧ IMM → CC					•	1	1	0	•		
ASL	ASLA ASLB ASL					08	6	2	68	6+	2+	78	7	3	48 58	2 2	1 1						8	1	1	1	1	
ASR	ASRA ASRB ASR					07	6	2	67	6+	2+	77	7	3	47 57	2 2	1 1						8	1	1	•	1	
BIT	BITA BITB	85 C5	2 2	2 2	95 D5	4 4	2 2	A5 E5	4+ 4+	2+ 2+	B5 F5	5 5	3 3				Bit Test A (M ∧ A) Bit Test B (M ∧ B)					•	1	1	0	•		
CLR	CLRA CLRB CLR					0F	6	2	6F	6+	2+	7F	7	3	4F 5F	2 2	1 1	0 → A 0 → B 0 → M					•	0	1	0	0	
CMP	CMPA CMPB CMPD CMPS CMPU CMPX CMPY	81 C1 10 83 11 8C 11 83 8C 10 8C	2 2 5 5 5 4 5 4 4 5 4	2 2 4 4 4 3 4 3 3 4	91 D1 10 93 11 9C 11 93 9C 10 9C	4 4 7 10 7 7 7 6 7 7	2 2 3 3 3 2 3 2 3 3	A1 E1 10 A3 11 AC 11 A3 AC 10 AC	4+ 4+ 7+ 4+ 7+ 6+ 7+ 6+ 7+ 7+ 6+	2+ 2+ 3+ 3+ 3+ 2+ 3+ 2+ 3+ 3+ 2+	B1 F1 10 B3 11 BC 11 B3 BC 10 BC	5 5 8 8 8 7 8 7 8 8	3 3 4 4 4 3 4 3 4 4				Compare M from A Compare M from B Compare M: M + 1 from D Compare M: M + 1 from S Compare M: M + 1 from U Compare M: M + 1 from X Compare M: M + 1 from Y					8	1	1	1	1		
COM	COMA COMB COM					03	6	2	63	6+	2+	73	7	3	43 53	2 2	1 1	$\bar{A} \rightarrow A$ $\bar{B} \rightarrow B$ $\bar{M} \rightarrow M$					•	1	1	0	1	
CWAI		3C	≥20	2													CC ∧ IMM → CC Wait for Interrupt									7		
DAA															19	2	1	Decimal Adjust A					•	1	1	0	1	
DEC	DECA DECB DEC					0A	6	2	6A	6+	2+	7A	7	3	4A 5A	2 2	1 1	A - 1 → A B - 1 → B M - 1 → M					•	1	1	1	•	
EOR	EORA EORB	88 C8	2 2	2 2	98 D8	4 4	2 2	A8 E8	4+ 4+	2+ 2+	B8 F8	5 5	3 3				$A \nabla M \rightarrow A$ $B \nabla M \rightarrow B$					•	1	1	0	•		
EXG	R1, R2	1E	8	2													$R1 \leftrightarrow R2^2$					•	•	•	•	•		
INC	INCA INCB INC					0C	6	2	6C	6+	2+	7C	7	3	4C 5C	2 2	1 1	A + 1 → A B + 1 → B M + 1 → M					•	1	1	1	•	
JMP						0E	3	2	6E	3+	2+	7E	4	3				$EA^3 \rightarrow PC$					•	•	•	•	•	
JSR						9D	7	2	AD	7+	2+	BD	8	3				Jump to Subroutine					•	•	•	•	•	
LD	LDA LDB LDD LDS LDU LDX LDY	86 C6 CC 10 CE CE 8E 10 8E	2 2 3 4 3 3 3 4 4	2 2 3 4 3 3 3 4 4	96 D6 DC 10 DE DE 9E 10 9E	4 4 5 6 5 5 5 6 6	2 2 2 3 2 2 2 3 3	A6 E6 EC 10 EE EE AE 10 AE	4+ 4+ 5+ 6+ 5+ 5+ 5+ 6+ 4+	2+ 2+ 2+ 3+ 2+ 2+ 2+ 3+ 2+	B6 F6 FC 10 FE FE BE 10 BE	5 5 6 7 6 6 6 7 7	3 3 3 4 3 3 3 4 4				M → A M → B M: M + 1 → D M: M + 1 → S M: M + 1 → U M: M + 1 → X M: M + 1 → Y					•	1	1	0	•		
LEA	LEAS LEAU LEAX LEAY								32 33 30 31	4+ 4+ 4+ 4+	2+ 2+ 2+ 2+							$EA^3 \rightarrow S$ $EA^3 \rightarrow U$ $EA^3 \rightarrow X$ $EA^3 \rightarrow Y$					•	•	•	•	•	

LEGEND:

OP	Operation Code (Hexadecimal)	$\bar{M}$	Complement of M	†	Test and set if true, cleared otherwise
~	Number of MPU Cycles	→	Transfer Into	•	Not Affected
#	Number of Program Bytes	H	Half-carry (from bit 3)	CC	Condition Code Register
+	Arithmetic Plus	N	Negative (sign bit)	:	Concatenation
-	Arithmetic Minus	Z	Zero result	V	Logical or
•	Multiply	V	Overflow, 2's complement	∧	Logical and
		C	Carry from ALU	∇	Logical Exclusive or

FIGURE 19 — PROGRAMMING AID (CONTINUED)

Instruction	Forms	Addressing Modes															Description	5	3	2	1	0				
		Immediate			Direct			Indexed <sup>1</sup>			Extended			Inherent				H	N	Z	V	C				
		Op	-	#	Op	-	#	Op	-	#	Op	-	#	Op	-	#										
LSL	LSLA LSLB LSL															48 58	2 2	1 1			•	•	•	•	•	
LSR	LSRA LSRB LSR				08	6	2	68	6+	2+	78	7	3				44 54	2 2	1 1			•	•	•	•	•
MUL																3D	11	1		A x B -> D (Unsigned)	•	•	•	•	•	
NEG	NEGA NEGB NEG				00	6	2	60	6+	2+	70	7	3				40 50	2 2	1 1		A + 1 - A B + 1 - B M + 1 - M	8	8	8	•	•
NOP																12	2	1		No Operation	•	•	•	•	•	
OR	ORA ORB ORCC	8A CA 1A	2 2 3	2	9A DA	4 4	2	2	AA EA	4+ 4+	2+ 2+	BA FA	5 5	3 3						A V M -> A B V M -> B CC V IMM -> CC	•	•	•	•	•	
PSH	PSHS PSHU	34 36	5+ <sup>4</sup> 5+ <sup>4</sup>	2																Push Registers on S Stack Push Registers on L Stack	•	•	•	•	•	
PUL	PULS PULU	35 37	5+ <sup>4</sup> 5+ <sup>4</sup>	2																Pull Registers from S Stack Pull Registers from L Stack	•	•	•	•	•	
ROL	ROLA ROLB ROL				09	6	2	69	6+	2+	79	7	3				49 59	2 2	1 1			•	•	•	•	•
ROR	RORA RORB ROR				06	6	2	66	6+	2+	76	7	3				46 56	2 2	1 1			•	•	•	•	•
RTI																3B	6, 15	1		Return From Interrupt	•	•	•	•	•	
RTS																39	5	1		Return from Subroutine	•	•	•	•	•	
SBC	SBCA SBCB	82 C2	2 2	2	92 D2	4 4	2	2	A2 E2	4+ 4+	2+ 2+	B2 F2	5 5	3 3						A - M - C -> A B - M - C -> B	8	8	•	•	•	
SEX																1D	2	1		Sign Extend B into A	•	•	•	•	•	
ST	STA STB STD STS  STU STX STY				97 D7 DD 10 DF 9F 10 9F	4 4 5 6 5 5 6	2	2	A7 E7 ED 10 EF AF 10 AF	4+ 4+ 5+ 6+ 5+ 5+ 6+ 6+	2+ 2+ 2+ 3+ 2+ 2+ 3+ 3+	B7 F7 FD 10 FF BF 10 BF	5 5 6 7 6 6 7 7	3 3 3 4 3 3 4 4						A - M B - M D - M M + 1 S - M M + 1  U - M M + 1 X - M M + 1 Y - M M + 1	•	•	•	•	•	
SUB	SUBA SUBB SUBD	80 C0 83	2 2 4	2	90 D0 93	4 4 6	2	2	A0 E0 A3	4+ 4+ 6+	2+ 2+ 2+	B0 F0 B3	5 5 7	3 3 3					A - M - A B - M - B D - M M + 1 - D	8	8	•	•	•		
SWI	SWI <sup>6</sup> SWI <sup>26</sup>  SWI <sup>36</sup>															3F 10 3F 11 3F	19 20  20	1 2  1		Software Interrupt 1 Software Interrupt 2  Software Interrupt 3	•	•	•	•	•	
SYNC																13	≥ 4	1		Synchronize to interrupt	•	•	•	•	•	
TFR	R1, R2	1F	6	2																R1 - R2 <sup>2</sup>	•	•	•	•	•	
TST	TSTA TSTB TST				0D	6	2	6D	6+	2+	7D	7	3				4D 5D	2 2	1 1		Test A Test B Test M	•	•	•	•	•

NOTES:

- This column gives a base cycle and byte count. To obtain total count, add the values obtained from the INDEXED ADDRESSING MODE table, Table 2.
- R1 and R2 may be any pair of 8 bit or any pair of 16 bit registers.  
The 8 bit registers are: A, B, CC, DP  
The 16 bit registers are: X, Y, U, S, D, PC
- EA is the effective address.
- The PSH and PUL instructions require 5 cycles plus 1 cycle for each byte pushed or pulled.
- 5(6) means: 5 cycles if branch not taken, 6 cycles if taken (Branch instructions).
- SWI sets I and F bits. SWI2 and SWI3 do not affect I and F.
- Conditions Codes set as a direct result of the instruction.
- Value of half-carry flag is undefined.
- Special Case — Carry set if b7 is SET.



FIGURE 19 – PROGRAMMING AID (CONTINUED)

Branch Instructions

Instruction	Forms	Addressing Mode			Description	5	3	2	1	0
		Relative								
		OP	~	#						
BCC	BCC LBCC	24	3	2	Branch C=0 Long Branch C=0	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		24								
BCS	BCS LBCS	25	3	2	Branch C=1 Long Branch C=1	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		25								
BEQ	BEQ LBEQ	27	3	2	Branch Z=1 Long Branch Z=0	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		27								
BGE	BGE LBGE	2C	3	2	Branch ≥ Zero Long Branch ≥ Zero	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		2C								
BGT	BGT LBGT	2E	3	2	Branch > Zero Long Branch > Zero	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		2E								
BHI	BHI LBHI	22	3	2	Branch Higher Long Branch Higher	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		22								
BHS	BHS LBHS	24	3	2	Branch Higher or Same Long Branch Higher or Same	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		24								
BLE	BLE LBLE	2F	3	2	Branch ≤ Zero Long Branch ≤ Zero	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		2F								
BLO	BLO LBLO	25	3	2	Branch lower Long Branch Lower	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		25								

Instruction	Forms	Addressing Mode			Description	5	3	2	1	0
		Relative								
		OP	~	#						
BLS	BLS LBLS	23	3	2	Branch Lower or Same Long Branch Lower or Same	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		23								
BLT	BLT LBLT	2D	3	2	Branch < Zero Long Branch < Zero	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		2D								
BMI	BMI LBMI	2B	3	2	Branch Minus Long Branch Minus	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		2B								
BNE	BNE LBNE	26	3	2	Branch Z=0 Long Branch Z≠0	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		26								
BPL	BPL LBPL	2A	3	2	Branch Plus Long Branch Plus	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		2A								
BRA	BRA LBRA	20	3	2	Branch Always Long Branch Always	•	•	•	•	•
		16	5	3		•	•	•	•	•
		20								
BRN	BRN LBRN	21	3	2	Branch Never Long Branch Never	•	•	•	•	•
		10	5	4		•	•	•	•	•
		21								
BSR	BSR LBSR	8D	7	2	Branch to Subroutine Long Branch to Subroutine	•	•	•	•	•
		17	9	3		•	•	•	•	•
		8D								
BVC	BVC LBVC	29	3	2	Branch V=0 Long Branch V=0	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		29								
BVS	BVS LBVS	29	3	2	Branch V=1 Long Branch V=1	•	•	•	•	•
		10	5(6)	4		•	•	•	•	•
		29								

SIMPLE BRANCHES

	OP	~	#
BRA	20	3	2
LBRA	16	5	3
BRN	21	3	2
LBRN	1021	5	4
BSR	8D	7	2
LBSR	17	9	3

SIMPLE CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
N=1	BMI	2B	BPL	2A
Z=1	BEQ	27	BNE	26
V=1	BVS	29	BVC	28
C=1	BCS	25	BCC	24

SIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BGT	2E	BLE	2F
r ≥ m	BGE	2C	BLT	2D
r = m	BEQ	27	BNE	26
r ≤ m	BLE	2F	BGT	2E
r < m	BLT	2D	BGE	2C

UNSIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BHI	22	BLS	23
r ≥ m	BHS	24	BLO	25
r = m	BEQ	27	BNE	26
r ≤ m	BLS	23	BHI	22
r < m	BLO	25	BHS	24