



INSTRUCTIONS

68HC11

Documentation interne ELE
D. MENESPLIER 1998

ABA Add B to A

Description: Ajoute l'accum A et l'accum B. Résultat dans A

Octets : 1

Cycles : 2

Exemple: ETIQ ABA

ABX Add B to X

Description: Ajoute l'accum B (non signé) à l'accum X. Résultat dans X

Octets : 1

Cycles : 3

Exemple: ETIQ ABX

ABY Add B to Y

Description: Ajoute l'accum B (non signé) à l'accum Y. Résultat dans Y

Octets : 2

Cycles : 4

Exemple: ETIQ ABY

ADC Add with Carry

Description: Ajoute l'accum A (ou l'accum B) avec le contenu de l'adresse mémoire et avec le bit C du CCR. Résultat dans A (ou B)

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	2	3	2	3
<i>Cycles</i>	2	3	4	4	5

Exemple: ETIQ ADCA MEM
 ADCB #\$41

ADD Add

Description: Ajoute l'accum A (ou l'accum B) avec le contenu de l'adresse mémoire. Résultat dans A (ou B)

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	2	3	2	3
<i>Cycles</i>	2	3	4	4	5

Exemple: ETIQ ADDA MEM
 ADDB #\$1E

ADDD Add D

Description: Ajoute l'accum D (16 bits) avec le contenu de l'adresse mémoire (8 bits de MSB) et celui de l'adresse mémoire + 1 (8 bits de LSB). Résultat dans D.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	3	2	3	2	3
<i>Cycles</i>	4	5	6	6	7

Exemple: ETIQ ADDD MEM

AND Logical AND

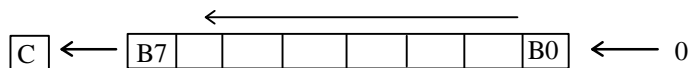
Description: Effectue un ET logique bit à bit entre le contenu de A (ou de B) et le contenu de l'adresse mémoire. Résultat dans A (ou B).

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	2	3	2	3
<i>Cycles</i>	2	3	4	4	5

Exemple: ETIQ ANDA MEM
 ANDB #\$B2

ASL Arithmetic Shift Left

Description: Décalage de 1 bit vers la gauche. Le MSB va dans bit C du CCR et le LSB = 0

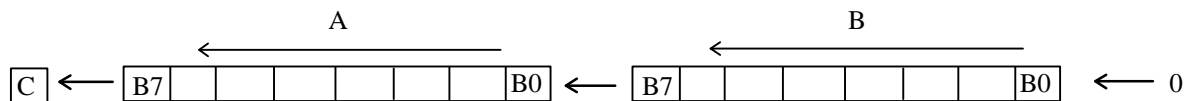


	Accu A	Accu B	Etendu	Ind. X	Ind. Y
<i>Octets</i>	1	1	3	2	3
<i>Cycles</i>	2	2	6	6	7

Exemple: ETIQ ASLA
 ASL MEM

ASLD Arithmetic Shift Left D

Description: Décalage de 1 bit vers la gauche de l'accum D. Le MSB va dans bit C du CCR et le LSB = 0

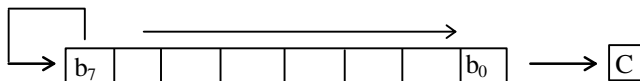


Octets : 1
Cycles : 3

Exemple: ETIQ ASLD

ASR Arithmetic Shift Right

Description: Décalage de 1 bit vers la droite. Le LSB va dans bit C du CCR et le MSB est recopié.



	Accu A	Accu B	Etendu	Ind. X	Ind. Y
<i>Octets</i>	1	1	3	2	3
<i>Cycles</i>	2	2	6	6	7

Exemple: ETIQ ASRA
ASR MEM

BCC Branch if Carry Clear

Description: Teste le bit C du CCR et effectue un branchement à l'adresse spécifiée en mode relatif s'il est à « 0 ».

Instruction identique à BHS.

Octets : 2

Cycles : 3

Exemple: ETIQ1 BCC ETIQ2

BCLR Bit Clear.

Description: Mise à « 0 » des bits de l'octet opérande en spécifiant par des « 1 » les bits voulus dans un masque. Les autres bits ne sont pas affectés.

L'octet opérande doit être spécifié en adressage direct ou indéxé. Le masque est une donnée immédiate qui doit être précédée du signe : #.

	Direct	Ind. X	Ind. Y
<i>Octets</i>	3	3	4
<i>Cycles</i>	6	7	8

Exemple: PORTB EQU \$04
LDX #\$1000
ETIQ BCLR PORTB,X,\$05

Les bits b0 et b2 du PORTB qui est à l'adresse \$1004 sont mis à « 0 ».

BCS Branch if Carry Set

Description: Teste le bit C du CCR et effectue un branchement à l'adresse spécifiée en mode relatif s'il est à « 1 ».

Instruction identique à BLO.

Octets : 2

Cycles : 3

BEQ Branch if Equal

Description: Teste le bit Z du CCR et effectue un branchement à l'adresse spécifiée en mode relatif s'il est à « 1 ».

Après un CMP cette instruction cause un branchement si les opérandes étaient égaux.

Octets : 2

Cycles : 3

Exemple: ETIQ1 CMPA #\$08
 BEQ ETIQ2

BGE Branch if Greater or Egal.

Description: Teste les bits N et V du CCR et effectue un branchement à l'adresse spécifiée en mode relatif si : (**N ou exclusif V**) = 0.

Après un CMP cette instruction cause un branchement si le registre est supérieur ou égal à l'opérande. Les nombres sont considérés comme signés.

Octets : 2

Cycles : 3

BGT Branch if Greater Than zero.

Description: Teste les bits Z, N et V du CCR et effectue un branchement à l'adresse spécifiée en mode relatif si : { **Z ou (N ou exclusif V)** } = 0.

Après un CMP cette instruction cause un branchement si le registre est strictement supérieur à l'opérande. Les nombres sont considérés comme signés.

Octets : 2

Cycles : 3

BHI Branch if Higher.

Description: Teste les bits C et Z du CCR et effectue un branchement à l'adresse spécifiée en mode relatif si : (**C ou Z**) = 0.

Après un CMP cette instruction cause un branchement si le registre est plus grand que l'opérande. Les nombres sont considérés comme non signés.

Octets : 2

Cycles : 3

BHS Branch if Higher or Same.

Description: Teste les bit C du CCR et effectue un branchement à l'adresse spécifiée en mode relatif s'il est à « 0 ».

Après un CMP cette instruction cause un branchement si le registre est supérieur ou égal à l'opérande. Les nombres sont considérés comme non signés.

Octets : 2

Cycles : 3

BIT Bit Test memory and accumulator

Description: Effectue un ET logique bit à bit entre le contenu de A (ou de B) et le contenu de l'adresse mémoire, et positionne les bits du CCR mais ne modifie pas l'accumulateur et la mémoire. Ne pas confondre avec AND qui elle modifie les opérandes.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	2	3	2	3
<i>Cycles</i>	2	3	4	4	5

Exemple: ETIQ BITA MEM
 BITB #\$B2

BLE Branch if Less than or Equal to zero.

Description: Teste les bits Z, N et V du CCR et effectue un branchement à l'adresse spécifiée en mode relatif si : **{ Z ou (N ou exclusif V) = 1 }**.

Après un CMP cette instruction cause un branchement si le registre est inférieur ou égal à l'opérande. Les nombres sont considérés comme signés.

Octets : 2

Cycles : 3

BLO Branch if Lower.

Description: Teste les bit C du CCR et effectue un branchement à l'adresse spécifiée en mode relatif s'il est à « 1 ».

Après un CMP cette instruction cause un branchement si le registre est plus petit que l'opérande. Les nombres sont considérés comme non signés.

Octets : 2

Cycles : 3

BLS Branch if Lower or Same.

Description: Teste les bits C et Z du CCR et effectue un branchement à l'adresse spécifiée en mode relatif si : **(C ou Z) = 1**.

Après un CMP cette instruction cause un branchement si le registre est plus petit ou égal à l'opérande. Les nombres sont considérés comme non signés.

Octets : 2

Cycles : 3

BLT Branch if Less Than zero.

Description: Teste les bits N et V du CCR et effectue un branchement à l'adresse spécifiée en mode relatif si : **(N ou exclusif V) = 1**.

Après un CMP cette instruction cause un branchement si le registre est strictement inférieur à l'opérande. Les nombres sont considérés comme signés.

Octets : 2

Cycles : 3

BMI Branch if Minus.

Description: Teste le bit N du CCR et effectue un branchement à l'adresse spécifiée en mode relatif s'il est à « 1 ».

Octets : 2

Cycles : 3

BNE Branch if Not Egal.

Description: Teste le bit Z du CCR et effectue un branchement à l'adresse spécifiée en mode relatif s'il est à « 0 ».

Après un CMP cette instruction cause un branchement si les opérandes étaient différents.

Octets : 2

Cycles : 3

Exemple: ETIQ1 CMPA #\$08
 BNE ETIQ2

BPL Branch if Plus.

Description: Teste le bit N du CCR et effectue un branchement à l'adresse spécifiée en mode relatif s'il est à « 0 ».

Octets : 2

Cycles : 3

BRA Branch Always.

Description: Effectue un branchement obligatoire à l'adresse spécifiée en mode relatif.

Octets : 2

Cycles : 3

BRCLR BRanch if bit CLear.

Description: Effectue un branchement relatif à l'adresse donnée, si les bits de l'octet opérande spécifiées par le masque sont à « 0 ».

L'octet opérande doit être spécifié en adressage direct ou indéré. Le masque est une donnée immédiate qui doit être précédée du signe : #.

	Direct	Ind. X	Ind. Y
<i>Octets</i>	4	4	5
<i>Cycles</i>	6	7	8

Exemple: PORTB EQU \$04
 LDX #\$1000
 ETIQ BRCLR PORTB,X,#\$05 ETIQ2

Branchement à ETIQ2 si les bits b0 et b2 du PORTB qui est à l'adresse \$1004 sont à « 0 ».

BRN Branch Never.

Description: Cette instruction ne fait rien. C'est un NOP codé sur 2 octets qui peut être utile dans une boucle de temporisation pour consommer du temps.

Octets : 2

Cycles : 3

BRSET BRanch if bit SET.

Description: Effectue un branchement relatif à l'adresse donnée, si les bits de l'octet opérande spécifiées par le masque sont à «1».

L'octet opérande doit être spécifié en adressage direct ou indirect. Le masque est une donnée immédiate qui doit être précédée du signe : #.

	Direct	Ind. X	Ind. Y
<i>Octets</i>	4	4	5
<i>Cycles</i>	6	7	8

Exemple: PORTB EQU \$04
 LDX #\$1000
 ETIQ BRSET PORTB,X,#\$05 ETIQ2

Branchement à ETIQ2 si les bits b0 et b2 du PORTB qui est à l'adresse \$1004 sont à « 1 ».

BSET Bit SET.

Description: Mise à « 1 » des bits de l'octet opérande en spécifiant par des « 1 » les bits voulus dans un masque. Les autres bits ne sont pas affectés.

L'octet opérande doit être spécifié en adressage direct ou indirect. Le masque est une donnée immédiate qui doit être précédée du signe : #.

	Direct	Ind. X	Ind. Y
<i>Octets</i>	3	3	4
<i>Cycles</i>	6	7	8

Exemple: PORTB EQU \$04
 LDX #\$1000
 ETIQ BSET PORTB,X,#\$05

Les bits b0 et b2 du PORTB qui est à l'adresse \$1004 sont mis à « 1 ».

BSR Branch to SubRoutine.

Description: Branchement au sous-programme dont l'adresse est spécifiée en mode relatif. La valeur du PC est sauvegardée dans la pile et sera restituée à la fin du sous-programme suite à l'instruction RTS.

Octets : 2

Cycles : 6

BVC Branch if oVerflow Clear.

Description: Teste l'état du bit V du CCR et cause un branchement à l'adresse indiquée en mode relatif s'il est à « 0 ».

Utilisée après une opération arithmétique sur des nombres signés, cette instruction cause un branchement si le résultat de l'opération est valide (pas de débordement).

Octets : 2

Cycles : 3

BVS Branch if oVerflow Set

Description: Teste l'état du bit V du CCR et cause un branchement à l'adresse indiquée en mode relatif s'il est à « 1 ».

Utilisée après une opération arithmétique sur des nombres signés, cette instruction cause un branchement si le résultat de l'opération est invalide (débordement).

Octets : 2

Cycles : 3

CBA Compare B to A

Description: Compare les accumulateurs A et B. Soustrait l'accum B de l'accum A et positionne les bits du CCR en fonction du résultat. Les accum A et B ne sont pas modifiés.

N est mis à « 1 » si le bit de poids fort du résultat de la soustraction est « 1 ».

Z est mis à « 1 » si le résultat de la soustraction est nul.

V est mis à « 1 » s'il y a débordement de la soustraction avec des nombres signés.

C est mis à « 1 » si la valeur absolue de A est plus grande que celle de B.

Octets : 1

Cycles : 2

Exemple: ETIQ CBA
 BEQ EGAL

CLC CLear Carry

Description: Mise à « 0 » du bit C du CCR.

Octets : 1

Cycles : 2

CLI CLear Interrupt mask

Description: Mise à « 0 » du bit I du CCR ce qui permet la prise en compte des interruptions.

Octets : 1

Cycles : 2

CLR CleaR

Description: Mise à « 0 » de l'accum A, de l'accum B ou de la mémoire en adressage indexé ou étendu.

	Accu A	Accu B	Étendu	Ind. X	Ind. Y
<i>Octets</i>	1	1	3	2	3
<i>Cycles</i>	2	2	6	6	7

Exemple: ETIQ CLRA
 CLRB
 CLR MEM

CLV CLear oVerflow

Description: Mise à « 0 » du bit V du CCR.

Octets : 1

Cycles : 2

CMP CoMpare accumulator with memory.

Description: Soustrait le contenu de la mémoire au contenu de l'accumulateur A ou B et positionne les bits du CCR. Ni la mémoire ni les accum ne sont modifiés.

	Imm.	Direct	Étendu	Ind. X	Ind. Y
<i>Octets</i>	2	2	3	2	3
<i>Cycles</i>	2	3	4	4	5

Exemple: ETIQ CMPA MEM
 CMPB #\$F5

COM COMplément logique.

Description: Réalise la complémentation logique bit à bit de l'accumulateur A ou B ou de la mémoire en adressage étendu ou indexé.

	Accu A	Accu B	Étendu	Ind. X	Ind. Y
<i>Octets</i>	1	1	3	2	3
<i>Cycles</i>	2	2	6	6	7

Exemple: ETIQ COM MEM
 COMA
 COMB

CPD ComPare accumulator D with memory.

Description: Soustrait le contenu des adresses mémoires M et M+1 au contenu de l'accumulateur D et positionne les bits du CCR. Ni la mémoire ni l'accu D ne sont modifiés.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	4	3	4	3	3
<i>Cycles</i>	5	6	7	7	7

Exemple: ETIQ CPD MEM
 BEQ EGAL
 CPD #A1F5
 BNE DIFF

CPX ComPare index X with memory.

Description: Soustrait le contenu des adresses mémoires M et M+1 au contenu du registre d'index X et positionne les bits du CCR. Ni la mémoire ni l'index X ne sont modifiés.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	3	2	3	2	3
<i>Cycles</i>	4	5	6	6	7

Exemple: ETIQ CPX MEM
 CPX #F1A5

CPY ComPare index Y with memory.

Description: Soustrait le contenu des adresses mémoires M et M+1 au contenu du registre d'index Y et positionne les bits du CCR. Ni la mémoire ni l'index Y ne sont modifiés.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	4	3	4	3	3
<i>Cycles</i>	5	6	7	7	7

Exemple: ETIQ CPY MEM
 CPY #F1A5

DAA Decimal Adjust A

Description: Ajoute au contenu de l'accumulateur A la valeur \$00 ou \$06 ou \$60 ou \$66 afin que l'addition qui vient d'être effectuée donne un résultat exact si l'on considère que l'octet dans A est constitué de deux chiffres codés en DCB sur 4 bits.

Octets : 1
Cycles : 2

DEC DECrement.

Description: Diminue de 1 le contenu de l'accumulateur A ou B ou de la mémoire en adressage étendu ou indexé.

	Accu A	Accu B	Étendu	Ind. X	Ind. Y
<i>Octets</i>	1	1	3	2	3
<i>Cycles</i>	2	2	6	6	7

Exemple: ETIQ DEC MEM
 DECA
 DECB

DES DEcrement Stack pointer.

Description: Diminue de 1 le contenu du pointeur de pile SP.

Octets : 1
Cycles : 3

DEX DEcrement index X.

Description: Diminue de 1 le contenu du registre d'index X.

Octets : 1
Cycles : 3

DEY DEcrement index Y.

Description: Diminue de 1 le contenu du registre d'index Y.

Octets : 2
Cycles : 4

EOR Exclusive OR

Description: Effectue un OU Exclusif logique bit à bit entre le contenu de A (ou de B) et le contenu de l'adresse mémoire, et place le résultat dans A (ou B).

	Imm.	Direct	Étendu	Ind. X	Ind. Y
<i>Octets</i>	2	2	3	2	3
<i>Cycles</i>	2	3	4	4	5

Exemple: ETIQ EORA MEM
 EORB #\$B2

FDIV Fractional DIVide

Description: Effectue une division fractionnaire non signée sur 16 bits. Le numérateur est contenu dans l'accumulateur D. Le dénominateur est contenu dans le registre d'index X. Le point décimal est supposé à la même position au numérateur et au dénominateur. Les bits du CCR sont positionnés en fonction du résultat. Le quotient est placé dans X et le reste dans D. Le point décimal est placé à gauche du bit 15 du quotient. Le numérateur est supposé inférieur au dénominateur. Dans le cas contraire ou en cas de division par 0, le quotient est forcé à \$FFFF et le reste est indéterminé.

D Quotient → X
X Reste → D

Octets : 1
 Cycles : 41

IDIV Integer DIVide

Description: Effectue une division entière non signée sur 16 bits. Le numérateur est contenu dans l'accumulateur D. Le dénominateur est contenu dans le registre d'index X. Le point décimal est supposé à la même position au numérateur et au dénominateur. Les bits du CCR sont positionnés en fonction du résultat. Le quotient est placé dans X et le reste dans D. Le point décimal est placé à droite du bit 0 du quotient. Dans le cas contraire ou en cas de division par 0, le quotient est forcé à \$FFFF et le reste est indéterminé.

Octets : 1
 Cycles : 41

INC INCrement.

Description: Augmente de 1 le contenu de l'accumulateur A ou B ou de la mémoire en adressage étendu ou indexé.

	Accu A	Accu B	Etendu	Ind. X	Ind. Y
Octets	1	1	3	2	3
Cycles	2	2	6	6	7

Exemple: ETIQ INC MEM
 INCA
 INCB

INS INcrement Stack pointer.

Description: Augmente de 1 le contenu du pointeur de pile SP.

Octets : 1
 Cycles : 3

INX INcrement index X.

Description: Augmente de 1 le contenu du registre d'index X.

Octets : 1

Cycles : 3

INY INcrement index Y.

Description: Augmente de 1 le contenu du registre d'index Y.

Octets : 2

Cycles : 4

JMP JuMP.

Description: Cause un saut obligatoire à l'adresse spécifiée en mode étendu ou indexé..

	Etendu	Ind. X	Ind. Y
<i>Octets</i>	3	2	3
<i>Cycles</i>	3	3	4

JSR Jump to SubRoutine

Description: Cause un saut à un sous programme dont l'adresse est spécifiée en mode direct, étendu ou indexé. La valeur du PC est sauvegardé dans la pile afin de permettre le retour en ce point suite à l'instruction RTS en fin de sous programme.

	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	3	2	3
<i>Cycles</i>	5	6	6	7

Exemple: ETIQ JSR SPROG

LDA LoaD Accumulator

Description: Charge l'accumulateur A (ou B) avec le contenu de l'adresse mémoire.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	2	3	2	3
<i>Cycles</i>	2	3	4	4	5

Exemple: ETIQ LDAA MEM
LDAB 0,X
LDAA #\$E6

LDD Load accumulator D

Description: Charge l'accumulateur D avec le contenu des adresses mémoire M et M+1. Le MSB qui était en M est placé dans l'accumulateur A et le LSB qui était dans M+1 est placé dans l'accumulateur B.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	3	2	3	2	3
<i>Cycles</i>	3	4	5	5	6

Exemple: ETIQ LDD MEM
LDD #\$A8E6

LDS Load Stack pointer

Description: Charge le pointeur de pile SP avec le contenu des adresses mémoire M et M+1.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	3	2	3	2	3
<i>Cycles</i>	3	4	5	5	6

Exemple: ETIQ LDS #\$00FF

LDX Load index X

Description: Charge le registre d'index X avec le contenu des adresses mémoire M et M+1.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	3	2	3	2	3
<i>Cycles</i>	3	4	5	5	6

Exemple: ETIQ LDX MEM
LDX #\$10FF

LDY Load index Y

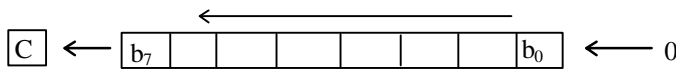
Description: Charge le registre d'index Y avec le contenu des adresses mémoire M et M+1.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	4	3	4	3	3
<i>Cycles</i>	4	5	6	6	6

Exemple: ETIQ LDY MEM
LDY #\$FA46

LSL Logical Shift Left.

Description: Effectue un décalage de 1 bit vers la gauche du contenu de l'opérande. Un « 0 » entre dans le bit de poids faible et le bit de poids fort va dans le bit C du CCR.



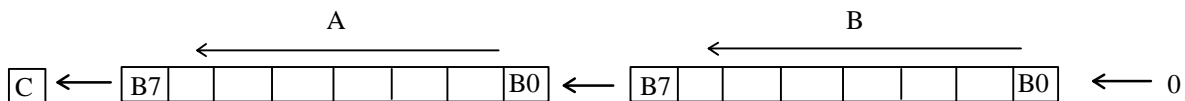
	Accu A	Accu B	Etendu	Ind. X	Ind. Y
Octets	1	1	3	2	3
Cycles	2	2	6	6	7

Exemple: ETIQ LSL MEM
LSLA
LSLB

Cette instruction est identique à ASL.

LSLD Logical Shift Left D

Description: Décalage de 1 bit vers la gauche de l'accu D. Le MSB va dans bit C du CCR et le LSB = 0

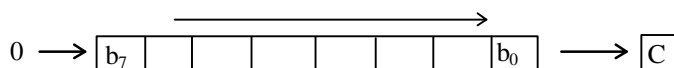


Octets : 1
Cycles : 3

Cette instruction est identique à ASLD.

LSR Logical Shift Right.

Description: Effectue un décalage de 1 bit vers la droite du contenu de l'opérande. Un « 0 » entre dans le bit de poids fort et le bit de poids faible va dans le bit C du CCR.

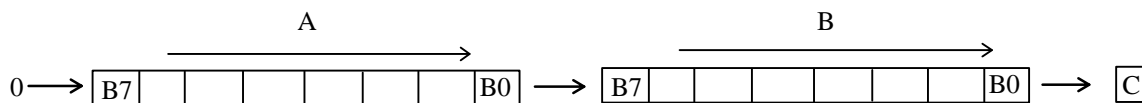


	Accu A	Accu B	Etendu	Ind. X	Ind. Y
Octets	1	1	3	2	3
Cycles	2	2	6	6	7

Exemple: ETIQ LSR MEM
LSRA
LSRB

LSRD Logical Shift Right D

Description: Décalage de 1 bit vers la droite de l'accum D. Place un « 0 » dans le bit de poids fort et fait sortir le bit de poids faible dans le bit C du CCR.



Octets : 1

Cycles : 3

MUL MULtiplied unsigned

Description: Multiplie les deux octets considérés comme non signés contenus dans l'accumulateur A et dans l'accumulateur B. Le résultat est placé dans l'accumulateur D.

Octets : 1

$$A \times B \longrightarrow D$$

Cycles : 10

NEG NEGate.

Description: Réalise le complément à deux de l'opérande, donc change le signe.

	Accu A	Accu B	Etendu	Ind. X	Ind. Y
Octets	1	1	3	2	3
Cycles	2	2	6	6	7

Exemple: ETIQ NEG MEM
 NEGA
 NEGB

NOP No OPeration.

Description: Cette instruction ne fait rien. Elle est utile dans les boucles de temporisation pour consommer du temps.

Octets : 1

Cycles : 2

ORA Inclusif OR

Description: Effectue un OU inclusif logique bit à bit entre le contenu de A (ou de B) et le contenu de l'adresse mémoire, et place le résultat dans A (ou B).

	Imm.	Direct	Etendu	Ind. X	Ind. Y
Octets	2	2	3	2	3
Cycles	2	3	4	4	5

Exemple: ETIQ ORAA MEM
 ORAB #B2

PSHA PuSH A onto stack.

Description: Sauvegarde le contenu de l'accumulateur A dans la pile et diminue le pointeur de pile d'une unité.

Octets : 1

Cycles : 3

PSHB PuSH B onto stack.

Description: Sauvegarde le contenu de l'accumulateur B dans la pile et diminue le pointeur de pile d'une unité.

Octets : 1

Cycles : 3

PSHX PuSH index X onto stack.

Description: Sauvegarde le contenu du registre d'index X dans la pile et diminue le pointeur de pile de deux unités.

Octets : 1

Cycles : 4

PSHY PuSH index Y onto stack.

Description: Sauvegarde le contenu du registre d'index Y dans la pile et diminue le pointeur de pile de deux unités.

Octets : 2

Cycles : 5

PULA PUL A from stack.

Description: Augmente le pointeur de pile d'une unité puis charge l'accumulateur A avec la data lue dans la pile.

Octets : 1

Cycles : 4

PULB PUL B from stack.

Description: Augmente le pointeur de pile d'une unité puis charge l'accumulateur B avec la data lue dans la pile.

Octets : 1

Cycles : 4

PULX PUL index X from stack.

Description: Augmente le pointeur de pile de deux unités puis charge le registre d'index X avec les deux octets lus dans la pile.

Octets : 1
Cycles : 5

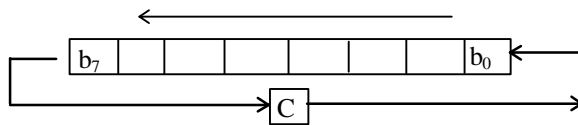
PULY PUL index Y from stack.

Description: Augmente le pointeur de pile de deux unités puis charge le registre d'index Y avec les deux octets lus dans la pile.

Octets : 2
Cycles : 6

ROL ROtate Left thru carry.

Description: Effectue un décalage d'un bit vers la gauche du contenu de l'opérande. Fait sortir le bit de poids fort dans le bit C du CCR puis le place à la place du bit de poids faible. On obtient ainsi une rotation de l'opérande sur lui même d'une position vers la gauche.

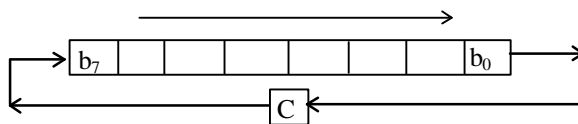


	Accu A	Accu B	Etendu	Ind. X	Ind. Y
Octets	1	1	3	2	3
Cycles	2	2	6	6	7

Exemple: ETIQ ROL MEM
ROLA
ROLB

ROR ROtate Right thru carry.

Description: Effectue un décalage d'un bit vers la droite du contenu de l'opérande. Fait sortir le bit de poids faible dans le bit C du CCR puis le place à la place du bit de poids fort. On obtient ainsi une rotation de l'opérande sur lui même d'une position vers la droite.



	Accu A	Accu B	Etendu	Ind. X	Ind. Y
Octets	1	1	3	2	3
Cycles	2	2	6	6	7

Exemple: ETIQ ROR MEM
RORA
RORB

RTI ReTurn from Interrupt.

Description: Effectue un retour d'interruption avec restauration du contexte. Le CCR, les accumulateurs A et B, les registres d'index X et Y et le compteur ordinal sont récupérés dans la pile où ils avaient été sauvegardés. Le pointeur de pile est incrémenté de 9.

Octets : 1

Cycles : 12

RTS ReTurn from Subroutine.

Description: Effectue un retour de sous programme en forçant le compteur ordinal à la valeur sauvegardée dans la pile lors de l'appel à ce sous programme. Le pointeur de pile est incrémenté de 2.

Octets : 1

Cycles : 5

SBA SuBstract Accumulators.

Description: Soustrait le contenu de l'accumulateur B à celui de l'accumulateur A et place le résultat dans A.

$$A - B \longrightarrow A$$

Octets : 1

Cycles : 2

SBC SuBstract with Carry.

Description: Soustrait le contenu de la mémoire et le bit C du CCR au contenu de l'accumulateur A (ou B) et place le résultat dans A (ou B).

$$A - M - C \longrightarrow A \quad \text{ou} \quad B - M - C \longrightarrow B$$

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	2	3	2	3
<i>Cycles</i>	2	3	4	4	5

Exemple: ETIQ SBCA MEM
 SBCB #B2

SEC SEt Carry.

Description: Mise à « 1 » du bit C du CCR.

Octets : 1

Cycles : 2

SEI SEt Interrupt mask.

Description: Mise à « 1 » du bit I du CCR. Cela interdit la prise en compte des interruptions masquables.

Octets : 1

Cycles : 2

SEV SEt oVerflow.

Description: Mise à « 1 » du bit V du CCR.

Octets : 1

Cycles : 2

STA STore Accumulator.

Description: Place le contenu de l'accumulateur A (ou B) dans la mémoire. Le contenu de l'accumulateur reste inchangé.

	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	3	2	3
<i>Cycles</i>	3	4	4	5

Exemple: ETIQ STAA MEM
 STAB 0,X

STD STore Double accumulator.

Description: Place le contenu de l'accumulateur D dans la mémoire M et M+1 avec les poids forts dans M et les poids faibles dans M+1. Le contenu de l'accumulateur D reste inchangé.

	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	3	2	3
<i>Cycles</i>	4	5	5	6

Exemple: ETIQ STD MEM

STOP STOP Mode.

Description: Si le bit C du CCR est à « 0 » la consommation est réduite en stoppant le fonctionnement des sous ensembles dynamiques. Le contenu des registres est sauvegardé ainsi que l'état des broches d'entrées/sorties. L'oscillateur est remis en service si l'entrée RESET ou une entrée d'interruption passe à « 0 ». Si le bit S du CCR est à « 1 » cette instruction est un **NOP**.

Octets : 1

Cycles : 2

STS STore Stack pointer.

Description: Place le contenu du pointeur de pile dans la mémoire M et M+1 avec les poids forts dans M et les poids faibles dans M+1. Le contenu de SP reste inchangé.

	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	3	2	3
<i>Cycles</i>	4	5	5	6

Exemple: ETIQ STS MEM

STX STore index X.

Description: Place le contenu du registre d'index X dans la mémoire M et M+1 avec les poids forts dans M et les poids faibles dans M+1. Le contenu de X reste inchangé.

	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	3	2	3
<i>Cycles</i>	4	5	5	6

Exemple: ETIQ STX MEM

STY STore index Y.

Description: Place le contenu du registre d'index Y dans la mémoire M et M+1 avec les poids forts dans M et les poids faibles dans M+1. Le contenu de Y reste inchangé.

	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	3	4	3	3
<i>Cycles</i>	5	6	6	6

Exemple: ETIQ STY MEM

SUB SUBstract.

Description: Soustrait le contenu de la mémoire au contenu de l'accumulateur A (ou B) et place le résultat dans A (ou B).

$$A - M \longrightarrow A \quad \text{ou} \quad B - M \longrightarrow B$$

	Imm.	Direct	Etendu	Ind. X	Ind. Y
<i>Octets</i>	2	2	3	2	3
<i>Cycles</i>	2	3	4	4	5

Exemple: ETIQ SUBA MEM
SUBB #\$B2

SUBD SUBstract Double accumulator.

Description: Soustrait le contenu des mémoires M et M+1 au contenu de l'accumulateur D et place le résultat dans D.

	Imm.	Direct	Etendu	Ind. X	Ind. Y
Octets	3	2	3	2	3
Cycles	4	5	6	6	7

SWI SoftWare Interrupt.

Description: Interruption logicielle. Le contexte est sauvegardé dans la pile. Le compteur ordinal est chargé avec le vecteur de SWI ce qui provoque l'exécution du sous programme d'interruption correspondant.

Octets : 1

Cycles : 14

TAB Transfer A to B

Description: Transfère le contenu de l'accumulateur A dans l'accumulateur B. L'accu A reste inchangé.

A → **B**

Octets : 1

Cycles : 2

TAP Transfer A to CCR

Description: Transfère le contenu de l'accumulateur A dans le CCR. L'accumulateur A reste inchangé. Le bit 0 de A correspond au bit C du CCR et le bit 7 correspond au bit S. Le bit X peut être mis à « 0 » par un TAP mais ne peut pas être mis à « 1 ».

A → **CCR**

Octets : 1

Cycles : 2

TBA Transfer B to A

Description: Transfère le contenu de l'accumulateur B dans l'accumulateur A. L'accu B reste inchangé.

B → **A**

Octets : 1

Cycles : 2

TPA Transfer CCR to A

Description: Transfère le contenu du CCR dans l'accumulateur A. Le contenu du CCR reste inchangé. Le bit C du CCR correspond au bit 0 de A et le bit S au bit 7.

CCR → **A**

Octets : 1

Cycles : 2

TYS Transfer Y to S.

Description: Charge le pointeur de pile SP avec le contenu du registre d'index Y diminué d'une unité afin que SP pointe ainsi une adresse mémoire libre dans la pile. Le contenu de Y reste inchangé.

Octets : 2
Cycles : 4

Y - 1 \longrightarrow **SP**

WAI WAIt for interrupt.

Description: L'unité centrale sauvegarde le contexte et s'arrête. Elle attend alors une interruption. Quand une interruption valide arrive, le vecteur correspondant est chargé et l'unité centrale redémarre pour exécuter le sous programme d'interruption.

Octets : 2
Cycles : 0

XGDX eXchanGe D with X.

Description: Echange le contenu de l'accumulateur D avec celui du registre d'index X.

X \longrightarrow **D** et **D** \longrightarrow **X**

Octets : 1
Cycles : 3

XGDY eXchanGe D with Y.

Description: Echange le contenu de l'accumulateur D avec celui du registre d'index Y.

Y \longrightarrow **D** et **D** \longrightarrow **Y**

Octets : 2
Cycles : 4