

***LE MICRO CONTROLEUR***

***MOTOROLA***

***68HC11***

***P. LECARDONNEL  
P. LETENNEUR  
GRANVILLE - 2002***

## SOMMAIRE

<b>1) CARACTÉRISTIQUES DU MICRO CONTROLEUR .....</b>	<b>4</b>
<b>2) ORGANISATION INTERNE.....</b>	<b>4</b>
<b>3) CONFIGURATION DU <math>\mu</math>C. ....</b>	<b>5</b>
3.1) QUATRE CONFIGURATIONS POSSIBLES. ....	5
3.2) CONFIGURATION MEMOIRE EN FONCTION DES MODES. ....	5
<b>4) LES MODES DE FONCTIONNEMENT DU <math>\mu</math>C. ....</b>	<b>6</b>
4.1) LE MODE RUN. ....	6
4.2) LE MODE STOP. ....	6
4.3) LE MODE WAIT. ....	6
<b>5) LES BROCHES DU <math>\mu</math>C.....</b>	<b>7</b>
5.1) RESET. ....	7
5.2) OSCILLATEUR : OSC1 ET OSC2 OU EXTAL ET XTAL. ....	8
5.3) ALIMENTATION : VDD ET VSS.....	8
5.4) INTERRUPTIONS : IRQ ET XIRQ.....	9
5.5) REFERENCE DE TENSION : VREFL ET VREFH.....	9
<b>6) L'UNITÉ CENTRALE. ....</b>	<b>10</b>
6.1) ORGANISATION.....	10
<b>7) JEU D'INSTRUCTIONS.....</b>	<b>11</b>
<b>8) LES MODES D'ADRESSAGE.....</b>	<b>17</b>
8.1) ADRESSAGE INHERENT OU IMPLICITE .....	17
8.2) ADRESSAGE IMMEDIAT.....	17
8.3) ADRESSAGES DIRECT ET ETENDU.....	17
8.4) ADRESSAGE RELATIF.....	17
8.5) ADRESSAGE INDEXE.....	18
8.6) MANIPULATION DE BITS.....	18
8.6.1) Forçage de bits.....	18
8.6.2) Test de bits.....	19
<b>9) LES PORTS D'ENTREES SORTIES.....</b>	<b>20</b>
<b>10) LE CONVERTISSEUR ANALOGIQUE NUMERIQUE.....</b>	<b>21</b>
10.1) ORGANISATION INTERNE. ....	21
10.2) FONCTIONNEMENT DU CONVERTISSEUR. ....	22
10.3) CYCLE DE FONCTIONNEMENT.....	23
10.4) REGISTRES.....	24
10.4.1) Le registre <i>OPTION</i> . ....	24
10.4.2) Le registre <i>ADCTL</i> . ....	24
10.4.3) Les registres <i>ADR1</i> , <i>ADR2</i> , <i>ADR3</i> et <i>ADR4</i> .....	24

# LE MICRO CONTROLEUR MC68HC811E2

<b>11) LA LIAISON SERIE SPI (SERIAL PERIPHERAL INTERFACE).....</b>	<b>25</b>
11.1) CARACTERISTIQUES.....	25
11.2) LE BUS.....	25
11.3) PRINCIPE DE FONCTIONNEMENT.....	26
11.4) ORGANISATION UTILISANT UN MAITRE ET PLUSIEURS ESCLAVES.....	26
11.5) ORGANISATION INTERNE.....	27
11.5.1) Le registre de contrôle SPCR (Serial Peripheral Control Register).....	28
11.5.2) Le registre d'état SPSR (Serial Peripheral Status Register).....	29
11.5.3) Le registre de données SPDR (Serial Peripheral Data Register).....	29
<b>12) LA LIAISON SERIE SCI (SERIAL COMMUNICATION INTERFACE).....</b>	<b>30</b>
12.1) LE REGISTRE SCDR.....	31
12.2) LE REGISTRE BAUD.....	31
12.3) LE REGISTRE DE CONTROLE SCCR1 (SERIAL COMMUNICATION CONTROL REGISTER 1).....	32
12.4) LE REGISTRE DE CONTROLE SCCR2 (SERIAL COMMUNICATION CONTROL REGISTER 2).....	32
12.5) LE REGISTRE SCSR (SERIAL COMMUNICATION STATUS REGISTER).....	34
<b>13) LE TIMER (TEMPORISATEUR).....</b>	<b>35</b>
13.1) LE REGISTRE FREE RUNNING COUNTER TCNT.....	36
13.2) LES REGISTRES TIMER OUTPUT COMPARE TOCX (X = 1, 2, 3, 4 OU 5).....	36
13.3) LES REGISTRES TIMER INPUT COMPARE TICX (X = 1, 2, 3 OU 4).....	36
13.4) LE REGISTRE TIMER CONTROL 2 : TCTL2.....	37
13.5) LE REGISTRE CONTROL FORCE OUTPUT COMPARE : CFORC.....	37
13.6) LE REGISTRE OUTPUT COMPARE 1 MASK : OC1M.....	37
13.7) LE REGISTRE OUTPUT COMPARE 1 DATA : OC1D.....	37
13.8) LE REGISTRE TIMER CONTROL 1 : TCTL1.....	38
13.9) LE REGISTRE TIMER MASK 1 : TMSK1.....	38
13.10) LE REGISTRE TIMER FLAG 1 : TFLG1.....	38
13.11) LE REGISTRE TIMER MASK 2 : TMSK2.....	39
13.12) LE REGISTRE TIMER FLAG 2 : TFLG2.....	39
<b>14) LES INTERRUPTIONS.....</b>	<b>40</b>
14.1) LES SOURCES D'INTERRUPTIONS.....	40
14.2) DEROULEMENT D'UNE INTERRUPTION.....	41
14.3) L'INTERRUPTION DEFAUT D'HORLOGE.....	41
14.4) L'INTERRUPTION COP : COMPUTER OPERATING PROPERLY.....	41
14.5) L'INTERRUPTION ILLEGAL OPCODE.....	41
14.6) L'INTERRUPTION EXTERNE XIRQ.....	41
14.7) L'INTERRUPTION EXTERNE IRQ.....	41
14.8) L'INTERRUPTION RTI.....	42
14.9) LES INTERRUPTIONS TIMER.....	42
14.10) L'INTERRUPTION SCI.....	42
14.11) L'INTERRUPTION SPI.....	42
14.12) VECTEURS DES DIFFERENTES INTERRUPTIONS.....	43
<b>15) LISTE DE TOUS LES REGISTRES.....</b>	<b>44</b>
<b>16) CARACTERISTIQUES MECANIQUES.....</b>	<b>45</b>
<b>17) REFERENCES.....</b>	<b>46</b>
<b>18) FAMILLE HC11.....</b>	<b>47</b>
<b>19) BIBLIOGRAPHIE.....</b>	<b>47</b>

# LE MICRO CONTROLEUR MC68HC811E2

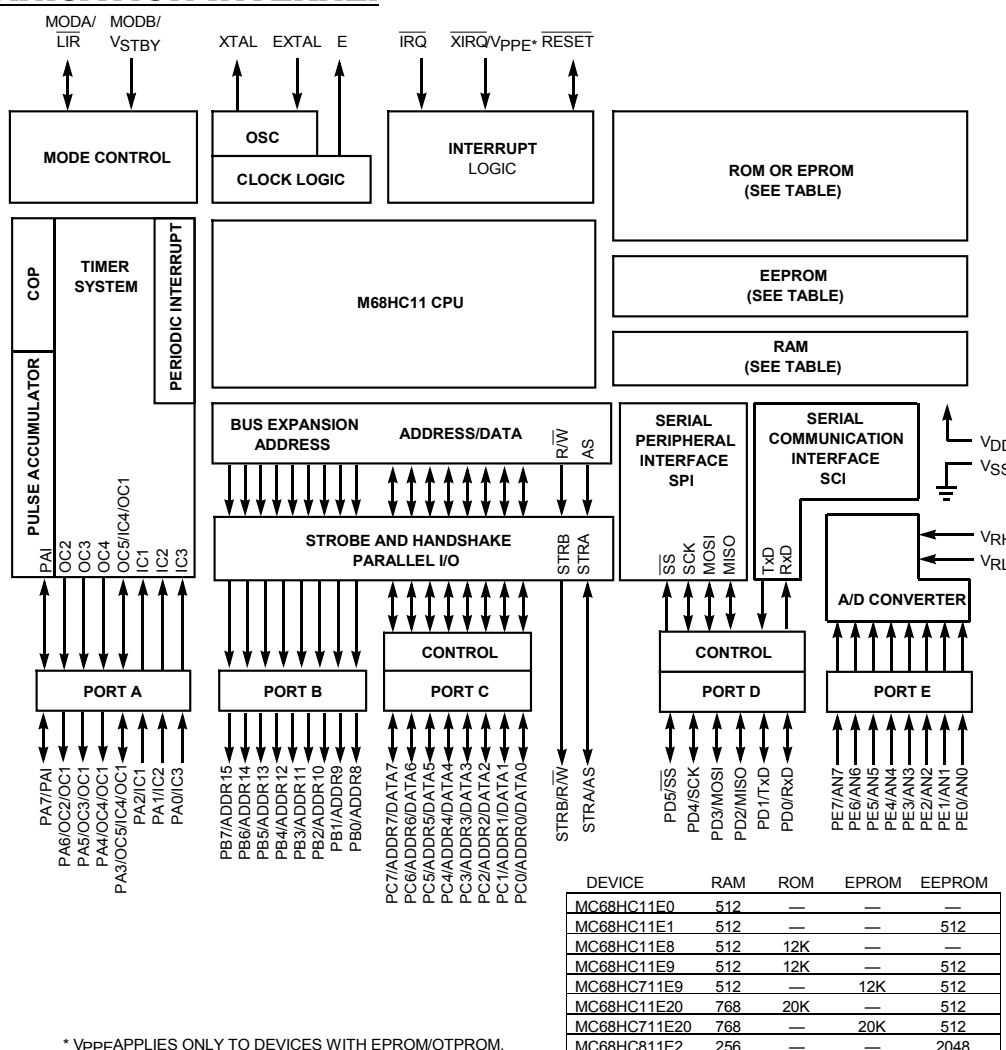
## 1) CARACTÉRISTIQUES DU MICRO CONTROLEUR

Le micro contrôleur **MC68HC811E2** de technologie **HCMOS** appartient à la famille des **68HC11**, le **8** signifie que c'est une version **EEPROM**, **C** le type et **2** la taille mémoire **2k octets**.

### Caractéristiques :

- Un oscillateur intégré.
- Une mémoire vive de 256 octets.
- Une mémoire morte EEPROM de 2048 octets.
- Chien de garde.
- Surveillance d'horloge.
- 38 Entrées et sorties : (16 entrées / sorties, 11 entrées et 11 sorties)
- Temporisateur 16 bits : ( 3 entrées de capture, 4 sorties de comparaison et une entrée de capture ou une sortie de comparaison PA3 (Sélection par logiciel).
- Interruption temps réel.
- Compteur d'événement 8 bits.
- Convertisseur analogique numérique 8 bits avec 8 entrées multiplexées.
- Une interface de communication série asynchrone. **(SCI) Serial Communication Interface**.
- Une interface de communication série synchrone. **(SPI) Serial Peripheral Interface**.
- Plusieurs modes de fonctionnement faible consommation.
- Une seule tension d'alimentation 3,3v ou 5V.
- Conservation des informations en mémoire vive jusqu'à 2V.

## 2) ORGANISATION INTERNE.



# LE MICRO CONTROLEUR MC68HC811E2

## 3) CONFIGURATION DU $\mu$ C.

### 3.1) Quatre configurations possibles.

La configuration du  $\mu$ C est définie par les broches **MODA** et **MODB**.

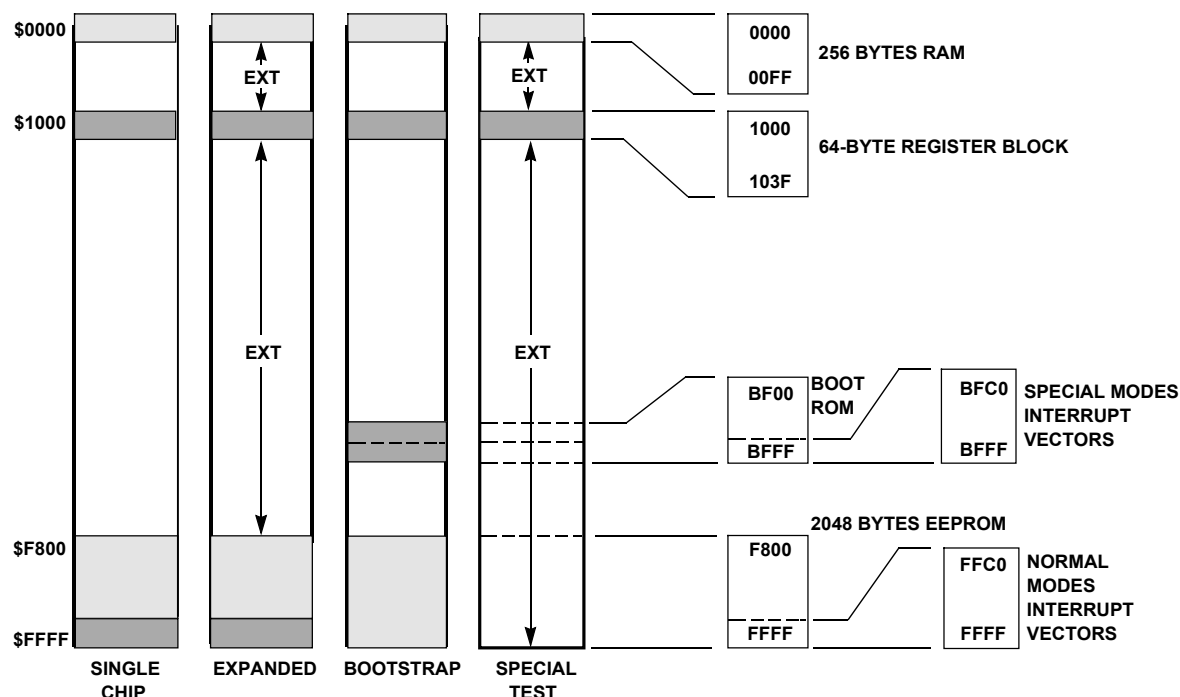
**MODA et MODB** déterminent l'une des quatre configurations suivantes :

MODB	MODA	Configuration
1	0	CIRCUIT SEUL (Mode micro contrôleur)
1	1	ÉTENDU (Mode micro processeur).
0	0	BOOTSTRAP (Mode autonome programme en RAM, dialogue avec un PC via un logiciel du type PCBUG11, ...)
0	1	TEST (Test en usine : ne pas utiliser, car on peut détruire certaine fonctionnalités du HC11)

Le  $\mu$ C est principalement utilisé en configuration **CIRCUIT SEUL**, ou **MONOCHIP**, ou encore **SINGLECHIP**. La configuration **ÉTENDU** permet entre autre d'augmenter la capacité mémoire du circuit ou de s'interfacer avec d'autres circuits programmables.

Seule la configuration **MONOCHIP** sera étudiée dans la suite du polycopié.

### 3.2) Configuration mémoire en fonction des modes.



## **4) LES MODES DE FONCTIONNEMENT DU $\mu$ C.**

Le  $\mu$ C peut fonctionner sous trois modes:

- RUN : NORMAL.
- STOP : ARRET.
- WAIT : ATTENTE.

### **4.1) Le mode RUN.**

Fonctionnement normal, c'est à dire que toutes les structures internes sont opérationnelles (SPI, SCI, COP et TIMER).

### **4.2) Le mode STOP.**

Il est déclenché par l'instruction **STOP**, il met le  $\mu$ C à l'arrêt, c'est à dire que toutes les structures internes (**CAN, SPI, SCI, COP et TIMER**) ne sont plus opérationnelles et de plus l'oscillateur interne est arrêté.

Le  $\mu$ C est en mode très faible consommation (quelques  $\mu$ A).

Le  $\mu$ C peut sortir de ce mode par :

- Un **RESET** externe.
- Une interruption **IRQ** ou **XIRQ**.

### **4.3) Le mode WAIT.**

Il est déclenché par l'instruction **WAIT**, il met le  $\mu$ C à l'arrêt. Le  $\mu$ C peut sortir de ce mode par une interruption (non masquée).

## 5) LES BROCHES DU $\mu$ C.

### 5.1) RESET.

Cette broche est bidirectionnelle, elle sert à initialiser le  $\mu$ C mais peut servir à initialiser des circuits périphériques lors de **RESETs** déclenchés par le chien de garde.

Le  $\mu$ C dispose de trois sources de **RESET** :

- **POR.**
- **EXTERNAL RESET.**
- **COP.**

**POR: (POWER ON RESET)** Mise sous tension.

A la mise sous tension du  $\mu$ C un front montant sur **RESET** déclenche l'initialisation du  $\mu$ C. Le temps nécessaire est de 4064 cycles machine.

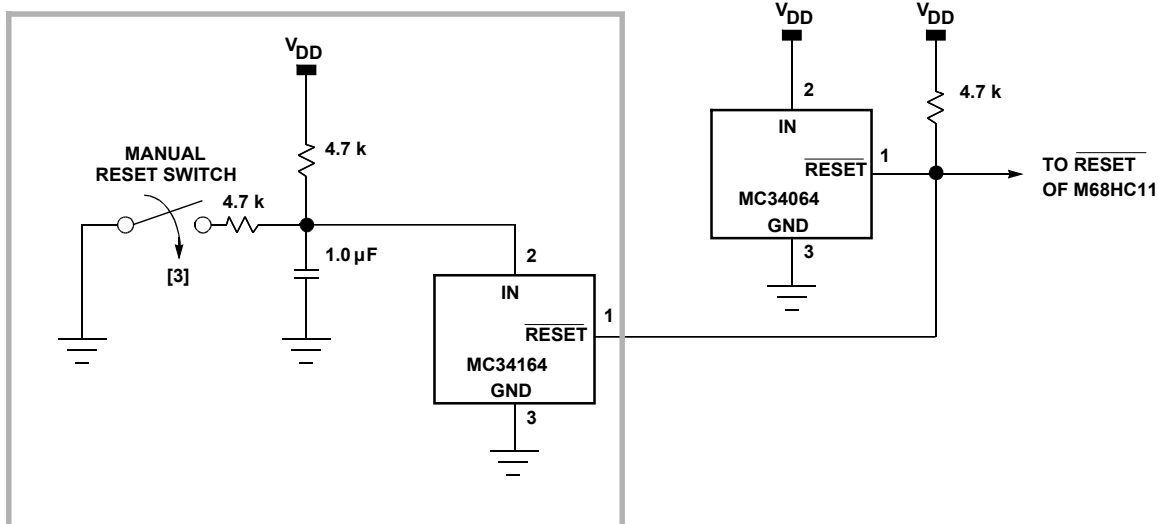
De plus, pour éviter des problèmes internes sur la mémoire EEPROM et sur les registres de même technologie, il faut que la broche **RESET** soit maintenue au niveau bas tant que la tension d'alimentation n'a pas atteint sa valeur nominale. Pour cela on utilise des circuits spécialisés de supervision d'alimentation (Exemple MC34064).

**EXTERNAL RESET.** Remise à zéro extérieure. Il faut appliquer un niveau bas sur l'entrée **RESET** pendant au moins 8 cycles machine pour que l'initialisation soit prise en compte.

**COP: (COMPUTER OPERATING PROPERLY):** Chien de garde.

Si le **COP** arrive à la fin du temps de garde sans avoir été rafraîchi alors il y a initialisation du  $\mu$ C, pour plus de détail voir le paragraphe sur le chien de garde.

#### Schéma structurel du câblage de la broche RESET.

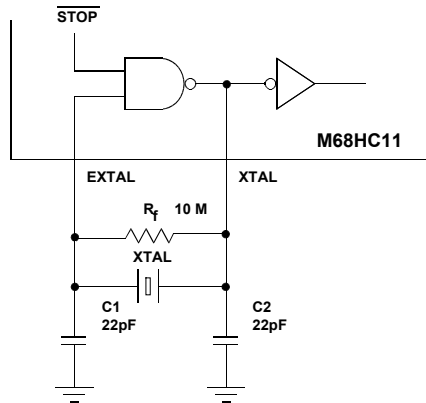


#### État du $\mu$ C au RESET :

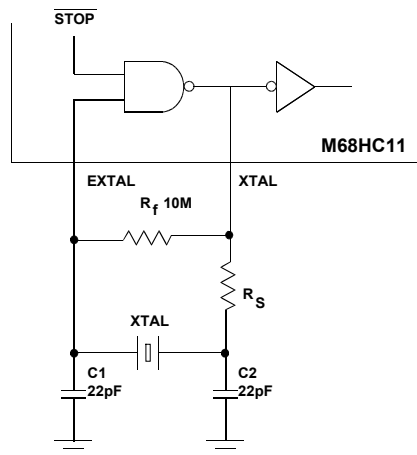
- Les masques d'interruptions, les structures internes (**SPI, SCI, COP et TIMER**) sont bloquées et les bits **I** et **X** du **CCR** sont à 1, ce qui bloque la prise en compte de toutes les interruptions.
- Le compteur programme **PC** est chargé avec la valeur du vecteur de **RESET** (\$FFFE et \$FFFF) et l'exécution des premières instructions peut commencer.

## 5.2) Oscillateur : OSC1 et OSC2 ou EXTAL et XTAL.

Ces broches permettent de faire fonctionner l'oscillateur interne du HC11.



High-Frequency Crystal Connections



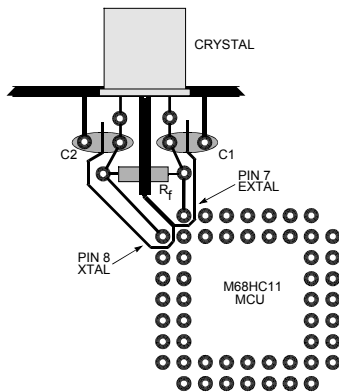
Low-Frequency Crystal Connections

**Remarque** : Le  $\mu\text{C}$  intègre en interne un diviseur par 4. Si la fréquence du **QUARTZ** est de **8MHz** la fréquence du bus interne sera de **2MHz**, en conséquence un cycle machine durera **0,5 $\mu\text{s}$** .

La fréquence **MAX** est de **2.1MHz** pour le  $\mu\text{C}$ .  
La fréquence **MIN** est le continu.

**Remarque** : La consommation du circuit sera d'autant plus faible que la fréquence sera petite, cela peut être intéressant pour des applications de faible consommation (alimentation autonome).

### Implantation souhaitable du circuit d'horloge :



Crystal Layout Example

## 5.3) Alimentation : VDD et VSS.

Ce sont les broches d'alimentation du circuit. Les tensions qui peuvent être appliquées sont **3,3V** ou **5V(+/-0.5V)** et l'intensité du courant consommé peut aller de **100 $\mu\text{A}$**  à **100mA**.

La consommation du  $\mu\text{C}$  sera fonction de :

- La tension d'alimentation.
- La fréquence interne.
- Le mode de fonctionnement.

De plus ces bornes doivent être découplées par deux condensateurs :

- 1 $\mu\text{F}$  électrolytique.



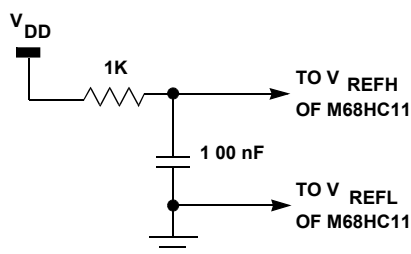
- 10nF céramique.

### **5.4) Interruptions : IRQ et XIRQ.**

Ces broches sont des entrées d'interruptions externes, elles servent à répondre à des événements extérieurs asynchrones (on ne peut prévoir le moment de l'événement). Ces broches doivent être maintenues à un niveau haut par l'intermédiaire de résistances de **4.7 k $\Omega$**  pour ne pas déclencher d'interruptions imprévues, cela permet aussi de relier plusieurs sources d'interruptions sur une même ligne (**OU CABLE**).

### **5.5) Référence de tension : VREFL et VREFH**

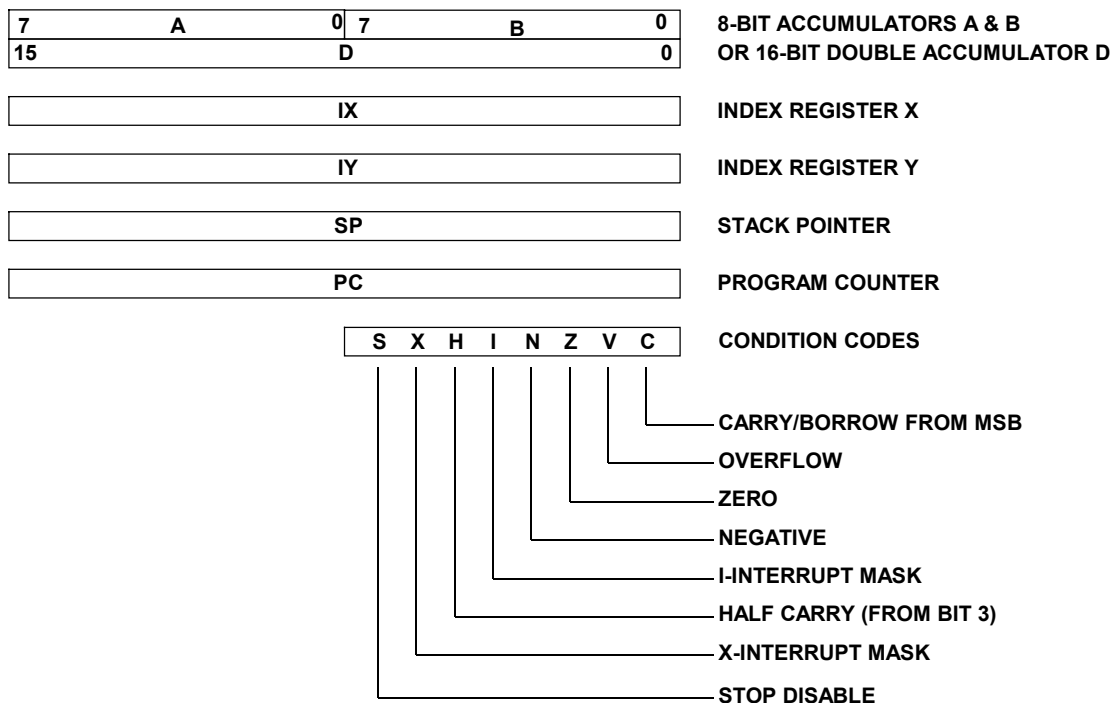
Ce sont les entrées de référence de tension du convertisseur A/N. Si la tension d'alimentation est correctement stabilisée, on peut utiliser le montage ci-dessous.



**Low-Pass Filter for A/D Reference Pins**

## 6) L'UNITÉ CENTRALE.

### 6.1) Organisation.



Elle est composée de :

- Deux accumulateurs 8 bits **A** et **B** qui peuvent se concaténer pour former un registre 16 bits **D** (**A** est l'octet de poids fort).
- Deux registres d'index 16 bits **X** et **Y**.
- Un registre de PILE 16 bits **SP**.
- Le compteur ordinal 16 bits **PC**.
- Un registre d'état 8 bits **CCR**.

	b7	b6	b5	b4	b3	b2	b1	b0
CCR	S	X	H	I	N	Z	V	C

- **S**: Interdiction du mode STOP.
- **X**: Masque d'interruption associée à la broche XIRQ.
- **H**: Demi retenue.
- **I**: Masque d'interruption associée à la broche IRQ.
- **N**: Négatif.
- **Z**: Zéro.
- **V**: Débordement
- **C**: Retenue.

# LE MICRO CONTROLEUR MC68HC811E2

## 7) JEU D'INSTRUCTIONS.

OPERATIONS ARITHMETIQUES		
additions	ADD	ABA ABX ABY ADCA ADCB ADDA ADDB ADDD
soustractions	SUBTRACT	SBA SBCA SBCB SUBA SUBB SUBD
comparaisons	COMPARE	CBA CMPA CMPB CPD CPX CPY
incrémentations	INCREMENT	INC INCA INCB INS INX INY
décrémentations	DECREMENT	DEC DECA DECB DES DEX DEY
négations (opposé)	NEGATION	NEG NEGA NEGB
multiplication-divisions		MUL FDIV IDIV
ajustement décimal		DAA

OPERATIONS LOGIQUES		
et	AND	ANDA ANDB
ou	OR	ORAA ORAB
ou exclusif	EXCLUSIVE OR	EORA EORB
complémentations	COMPLEMENT	COM COMA COMB
décalages	ARITHMETIC SHIFT LOGICAL SHIFT	ASL ASLA ASLB ASLD ASR ASRA ASRB LSL LSLA LSLB LSLD LSR LSRA LSRB LSRD
rotations	ROTATE	ROL ROLA ROLB ROR RORA RORB
mise à 0	CLEAR	CLR CLRA CLRB
tests	BIT TEST	BITA BITB TST TSTA TSTB

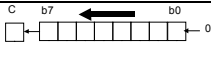
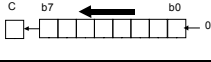

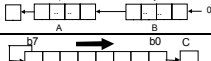
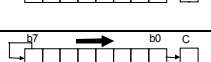

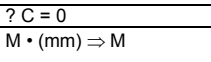
TRANSFERTS DE DONNEES		
chargements de registres	LOAD	LDAA LDAB LDD LDS LDX LDY
stockages de registres	STORE	STAA STAB STD STS STX STY
transferts de registres	TRANSFER	TAB TBA TPA TAP TSX TSY TXS TYS
échange de registres	EXCHANGE	XGDX XGDY
sauvegardes dans la pile	PUSH	PSHA PSHB PSHX PSHY
restitution à partir de la pile	PULL	PULA PULB PULX PULY

MANIPULATIONS DE BITS		
forçage de bits	BIT CLEAR BIT SET	BCLR BSET
tests de bits	BRANCH IF BIT CLEAR BRANCH IF BIT SET	BRCLR BRSET
contrôle des bits du CCR	CLEAR / SET	CLC CLI CLV SEC SEI SEV

TESTS ET BRANCHEMENTS		
	BRANCH IF	BCC BCS BVC BVS BEQ BNE BMI BPL BGT BLE BGE BLT BHI BLS BHS BLO

DIVERS		
		BSR JSR JMP BRA BRN RTS RTI NOP STOP SWI TEST WAI

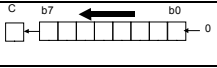
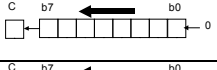

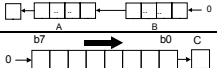

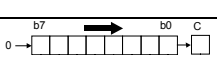
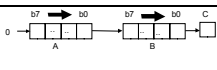
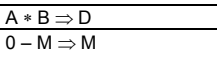
# LE MICRO CONTROLEUR MC68HC811E2

Mnémonique	Opération	Description	Mode adressage	Instruction			Registre Code Condition							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
ABA	Add Accumulators	$A + B \Rightarrow A$	INH	1B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to X	$IX + (00 : B) \Rightarrow IX$	INH	3A	—	3	—	—	—	—	—	—	—	—
ABY	Add B to Y	$IY + (00 : B) \Rightarrow IY$	INH	18 3A	—	4	—	—	—	—	—	—	—	—
ADCA (opr)	Add with Carry to A	$A + M + C \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	89 99 B9 A9 18 A9	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADCB (opr)	Add with Carry to B	$B + M + C \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C9 D9 F9 E9 18 E9	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDA (opr)	Add Memory to A	$A + M \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	8B 9B BB AB 18 AB	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDB (opr)	Add Memory to B	$B + M \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	CB DB FB EB 18 EB	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDD (opr)	Add 16-Bit to D	$D + (M : M + 1) \Rightarrow D$	IMM DIR EXT IND,X IND,Y	C3 D3 F3 E3 18 E3	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ANDA (opr)	AND A with Memory	$A \cdot M \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	84 94 B4 A4 18 A4	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ANDB (opr)	AND B with Memory	$B \cdot M \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C4 D4 F4 E4 18 E4	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ASL (opr)	Arithmetic Shift Left		EXT IND,X IND,Y	78 68 18 68	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ASLA	Arithmetic Shift Left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic Shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic Shift Left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
ASR	Arithmetic Shift Right		EXT IND,X IND,Y	77 67 18 67	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ASRA	Arithmetic Shift Right A		A INH	47	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRB	Arithmetic Shift Right B		B INH	57	—	2	—	—	—	—	Δ	Δ	Δ	Δ
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	rr	3	—	—	—	—	—	—	—	—
BCLR (opr) (msk)	Clear Bit(s)	$M \cdot (mm) \Rightarrow M$	DIR IND,X IND,Y	15 1D 18 1D	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
BCS (rel)	Branch if Carry Set	? C = 1	REL	25	rr	3	—	—	—	—	—	—	—	—
BEQ (rel)	Branch if = Zero	? Z = 1	REL	27	rr	3	—	—	—	—	—	—	—	—
BGE (rel)	Branch if Δ Zero	? N ⊕ V = 0	REL	2C	rr	3	—	—	—	—	—	—	—	—
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	REL	2E	rr	3	—	—	—	—	—	—	—	—
BHI (rel)	Branch if Higher	? C + Z = 0	REL	22	rr	3	—	—	—	—	—	—	—	—
BHS (rel)	Branch if Higher or Same	? C = 0	REL	24	rr	3	—	—	—	—	—	—	—	—
BITA (opr)	Bit(s) Test A with Memory	$A \cdot M$	A IMM A DIR A EXT A IND,X A IND,Y	85 95 B5 A5 18 A5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BITB (opr)	Bit(s) Test B with Memory	$B \cdot M$	B IMM B DIR B EXT B IND,X B IND,Y	C5 D5 F5 E5 18 E5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—

# LE MICRO CONTROLEUR MC68HC811E2

Mnémonique	Opération	Description	Mode adressage	Instruction			Registre Code Condition							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
BLE (rel)	Branch if $\Delta$ Zero	$? Z + (N \oplus V) = 1$	REL	2F	rr	3	—	—	—	—	—	—	—	—
BLO (rel)	Branch if Lower	$? C = 1$	REL	25	rr	3	—	—	—	—	—	—	—	—
BLS (rel)	Branch if Lower or Same	$? C + Z = 1$	REL	23	rr	3	—	—	—	—	—	—	—	—
BLT (rel)	Branch if < Zero	$? N \oplus V = 1$	REL	2D	rr	3	—	—	—	—	—	—	—	—
BMI (rel)	Branch if Minus	$? N = 1$	REL	2B	rr	3	—	—	—	—	—	—	—	—
BNE (rel)	Branch if not = Zero	$? Z = 0$	REL	26	rr	3	—	—	—	—	—	—	—	—
BPL (rel)	Branch if Plus	$? N = 0$	REL	2A	rr	3	—	—	—	—	—	—	—	—
BRA (rel)	Branch Always	$? 1 = 1$	REL	20	rr	3	—	—	—	—	—	—	—	—
BRCLR (opr) (msk) (rel)	Branch if Bit(s) Clear	$? M \cdot mm = 0$	DIR IND,X IND,Y	13 1F 18 1F	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BRN (rel)	Branch Never	$? 1 = 0$	REL	21	rr	3	—	—	—	—	—	—	—	—
BRSET (opr) (msk) (rel)	Branch if Bit(s) Set	$? (M) \cdot mm = 0$	DIR IND,X IND,Y	12 1E 18 1E	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BSET (opr) (msk)	Set Bit(s)	$M + mm \Rightarrow M$	DIR IND,X IND,Y	14 1C 18 1C	dd mm ff mm ff mm	6 7 8	—	—	—	—	$\Delta$	$\Delta$	0	—
BSR (rel)	Branch to Subroutine	See Figure 3-2	REL	8D	rr	6	—	—	—	—	—	—	—	—
BVC (rel)	Branch if Overflow Clear	$? V = 0$	REL	28	rr	3	—	—	—	—	—	—	—	—
BVS (rel)	Branch if Overflow Set	$? V = 1$	REL	29	rr	3	—	—	—	—	—	—	—	—
CBA	Compare A to B	$A - B$	INH	11	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
CLC	Clear Carry Bit	$0 \Rightarrow C$	INH	0C	—	2	—	—	—	—	—	—	—	0
CLI	Clear Interrupt Mask	$0 \Rightarrow I$	INH	0E	—	2	—	—	—	0	—	—	—	—
CLR (opr)	Clear Memory Byte	$0 \Rightarrow M$	EXT IND,X IND,Y	7F 6F 18 6F	hh ll ff ff	6 6 7	—	—	—	—	0	1	0	0
CLRA	Clear Accumulator A	$0 \Rightarrow A$	A INH	4F	—	2	—	—	—	—	0	1	0	0
CLRB	Clear Accumulator B	$0 \Rightarrow B$	B INH	5F	—	2	—	—	—	—	0	1	0	0
CLV	Clear Overflow Flag	$0 \Rightarrow V$	INH	0A	—	2	—	—	—	—	—	—	—	0
CPMA (opr)	Compare A to Memory	$A - M$	A IMM A DIR A EXT A IND,X A IND,Y	81 91 B1 A1 18 A1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
CMPB (opr)	Compare B to Memory	$B - M$	B IMM B DIR B EXT B IND,X B IND,Y	C1 D1 F1 E1 18 E1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
COM (opr)	Ones Complement Memory Byte	$\$FF - M \Rightarrow M$	EXT IND,X IND,Y	73 63 18 63	hh ll ff ff	6 6 7	—	—	—	—	$\Delta$	$\Delta$	0	1
COMA	Ones Complement A	$\$FF - A \Rightarrow A$	A INH	43	—	2	—	—	—	—	$\Delta$	$\Delta$	0	1
COMB	Ones Complement B	$\$FF - B \Rightarrow B$	B INH	53	—	2	—	—	—	—	$\Delta$	$\Delta$	0	1
CPD (opr)	Compare D to Memory 16-Bit	$D - M : M + 1$	IMM DIR EXT IND,X IND,Y	1A 83 1A 93 1A B3 1A A3 CD A3	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
CPX (opr)	Compare X to Memory 16-Bit	$IX - M : M + 1$	IMM DIR EXT IND,X IND,Y	8C 9C BC AC CD AC	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
CPY (opr)	Compare Y to Memory 16-Bit	$IY - M : M + 1$	IMM DIR EXT IND,X IND,Y	18 8C 18 9C 18 BC 1A AC 18 AC	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
DEC (opr)	Decrement Memory Byte	$M - 1 \Rightarrow M$	EXT IND,X IND,Y	7A 6A 18 6A	hh ll ff ff	6 6 7	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	—
DECA	Decrement Accumulator A	$A - 1 \Rightarrow A$	A INH	4A	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	—
DECB	Decrement Accumulator B	$B - 1 \Rightarrow B$	B INH	5A	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	—
DES	Decrement Stack Pointer	$SP - 1 \Rightarrow SP$	INH	34	—	3	—	—	—	—	—	—	—	—
DEX	Decrement Index Register X	$IX - 1 \Rightarrow IX$	INH	09	—	3	—	—	—	—	$\Delta$	—	—	—
DEY	Decrement Index Register Y	$IY - 1 \Rightarrow IY$	INH	18 09	—	4	—	—	—	—	$\Delta$	—	—	—
EORA (opr)	Exclusive OR A with Memory	$A \oplus M \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	88 98 B8 A8 18 A8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	$\Delta$	$\Delta$	0	—
EORB (opr)	Exclusive OR B with Memory	$B \oplus M \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C8 D8 F8 E8 18 E8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	$\Delta$	$\Delta$	0	—

# LE MICRO CONTROLEUR MC68HC811E2

Mnémonique	Opération	Description	Mode adressage	Instruction			Registre Code Condition							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
FDIV	Fractional Divide 16 by 16	$D / IX \Rightarrow IX; r \Rightarrow D$		03	—	41	—	—	—	—	—	$\Delta$	$\Delta$	$\Delta$
IDIV	Integer Divide 16 by 16	$D / IX \Rightarrow IX; r \Rightarrow D$	INH	02	—	41	—	—	—	—	—	$\Delta$	0	$\Delta$
INC (opr)	Increment Memory Byte	$M + 1 \Rightarrow M$	EXT IND,X IND,Y	7C 6C 18 6C	hh ll ff ff	6 6 7	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	—
INCA	Increment Accumulator A	$A + 1 \Rightarrow A$	A INH	4C	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	—
INCB	Increment Accumulator B	$B + 1 \Rightarrow B$	B INH	5C	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	—
INS	Increment Stack Pointer	$SP + 1 \Rightarrow SP$	INH	31	—	3	—	—	—	—	—	—	—	—
INX	Increment Index Register X	$IX + 1 \Rightarrow IX$	INH	08	—	3	—	—	—	—	—	$\Delta$	—	—
INY	Increment Index Register Y	$IY + 1 \Rightarrow IY$	INH	18 08	—	4	—	—	—	—	—	$\Delta$	—	—
JMP (opr)	Jump	See Figure 3-2	EXT IND,X IND,Y	7E 6E 18 6E	hh ll ff ff	3 4 4	—	—	—	—	—	—	—	—
JSR (opr)	Jump to Subroutine	See Figure 3-2	DIR EXT IND,X IND,Y	9D BD AD 18 AD	dd hh ll ff ff	5 6 6 7	—	—	—	—	—	—	—	—
LDA (opr)	Load Accumulator A	$M \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	86 96 B6 A6 18 A6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	$\Delta$	$\Delta$	0	—
LDAB (opr)	Load Accumulator B	$M \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C6 D6 F6 E6 18 E6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	$\Delta$	$\Delta$	0	—
LDD (opr)	Load Double Accumulator D	$M \Rightarrow A, M + 1 \Rightarrow B$	IMM DIR EXT IND,X IND,Y	CC DC FC EC 18 EC	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	$\Delta$	$\Delta$	0	—
LDS (opr)	Load Stack Pointer	$M : M + 1 \Rightarrow SP$	IMM DIR EXT IND,X IND,Y	8E 9E BE AE 18 AE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	$\Delta$	$\Delta$	0	—
LDX (opr)	Load Index Register X	$M : M + 1 \Rightarrow IX$	IMM DIR EXT IND,X IND,Y	CE DE FE EE CD EE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	$\Delta$	$\Delta$	0	—
LDY (opr)	Load Index Register Y	$M : M + 1 \Rightarrow IY$	IMM DIR EXT IND,X IND,Y	18 CE 18 DE 18 FE 1A EE 18 EE	jj kk dd hh ll ff ff	4 5 6 6 6	—	—	—	—	$\Delta$	$\Delta$	0	—
LSL (opr)	Logical Shift Left		EXT IND,X IND,Y	78 68 18 68	hh ll ff ff	6 6 7	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
LSLA	Logical Shift Left A		A INH	48	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
LSLB	Logical Shift Left B		B INH	58	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
LSLD	Logical Shift Left Double		INH	05	—	3	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
LSR (opr)	Logical Shift Right		EXT IND,X IND,Y	74 64 18 64	hh ll ff ff	6 6 7	—	—	—	—	0	$\Delta$	$\Delta$	$\Delta$
LSRA	Logical Shift Right A		A INH	44	—	2	—	—	—	—	0	$\Delta$	$\Delta$	$\Delta$
LSRB	Logical Shift Right B		B INH	54	—	2	—	—	—	—	0	$\Delta$	$\Delta$	$\Delta$
LSRD	Logical Shift Right Double		INH	04	—	3	—	—	—	—	0	$\Delta$	$\Delta$	$\Delta$
MUL	Multiply 8 by 8	$A * B \Rightarrow D$	INH	3D	—	10	—	—	—	—	—	—	—	$\Delta$
NEG (opr)	Two's Complement Memory Byte	$0 - M \Rightarrow M$	EXT IND,X IND,Y	70 60 18 60	hh ll ff ff	6 6 7	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
NEGA	Two's Complement A	$0 - A \Rightarrow A$	A INH	40	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
NEGB	Two's Complement B	$0 - B \Rightarrow B$	B INH	50	—	2	—	—	—	—	$\Delta$	$\Delta$	$\Delta$	$\Delta$
NOP	No operation	No Operation	INH	01	—	2	—	—	—	—	—	—	—	—

# LE MICRO CONTROLEUR MC68HC811E2

Mnémonique	Opération	Description	Mode adressage	Instruction			Registre Code Condition							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
ORAA (opr)	OR Accumulator A (Inclusive)	$A + M \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	8A	ii	2	—	—	—	—	—	Δ	Δ	0
				9A	dd	3	—	—	—	—	—	—	—	—
				BA	hh ll	4	—	—	—	—	—	—	—	—
				AA	ff	4	—	—	—	—	—	—	—	—
				18 AA	ff	5	—	—	—	—	—	—	—	—
ORAB (opr)	OR Accumulator B (Inclusive)	$B + M \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	CA	ii	2	—	—	—	—	—	Δ	Δ	0
				DA	dd	3	—	—	—	—	—	—	—	—
				FA	hh ll	4	—	—	—	—	—	—	—	—
				EA	ff	4	—	—	—	—	—	—	—	—
				18 EA	ff	5	—	—	—	—	—	—	—	—
PSHA	Push A onto Stack	$A \Rightarrow \text{Stk}, SP = SP - 1$	A INH	36	—	3	—	—	—	—	—	—	—	—
PSHB	Push B onto Stack	$B \Rightarrow \text{Stk}, SP = SP - 1$	B INH	37	—	3	—	—	—	—	—	—	—	—
PSHX	Push X onto Stack (Low First)	$IX \Rightarrow \text{Stk}, SP = SP - 2$	INH	3C	—	4	—	—	—	—	—	—	—	—
PSHY	Push Y onto Stack (Low First)	$IY \Rightarrow \text{Stk}, SP = SP - 2$	INH	18 3C	—	5	—	—	—	—	—	—	—	—
PULA	Pull A from Stack	$SP = SP + 1, A \Leftarrow \text{Stk}A$	INH	32	—	4	—	—	—	—	—	—	—	—
PULB	Pull B from Stack	$SP = SP + 1, B \Leftarrow \text{Stk}B$	INH	33	—	4	—	—	—	—	—	—	—	—
PULX	Pull X From Stack (Hi First)	$SP = SP + 2, IX \Leftarrow \text{Stk}$	INH	38	—	5	—	—	—	—	—	—	—	—
PULY	Pull Y from Stack (Hi First)	$SP = SP + 2, IY \Leftarrow \text{Stk}$	INH	18 38	—	6	—	—	—	—	—	—	—	—
ROL (opr)	Rotate Left		EXT IND,X IND,Y	79	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
				69	ff	6	—	—	—	—	—	—	—	—
				18 69	ff	7	—	—	—	—	—	—	—	—
ROLA	Rotate Left A		A INH	49	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLB	Rotate Left B		B INH	59	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROR (opr)	Rotate Right		EXT IND,X IND,Y	76	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
				66	ff	6	—	—	—	—	—	—	—	—
				18 66	ff	7	—	—	—	—	—	—	—	—
RORA	Rotate Right A		A INH	46	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORB	Rotate Right B		B INH	56	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RTI	Return from Interrupt	See Figure 3-2	INH	3B	—	12	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
RTS	Return from Subroutine	See Figure 3-2	INH	39	—	5	—	—	—	—	—	—	—	—
SBA	Subtract B from A	$A - B \Rightarrow A$	INH	10	—	2	—	—	—	—	Δ	Δ	Δ	Δ
SBCA (opr)	Subtract with Carry from A	$A - M - C \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	82	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
				92	dd	3	—	—	—	—	—	—	—	—
				B2	hh ll	4	—	—	—	—	—	—	—	—
				A2	ff	4	—	—	—	—	—	—	—	—
				18 A2	ff	5	—	—	—	—	—	—	—	—
SBCB (opr)	Subtract with Carry from B	$B - M - C \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C2	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
				D2	dd	3	—	—	—	—	—	—	—	—
				F2	hh ll	4	—	—	—	—	—	—	—	—
				E2	ff	4	—	—	—	—	—	—	—	—
				18 E2	ff	5	—	—	—	—	—	—	—	—
SEC	Set Carry	$1 \Rightarrow C$	INH	0D	—	2	—	—	—	—	—	—	—	1
SEI	Set Interrupt Mask	$1 \Rightarrow I$	INH	0F	—	2	—	—	1	—	—	—	—	—
SEV	Set Overflow Flag	$1 \Rightarrow V$	INH	0B	—	2	—	—	—	—	—	—	1	—
STAA (opr)	Store Accumulator A	$A \Rightarrow M$	A DIR A EXT A IND,X A IND,Y	97	dd	3	—	—	—	—	Δ	Δ	0	—
				B7	hh ll	4	—	—	—	—	—	—	—	—
				A7	ff	4	—	—	—	—	—	—	—	—
				18 A7	ff	5	—	—	—	—	—	—	—	—
STAB (opr)	Store Accumulator B	$B \Rightarrow M$	B DIR B EXT B IND,X B IND,Y	D7	dd	3	—	—	—	—	Δ	Δ	0	—
				F7	hh ll	4	—	—	—	—	—	—	—	—
				E7	ff	4	—	—	—	—	—	—	—	—
				18 E7	ff	5	—	—	—	—	—	—	—	—
STD (opr)	Store Accumulator D	$A \Rightarrow M, B \Rightarrow M + 1$	DIR EXT IND,X IND,Y	DD	dd	4	—	—	—	—	Δ	Δ	0	—
				FD	hh ll	5	—	—	—	—	—	—	—	—
				ED	ff	5	—	—	—	—	—	—	—	—
				18 ED	ff	6	—	—	—	—	—	—	—	—
STOP	Stop Internal Clocks	—	INH	CF	—	2	—	—	—	—	—	—	—	—
STS (opr)	Store Stack Pointer	$SP \Rightarrow M : M + 1$	DIR EXT IND,X IND,Y	9F	dd	4	—	—	—	—	Δ	Δ	0	—
				BF	hh ll	5	—	—	—	—	—	—	—	—
				AF	ff	5	—	—	—	—	—	—	—	—
				18 AF	ff	6	—	—	—	—	—	—	—	—
STX (opr)	Store Index Register X	$IX \Rightarrow M : M + 1$	DIR EXT IND,X IND,Y	DF	dd	4	—	—	—	—	Δ	Δ	0	—
				FF	hh ll	5	—	—	—	—	—	—	—	—
				EF	ff	5	—	—	—	—	—	—	—	—
				CD EF	ff	6	—	—	—	—	—	—	—	—
STY (opr)	Store Index Register Y	$IY \Rightarrow M : M + 1$	DIR EXT IND,X IND,Y	18 DF	dd	5	—	—	—	—	Δ	Δ	0	—
				18 FF	hh ll	6	—	—	—	—	—	—	—	—
				1A EF	ff	6	—	—	—	—	—	—	—	—
				18 EF	ff	6	—	—	—	—	—	—	—	—
SUBA (opr)	Subtract Memory from A	$A - M \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	80	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
				90	dd	3	—	—	—	—	—	—	—	—
				B0	hh ll	4	—	—	—	—	—	—	—	—
				A0	ff	4	—	—	—	—	—	—	—	—
				18 A0	ff	5	—	—	—	—	—	—	—	—

# LE MICRO CONTROLEUR MC68HC811E2

Mnémonique	Opération	Description	Mode adressage	Instruction			Registre Code Condition							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
SUBB (opr)	Subtract Memory from B	$B - M \Rightarrow B$	A IMM A DIR A EXT A IND,X A IND,Y	C0	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
				D0	dd	3								
				F0	hh ll	4								
				E0	ff	4								
				18 E0	ff	5								
SUBD (opr)	Subtract Memory from D	$D - M : M + 1 \Rightarrow D$	IMM DIR EXT IND,X IND,Y	83	jj kk	4	—	—	—	—	Δ	Δ	Δ	Δ
				93	dd	5								
				B3	hh ll	6								
				A3	ff	6								
				18 A3	ff	7								
SWI	Software Interrupt	See Figure 3-2	INH	3F	—	14	—	—	—	1	—	—	—	—
TAB	Transfer A to B	$A \Rightarrow B$	INH	16	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CC Register	$A \Rightarrow CCR$	INH	06	—	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	$B \Rightarrow A$	INH	17	—	2	—	—	—	—	Δ	Δ	0	—
TEST	TEST (Only in Test Modes)	Address Bus Counts	INH	00	—	*	—	—	—	—	—	—	—	—
TPA	Transfer CC Register to A	$CCR \Rightarrow A$	INH	07	—	2	—	—	—	—	—	—	—	—
TST (opr)	Test for Zero or Minus	$M - 0$	EXT IND,X IND,Y	7D	hh ll	6	—	—	—	—	Δ	Δ	0	0
				6D	ff	6								
				18 6D	ff	7								
TSTA	Test A for Zero or Minus	$A - 0$	A INH	4D	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for Zero or Minus	$B - 0$	B INH	5D	—	2	—	—	—	—	Δ	Δ	0	0
TSX	Transfer Stack Pointer to X	$SP + 1 \Rightarrow IX$	INH	30	—	3	—	—	—	—	—	—	—	—
TSY	Transfer Stack Pointer to Y	$SP + 1 \Rightarrow IY$	INH	18 30	—	4	—	—	—	—	—	—	—	—
TXS	Transfer X to Stack Pointer	$IX - 1 \Rightarrow SP$	INH	35	—	3	—	—	—	—	—	—	—	—
TYS	Transfer Y to Stack Pointer	$IY - 1 \Rightarrow SP$	INH	18 35	—	4	—	—	—	—	—	—	—	—
WAI	Wait for Interrupt	Stack Regs & WAIT	INH	3E	—	**	—	—	—	—	—	—	—	—
XGDX	Exchange D with X	$IX \Rightarrow D, D \Rightarrow IX$	INH	8F	—	3	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	$IY \Rightarrow D, D \Rightarrow IY$	INH	18 8F	—	4	—	—	—	—	—	—	—	—

LEGENDE DES TABLEAUX D'INSTRUCTIONS			
dd	Adressage direct (8 bits)	INH	Adressage inhérent
ff	Offset positif sur 8 bits	IMM	Adressage immédiat
hh	Octet de poids fort d'une adresse sur 16 bits	DIR	Adressage direct
ii	Octet de donnée en mode immédiat	EXT	Adressage étendu
jj	Octet de poids fort d'une donnée immédiate sur 16 bits	IND	Adressage indexé
kk	Octet de poids faible d'une donnée immédiate sur 16 bits	-	Bit non affecté
ll	Octet de poids faible d'une adresse sur 16 bits	0	Bit mis à 0
mm	Masque sur 8 bits	1	Bit mis à 1
rr	Offset signé sur 8 bits	Δ	Bit mis à 0 ou 1 selon le résultat



## 8) LES MODES D'ADRESSAGE

### 8.1) Adressage inhérent ou implicite

#### 8.1.1) Description.

Le mnémonique de l'instruction mentionne la donnée sur laquelle porte l'opération (contenu des registres), ou aucune donnée n'est nécessaire.

#### 8.1.2) Syntaxe.

MNEMONIQUE

#### 8.1.3) Exemples.

INCA	* incrémentation de l'accumulateur A
ABX	* addition de l'accumulateur B et du registre d'index X (résultat dans X)
NOP	* aucune opération (temporisation)

### 8.2) Adressage immédiat.

#### 8.2.1) Description.

L'instruction porte sur une valeur constante indiquée immédiatement après le mnémonique.

#### 8.2.2) Syntaxe.

MNEMONIQUE      #constante

#### 8.2.3) Exemples.

Maxi	EQU	100	
	LDAA	#255	* charge \$FF dans l'accumulateur A
	LDD	#Maxi	* charge \$0064 dans le registre D

### 8.3) Adressages direct et étendu.

#### 8.3.1) Description.

L'instruction porte sur le contenu de l'adresse mentionnée après le mnémonique.

L'adressage direct permet d'accéder au contenu des adresses \$0000 à \$00FF (le poids fort est obligatoirement \$00).

L'adressage étendu permet d'accéder à l'intégralité du plan mémoire (\$0000 à \$FFFF).

L'assembleur utilise le mode d'adressage adapté :

- valeur précisée < 256 : adressage direct
- valeur précisée ≥ 256 : adressage étendu

#### 8.3.2) Syntaxe.

MNEMONIQUE      Adresse

#### 8.3.3) Exemples.

PORTB	EQU	\$1004	
VARIABLE	EQU	\$0010	
	LDAB	VARIABLE	* charge la variable dans B (adressage direct)
	STAB	PORTB	* stocke le contenu de B sur le PortB (adressage étendu)

### 8.4) Adressage relatif.

#### 8.4.1) Description.

Ce mode d'adressage est utilisé pour les branchements à la suite de tests.

Il précise le déplacement à effectuer pour déterminer l'adresse de la prochaine instruction à exécuter. Le déplacement est une valeur signée (-128 à +127).

Si le résultat du test est positif

alors il y a branchement à l'adresse de destination précisée  
sinon l'instruction suivante est exécutée.

## 8.4.2) Syntaxe.

MNEMONIQUEduTEST                      EtiquetteDeDestination

## 8.4.3) Exemples.

Compteur EQU	\$0010		
Tempo LDAB	Compteur	* adressage direct	
BEQ	Fin	* adressage relatif	
Boucle DECB		* adressage inhérent	
BNE	Boucle	* adressage relatif	
Fin RTS		* adressage inhérent	

## 8.5) Adressage indexé

### 8.5.1) Description.

L'opérande précise l'adresse de la donnée à traiter sous la forme "Constante + Adresse de Base".  
L'adresse de base est en fait le contenu d'un registre d'index (X ou Y : 16 bits).  
La constante est une valeur codée sur 8 bits (0 à 255).

### 8.5.2) Syntaxe.

MNEMONIQUE	Constante, X	pour le registre d'index X
MNEMONIQUE	Constante, Y	pour le registre d'index Y

### 8.5.3) Exemples.

Decalage EQU	10	
ORG	\$F800	
LDX	#\$1000	
LDY	#100	
LDAB	0, X	* charge dans B le contenu de l'adresse \$1000 (Port A)
LDAA	4, X	* charge dans A le contenu de l'adresse \$1004 (Port B)
STD	Decalage, Y	* stocke le contenu de D à l'adresse \$006E et \$006F

## 8.6) Manipulation de bits.

### 8.6.1) Forçage de bits.

#### 8.6.1.1) Description.

Il s'agit de 2 instructions permettant de mettre à 0 ou 1 un ou plusieurs bits d'un octet de l'espace mémoire. Elles sont le plus souvent utilisées pour positionner des bits des registres du µC.

#### 8.6.1.2) Syntaxe.

	BSET	adresse, #masque	pour mettre à 1
ou	BCLR	adresse, #masque	pour mettre à 0

#### 8.6.1.3) Mode d'adressage.

Ces 2 instructions supportent les modes d'adressage :

- direct : l'espace mémoire accessible est donc \$0000 à \$ 00FF
- indexé : l'espace mémoire accessible est donc \$0000 à \$FFFF grâce à l'utilisation d'un registre d'index (X ou Y : 16 bits).

#### 8.6.1.4) Masque.

Le masque permet d'identifier le ou les bits à forcer. C'est une valeur constante, il est donc spécifié avec le symbole immédiat : #

## 8.6.1.5) Exemples.

```

QuartH EQU    %11110000
Entrees EQU    %01100001
        LDX     #$1000
        BCLR    0,X,#QuartH  * met à 0 les 4 bits de poids fort du registre d'adresse
                               $1000 : PortA
        BSET    4,X,#Entrees * met à 1 les bits 6, 5 et 0 du registre d'adresse $1004 :
                               PortB

```

## 8.6.2) Test de bits.

### 8.6.2.1) Description.

Il s'agit de 2 instructions permettant de tester un ou plusieurs bits d'un octet de l'espace mémoire. Elles sont le plus souvent utilisées pour déterminer l'état des bits des registres du µC.

En fonction du résultat du test :

- le programme se poursuit avec l'instruction suivante (résultat du test faux)
- le programme se poursuit avec l'instruction dont l'adresse est précisée : il s'agit d'un branchement (résultat du test vrai).

Les modes d'adressage et le masque ont été décrit pour les instructions de forçage de bits.

### 8.6.2.2) Syntaxe.

```

        BRSET    adresse,#masque,EtiquetteDuBranchement
                               pour le test de bits à 1
ou      BRCLR    adresse,#masque,EtiquetteDuBranchement
                               pour le test de bits à 0

```

### 8.6.2.3) Exemples.

```

        LDX      #$1000
Attente BRSET    0,X,#4,Attente
        BRCLR    3,X,#129,Attente

```

L'instruction BRSET boucle sur elle même tant que le bit 2 (masque = % 0000 0100) du registre d'adresse \$1000 (PortA) est à 1

L'instruction BRCLR boucle à l'étiquette Attente tant que les bits 7 et 0 (masque = % 1000 0001) du registre d'adresse \$1003 (PortC) sont à 0

Ces instructions permettent donc d'attendre que PA2 passe à 0, PC7 passe à 1 et PC0 passe à 1.

# LE MICRO CONTROLEUR MC68HC811E2

## 9) LES PORTS D'ENTREES SORTIES

Le  $\mu$ C dispose de 5 PORTS:

- Bidirectionnels (D et C).
- Unidirectionnels (A, B et E).

**Port A** : les broches PA7 à PA0 sont utilisées conjointement par le Timer

Sens	E/S	S	S	S	E/S	E	E	E	
<b>PORTA</b>	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	\$1000
Reset	E	?	?	?	E	E	E	E	

Le sens de PA7 et PA3 est défini par les bits DDRA7 et DDRA3 du registre PACTL

<b>PACTL</b>	DDRA7	-	-	-	DDRA3	-	-	-	\$1026
Reset	0	-	-	-	0	-	-	-	

**Port B** : les broches PB7 à PB0 sont toutes des sorties

Sens	S	S	S	S	S	S	S		
<b>PORTB</b>	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	\$1004
Reset	0	0	0	0	0	0	0	0	

**Port C** : les broches PC7 à PC0 sont toutes bidirectionnelles

Sens	E/S	E/S	E/S	E/S	E/S	E/S	E/S	E/S	
<b>PORTC</b>	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	\$1003
Reset	E	E	E	E	E	E	E	E	

Le sens des broches PC7 à PC0 est défini par le registre DDRC

<b>DDRC</b>	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	\$1007
Reset	0	0	0	0	0	0	0	0	

**Port D** : les broches PD5 à PD0 sont utilisées conjointement par les liaisons SPI et SCI

Sens			E/S	E/S	E/S	E/S	E/S	E/S	
<b>PORTD</b>	-	-	PD5	PD4	PD3	PD2	PD1	PD0	\$1008
Reset			E	E	E	E	E	E	

Le sens des broches PD5 à PD0 est défini par le registre DDRD

<b>DDRD</b>	-	-	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	\$1009
Reset			0	0	0	0	0	0	

**Port E** : les broches PE7 à PE0 sont utilisées conjointement par le CAN. Elles peuvent être utilisées simultanément comme des entrées logiques.

Sens	E	E	E	E	E	E	E		
<b>PORTE</b>	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	\$100A
Reset	E	E	E	E	E	E	E	E	

### Remarques:

- Un bit **DDRXi** à 1 permet de configurer la broche correspondante **PXi** en sortie.
- A l'initialisation le registre de données **PORTX** peut contenir n'importe quelle valeur, donc il faut l'initialiser avant le registre de direction **DDRX** pour éviter d'avoir sur le port n'importe quelle valeur.
- Au **RESET** les broches bidirectionnelles sont toutes fixées en entrée (**DDRX=\$00**).

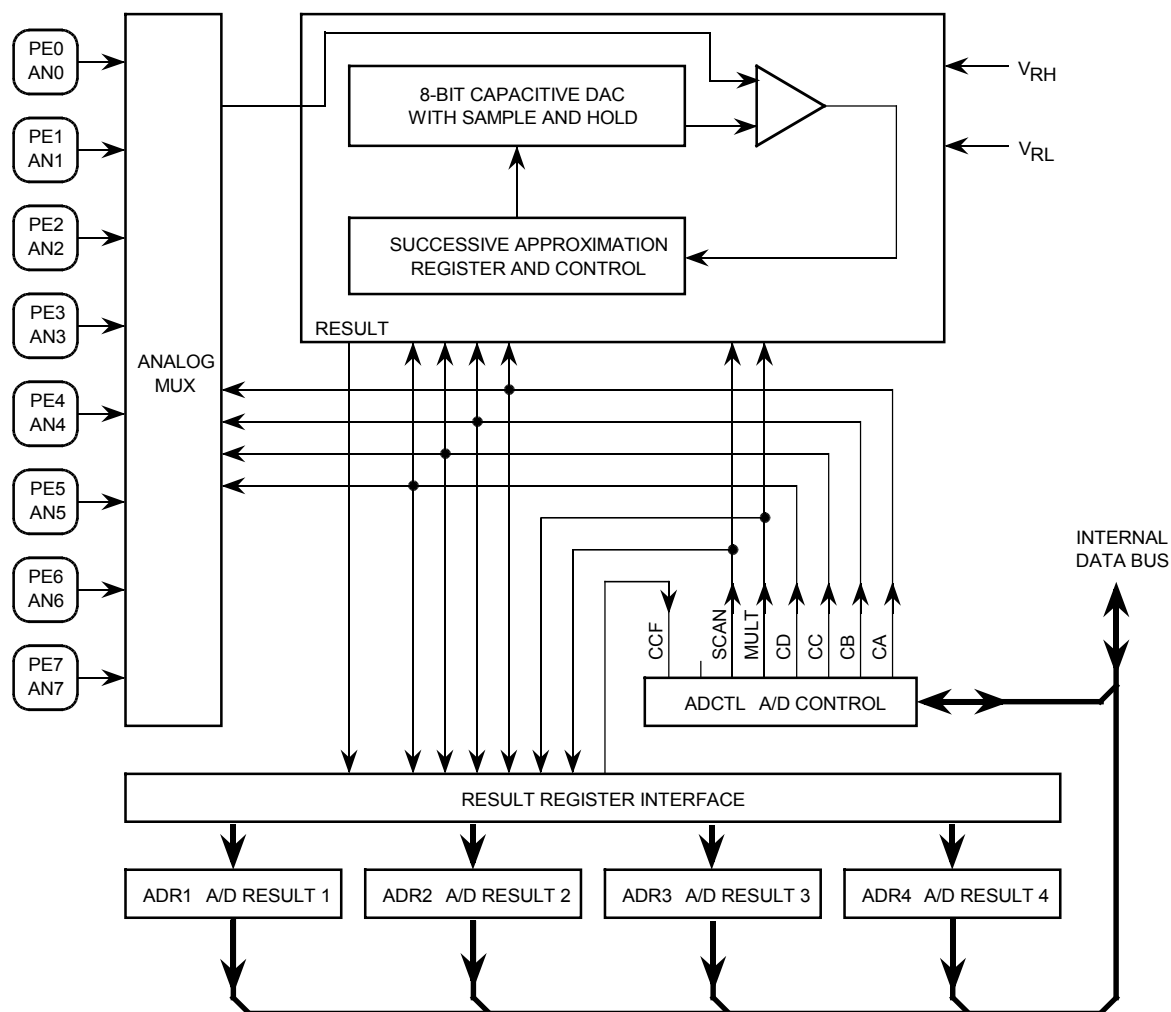
## 10) LE CONVERTISSEUR ANALOGIQUE NUMERIQUE.

Le convertisseur analogique numérique est à approximation successive et il possède une résolution de 8 bits, il est précédé d'un multiplexeur analogique 8 voies.

La conversion peut être synchronisée :

- Par l'horloge interne E si sa fréquence est supérieure à 750 kHz.
- Par une horloge propre au convertisseur si la fréquence de E est inférieure à 750 kHz.

### 10.1) Organisation interne.



# LE MICRO CONTROLEUR MC68HC811E2

## 10.2) Fonctionnement du convertisseur.

Une conversion commence toujours par une écriture dans le registre de contrôle **ADCTL**, elle peut se répéter si le bit **SCAN** est à un, sinon elle ne sera effectuée qu'une seule fois.

L'entrée ou les entrées converties dépendent de la valeur du bit **MULT** et des bits **CD** à **CA** du registre **ADCTL**.

- Si **MULT=0** : l'entrée définie par les bits **CD-CA** est convertie 4 fois et les 4 résultats consécutifs se trouvent dans **ADR1**, **ADR2**, **ADR3** et **ADR4**.

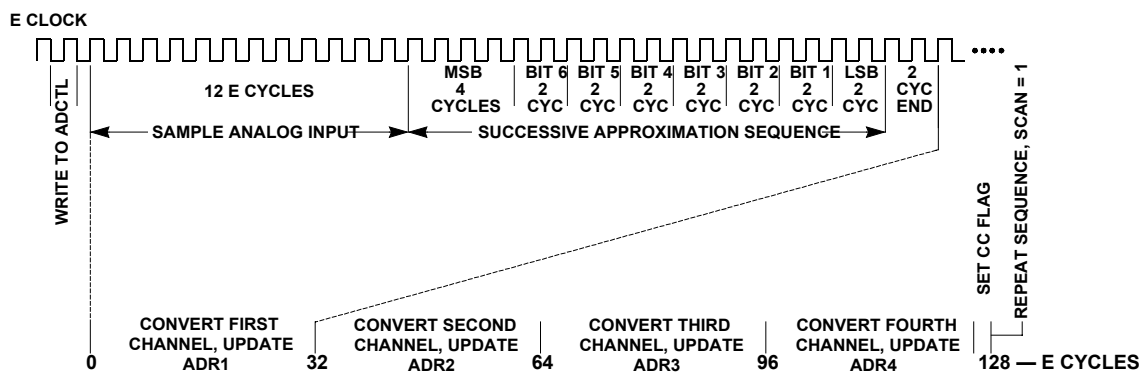
Channel Select Control Bits				Channel Signal	Result
CD	CC	CB	CA		
0	0	0	0	PE0	ADR1,ADR2,ADR3, ADR4
0	0	0	1	PE1	ADR1,ADR2,ADR3, ADR4
0	0	1	0	PE2	ADR1,ADR2,ADR3, ADR4
0	0	1	1	PE3	ADR1,ADR2,ADR3, ADR4
0	1	0	0	PE4	ADR1,ADR2,ADR3, ADR4
0	1	0	1	PE5	ADR1,ADR2,ADR3, ADR4
0	1	1	0	PE6	ADR1,ADR2,ADR3, ADR4
0	1	1	1	PE7	ADR1,ADR2,ADR3, ADR4
1	0	0	0	Reserved	-
1	0	0	1	Reserved	-
1	0	1	0	Reserved	-
1	0	1	1	Reserved	-
1	1	0	0	VRH	ADR1,ADR2,ADR3, ADR4
1	1	0	1	VRL	ADR1,ADR2,ADR3, ADR4
1	1	1	0	(VRH)/2	ADR1,ADR2,ADR3, ADR4
1	1	1	1	Reserved	-

- Si **MULT=1**: Le groupe d'entrées défini par **CD** et **CC** est converti et les résultats se trouvent dans **ADR1**, **ADR2**, **ADR3** et **ADR4**.

Channel Select Control Bits				Channel Signal	Result
CD	CC	CB	CA		
0	0	X	X	PE0	ADR1
				PE1	ADR2
				PE2	ADR3
				PE3	ADR4
0	1	X	X	PE4	ADR1
				PE5	ADR2
				PE6	ADR3
				PE7	ADR4
1	0	X	X	Reserved	-
				Reserved	-
				Reserved	-
				Reserved	-
1	1	X	X	VRH	ADR1
				VRL	ADR2
				(VRH)/2	ADR3
				Reserved	-

## 10.3) Cycle de fonctionnement.

La lecture des registres **ADR1**, **ADR2**, **ADR3** et **ADR4** doit être précédée par une lecture du registre **ADCTL** pour savoir si la conversion est terminée (**CCF=1**).



## 10.4) Registres.

### 10.4.1) Le registre OPTION.

Il permet d'activer le convertisseur et de choisir le type d'horloge. Toute utilisation doit commencer par initialiser ce registre.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>OPTION</b>	ADPU	CSEL	-	-	-	-	-	-	<b>\$1039</b>
Reset	0	0	0	1	0	0	0	0	

• **ADPU: A/D POWER UP.**

1: Mise en fonctionnement du convertisseur (attendre 100µS pour que l'alimentation soit stabilisée).  
0: Arrêt du convertisseur.

• **CSEL: CLOCK SELECT.**

1 : Le convertisseur utilise l'horloge interne (F < 750 kHz).  
0 : Le convertisseur utilise l'horloge E (F > 750 kHz).

### 10.4.2) Le registre ADCTL.

C'est le registre de contrôle et d'état du convertisseur Analogique / Numérique.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>ADCTL</b>	CCF	0	SCAN	MULT	CD	CC	CB	CA	<b>\$1030</b>
Reset	1	0	1	1	1	1	1	1	

• **CCF: Conversion Complete Flag.**

Ce bit est mis à un à la fin de chaque phase de conversion (4 Conversions), il est remis à zéro automatiquement au début d'une nouvelle conversion.

• **SCAN: Continuous Scan Control.**

0 : Le convertisseur effectue une seule conversion.  
1 : Le convertisseur effectue des conversions en permanence.

• **MULT: Multiple channel / Single Channel Control.**

0 : Le convertisseur effectue la conversion de l'entrée sélectionnée par les bits **CD** à **CA**.  
1 : Le convertisseur effectue la conversion du groupe d'entrées sélectionné par les bits **CD** et **CC**.

• **CD - CA: Channel Selects.**

### 10.4.3) Les registres ADR1, ADR2, ADR3 et ADR4.

Ces registres contiennent les résultats de la conversion. Ils sont à lecture seule.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>ADR1</b>	R7	R6	R5	R4	R3	R2	R1	R0	<b>\$1031</b>

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>ADR2</b>	R7	R6	R5	R4	R3	R2	R1	R0	<b>\$1032</b>

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>ADR3</b>	R7	R6	R5	R4	R3	R2	R1	R0	<b>\$1033</b>

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>ADR4</b>	R7	R6	R5	R4	R3	R2	R1	R0	<b>\$1034</b>



## **11) LA LIAISON SERIE SPI (SERIAL PERIPHERAL INTERFACE).**

C'est une liaison de type série synchrone fonctionnant en full duplex. Elle permet de communiquer avec d'autres microcontrôleurs ou circuits SPI. Elle est utilisée en général pour simplifier le routage du circuit imprimé d'un Objet Technique. Elle est utilisée pour de faibles distances.

### **11.1) Caractéristiques.**

- 4 Fils de communication.
- Le  $\mu$ C peut être Maître ou Esclave.
- Vitesse maximale en Maître 1,05 MHz.
- Vitesse maximale en Esclave 2,1 MHz.
- Polarité et Phase de l'horloge programmables.

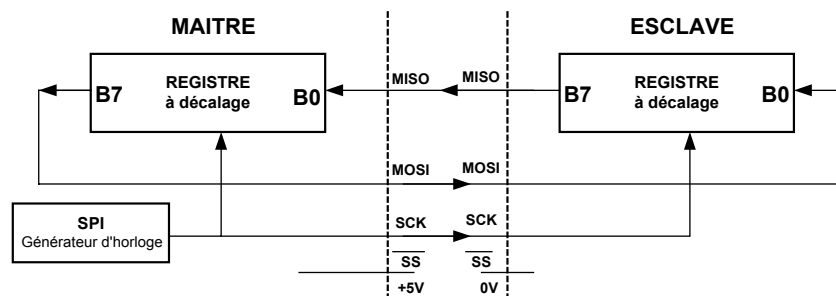
### **11.2) Le bus.**

Il utilise quatre broches du port D.

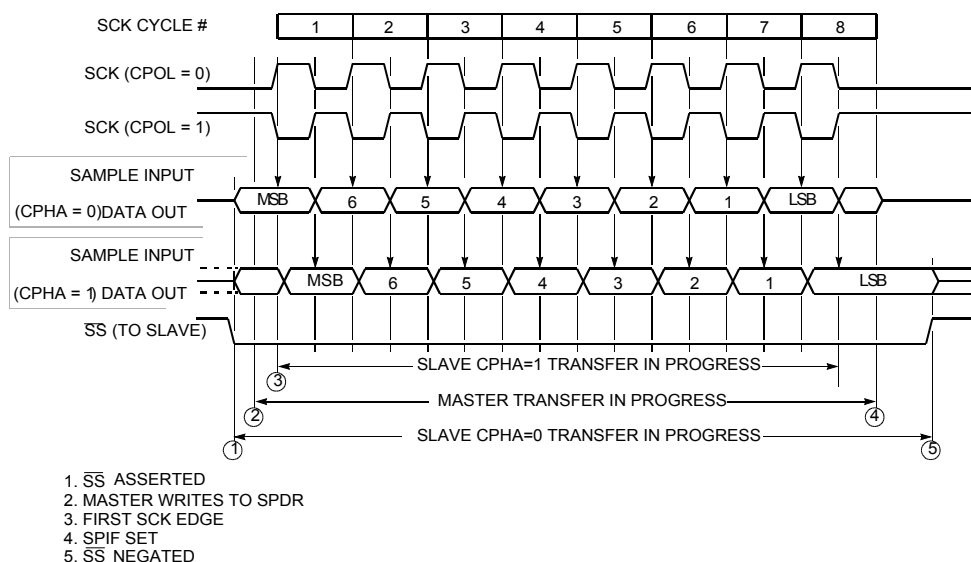
- PD3, MOSI: (Master Out Slave In).  
Sortie si le  $\mu$ C est maître.  
Entrée si le  $\mu$ C est esclave.
- PD2, MISO: (Master In Slave Out).  
Entrée si le  $\mu$ C est maître.  
Sortie si le  $\mu$ C est esclave.
- PD4, SCK: (Serial Clock).  
Entrée si le  $\mu$ C est esclave.  
Sortie si le  $\mu$ C est maître.
- PD5, SS\ (Slave Select).  
0 : Le  $\mu$ C est esclave.  
1 : Le  $\mu$ C est maître.

# LE MICRO CONTROLEUR MC68HC811E2

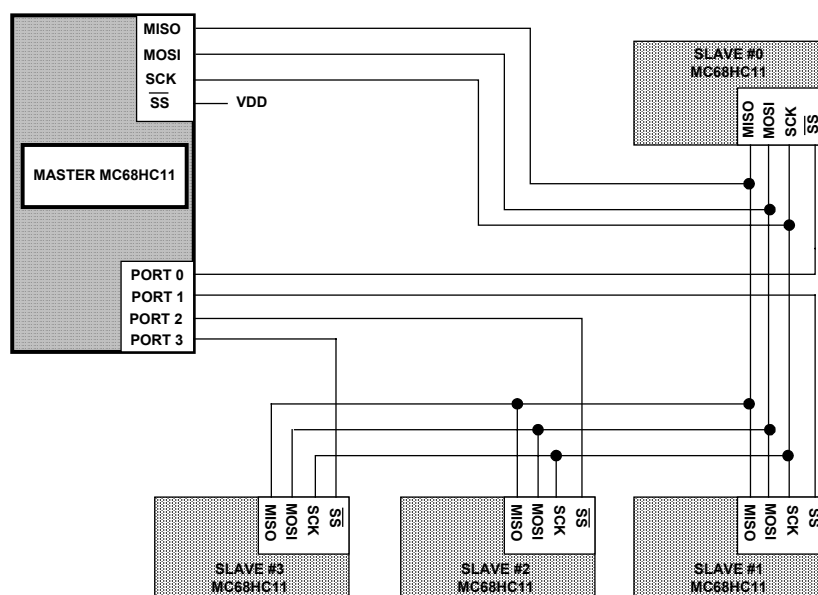
## 11.3) Principe de fonctionnement.



Le contenu du registre de données du maître est transféré dans le registre de données de l'esclave et le registre de données de l'esclave est transféré dans le registre de données du maître, bit après bit. Le transfert est synchronisé par l'horloge du maître.



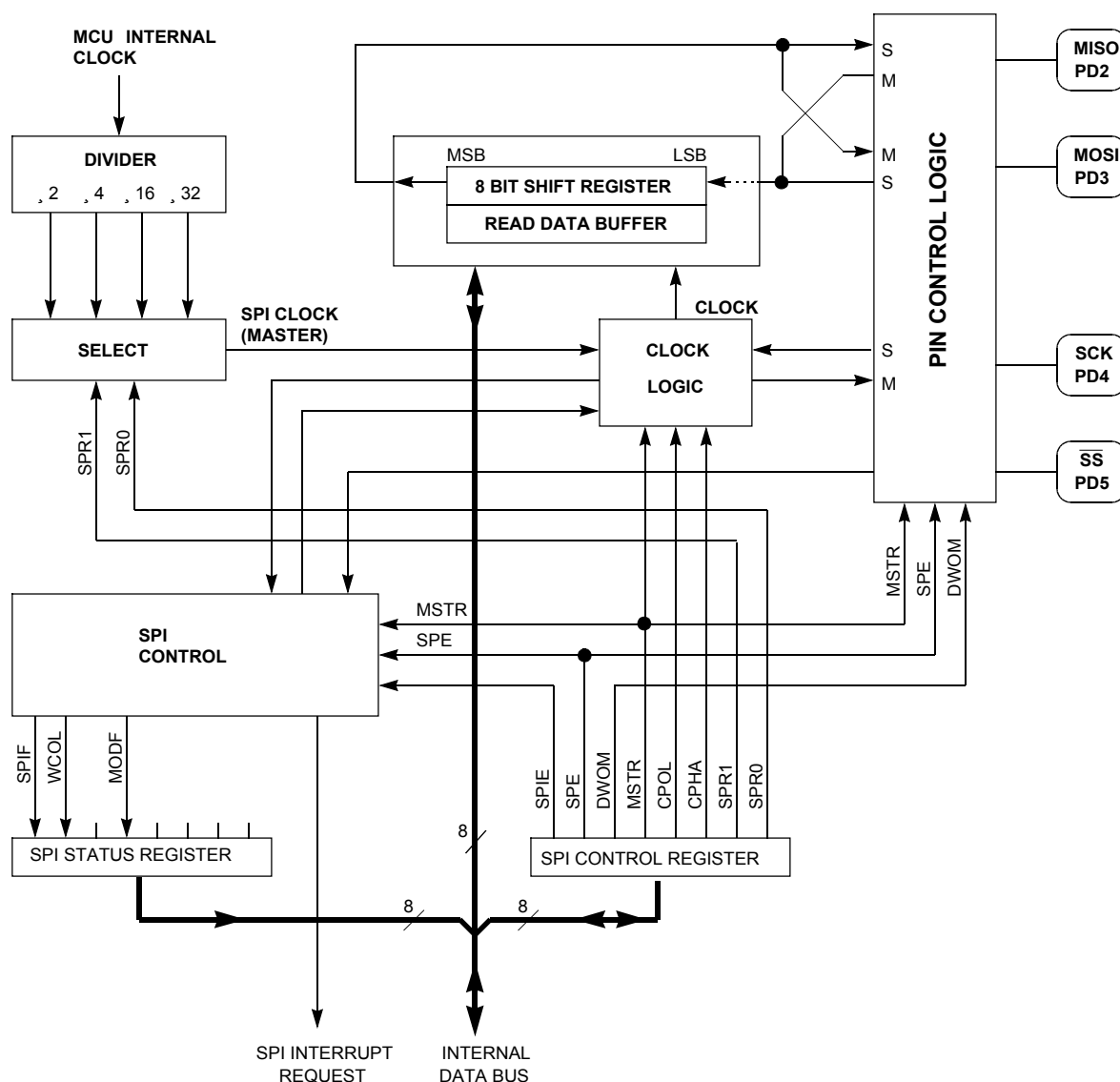
## 11.4) Organisation utilisant un maître et plusieurs esclaves.



## 11.5) Organisation Interne.

La liaison SPI est gérée par trois registres de :  
**Données** : **SPDR** : Serial Peripheral Data Register.  
**Contrôle** : **SPCR** : Serial Peripheral Control Register.  
**Etat** : **SPSR** : Serial Peripheral State Register..

L'utilisation de la liaison SPI nécessite le recours aux interruptions. Voir le chapitre correspondant.



## 11.5.1) Le registre de contrôle SPCR (Serial Peripheral Control Register).

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>SPCR</b>	<b>SPIE</b>	<b>SPE</b>	<b>0</b>	<b>MSTR</b>	<b>CPOL</b>	<b>CPHA</b>	<b>SR1</b>	<b>SR0</b>	<b>\$1028</b>
Reset	0	0	0	0	0	1	0	0	

• **SPIE**: Serial Peripheral Interrupt Enable.

1: Interruption autorisée, si le bit **SPIF** du **SPSR** est à 1 alors une interruption **SPI** sera demandée au  $\mu$ C.

0: Interruption inhibée.

• **SPE**: Serial Peripheral Enable.

1: Liaison **SPI** en fonctionnement.

0: Liaison **SPI** inhibée.

• **MSTR**: Master Mode Select.

1 : La liaison **SPI** est configurée en maître.

0 : La liaison **SPI** est configurée en esclave.

• **CPOL**: Clock Polarity.

1 : Le signal **SCK** est actif à l'état haut.

0 : Le signal **SCK** est actif à l'état bas.

• **CPHA**: Clock Phase.

1 : Les données sont échantillonnées sur le front montant de **SCK** si **CPOL**=1 ou sur le front descendant si **CPOL**=0.

0 : Les données sont échantillonnées sur le front descendant de **SCK** si **CPOL**=1 ou sur le front montant si **CPOL**=0.

• **SPR1** & **SPR0**: Serial Peripheral Rate Bits.

Ils déterminent la vitesse des échanges de données.

<b>SPR1</b>	<b>SPR0</b>	<b>HORLOGE du <math>\mu</math>C Divisée par</b>
<b>0</b>	<b>0</b>	<b>2</b>
<b>0</b>	<b>1</b>	<b>4</b>
<b>1</b>	<b>0</b>	<b>16</b>
<b>1</b>	<b>1</b>	<b>32</b>

## 11.5.2) Le registre d'état SPSR (Serial Peripheral Status Register).

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>SPSR</b>	SPIF	WCOL	-	MODF	-	-	-	-	<b>\$1029</b>
Reset	0	0	0	0	0	0	0	0	

- **SPIF**: Serial Peripheral Interface Flag.

1 : Il indique que la transmission ou la réception de données est terminée, elle est disponible dans le registre **SPDR**. Si le bit **SPIE** du registre **SPCR** est à 1 alors une interruption **SPI** est demandée au  $\mu$ C.

0 : Il est remis à 0 par une lecture de **SPSR** suivie d'une lecture ou d'une écriture dans le **SPDR**.

- **WCOL**: Write Collision.

1 : Indique que l'on a écrit dans le registre **SPDR** pendant qu'une transmission était en cours.

0 : Il est remis à 0 par une lecture de **SPSR** suivie d'une lecture ou d'une écriture dans le **SPDR**.

- **MODF**: Mode Fault Flag.

1 : Indique que plusieurs maîtres sont détectés simultanément sur la liaison **SPI**.

0 : Il est remis à 0 par une lecture de **SPSR** suivie d'une lecture ou d'une écriture dans le **SPCR**.

## 11.5.3) Le registre de données SPDR (Serial Peripheral Data Register).

Registre à lecture ou écriture.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>SPDR</b>	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	<b>\$102A</b>
Reset	X	X	X	X	X	X	X	X	

### Si le $\mu$ C est maître:

- l'écriture dans ce registre provoque la transmission de données sur le bus.
- la donnée lue après un transfert est celle envoyée par l'esclave.

### Si le $\mu$ C est esclave:

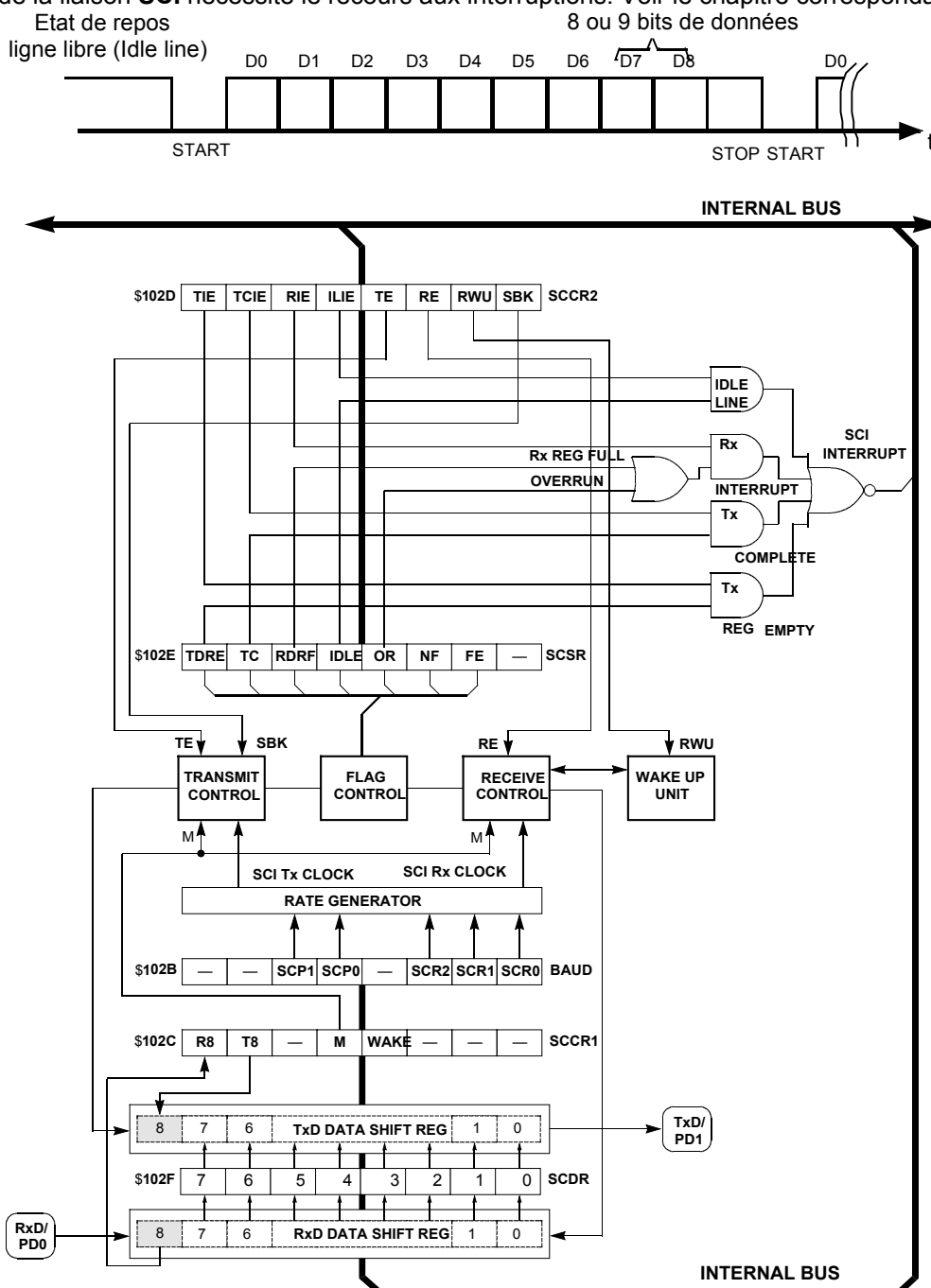
- L'écriture dans ce registre pré positionne la donnée à envoyer au maître lors du prochain transfert.
- La donnée lue après un transfert est celle envoyée par le maître.

## 12) LA LIAISON SERIE SCI (SERIAL COMMUNICATION INTERFACE).

La liaison série **SCI** est une interface série asynchrone de type **START / STOP**. Elle permet d'effectuer des communications avec d'autres systèmes ou objets techniques sur de longues distances. Elle dispose des fonctionnalités suivantes:

- Fonctionnement en Full Duplex, c'est à dire émission et réception de données en même temps.
- Transmission et réception de données (compatibles avec la norme **RS232** en utilisant une fonction d'adaptation de niveaux).
- 32 Vitesses de transmission et de réception disponibles.
- Contrôle des erreurs de transmission et de réception.
- Mode réveil automatique lors de la réception de signaux valides.
- 4 Sources d'interruptions possibles.

L'utilisation de la liaison **SCI** nécessite le recours aux interruptions. Voir le chapitre correspondant.



# LE MICRO CONTROLEUR MC68HC811E2

## 12.1) Le registre SCDR .

Ce registre sert à l'émission et la réception. Il est constitué physiquement de deux registres, l'un pour la réception et l'autre pour l'émission, la sélection entre les deux se fait par le signal **R/W**.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>SCDR</b>	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	<b>\$102F</b>
Reset	X	X	X	X	X	X	X	X	

## 12.2) Le registre BAUD .

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>BAUD</b>	0	0	SCP1	SCP0	0	SCR2	SCR1	SCR0	<b>\$102B</b>
Reset	0	0	0	0	0	X	X	X	

Ce registre détermine la vitesse de transmission et de réception de l'interface.  
Deux bits servent à pré diviser l'horloge du **µC** à savoir **SCP0** et **SCP1**.  
Trois bits déterminent la vitesse **SCR0**, **SCR1** et **SCR2**.

Baud Rates by Crystal Frequency, SCP[1:0] and SCR[2:0]

SCP1	SCP0	SCR2	SCR1	SCR0	Crystal Frequency				
					2 <sup>23</sup> Hz	8 MHz	4.9152 MHz	4 MHz	3.6864 MHz
					Baud Rates				
0	0	0	0	0	131.072K Baud	125.00K Baud	76.80K Baud	62.50K Baud	57.60K Baud
0	0	0	0	1	65.536K Baud	62.50K Baud	38.40K Baud	31.25K Baud	28.80K Baud
0	0	0	1	0	32.768K Baud	31.25K Baud	19.20K Baud	15.625K Baud	14.40K Baud
0	0	0	1	1	16.384K Baud	15.625K Baud	9600 Baud	7812.5 Baud	7200 Baud
0	0	1	0	0	8192 Baud	7812.5 Baud	4800 Baud	3906 Baud	3600 Baud
0	0	1	0	1	4096 Baud	3906 Baud	2400 Baud	1953 Baud	1800 Baud
0	0	1	1	0	2048 Baud	1953 Baud	1200 Baud	977 Baud	900 Baud
0	0	1	1	1	1024 Baud	977 Baud	600 Baud	488 Baud	450 Baud
0	1	0	0	0	43.691K Baud	41.666K Baud	25.60K Baud	20.833K Baud	19.20K Baud
0	1	0	0	1	21.845K Baud	20.833K Baud	12.80K Baud	10.417K Baud	9600 Baud
0	1	0	1	0	10.923K Baud	10.417K Baud	6400 Baud	5208 Baud	4800 Baud
0	1	0	1	1	5461 Baud	5208 Baud	3200 Baud	2604 Baud	2400 Baud
0	1	1	0	0	2731 Baud	2604 Baud	1600 Baud	1302 Baud	1200 Baud
0	1	1	0	1	1365 Baud	1302 Baud	800 Baud	651 Baud	600 Baud
0	1	1	1	0	683 Baud	651 Baud	400 Baud	326 Baud	300 Baud
0	1	1	1	1	341 Baud	326 Baud	200 Baud	163 Baud	150 Baud
1	0	0	0	0	32.768K Baud	31.250K Baud	19.20K Baud	15.625K Baud	14.40K Baud
1	0	0	0	1	16.384K Baud	15.625K Baud	9600 Baud	7812.5 Baud	7200 Baud
1	0	0	1	0	8192 Baud	7812.5 Baud	4800 Baud	3906 Baud	3600 Baud
1	0	0	1	1	4096 Baud	3906 Baud	2400 Baud	1953 Baud	1800 Baud
1	0	1	0	0	2048 Baud	1953 Baud	1200 Baud	977 Baud	900 Baud
1	0	1	0	1	1024 Baud	977 Baud	600 Baud	488 Baud	450 Baud
1	0	1	1	0	512 Baud	488 Baud	300 Baud	244 Baud	225 Baud
1	0	1	1	1	256 Baud	244 Baud	150 Baud	122 Baud	112.5 Baud
1	1	0	0	0	10.082K Baud	9600 (+0.16%)	5908 Baud	4800 (+0.16%)	4431 Baud
1	1	0	0	1	5041 Baud	4800 Baud	2954 Baud	2400 Baud	2215 Baud
1	1	0	1	0	2521 Baud	2400 Baud	1477 Baud	1200 Baud	1108 Baud
1	1	0	1	1	1260 Baud	1200 Baud	738 Baud	600 Baud	554 Baud
1	1	1	0	0	630 Baud	600 Baud	369 Baud	300 Baud	277 Baud
1	1	1	0	1	315 Baud	300 Baud	185 Baud	150 Baud	138 Baud
1	1	1	1	0	158 Baud	150 Baud	92 Baud	75 Baud	69 Baud
1	1	1	1	1	79 Baud	75 Baud	46 Baud	38 Baud	35 Baud
					2.1 MHz	2 MHz	1.2288 MHz	1 MHz	921.6 kHz
					Bus Frequency (E clock)				

## 12.3) Le registre de contrôle SCCR1 (Serial Communication Control Register 1).

Ce registre permet de configurer la liaison **SCI**, il est utilisé pour contrôler la longueur des données à traiter et le mode réveil.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>SCCR1</b>	R8	T8	0	M	WAKE	0	0	0	<b>\$102C</b>
Reset	X	X	0	0	0	0	0	0	

- **R8**: Ce bit contient le neuvième bit de données reçu si **M**=1.
- **T8**: Ce bit contient le neuvième bit de données à transmettre si **M**=1.
- **M**: détermine le nombre de bits à transmettre.  
 1 : Transmission avec un bit de start, 9 bits de données et un bit de stop.  
 0 : Transmission avec un bit de start, 8 bits de données et un bit de stop.
- **WAKE**: Détermine le type d'action qui va réveiller le récepteur.  
 1 : Le récepteur sera réveillé si le **MSB** de la donnée reçue est à 1.  
 0 : Le récepteur sera réveillé si la ligne **RDO** est au repos (**IDLE LINE**).

## 12.4) Le registre de contrôle SCCR2 (Serial Communication Control Register 2).

Il est le deuxième registre de configuration de la liaison **SCI**.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>SCCR2</b>	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	<b>\$102D</b>
Reset	0	0	0	0	0	0	0	0	

- **TIE**: Transmit Interrupt Enable.  
 1 : Interruption générée si le bit **TDRE** du registre **SCSR** est à un.  
 0 : Interruption inhibée.
- **TCIE**: Transmit Complete Interrupt Enable.  
 1 : Interruption générée si le bit **TC** du registre **SCSR** est à un.  
 0 : Interruption inhibée.
- **RIE**: Receive Interrupt Enable.  
 1 : Interruption générée si le bit **RDRF** du registre **SCSR** est à un.  
 0 : Interruption inhibée.
- **ILIE**: Idle Line Interrupt Enable.  
 1 : Interruption générée si le bit **ILIE** du registre **SCSR** est à un.  
 0 : Interruption inhibée.
- **TE**: Transmitter Enable.  
 1 : Validation de l'émetteur **SCI**.  
 0 : L'émetteur de la **SCI** est inhibé.
- **RE**: Receive Enable.  
 1 : Validation du récepteur **SCI**.  
 0 : Le récepteur de la **SCI** est inhibé.



## LE MICRO CONTROLEUR MC68HC811E2

- **RWU**: Receive Wake Up.

1 : Met le récepteur de la **SCI** en mode sommeil. Il se réveillera conformément au choix fait par le bit **WAKE** du **SCCR1**.

0 : Fonction réveil désactivé.

- **SBK**: Send Break.

1 : L'émetteur envoie en permanence des blocs de 10 ou 11 zéros.

0 : L'émetteur envoie des blocs de 10 (M=0) ou de 11 zéros (M=1), et passe ensuite en mode transmission normale.

## 12.5 ) Le registre SCSR (Serial Communication Status Register).

C'est le registre d'état de l'interface **SCI**.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>SCSR</b>	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	\$102E
Reset	1	1	0	0	0	0	0	0	

• **TDRE** : Transmit Data Register Empty.

1 : Le registre de transmission est vide, une donnée à transmettre peut être écrite dans le registre **SCDR**, il est remis à zéro par une lecture du **SCSR** suivie d'une écriture dans le **SCDR**.

0 : Le registre de transmission n'est pas libre.

• **TC** : Transmit Complete

1 : Indique que la transmission d'une donnée est complètement terminée et que par conséquent le registre de transmission est vide. Il est remis à zéro par une lecture du **SCSR** suivie d'une écriture dans le **SCDR**.

0 : La transmission n'est pas complètement terminée.

• **RDRF** : Receive Data Register Full.

1 : Le registre de réception est plein, la donnée reçue peut être lue dans le registre **SCDR**, il est remis à zéro par une lecture du **SCSR** suivie d'une lecture du registre **SCDR**.

0 : Le registre de réception est vide.

• **IDLE** : Idle Line Detect

1 : Ce bit est mis à 1 quand la ligne **RDI** est au repos c'est à dire à 1. Il est remis à zéro par une lecture du **SCSR** suivie d'une lecture du registre **SCDR**. Il ne peut ensuite passer à nouveau à 1 que si au moins une donnée est reçue avant que la ligne ne devienne à nouveau inactive.

• **OR** : Overrun Error

1 : Indique une erreur de débordement du récepteur. Ceci se présente lorsqu'une donnée arrive alors que la précédente n'a pas encore été lue. Il est remis à zéro par une lecture du **SCSR** suivie d'une lecture du registre **SCDR**.

• **NF** : Noise Flag

1 : Indique la présence de bruit sur la ligne **RDI**. Il est remis à zéro par une lecture du **SCSR** suivie d'une lecture du registre **SCDR**.

• **FE** : Framing error.

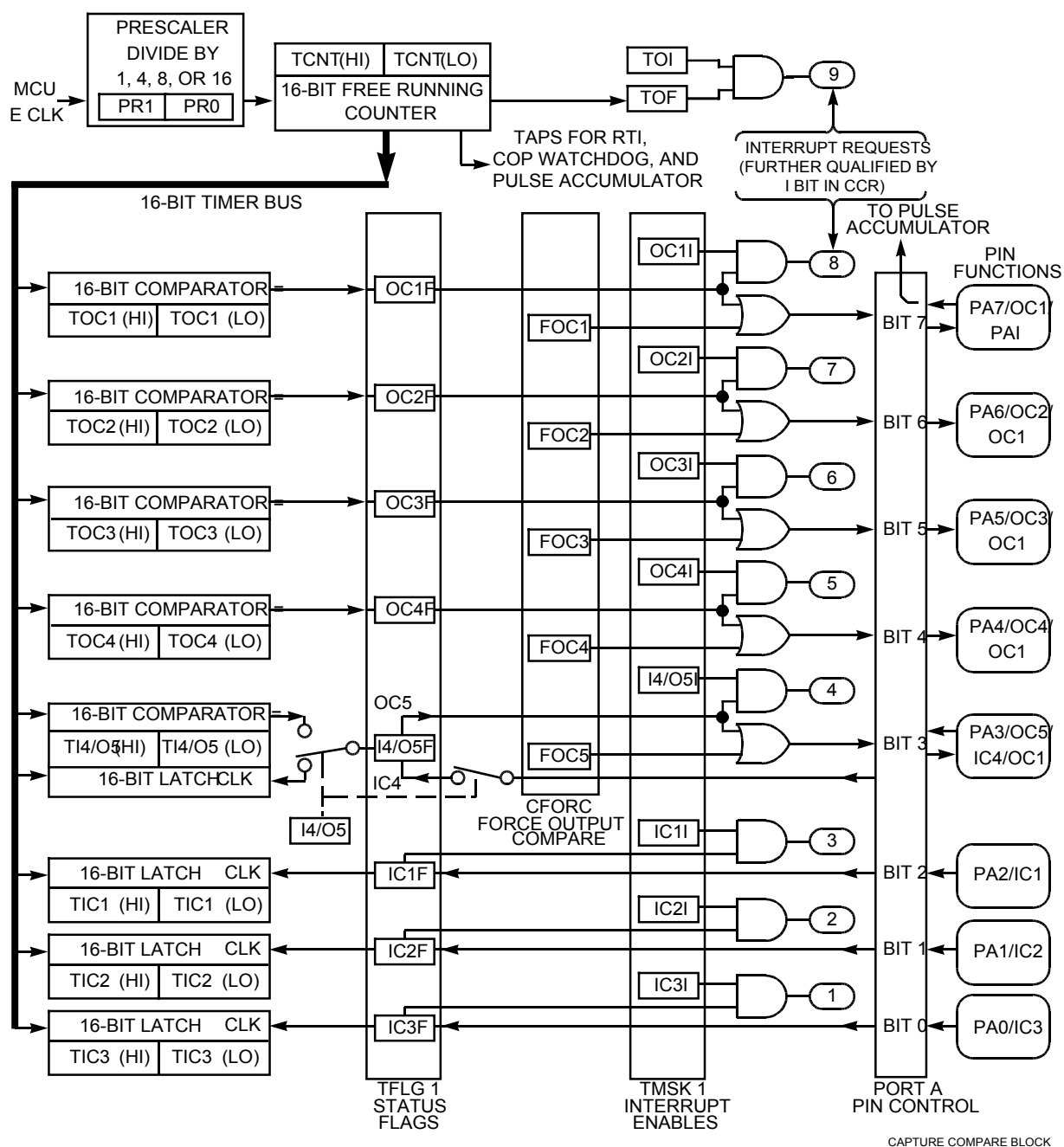
1 : Indique une erreur de format caractérisée par la non détection de bit de stop. Il est remis à zéro par une lecture du **SCSR** suivie d'une lecture du registre **SCDR**. La réception est bloquée tant que ce bit n'est pas remis à zéro.

## 13) LE TIMER (TEMPORISATEUR).

Il est implanté autour d'un compteur 16bits **TCNT (FREE RUNNING COUNTER)** incrémenté automatiquement par un pré diviseur (la pré division peut être de 1, 2, 4 ou 16 suivant les valeurs des bits **PR1** et **PR0** du registre **TMSK2**). Ce compteur, associé aux registres de comparaison de sortie (**Output Compare Register : OCR**), aux registres de comparaison d'entrée 16 bits (**Input Capture Register : ICR**) eux-mêmes en liaison avec les broches **ICX** et **OCX**, plusieurs registres de contrôle (**TCTL2**, **CFORC**, **OC1M**, **OC1D**, **TCTL1**, **TMSK1** et **TMSK2**) et aux deux registres d'états (**TFLG1** et **TFLG2**), permet de réaliser un grand nombre de fonctions:

- Générateur de créneaux de **PWM (Pulse Width Modulation)**.
- Compteur d'événements.
- Mesure de temps.
- Générateur d'interruptions cycliques.

L'utilisation du **TIMER** nécessite le recours aux interruptions. Voir le chapitre correspondant.



## 13.1) Le registre Free Running Counter TCNT.

Après un **RESET**, le compteur **TCNT** est positionné à **\$0000**, puis il est incrémenté. Au passage de **\$FFFF** à **\$0000**, le bit **TOF (Timer Overflow Flag)** du registre d'état **TFLG2** est positionné à un.

Quand on effectue une lecture du registre **TCNT** on doit commencer par le registre haut **TCRH** puis de **TCRL** car seulement la lecture du registre de poids fort provoque la mémorisation du registre de poids faible pour une lecture ultérieure, on utilise souvent le registre **D** car il permet de lire ou d'écrire 16 bits en une seule instruction.

<b>TCNT</b>	<i>TCNT<sub>H</sub></i>	15	14	13	12	11	10	9	8	<b>\$100E</b>
	<i>TCNT<sub>L</sub></i>	7	6	5	4	3	2	1	0	

## 13.2) Les registres Timer Output Compare TOCx (x = 1, 2, 3, 4 ou 5).

Ces registres 16 bits sont chargés avec la valeur de votre choix et lorsque le **TCNT** atteint cette valeur, le bit **OCXF (Output Compare Flag)** du registre d'état **TFLG1** est mis à un et la broche **OCX** est positionnée à l'état haut ou bas suivant l'état des bits **OMX** et **OLX** du registre de contrôle **TCTL1**. Le bit **OCXF** peut générer une interruption si le bit **OCXI (Output Compare Interrupt)** du registre de contrôle **TMSK1** est à un. Le bit **OCXF** est remis à zéro par l'écriture d'un 1 logique sur le bit **OCXF** du registre **TFLG1**.

Pour écrire dans ce registre, on doit commencer par le registre de poids fort **TOCXH** puis le registre de poids faible **TOCXL**, on utilise souvent le registre **D** car il permet de lire ou d'écrire 16 bits en une seule instruction.

<b>TOC1</b>	<i>TOC1<sub>H</sub></i>	15	14	13	12	11	10	9	8	<b>\$1016</b>
	<i>TOC1<sub>L</sub></i>	7	6	5	4	3	2	1	0	
<b>TOC2</b>	<i>TOC2<sub>H</sub></i>	15	14	13	12	11	10	9	8	<b>\$1018</b>
	<i>TOC2<sub>L</sub></i>	7	6	5	4	3	2	1	0	
<b>TOC3</b>	<i>TOC3<sub>H</sub></i>	15	14	13	12	11	10	9	8	<b>\$101A</b>
	<i>TOC3<sub>L</sub></i>	7	6	5	4	3	2	1	0	
<b>TOC4</b>	<i>TOC4<sub>H</sub></i>	15	14	13	12	11	10	9	8	<b>\$101C</b>
	<i>TOC4<sub>L</sub></i>	7	6	5	4	3	2	1	0	

## 13.3) Les registres Timer Input Compare TICx (x = 1, 2, 3 ou 4).

Ces registres 16 bits sont chargés avec la valeur de votre choix et lorsqu'une transition valide (front montant ou descendant suivant l'état des bits **EDGXB** et **EDGXA** du registre de contrôle **TCTL2**) intervient sur la broche **ICX** alors la valeur du registre **TCNT** est transférée dans le registre **TICx** et le bit **ICXF (Input Capture Flag)** du registre d'état **TFLG1** est mis à un. Le bit **ICXF** peut générer une interruption si le bit **ICXI (Input Capture Interrupt Compare)** du registre de contrôle **TMSK1** est mis à un. Le bit **ICXF** est remis à zéro par l'écriture d'un 1 logique sur le bit **ICXF** du registre **TFLG1**.

Pour écrire dans ce registre, on doit commencer par le registre de poids fort **TICXH** puis le registre de poids faible **TICXL**, on utilise souvent le registre **D** car il permet de lire ou d'écrire 16 bits en une seule instruction.

<b>TIC1</b>	<i>TIC1<sub>H</sub></i>	15	14	13	12	11	10	9	8	<b>\$1010</b>
	<i>TIC1<sub>L</sub></i>	7	6	5	4	3	2	1	0	
<b>TIC2</b>	<i>TIC2<sub>H</sub></i>	15	14	13	12	11	10	9	8	<b>\$1012</b>
	<i>TIC2<sub>L</sub></i>	7	6	5	4	3	2	1	0	
<b>TIC3</b>	<i>TIC3<sub>H</sub></i>	15	14	13	12	11	10	9	8	<b>\$1014</b>
	<i>TIC3<sub>L</sub></i>	7	6	5	4	3	2	1	0	
<b>TIC4/TOC5</b>	<i>TIC4/TOC5<sub>H</sub></i>	15	14	13	12	11	10	9	8	<b>\$101E</b>
	<i>TIC4/TOC5<sub>L</sub></i>	7	6	5	4	3	2	1	0	

# LE MICRO CONTROLEUR MC68HC811E2

## 13.4) Le registre Timer Control 2 : TCTL2.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>TCTL2</b>	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	<b>\$1021</b>
Reset	0	0	0	0	0	0	0	0	

La valeur des bits **EDGXB** et **EDGXA** détermine la configuration des entrées de capture.

EDGXB	EDGXA	Configuration
0	0	Capture de front inhibée.
0	1	Capture sur front montant seulement
1	0	Capture sur front descendant seulement.
1	1	Capture sur front montant et descendant.

## 13.5) Le registre Control Force Output Compare : CFORC.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>CFORC</b>	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	<b>\$100B</b>
Reset	0	0	0	0	0	0	0	0	

Ce registre permet de forcer la comparaison **OCX** sans attendre l'égalité de **TCNT** et de **TOCX**, la valeur de la broche **OCX** est fonction des bits **OMX** et **OLX** du registre **TCTL1**.

## 13.6) Le registre Output Compare 1 Mask : OC1M.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>OC1M</b>	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	<b>\$100C</b>
Reset	0	0	0	0	0	0	0	0	

Ce registre permet de déterminer les bits affectés par la sortie **OC1** lors de l'égalité entre le registre **TOC1** et le registre **TCNT**, en conjonction avec le registre **OC1D**.

## 13.7) Le registre Output Compare 1 Data : OC1D.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>OC1D</b>	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	<b>\$100D</b>
Reset	0	0	0	0	0	0	0	0	

Si le bit **OC1MX** du registre **OC1M** est à un, la sortie **OCX** prendra la valeur du bit **OC1DX**.

## 13.8) Le registre Timer Control 1 : TCTL1.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>TCTL1</b>	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	<b>\$1020</b>
Reset	0	0	0	0	0	0	0	0	

La valeur des bits **OMX** et **OLX** détermine l'état logique de la sortie **OCX**, lorsqu'il y a égalité entre **TOCX** et **TCNT**.

OMX	OLX	Etat futur de OCX
0	0	La sortie OCX n'est pas affectée par l'égalité
0	1	Changement d'état: 0 donne 1 et 1 donne 0
1	0	Met à zéro OCX
1	1	Met à un OCX

## 13.9) Le registre Timer Mask 1 : TMSK1.

Il permet d'autoriser ou d'inhiber les interruptions des **OCX** ou **ICX**.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>TMSK1</b>	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	<b>\$1022</b>
Reset	0	0	0	0	0	0	0	0	

- **OC1I-OC4I**: Autorisation de l'interruption **OCX**.  
1 : Interruption générée si **OCXF** à un.  
0 : Interruption inhibée.
- **I4/O5I**: Autorisation de l'interruption **OC5** ou **IC4**.  
1 : Interruption générée si **ICXF** à un.  
0 : Interruption inhibée.
- **IC1I-IC3I**: Autorisation de l'interruption **ICX**.  
1 : Interruption générée si **ICXF** à un.  
0 : Interruption inhibée.

## 13.10) Le registre Timer Flag 1 : TFLG1.

C'est le registre d'état des **OCX** et **ICX**.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>TFLG1</b>	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	<b>\$1023</b>
Reset	0	0	0	0	0	0	0	0	

Tous ces bits sont remis à zéro par écriture d'un 1 logique sur l'indicateur concerné.

- **OC1F-OC4F**: Indicateur de **OCX**.  
Mis à un à chaque égalité entre **TOCX** et **TCNT**.
- **I4-O5F**: Indicateur de **OC5** ou de **IC4**.  
Mis à un à chaque égalité entre **TOC5** et **TCNT**, ou changement d'état de **IC4**.
- **IC1F-IC3F**: Indicateur de **ICX**.  
Mis à un 1 si l'état de **ICX** est conforme à la configuration programmée par les bits **EDGXB** et **EDGXA** du registre **TCTL2**.

## 13.11) Le registre Timer Mask 2 : TMSK2.

Ce registre permet d'autoriser ou d'inhiber les interruptions suivantes :

	b7	b6	b5	b4	b3	B2	b1	b0	
<b>TMSK2</b>	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	<b>\$1024</b>
Reset	0	0	0	0	0	0	0	0	

• **TOI** : Autorisation ou inhibition de l'interruption de **TOI : Timer Overflow Interrupt**.

1 : Interruption autorisée.

0 : Interruption inhibée.

• **RTII** : Autorisation ou inhibition de l'interruption de **RTI : Real Time Interrupt**.

1 : Interruption autorisée.

0 : Interruption inhibée.

• **PAOVI** : Autorisation ou inhibition de l'interruption de **PAOVI : Pulse Accumulator Overflow Interrupt**.

1 : Interruption autorisée.

0 : Interruption inhibée.

• **PAII** : Autorisation ou inhibition de l'interruption de **PAI : Pulse Accumulator Input Edge Interrupt**.

1 : Interruption autorisée.

0 : Interruption inhibée.

**PR1** et **PR0** déterminent la valeur de la pré division de l'horloge du **TIMER**

PR[1:0]	Division
00	1
01	4
10	8
11	16

## 13.12) Le registre Timer Flag 2 : TFLG2.

C'est le registre d'état de **TO**.

	b7	b6	b5	b4	b3	b2	b1	b0	
<b>TFLG2</b>	TOF	RTIF	PAOVF	PAIF	-	-	-	-	<b>\$1025</b>
Reset	0	0	0	0	0	0	0	0	

Tous ces bits sont remis à zéro par écriture d'un 1 logique sur l'indicateur concerné.

• **TOF** : Indicateur de **TOI : Timer Overflow Interrupt**

Mis à 1 quand **TCNT** passe de \$FFFF à \$0000.

• **RTIF** : Indicateur de **RTI : Real Time Interrupt**.

• **PAOVF** : Indicateur de **PAOVI : Pulse Accumulator Overflow Interrupt**.

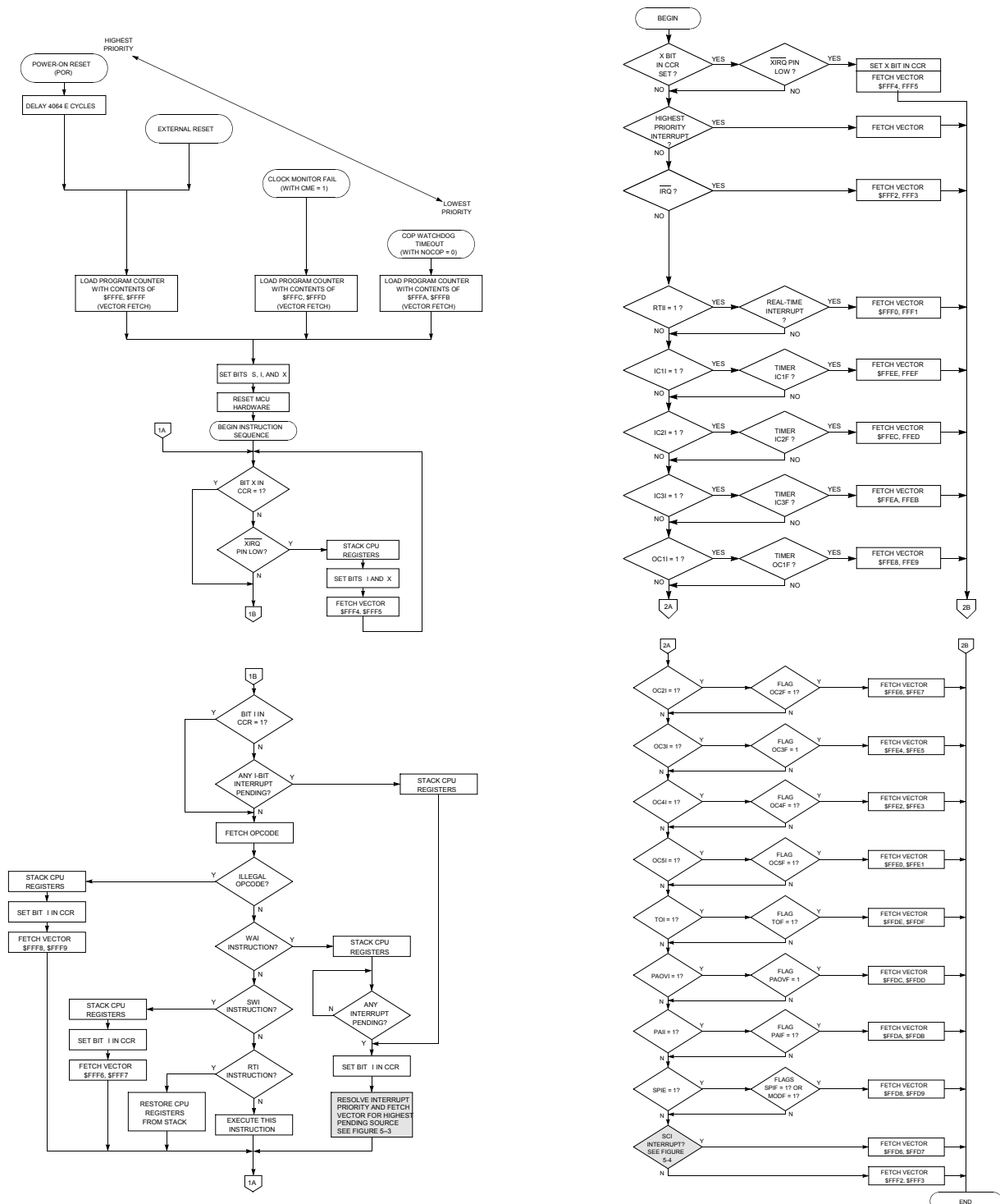
• **PAIF** : Indicateur de **PAI : Pulse Accumulator Input Edge Interrupt**.

## 14) LES INTERRUPTIONS.

### 14.1) Les sources d'interruptions.

Le  $\mu C$  dispose de 20 sources d'interruptions.

- Les Interruptions externes, action sur les broches **IRQ** et **XIRQ**.
- L'instruction **SWI (SOFTWARE INTERRUPT)**. Elle sert en général à un programme moniteur pour visualiser les registres du  $\mu C$ . Elle ne peut être inhibée.
- 14 interruptions générées par les fonctionnalités internes (**SPI**, **SCI** et **TIMER**), elles peuvent être inhibées par le bit **I** du **CCR**.
- 3 Interruptions d'initialisation **COP** et **RESET**.
- 1 Interruption de code instruction illégal.

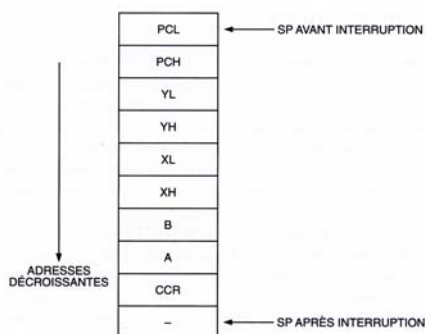




## 14.2) Déroulement d'une interruption.

En présence d'une interruption :

- Le  $\mu\text{C}$  termine l'exécution de l'instruction en cours.
- Il sauvegarde ensuite le contexte dans la pile, c'est à dire qu'il met dans la pile le contenu de tous registres dans l'ordre suivant :



- Il met le bit **I** ou **X** du **CCR** à 1 pour empêcher la prise en compte d'une autre interruption.
- Il charge le compteur programme **PC** avec le vecteur correspondant à l'interruption.
- Il exécute le programme d'interruption.
- Lorsqu'il rencontre l'instruction **RTI**, il restitue le contexte en dépilant le contenu de tous les registres.
- Il reprend le programme où il avait été interrompu.

## 14.3) L'interruption Défaut d'horloge.

Un circuit interne au  $\mu\text{C}$  surveille le bon fonctionnement du circuit d'horloge. En cas de fréquence insuffisante, une interruption « Clock Monitor Fail » est générée. Le bit **CME** (**Clock Monitor Enable**) du registre **OPTION** permet de masquer cette interruption.

## 14.4) L'interruption COP : Computer Operating Properly.

Ce « chien de garde » n'est autre qu'un compteur devant être réinitialisé régulièrement par logiciel. En cas d'initialisation non réalisée à temps, une interruption « COP Watchdog Time-Out » est générée. Le bit **NOCOP** du registre **CONFIG** permet de masquer cette interruption.

## 14.5) L'interruption Illegal Opcode.

Lors de la lecture d'un code opératoire incorrect (logiciel fonctionnant incorrectement), une interruption « Illegal Opcode » est générée.

## 14.6) L'interruption externe XIRQ.

Elle est la plus prioritaire des interruptions.

Le bit **X** du registre **CCR** permet le masquage de cette interruption.

## 14.7) L'interruption externe IRQ.

Elle peut être sensible à un front descendant ou à un niveau bas suivant la valeur du bit **IRQE** du registre **OPTION** :

- Si le bit **IRQE** = 0 alors sensible à un niveau bas.
- Si le bit **IRQE** = 1 alors sensible à un front descendant.

Le bit **I** du registre **CCR** permet le masquage de cette interruption.

## 14.8) L'interruption RTI.

Elle correspond à l'interruption de l'horloge temps réel. Si elle est autorisée (bit I du registre CCR à zéro), elle est déclenchée cycliquement.

## 14.9) Les interruptions TIMER.

Ces interruptions peuvent être déclenchées par plusieurs sources locales.

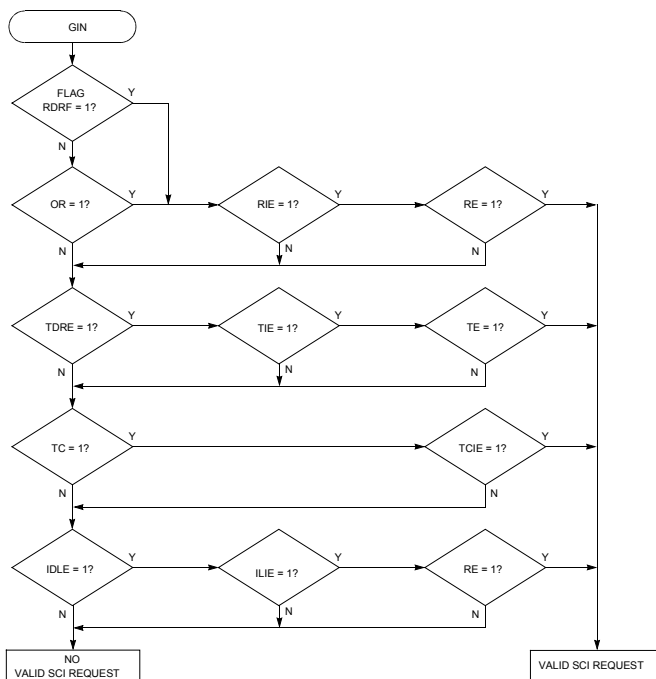
- Le changement d'état sur les broches **INPUT CAPTURE (TIC1, TIC2, TIC3 ou TIC4)**.
- L'égalité entre le **TIMER** et les registres d'**OUTPUT COMPARE (TOC1, TOC2, TOC3, TOC4 ou TOC5)**.
- Le dépassement de valeur du **TIMER**, passage de \$FFFF à \$0000 du registre TCNT.

## 14.10) L'interruption SCI.

Elle peut être provoquée par les **5** sources locales d'interruptions suivantes :

- La fin de transmission d'une valeur.
- Le registre de transmission est vide.
- La ligne de transmission est libre.
- Le registre de transmission est plein.
- Une valeur n'a pas été relue dans le registre de réception.

S'il y a au moins deux sources locales d'interruptions autorisées, étant donné que le  $\mu C$  ne dispose que d'un seul vecteur d'interruption pour la **SCI**, le programme d'interruption lors des premières instructions devra déterminer qu'elle est la source locale qui a déclenché l'interruption par une lecture du registre d'état **SCSR(\$102E)**.



## 14.11) L'interruption SPI.

Elle peut être provoquée par une seule source d'interruption:

- La fin d'émission ou de réception de donnée .

## 14.12) Vecteurs des différentes interruptions.

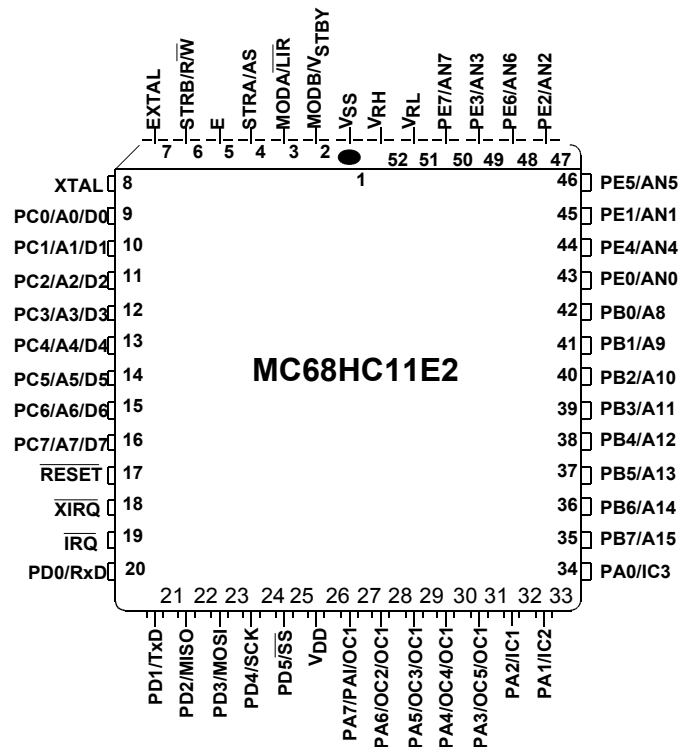
Vector Address	Interrupt Source	CCR Mask Bit	Local Mask
FFC0, C1 – FFD4, D5	Reserved	—	—
FFD6, D7	SCI Serial System <ul style="list-style-type: none"> <li>• SCI Receive Data Register Full</li> <li>• SCI Receiver Overrun</li> <li>• SCI Transmit Data Register Empty</li> <li>• SCI Transmit Complete</li> <li>• SCI Idle Line Detect</li> </ul>	I	RIE RIE TIE TCIE ILIE
FFD8, D9	SPI Serial Transfer Complete	I	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I	PAII
FFDC, DD	Pulse Accumulator Overflow	I	PAOVI
FFDE, DF	Timer Overflow	I	TOI
FFE0, E1	Timer Input Capture 4/Output Compare 5	I	I4/O5I
FFE2, E3	Timer Output Compare 4	I	OC4I
FFE4, E5	Timer Output Compare 3	I	OC3I
FFE6, E7	Timer Output Compare 2	I	OC2I
FFE8, E9	Timer Output Compare 1	I	OC1I
FFEA, EB	Timer Input Capture 3	I	IC3I
FFEC, ED	Timer Input Capture 2	I	IC2I
FFEE, EF	Timer Input Capture 1	I	IC1I
FFF0, F1	Real-Time Interrupt	I	RTII
FFF2, F3	$\overline{\text{IRQ}}$ (External Pin)	I	None
FFF4, F5	$\overline{\text{XIRQ}}$ Pin	X	None
FFF6, F7	Software Interrupt	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure	None	NOCOP
FFFC, FD	Clock Monitor Fail	None	CME
FFFE, FF	RESET	None	None

# LE MICRO CONTROLEUR MC68HC811E2

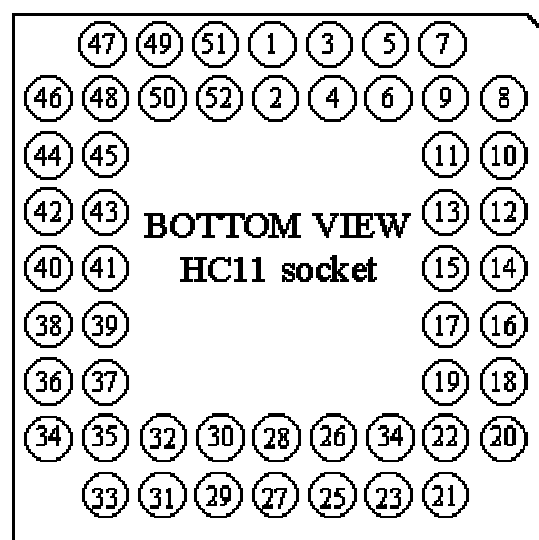
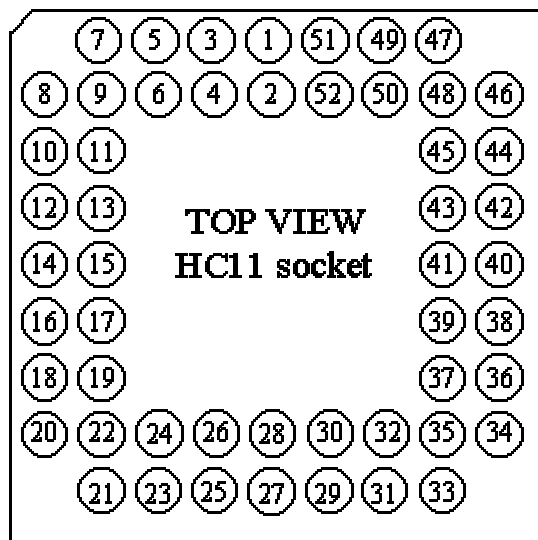
## 15) LISTE DE TOUS LES REGISTRES.

	Bit 7	6	5	4	3	2	1	Bit 0	
\$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
\$1001									Reserved
\$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC
\$1003	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
\$1004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
\$1005	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	PORTCL
\$1006									Reserved
\$1007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
\$1008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
\$1009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
\$100A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
\$100B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
\$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
\$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
\$100E	Bit 15	14	13	12	11	10	9	Bit 8	TCNT (High)
\$100F	Bit 7	6	5	4	3	2	1	Bit 0	TCNT (Low)
\$1010	Bit 15	14	13	12	11	10	9	Bit 8	TIC1 (High)
\$1011	Bit 7	6	5	4	3	2	1	Bit 0	TIC1 (Low)
\$1012	Bit 15	14	13	12	11	10	9	Bit 8	TIC2 (High)
\$1013	Bit 7	6	5	4	3	2	1	Bit 0	TIC2 (Low)
\$1014	Bit 15	14	13	12	11	10	9	Bit 8	TIC3 (High)
\$1015	Bit 7	6	5	4	3	2	1	Bit 0	TIC3 (Low)
\$1016	Bit 15	14	13	12	11	10	9	Bit 8	TOC1 (High)
\$1017	Bit 7	6	5	4	3	2	1	Bit 0	TOC1 (Low)
\$1018	Bit 15	14	13	12	11	10	9	Bit 8	TOC2 (High)
\$1019	Bit 7	6	5	4	3	2	1	Bit 0	TOC2 (Low)
\$101A	Bit 15	14	13	12	11	10	9	Bit 8	TOC3 (High)
\$101B	Bit 7	6	5	4	3	2	1	Bit 0	TOC3 (Low)
\$101C	Bit 15	14	13	12	11	10	9	Bit 8	TOC4 (High)
\$101D	Bit 7	6	5	4	3	2	1	Bit 0	TOC4 (Low)
\$101E	Bit 15	14	13	12	11	10	9	Bit 8	TI4/O5 (High)
\$101F	Bit 7	6	5	4	3	2	1	Bit 0	TI4/O5 (Low)
\$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
\$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
\$1022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	TMSK1
\$1023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	TFLG1
\$1024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	TMSK2
\$1025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
\$1026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0	PACTL
\$1027	Bit 7	6	5	4	3	2	1	Bit 0	PACNT
\$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
\$1029	SPIF	WCOL	0	MODF	0	0	0	0	QSPSR
\$102A	Bit 7	6	5	4	3	2	1	Bit 0	SPDR
\$102B	TCLR		SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
\$102C	R8	T8	0	M	WAKE	0	0	0	SCCR1
\$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
\$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
\$102F	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	SCDR
\$1030	CCF	0	SCAN	MULT	CD	CC	CB	CA	ADCTL
\$1031	Bit 7	6	5	4	3	2	1	Bit 0	ADR1
\$1032	Bit 7	6	5	4	3	2	1	Bit 0	ADR2
\$1033	Bit 7	6	5	4	3	2	1	Bit 0	ADR3
\$1034	Bit 7	6	5	4	3	2	1	Bit 0	ADR4
\$1035	0	0	0	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT
\$1036									Reserved
\$1037									Reserved
\$1038									Reserved
\$1039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0	OPTION
\$103A	Bit 7	6	5	4	3	2	1	Bit 0	COPRST
\$103B	ODD	EVEN	ELAT(3)	BYTE	ROW	ERASE	EELAT	EPGM	PPROG
\$103C	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
\$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
\$103E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1
\$103F	EE3(4)	EE2(4)	EE1(4)	EE0(4)	NOSEC	NOCOP	ROMON	EEON	CONFIG

## 16) CARACTERISTIQUES MECANIKES.



MC68HC811E2 Pin Assignments (52-Pin PLCC)



# LE MICRO CONTROLEUR MC68HC811E2

## 17) REFERENCES.

	MC	68	HC	P	7	11XX	B	C	FN	3	R2
<b>QUALIFICATION LEVEL</b> MC — FULLY SPECIFIED AND QUALIFIED XC — PILOT PRODUCTION DEVICE PC — ENGINEERING SAMPLE											
<b>NUMERIC DESIGNATOR (OPTIONAL)</b>											
<b>OPERATING VOLTAGE RANGE</b> HC — HCMOS ( $V_{DD}=5.0\text{ VDC}\pm 10\%$ ) L — HCMOS ( $V_{DD}=3.0\text{ VDC TO }5.5\text{ VDC}$ )											
<b>COP OPTION (ONLY ON A-SERIES DEVICES)</b> NONE — COP DISABLED P — COP ENABLED											
<b>MEMORY TYPE</b> BLANK — MASKED ROM OR NO ROM 7 — EPROM/OTPROM 8 — EEPROM											
<b>BASE PART NUMBER</b> 11A8, 11D3, 11E9, 11K4, ETC.											
<b>MONITOR MASK</b> NONE — BLANK B — BUFFALO											
<b>TEMPERATURE RANGE</b> NONE — $0^{\circ}\text{C TO }70^{\circ}\text{C}$ C — $-40^{\circ}\text{C TO }85^{\circ}\text{C}$ V — $-40^{\circ}\text{C TO }105^{\circ}\text{C}$ M — $-40^{\circ}\text{C TO }125^{\circ}\text{C}$											
<b>PACKAGE TYPE</b> FN — 44/52/68/84-PIN PLCC FS — 44/52/68/84-PIN CLCC FU — 64/80-PIN QFP FB — 44-PIN QFP PV — 112-PIN TQFP PU — 80/100-PIN TQFP PB — 52-PIN TQFP P — 40/48-PIN DIP S — 48-PIN SDIP											
<b>MAXIMUM SPECIFIED CLOCK SPEED</b> 2 — 2.0 MHz 3 — 3.0 MHz 4 — 4.0 MHz											
<b>TAPE AND REEL OPTION</b> NONE — STANDARD PACKAGING R2 — TAPE AND REEL PACKAGING											

# LE MICRO CONTROLEUR MC68HC811E2

## **18) FAMILLE HC11.**

Part Number	EPROM	ROM	EEPROM	RAM	CONFIG <sup>2</sup>	Comments
MC68HC11A8	—	—	512	256	\$0F	Family Built Around This Device
MC68HC11A1	—	—	512	256	\$0D	'A8 with ROM Disabled
MC68HC11A0	—	—	—	256	\$0C	'A8 with ROM and EEPROM Disabled
MC68HC811A8	—	—	8K + 512	256	\$0F	EEPROM Emulator for 'A8
MC68HC11E9	—	12K	512	512	\$0F	Four Input Capture/Bigger RAM 12K ROM
MC68HC11E1	—	—	512	512	\$0D	'E9 with ROM Disabled
MC68HC11E0	—	—	—	512	\$0C	'E9 with ROM and EEPROM Disabled
MC68HC811E2	—	—	2K <sup>1</sup>	256	\$FF <sup>3</sup>	No ROM Part for Expanded Systems
MC68HC711E9	12K	—	512	512	\$0F	One-Time Programmable Version of 'E9
MC68HC11D3	—	4K	—	192	N/A	Low-Cost 40-Pin Version
MC68HC711D9	4K	—	—	192	N/A	One-Time Programmable Version of 'D3
MC68HC11F1	—	—	512 <sup>1</sup>	1K	\$FF <sup>3</sup>	High-Performance Non-Multiplexed 68-Pin
MC68HC11K4	—	24K	640	768	\$FF	> 1 Mbyte memory space, PWM, C <sub>S</sub> , 84-Pin
MC68HC711K4	24K	—	640	768	\$FF	One-Time Programmable Version of 'K4
MC68HC11L6	—	16K	512	512	\$0F	Like 'E9 with more ROM and more I/O, 64/68
MC68HC711L6	16K	—	512	512	\$0F	One-Time Programmable Version of 'L4

1. The EEPROM is relocatable to the top of any 4 Kbyte memory page. Relocation is done with the upper four bits of the CONFIG register.
2. CONFIG register values in this table reflect the value programmed prior to shipment from Motorola.
3. At the time of this printing a change was being considered that would make this value \$0F.

## **19) BIBLIOGRAPHIE.**

**M68HC11 REFERENCE MANUAL** (Motorola) : *Hc11rm.pdf*

**M68HC11 E SERIES HCMOS MICROCONTROLLER UNIT** (Motorola) : *E\_series.pdf*

**M68HC11A8 MICROCONTROLLER UNIT** (Motorola) : *11a8bk.pdf*