

Famille PIC18F...



Sommaire

Présentation Générale	1
Les différents types d'horloges et leurs paramétrages	3
Le chien de garde (WATCHDOG)	5
Architecture interne et registre d'état (STATUS)	6
Organisation Mémoire Programme	8
Les registres de Configuration	9
Organisation Mémoire de Données	10
Registre de la zone SFR (SPECIAL FUNCTION REGISTER)	11
Liste des registres d'I/O	12
Modes d'Adressage	16
Le Jeu d'instructions	18
 Les Entrées – Sorties	
Les ports parallèles	19
Logique des interruptions	20
Les Timers 0-1-2-3	22
Le Convertisseur Analogique-Digital	26
Les Ports Série EUSART	29
 Le KITPIC du GE2i	
• Table de câblage	33
• µC 18F67J11	35
• Afficheurs 7 segments	36
• Ecran LCD, Codeur incrémental, BP	37
• Autres IO	38
• Moteurs CC et PàP	39
• Alimentations	40
• µC 18F66J60 (Ethernet)	41-42

Présentation générale de la Famille PIC 1x

8-bit PIC® Architectures				PIC18 Architecture
	Baseline Architecture	Midrange Architecture	Enhanced Midrange Architecture	
				
Pin count	6 – 40	8 – 64	8 – 64	18 – 100
Interrupts	No	Single Interrupt Capability	Single Interrupt Capability with Hardware Context Save	Multiple Interrupt Capability with Hardware Context Save
Operating Performance	5 MIPS	5 MIPS	8 MIPS	10 – 16 MIPS
Instructions	33, 12-bit instructions	35, 14-bit instructions	49, 14-bit instructions	75 – 83, 16-bit instructions
Program Memory	Up to 3 KB	Up to 14 KB	Up to 56 KB	Up to 128 KB
Data Memory	Up to 138 Bytes	Up to 368 Bytes	Up to 4 KB	Up to 4 KB
Features	<ul style="list-style-type: none"> Smallest form factor Lowest cost Ideal for battery operated or space constrained applications Easy to learn & use 	<ul style="list-style-type: none"> Optimal cost-to-performance ratio Integrated peripherals including SPI, I²C™, UART, LCD, ADC 	<ul style="list-style-type: none"> C-code Optimized Enhanced 16 Level Hardware Stack Enhanced Indirect Addressing Reduced Interrupt Latency Simplified Memory Map 	<ul style="list-style-type: none"> 32 level deep stack, 8x8 hardware multiplier C-code optimized Advanced peripherals including CAN, USB, Ethernet, touch sensing, and LCD drivers
Families	Includes PIC10 , PIC12 and PIC16	Includes PIC12 and PIC16	Includes PIC12F1xxx & PIC16F1xxx	PIC18 J-series for cost-sensitive applications with high levels of integration PIC18 K-series for low power, high-performance applications

Famille PIC 18FxxJxx

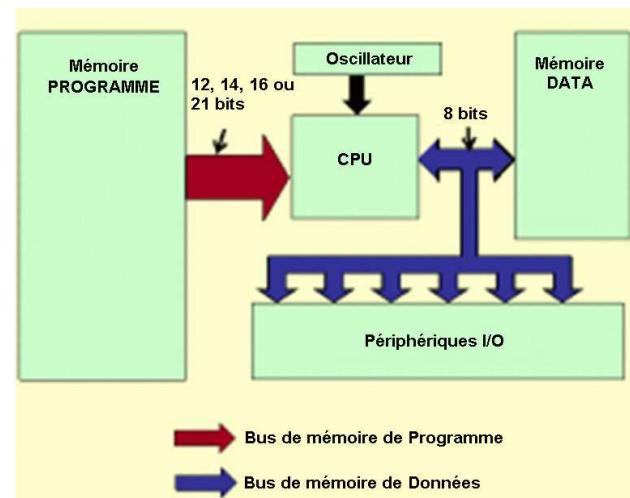
Features	PIC18F66J11	PIC18F66J16	PIC18F67J11
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	64K	96K	128K
Program Memory (Instructions)	32768	49152	65536
Data Memory (Bytes)	3930	3930	3930
Interrupt Sources	29		
I/O Ports	Ports A, B, C, D, E, F, G		
Timers	5		
Capture/Compare/PWM Modules	2		
Enhanced Capture/Compare/PWM Modules	3		
Serial Communications	MSSP (2), Enhanced USART (2)		
Parallel Communications (PMP)	Yes		
10-Bit Analog-to-Digital Module	11 Input Channels		
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST)		
Instruction Set	75 Instructions, 83 with Extended Instruction Set Enabled		
Packages	64-Pin TQFP		

NOTA : Les images et tableaux sont recopierés du DataSheet de la famille PIC18F87J11 disponible sur www.microchip.com
 Ouvrage servant de support : [Microcontrôleurs PIC18](#) de Christian Tavernier Edition DUNOD

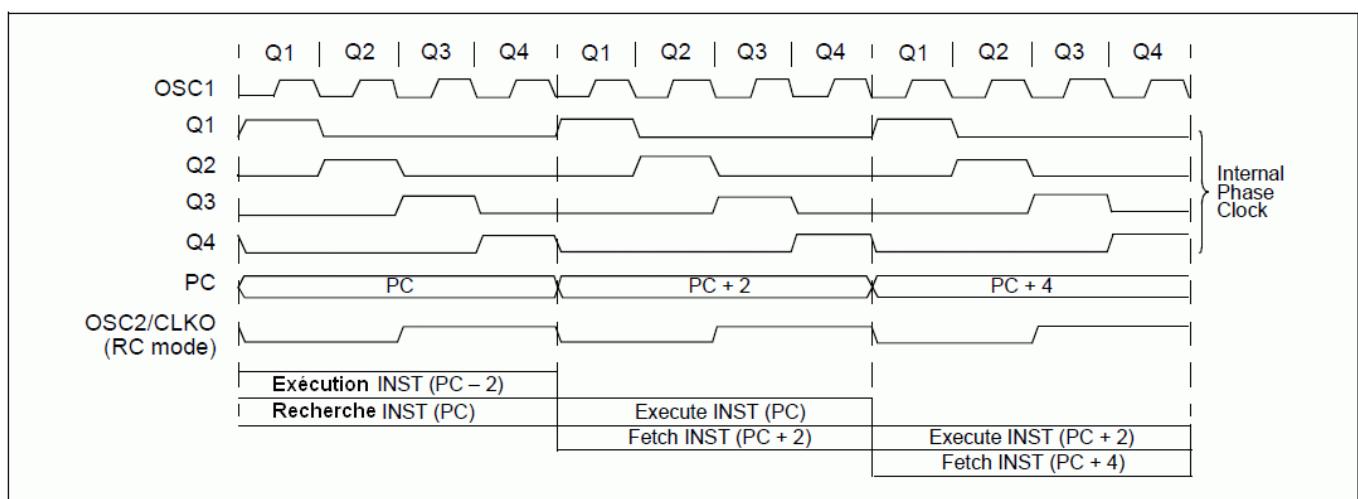
- Architecture HARVARD



- Horloge max de 48 MHz par multiplication interne avec PLL (avec quartz 12MHz)
- PIC18F avec rom programme FLASH
- Programmation et debug sur cible
- Chaque instruction sur 16 bits
- Mémoire EEPROM à accès aléatoire (sauvegarde de données)
- TIMERS et Compteurs
- Mesure de temps : Capture – Compare - PWM
- SPI : communication série synchrone sans protocole logiciel
- I2C : standard Philips, communication série synchrone avec protocole logiciel
- USART : communication série asynchrone (RS232 et RS485)
- CAN : convertisseur analogique numérique à 10 entrées multiplexées
- Multiplication 8x8 matérielle (pour par ex. le traitement numérique du signal)



Dans une **architecture Harvard** les instructions et les données étant clairement différencierées (bus distincts), on peut en un **seul cycle machine** exécuter une instruction et rechercher le code de l'instruction suivante. => **Gain de temps plus qu'appreciable.**

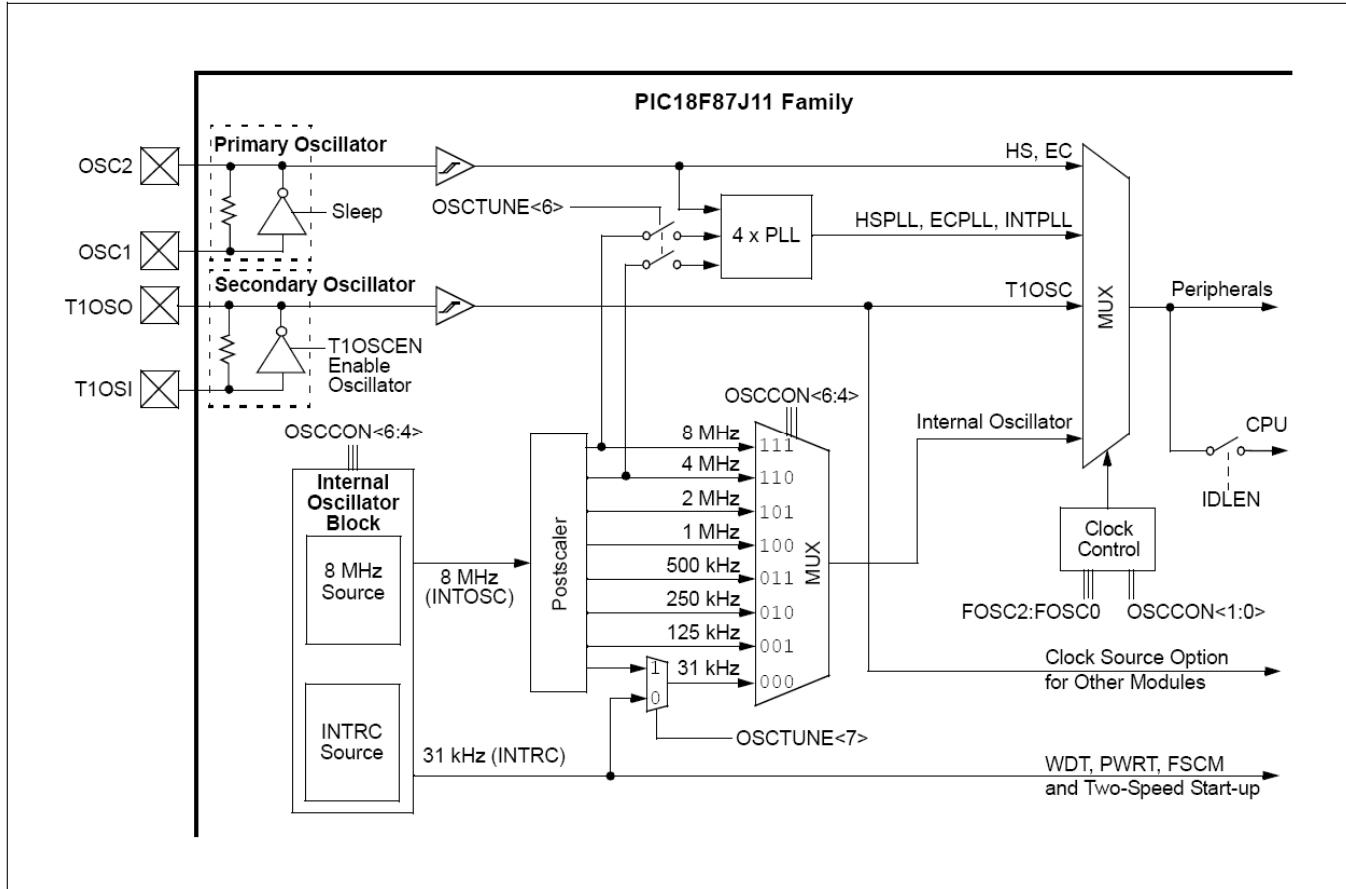


Quelques membres de la famille :

Device	Flash Program Memory (bytes)	SRAM Data Memory (bytes)	I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	MSSP		EUSART	Comparators	Timers 8/16-Bit	External Bus	PMP/EPSP	
						SPI	Master I ² C™						
PIC18F66J11	64 kB	3930	52	11	2/3	2	Y	Y	2	2	2/3	N	Y
PIC18F66J16	96 kB	3930	52	11	2/3	2	Y	Y	2	2	2/3	N	Y
PIC18F67J11	128 kB	3930	52	11	2/3	2	Y	Y	2	2	2/3	N	Y
PIC18F86J11	64 kB	3930	68	15	2/3	2	Y	Y	2	2	2/3	Y	Y
PIC18F86J16	96 kB	3930	68	15	2/3	2	Y	Y	2	2	2/3	Y	Y
PIC18F87J11	128 kB	3930	68	15	2/3	2	Y	Y	2	2	2/3	Y	Y

Les horloges

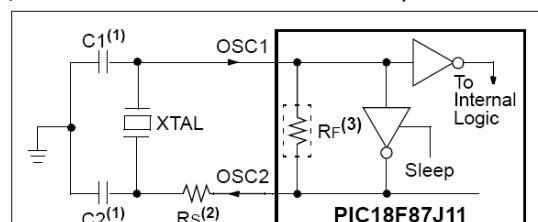
PIC18F87J11 FAMILY CLOCK DIAGRAM



Les principaux types d'horloges :

LP	basse vitesse (Low Power)	$f < 200 \text{ kHz}$
XT	vitesse standard	$f < 4 \text{ MHz}$
HS	haute vitesse	$f < 48 \text{ MHz}$
HSPLL	HS avec boucle à verrouillage de phase interne (permet un x4)	
RC		
EC	horloge externe sur CLK1	
INTIO	horloge interne	

CRYSTAL/CERAMIC RESONATOR OPERATION (HS OR HSPLL CONFIGURATION)



Note 1: See Table 2-1 and Table 2-2 for initial values of C1 and C2.

2: A series resistor (Rs) may be required for AT strip cut crystals.

3: RF varies with the oscillator mode chosen.

Le choix de l'horloge primaire s'effectue par la programmation du registre CONFIG2L

La commutation d'horloge pendant l'exécution peut s'effectuer par la programmation du registre OSCCON

OSCCON: OSCILLATOR CONTROL REGISTER⁽¹⁾

R/W-0	R/W-1	R/W-1	R/W-0	R ⁽²⁾	U-1	R/W-0	R/W-0
IDLEN	IRCF2 ⁽³⁾	IRCF1 ⁽³⁾	IRCF0 ⁽³⁾	OSTS	—	SCS1 ⁽⁵⁾	SCS0 ⁽⁵⁾
bit 7					—		bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7	IDLEN: Idle Enable bit
	1 = Device enters an Idle mode when a SLEEP instruction is executed
	0 = Device enters Sleep mode when a SLEEP instruction is executed
bit 6-4	IRCF2:IRCF0: INTOSC Source Frequency Select bits ⁽³⁾
	111 = 8 MHz (INTOSC drives clock directly)
	110 = 4 MHz (default)
	101 = 2 MHz
	100 = 1 MHz
	011 = 500 kHz
	010 = 250 kHz
	001 = 125 kHz
	000 = 31 kHz (from either INTOSC/256 or INTRC) ⁽⁴⁾
bit 3	OSTS: Oscillator Start-up Timer Time-out Status bit ⁽²⁾
	1 = Oscillator Start-up Timer (OST) time-out has expired; primary oscillator is running
	0 = Oscillator Start-up Timer (OST) time-out is running; primary oscillator is not ready
bit 2	Unimplemented: Read as '1'
bit 1-0	SCS1:SCS0: System Clock Select bits ⁽⁵⁾
	11 = Internal oscillator block
	10 = Primary oscillator
	01 = Timer1 oscillator
	00 = Default primary oscillator (as defined by FOSC2:FOSC0 Configuration bits)

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

2: Reset state depends on state of the IESO Configuration bit.

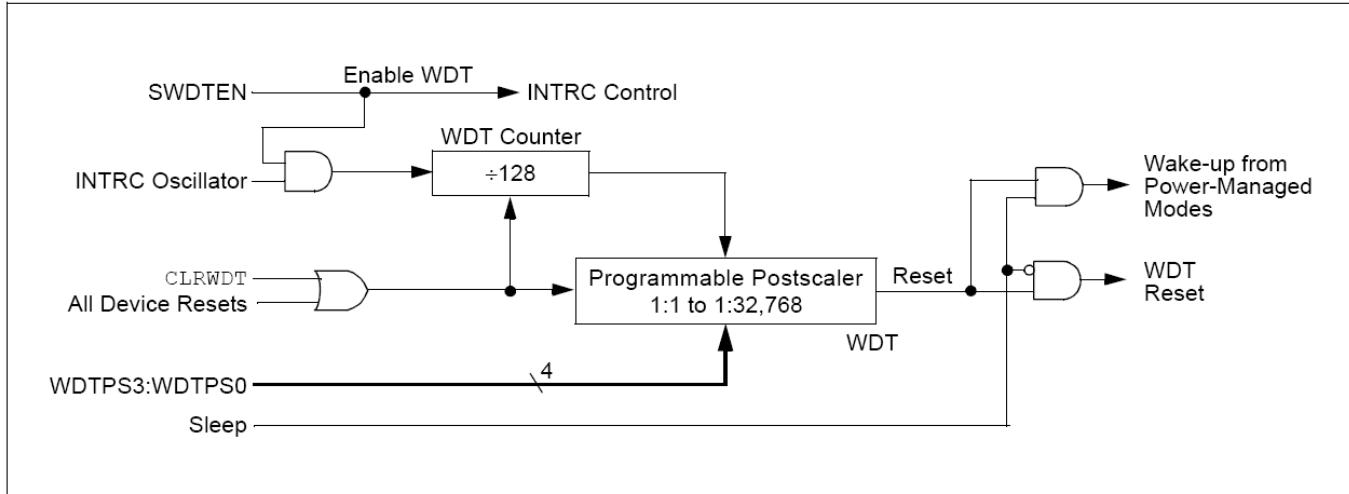
3: Modifying these bits will cause an immediate clock frequency switch if the internal oscillator is providing the device clocks.

4: Source selected by the INTSRC bit (OSCTUNE<7>), see text.

5: Modifying these bits will cause an immediate clock source switch.

Un timer chien de garde (WATCHDOG) est disponible et configurable par CONFIG2H

WDT BLOCK DIAGRAM



WDTCON: WATCHDOG TIMER CONTROL REGISTER

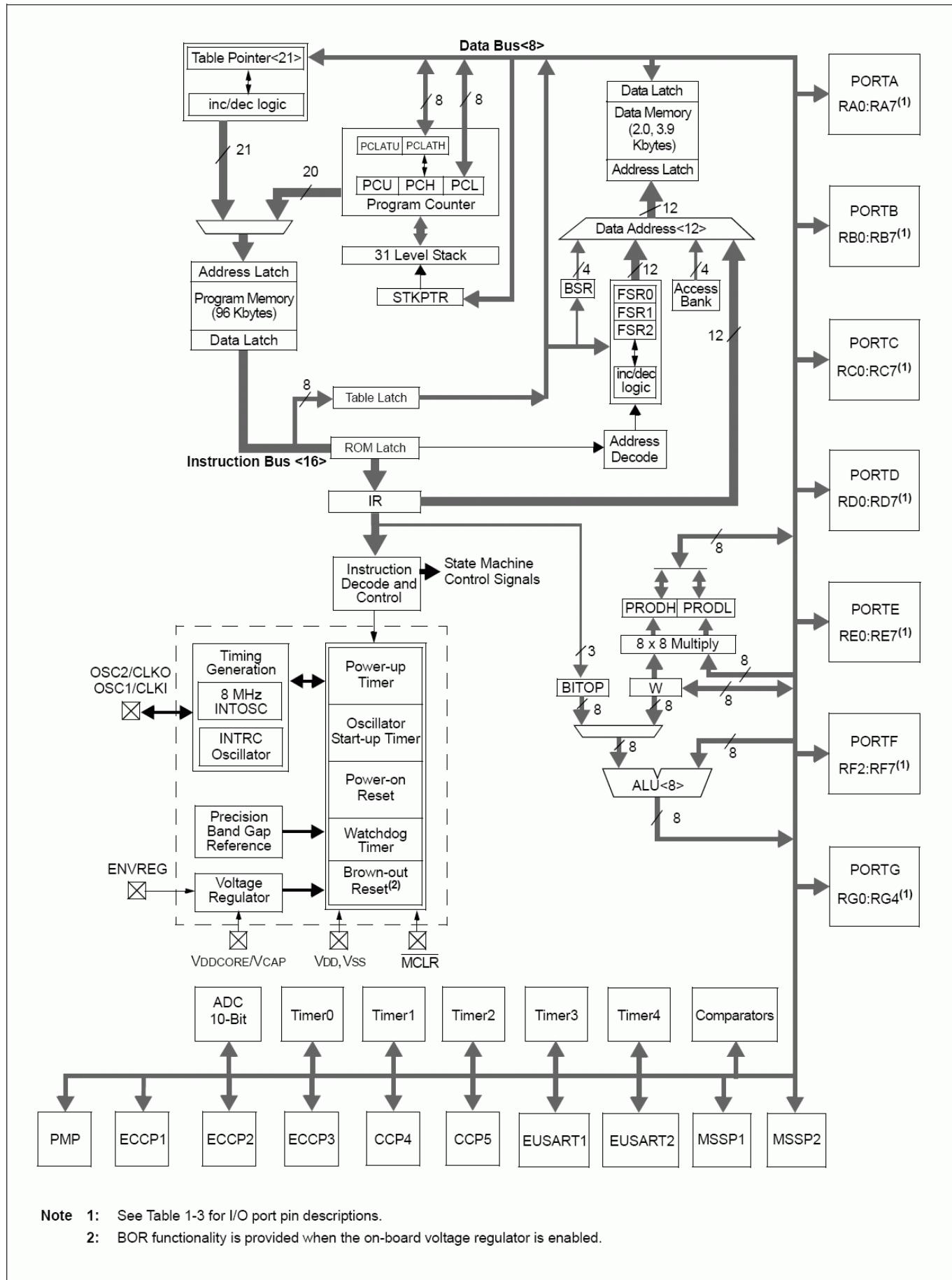
R/W-0	R-x	U-0	R/W-0	U-0	U-0	U-0	U-0
REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN
bit 7	bit 0						

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

- bit 7 **REGSLP:** Voltage Regulator Low-Power Operation Enable bit
For details of bit operation, see Register 24-9.
- bit 6 **LVDSTAT:** LVD Status bit
1 = VDDCORE > 2.45V
0 = VDDCORE < 2.45V
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **ADSHR:** Shared Address SFR Select bit
1 = Alternate SFR is selected
0 = Default (Legacy) SFR is selected
- bit 3-1 **Unimplemented:** Read as '0'
- bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit
For details of bit operation, see Register 24-9.

Architecture interne



Organisation de l'**unité centrale**

Toute manipulation de l'information s'effectue par l'intermédiaire de l'**U.A.L.** (Unité Arithmétique et Logique) de taille 8 bits d'où le nom de **Microcontrôleur 8 bits** pour cette famille de composants.

- Lors d'une opération, le premier opérande est stocké dans le **registre W** ou « registre de travail » appelé aussi chez certains constructeurs « Accumulateur » ; il est naturellement de 8 bits.
 - Le second opérande vient directement d'un octet mémoire par le bus de données.
 - Le résultat de cette opération se trouvera, par le choix de l'instruction, soit dans W, soit réécrit dans l'octet mémoire.
 - Un état de l'opération effectuée est mémorisé dans le registre STATUS. Chaque bit de celui-ci a une signification particulière :

STATUS REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC ⁽¹⁾	C ⁽²⁾
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **N**: Negative bit

This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1).

1 = Result was negative
0 = Result was positive

bit 3 **OV**: Overflow bit

This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7) to change state.

1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
0 = No overflow occurred

bit 2 Z: Zero bit

- 1 = The result of an arithmetic or logic operation is zero
- 0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit carry/borrow bit⁽¹⁾

For ADDWF, ADDLW, SUBLW and SUBWF instructions:

- 1 = A carry-out from the 4th low-order bit of the result occurred
- 0 = No carry-out from the 4th low-order bit of the result

bit 0 C: Carry/borrow bit⁽²⁾

For ADDWF, ADDLW, SUBLW and SUBWF instructions:

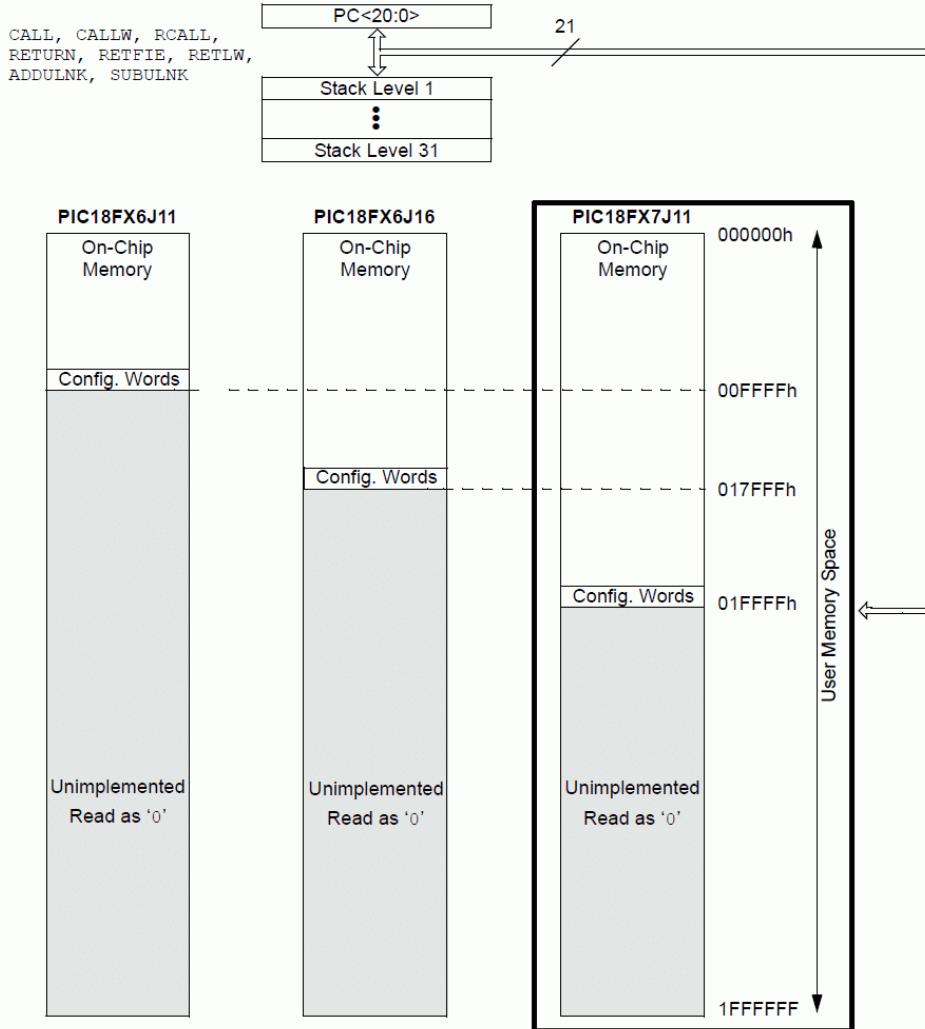
- A carry-out from the Most Significant bit of the result occurred
- No carry-out from the Most Significant bit of the result occurred

Note 1: For borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.

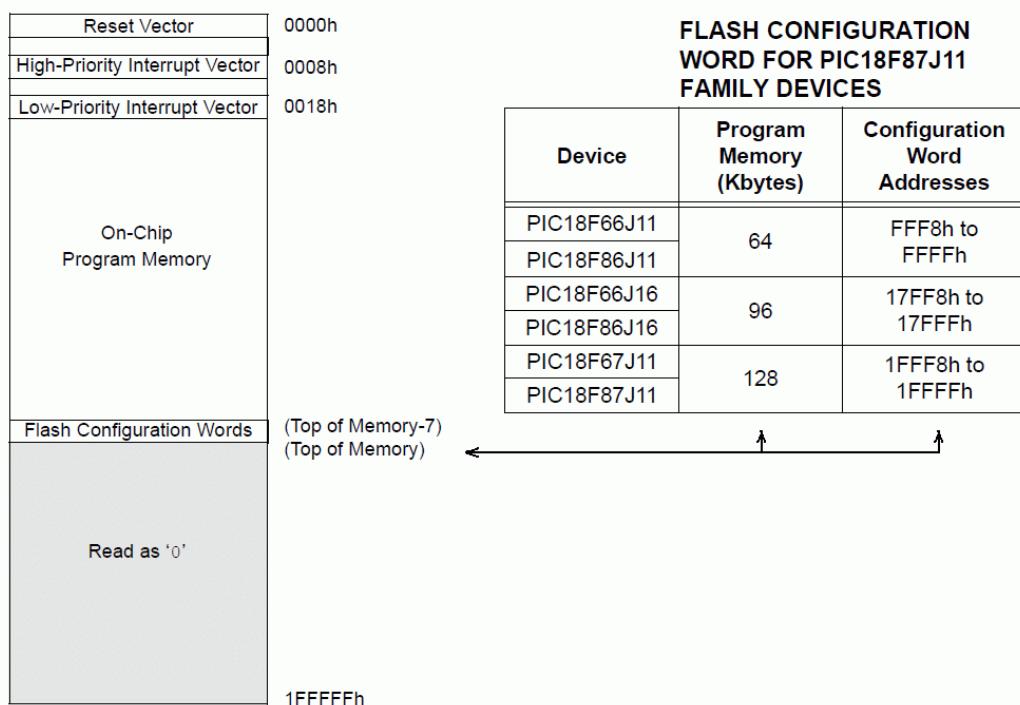
2: For borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low-order bit of the source register.

Organisation de la mémoire de programme

Dans notre cas (PIC18F67J11) l'espace mémoire est de 128 Koct. Les registres de configuration CONFIGxx sont donc implantés de @1FFF8h à @1FFFFh



Note: Sizes of memory areas are not to scale. Sizes of program memory areas are enhanced to show detail.



PIC18F87J11 FAMILY

PIC18F87J11 FAMILY

TABLE 24-1: MAPPING OF THE FLASH CONFIGURATION WORDS TO THE CONFIGURATION REGISTERS

Configuration Byte	Code Space Address	Configuration Register Address
CONFIG1L	XXXXFBh	300000Dh
CONFIG1H	XXXXFB0h	3000011h
CONFIG2L	XXXXFAh	3000022h
CONFIG2H	XXXXFBh	3000033h
CONFIG3L	XXXXFC0h	3000044h
CONFIG3H	XXXXFC0h	3000055h
CONFIG4L (1)	XXXXFEh	3000068h
CONFIG4H (1)	XXXXFFh	3000077h

Note 1: Unimplemented in PIC18F87J11 Family devices.

TABLE 24-2: CONFIGURATION BITS AND DEVICE IDs

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/Unprogrammed Value[1]
300000h	CONFIG1L	DEBUG	XINST	STVREN	—	—	—	WDTEN	1111----1
300001h	CONFIG1H	—[2]	—[2]	—[2]	—	—	CP0	—	1111-1111
300002h	CONFIG2L	IESO	FCMEN	—	—	—	FOSC2	FOSC1	11---1111
300003h	CONFIG2H	—[2]	—[2]	—[2]	—	WDTFS3	WDTFS2	WDTFS1	11111111
300004h	CONFIG3L	WAIT[3]	BW[5]	EMB0[3]	EASHFT[9]	—	—	—	111111111
300005h	CONFIG3H	—[2]	—[2]	—[2]	—	MSSPMISK	PMMPMX[3]	ECCP2MX[8]	CCP2MX
3FFFFFFh	DEV1D1	DEV1	DEV2	DEV0	REV4	REV3	REV2	REV1	REV0
3FFFFFFh	DEV1D2	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	xxxx00000[4]

Legend: xx = unknown; u = unchanged; - = unimplemented. Shaded cells are unimplemented, read as '0'.

Note 1: Values reflect the unprogrammed state as received from the factory and following Power-on Resets. In all other Reset states, the configuration bytes maintain their previous programmed states.

2: The value of these bits in program memory should always be '1'. This ensures that the location is executed as a NOP if it is accidentally executed.

3: Implemented in 80-pin devices only.

4: See Register 24-7 and Register 24-8 for DEVID values. These registers are read-only and cannot be programmed by the user.

REGISTER 24-3: CONFIG2: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

	R/W0-1 IESO	R/W0-1 FCMEN	U-0 —	U-0 —	U-0 —	R/W0-1 FOSC1	R/W0-1 FOSC0
bit 7							bit 0

Legend:

R = Readable bit
 -n = Value at POR
 '1' = Bit is set
 '0' = Bit is cleared
 x = Bit is unknown

IESO: Two-Speed Start-up (Internal/External Oscillator Switchover) Control bit

1 = Two-Speed Start-up enabled
 0 = Two-Speed Start-up disabled

FCMEN: Fail-Safe Clock Monitor Enable bit

1 = Fail-Safe Clock Monitor enabled
 0 = Fail-Safe Clock Monitor disabled

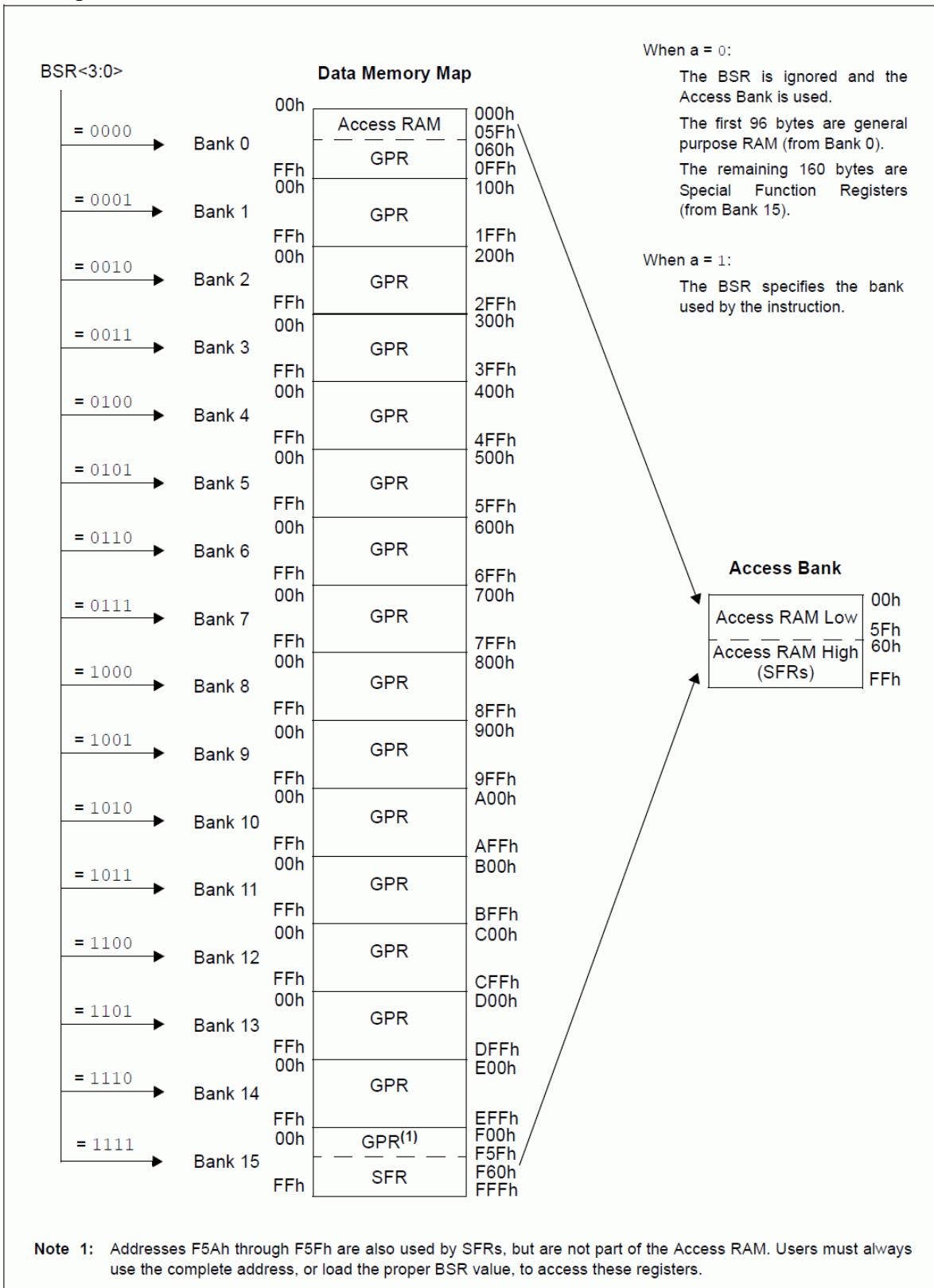
Unimplemented: Read as '0'

FOSC2:FOSC0: Oscillator Selection bits

111 = EC oscillator with PLL enabled; CLKO on RA6 (ECPLL)
 110 = EC oscillator; CLKO on RA6 (EC)
 101 = HS oscillator with PLL enabled (HSPLL)
 100 = HS oscillator (HS)
 011 = Internal oscillator with PLL enabled; CLKO on RA6, port function on RA7 (INTPLL1)
 010 = Internal oscillator with PLL enabled; port function on RA6 and RA7 (INTPLL2)
 001 = Internal oscillator block; CLKO on RA6, port function on RA7 (INTIO1)
 000 = Internal oscillator block ; port function on RA6 and RA7 (INTIO2)

Organisation de la mémoire de données

- Elle est divisée en 16 pages de 256 octets appelée « bank ». Après avoir défini la **page active** (reg. BSR: Bank Select Register) il suffit donc de préciser le complément d'@ sur **8 bits** pour pointer sur tout élément de cette page. (voir les modes d'adressage)
- Une **page virtuelle** « access bank » est toujours accessible sans tenir compte du contenu de BSR ; on y retrouve les 96 octets « access RAM » implantée de @000 à @050h et la zone SFR. Il est donc recommandé de travailler avec cet espace mémoire toujours accessible quelque soit la page active par BSR.



Liste des registres de la **zone SFR** (@ F60h à @ FFFh)

SPECIAL FUNCTION REGISTER MAP FOR PIC18F87J11 FAMILY DEVICES

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	ECCP1AS	F9Fh	IPR1	F7Fh	SPBRGH1	F5Fh	PMDIN2H
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	ECCP1DEL	F9Eh	PIR1	F7Eh	BAUDCON1	F5Eh	PMDIN2L
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCPR1H	F9Dh	PIE1	F7Dh	SPBRGH2	F5Dh	PMEH
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR1L	F9Ch	RCSTA2	F7Ch	BAUDCON2	F5Ch	PMEL
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCP1CON	F9Bh	OSCTUNE	F7Bh	TMR3H	F5Bh	PMSTATH
FFAh	PCLATH	FDAh	FSR2H	FBAh	ECCP2AS	F9Ah	TRISJ ⁽²⁾	F7Ah	TMR3L	F5Ah	PMSTATL
FF9h	PCL	FD9h	FSR2L	FB9h	ECCP2DEL	F99h	TRISH ⁽²⁾	F79h	T3CON	F59h	—
FF8h	TBLPTRU	FD8h	STATUS	FB8h	CCPR2H	F98h	TRISG	F78h	TMR4	F58h	—
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	CCPR2L	F97h	TRISF	F77h	PR4 ⁽³⁾	F57h	—
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	CCP2CON	F96h	TRISE	F76h	T4CON	F56h	—
FF5h	TABLAT	FD5h	T0CON	FB5h	ECCP3AS	F95h	TRISD	F75h	CCPR4H	F55h	—
FF4h	PRODH	FD4h	—	FB4h	ECCP3DEL	F94h	TRISC	F74h	CCPR4L	F54h	—
FF3h	PRODL	FD3h	OSCCON ⁽³⁾	FB3h	CCPR3H	F93h	TRISB	F73h	CCP4CON	F53h	—
FF2h	INTCON	FD2h	CM1CON	FB2h	CCPR3L	F92h	TRISA	F72h	CCPR5H	F52h	—
FF1h	INTCON2	FD1h	CM2CON	FB1h	CCP3CON	F91h	LATJ ⁽²⁾	F71h	CCPR5L	F51h	—
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRG1	F90h	LATH ⁽²⁾	F70h	CCP5CON	F50h	—
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H ⁽³⁾	FAFh	RCREG1	F8Fh	LATG	F6Fh	SSP2BUF	F4Fh	—
FEEh	POSTINC0 ⁽¹⁾	FCEh	TMR1L ⁽³⁾	FAEh	TXREG1	F8Eh	LATF	F6Eh	SSP2ADD	F4Eh	—
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON ⁽³⁾	FADh	TXSTA1	F8Dh	LATE	F6Dh	SSP2STAT	F4Dh	—
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2 ⁽³⁾	FACh	RCSTA1	F8Ch	LATD	F6Ch	SSP2CON1	F4Ch	—
FEKh	PLUSW0 ⁽¹⁾	FCBh	PR2 ⁽³⁾	FABh	SPBRG2	F8Bh	LATC	F6Bh	SSP2CON2	F4Bh	—
FEAh	FSR0H	FCAh	T2CON	FAAh	RCREG2	F8Ah	LATB	F6Ah	CMSTAT	F4Ah	—
FE9h	FSR0L	FC9h	SSP1BUF	FA9h	TXREG2	F89h	LATA	F69h	PMADDRH ⁽⁴⁾	F49h	—
FE8h	WREG	FC8h	SSP1ADD	FA8h	TXSTA2	F88h	PORTJ ⁽²⁾	F68h	PMADDRL ⁽⁴⁾	F48h	—
FE7h	INDF1 ⁽¹⁾	FC7h	SSP1STAT	FA7h	EECON2	F87h	PORTH ⁽²⁾	F67h	PMDIN1H	F47h	—
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSP1CON1	FA6h	EECON1	F86h	PORTG	F66h	PMDIN1L	F46h	—
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSP1CON2	FA5h	IPR3	F85h	PORTF	F65h	PMCONH	F45h	—
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE	F64h	PMCONL	F44h	—
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD	F63h	PMMODEH	F43h	—
FE2h	FSR1H	FC2h	ADCON0 ⁽³⁾	FA2h	IPR2	F82h	PORTC	F62h	PMMODEL	F42h	—
FE1h	FSR1L	FC1h	ADCON1 ⁽³⁾	FA1h	PIR2	F81h	PORTB	F61h	PMDOUT2H	F41h	—
FE0h	BSR	FC0h	WDTCON	FA0h	PIE2	F80h	PORTA	F60h	PMDOUT2L	F40h	—

Note 1: This is not a physical register.

2: This register is not available on 64-pin devices.

3: This register shares the same address with another register (see Table 5-4 for alternate register).

4: The PMADDRH/L and PMDOU1H/L register pairs share the same address. PMADDR is used in Master modes and PMDOU1 is used in Slave modes.

SHARED SFR ADDRESSES FOR PIC18F87J11 FAMILY DEVICES

Address	Name	Address	Name	Address	Name
FD3h	(D) OSCCON	FCDh	(D) T1CON	FC2h	(D) ADCON0
	(A) REFOCON		(A) ODCON3		(A) ANCON1
FCFh	(D) TMR1H	FCCh	(D) TMR2	FC1h	(D) ADCON1
	(A) ODCON1		(A) PADCFG1		(A) ANCON0
FCEh	(D) TMR1L	FCBh	(D) PR2	F77h	(D) PR4
	(A) ODCON2		(A) MEMCON ⁽¹⁾		(A) CVRCON

Legend: (D) = Default SFR, accessible only when ADSHR = 0; (A) = Alternate SFR, accessible only when ADSHR = 1.

Note 1: Implemented in 80-pin devices only.

Liste des registres (les notes de renvoi sont à consulter directement dans le DataSheet)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:
TOSU	—	—	—	Top-of-Stack Upper Byte (TOS<20:16>)						---0 0000 55, 65
TOSH	Top-of-Stack High Byte (TOS<15:8>)						0000 0000 55, 65			
TOSL	Top-of-Stack Low Byte (TOS<7:0>)						0000 0000 55, 65			
STKPTR	STKFUL	STKUNF	—	SP4	SP3	SP2	SP1	SP0	00-0 0000 55, 66	
PCLATU	—	—	bit 21 ⁽¹⁾	Holding Register for PC<20:16>						---0 0000 55, 65
PCLATH	Holding Register for PC<15:8>						0000 0000 55, 65			
PCL	PC Low Byte (PC<7:0>)						0000 0000 55, 65			
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)						--00 0000 55, 96
TBLPTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)						0000 0000 55, 96			
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)						0000 0000 55, 96			
TABLAT	Program Memory Table Latch						0000 0000 55, 96			
PRODH	Product Register High Byte						xxxx xxxx 55, 109			
PRODL	Product Register Low Byte						xxxx xxxx 55, 109			
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x 55, 113	
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	1111 1111 55, 113	
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	1100 0000 55, 113	
INDF0	Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register)						N/A 55, 82			
POSTINC0	Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register)						N/A 55, 83			
POSTDEC0	Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register)						N/A 55, 83			
PREINC0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)						N/A 55, 83			
PLUSW0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by W						N/A 55, 83			
FSR0H	—	—	—	—	Indirect Data Memory Address Pointer 0 High Byte				---- xxxx 55, 82	
FSR0L	Indirect Data Memory Address Pointer 0 Low Byte						xxxx xxxx 55, 82			
WREG	Working Register						xxxx xxxx 55, 67			
INDF1	Uses contents of FSR1 to address data memory – value of FSR1 not changed (not a physical register)						N/A 55, 82			
POSTINC1	Uses contents of FSR1 to address data memory – value of FSR1 post-incremented (not a physical register)						N/A 55, 83			
POSTDEC1	Uses contents of FSR1 to address data memory – value of FSR1 post-decremented (not a physical register)						N/A 55, 83			
PREINC1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register)						N/A 55, 83			
PLUSW1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR1 offset by W						N/A 55, 83			
FSR1H	—	—	—	—	Indirect Data Memory Address Pointer 1 High Byte				---- xxxx 55, 82	
FSR1L	Indirect Data Memory Address Pointer 1 Low Byte						xxxx xxxx 55, 82			
BSR	—	—	—	—	Bank Select Register				---- 0000 55, 70	
INDF2	Uses contents of FSR2 to address data memory – value of FSR2 not changed (not a physical register)						N/A 56, 82			
POSTINC2	Uses contents of FSR2 to address data memory – value of FSR2 post-incremented (not a physical register)						N/A 56, 83			
POSTDEC2	Uses contents of FSR2 to address data memory – value of FSR2 post-decremented (not a physical register)						N/A 56, 83			
PREINC2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register)						N/A 56, 83			
PLUSW2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) – value of FSR2 offset by W						N/A 56, 83			
FSR2H	—	—	—	—	Indirect Data Memory Address Pointer 2 High Byte				---- xxxx 56, 82	
FSR2L	Indirect Data Memory Address Pointer 2 Low Byte						xxxx xxxx 56, 82			
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx 56, 80	
TMR0H	Timer0 Register High Byte						0000 0000 56, 179			
TMR0L	Timer0 Register Low Byte						xxxx xxxx 56, 179			
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111 56, 178	
OSCCON ⁽²⁾ /REFOCON ⁽³⁾	IDLEN	IRCF2	IRCF1	IRCF0	OSTS ⁽⁴⁾	—	SCS1	SCS0	0110 q100 56, 32	
	ROON	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	0-00 0000 56, 39	
CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111 56, 302	
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111 56, 302	
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	0-11 1100 54, 56, 125	

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:
TMR1H ⁽²⁾ /ODCON1 ⁽³⁾	Timer1 Register High Byte								xxxx xxxx	56, 182
	—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD	---0 0000	56, 129
TMR1L ⁽²⁾ /ODCON2 ⁽³⁾	Timer1 Register Low Byte								xxxx xxxx	56, 182
	—	—	—	—	—	—	U2OD	U1OD	---- --00	56, 129
T1CON ⁽²⁾ /ODCON3 ⁽³⁾	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	56, 182
	—	—	—	—	—	—	SPI2OD	SPI1OD	---- --00	56, 129
TMR2 ⁽²⁾ /PADCFG1 ⁽³⁾	Timer2 Register								0000 0000	56, 187
	—	—	—	—	—	—	—	PMPTTL	---- ---0	56, 130
PR2 ⁽²⁾ /MEMCON ^(3.7)	Timer2 Period Register								1111 1111	56, 187
	EDBIS	—	WAIT1	WAIT0	—	—	WM1	WM0	0-00 --00	56, 98
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	56, 187
SSP1BUF	MSSP1 Receive Buffer/Transmit Register								xxxx xxxx	56, 222, 231
SSP1ADD/SSP1MSK ⁽⁵⁾	MSSP1 Address Register (² C™ Slave mode), MSSP1 Baud Rate Reload Register (² C Master mode)								0000 0000	56, 231
	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	0000 0000	56, 238
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	56, 222, 232
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	56, 223, 233
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN/	SEN	0000 0000	56, 234, 268
	GCEN	ACKSTAT	ADMSK5 ⁽⁶⁾	ADMSK4 ⁽⁶⁾	ADMSK3 ⁽⁶⁾	ADMSK2 ⁽⁶⁾	ADMSK1 ⁽⁶⁾	SEN		
ADRESH	A/D Result Register High Byte								xxxx xxxx	57, 291
ADRESL	A/D Result Register Low Byte								xxxx xxxx	57, 291
ADCON0 ⁽²⁾ /ANCON1 ⁽³⁾	VCFG1	VCFG0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	0000 0000	57, 291
	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	0000 0000	57, 293
ADCON1 ⁽²⁾ /ANCON0 ⁽³⁾	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0000 0000	57, 292
	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	00-0 0000	57, 293
WDTCON	REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN	0x-0 ---0	57, 321
ECCP1AS	ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	0000 0000	57, 219
ECCP1DEL	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	0000 0000	57, 219
CCPR1H	Capture/Compare/PWM Register 1 Hlgh Byte								xxxx xxxx	57, 219
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	57, 219
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	57, 219
ECCP2AS	ECCP2ASE	ECCP2AS2	ECCP2AS1	ECCP2AS0	PSS2AC1	PSS2AC0	PSS2BD1	PSS2BD0	0000 0000	57, 219
ECCP2DEL	P2RSEN	P2DC6	P2DC5	P2DC4	P2DC3	P2DC2	P2DC1	P2DC0	0000 0000	57, 219
CCPR2H	Capture/Compare/PWM Register 2 High Byte								xxxx xxxx	57, 219
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								xxxx xxxx	57, 219
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	0000 0000	57, 219
ECCP3AS	ECCP3ASE	ECCP3AS2	ECCP3AS1	ECCP3AS0	PSS3AC1	PSS3AC0	PSS3BD1	PSS3BD0	0000 0000	57, 219
ECCP3DEL	P3RSEN	P3DC6	P3DC5	P3DC4	P3DC3	P3DC2	P3DC1	P3DC0	0000 0000	57, 219
CCPR3H	Capture/Compare/PWM Register 1 High Byte								xxxx xxxx	57, 219
CCPR3L	Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	57, 219
CCP3CON	P3M1	P3M0	DC3B1	DC3B0	CCP3M3	CCP3M2	CCP3M1	CCP3M0	0000 0000	57, 219
SPBRG1	EUSART1 Baud Rate Generator Register Low Byte								0000 0000	57, 273
RCREG1	EUSART1 Receive Register								0000 0000	57, 281, 282
TXREG1	EUSART1 Transmit Register								xxxx xxxx	57, 279, 280
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDDB	BRGH	TRMT	TX9D	0000 0010	57, 279
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	57, 281
SPBRG2	EUSART2 Baud Rate Generator Register Low Byte								0000 0000	57, 273

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:
RCREG2	EUSART2 Receive Register								0000 0000	57, 281, 282
TXREG2	EUSART2 Transmit Register								0000 0000	57, 279, 280
TXSTA2	CSRC	TX9	TXEN	SYNC	SEND _B	BRGH	TRMT	TX9D	0000 0010	57, 279
EECON2	Program Memory Control Register 2 (not a physical register)								---- ----	57, 88
EECON1	—	—	WPROG	FREE	WRERR	WREN	WR	—	--00 x00-	57, 88
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	1111 1111	58, 122
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	0000 0000	58, 116
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	0000 0000	58, 119
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	111- 1111	58, 122
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	000- 0000	58, 116
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	000- 0000	58, 119
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	1111 1111	58, 122
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	58, 116
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	58, 119
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	58, 281
OSCTUNE	INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000 0000	58, 33
TRISJ ⁽⁷⁾	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	1111 1111	58, 150
TRISH ⁽⁷⁾	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	1111 1111	58, 148
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	---1 1111	58, 146
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	1111 111-	58, 144
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	1111 1111	58, 141
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	58, 138
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISCO	1111 1111	58, 136
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	58, 134
TRISA	TRISA7 ⁽⁸⁾	TRISA6 ⁽⁸⁾	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	58, 132
LATJ ⁽⁷⁾	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	xxxx xxxx	58, 150
LATH ⁽⁷⁾	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	xxxx xxxx	58, 148
LATG	—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0	---x xxxx	58, 146
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	xxxx xxxx-	58, 144
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx xxxx	58, 141
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxx	58, 138
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	58, 136
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx	58, 134
LATA	LATA7 ⁽⁸⁾	LATA6 ⁽⁸⁾	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx xxxx	58, 132
PORTJ ⁽⁷⁾	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	xxxx xxxx	59, 150
PORTH ⁽⁷⁾	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	0000 xxxx	59, 148
PORTG	RDPU	REPU	RJPU ⁽⁷⁾	RG4	RG3	RG2	RG1	RG0	000x xxxx	59, 146
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	x000 000-	59, 144
PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	xxxx xxxx	59, 141
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	59, 138
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	59, 136
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	59, 134
PORTA	RA7 ⁽⁸⁾	RA6 ⁽⁸⁾	RA5	RA4	RA3	RA2	RA1	RA0	000x 0000	59, 132
SPBRGH1	EUSART1 Baud Rate Generator Register High Byte								0000 0000	59, 273
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	0100 0-00	59, 273
SPBRGH2	EUSART2 Baud Rate Generator Register High Byte								0000 0000	59, 273
BAUDCON2	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	0100 0-00	59, 273
TMR3H	Timer3 Register High Byte								xxxx xxxx	59, 194
TMR3L	Timer3 Register Low Byte								xxxx xxxx	59, 194

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	0000 0000	59, 194
TMR4	Timer4 Register								0000 0000	59, 193
PR4 ⁽²⁾ /	Timer4 Period Register								1111 1111	59, 194
CVRCON ⁽³⁾	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	59, 310
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	-000 0000	59, 193
CCPR4H	Capture/Compare/PWM Register 4 High Byte								xxxxx xxxx	59, 196
CCPR4L	Capture/Compare/PWM Register 4 Low Byte								xxxxx xxxx	59, 196
CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	--00 0000	59, 196
CCPR5H	Capture/Compare/PWM Register 5 High Byte								xxxxx xxxx	59, 196
CCPR5L	Capture/Compare/PWM Register 5 Low Byte								xxxxx xxxx	59, 196
CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	--00 0000	59, 196
SSP2BUF	MSSP2 Receive Buffer/Transmit Register								xxxxx xxxx	59, 222, 231
SSP2ADD/	MSSP2 Address Register (I^2C ™ Slave mode), MSSP2 Baud Rate Reload Register (I^2C Master mode)								0000 0000	59, 231
SSP2MSK ⁽⁵⁾	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	0000 0000	59, 238
SSP2STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	59, 222, 232
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	59, 223, 233
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN/	SEN	0000 0000	59, 234, 268
GCEN	ACKSTAT	ADMSK5 ⁽⁶⁾	ADMSK4 ⁽⁶⁾	ADMSK3 ⁽⁶⁾	ADMSK2 ⁽⁶⁾	ADMSK1 ⁽⁶⁾	SEN			
CMSTAT	—	—	—	—	—	—	COUT2	COUT1	---- --11	59, 303
PMADDRH /	CS2	CS1	Parallel Master Port Address High Byte						0000 0000	60, 158
PMDOUT1H ⁽⁹⁾	Parallel Port Out Data High Byte (Buffer 1)								0000 0000	60, 161
PMADDRL /	Parallel Master Port Address Low Byte								0000 0000	60, 158
PMDOUT1L ⁽⁹⁾	Parallel Port Out Data Low Byte (Buffer 0)								0000 0000	60, 158
PMDIN1H	Parallel Port In Data High Byte (Buffer 1)								0000 0000	60, 158
PMDIN1L	Parallel Port In Data Low Byte (Buffer 0)								0000 0000	60, 158
PMCONH	PMPPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN	0-00 0000	60, 152
PMCONL	CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP	0000 0000	60, 153
PMMODEH	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0	0000 0000	60, 154
PMMODEL	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	0000 0000	60, 155
PMDOUT2H	Parallel Port Out Data High Byte (Buffer 3)								0000 0000	60, 158
PMDOUT2L	Parallel Port Out Data Low Byte (Buffer 2)								0000 0000	60, 158
PMDIN2H	Parallel Port In Data High Byte (Buffer 3)								0000 0000	60, 158
PMDIN2L	Parallel Port In Data Low Byte (Buffer 2)								0000 0000	60, 158
PMEH	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	0000 0000	60, 155
PMEL	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	0000 0000	60, 156
PMSTATH	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	00-- 0000	60, 156
PMSTATL	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	10-- 1111	60, 157

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. **Bold** indicates shared access SFRs.

Les modes d'adressage :

Ils permettent de définir le moyen d'accéder à la donnée située ou non en mémoire.

- Le plus simple est l'**adressage inhérent** (ou littéral)

Il est aussi appelé chez d'autres constructeur adressage Immédiat. La donnée est encodée dans l'instruction et donc ne peut plus être modifiée pendant l'exécution. On le réserve donc aux données constantes.

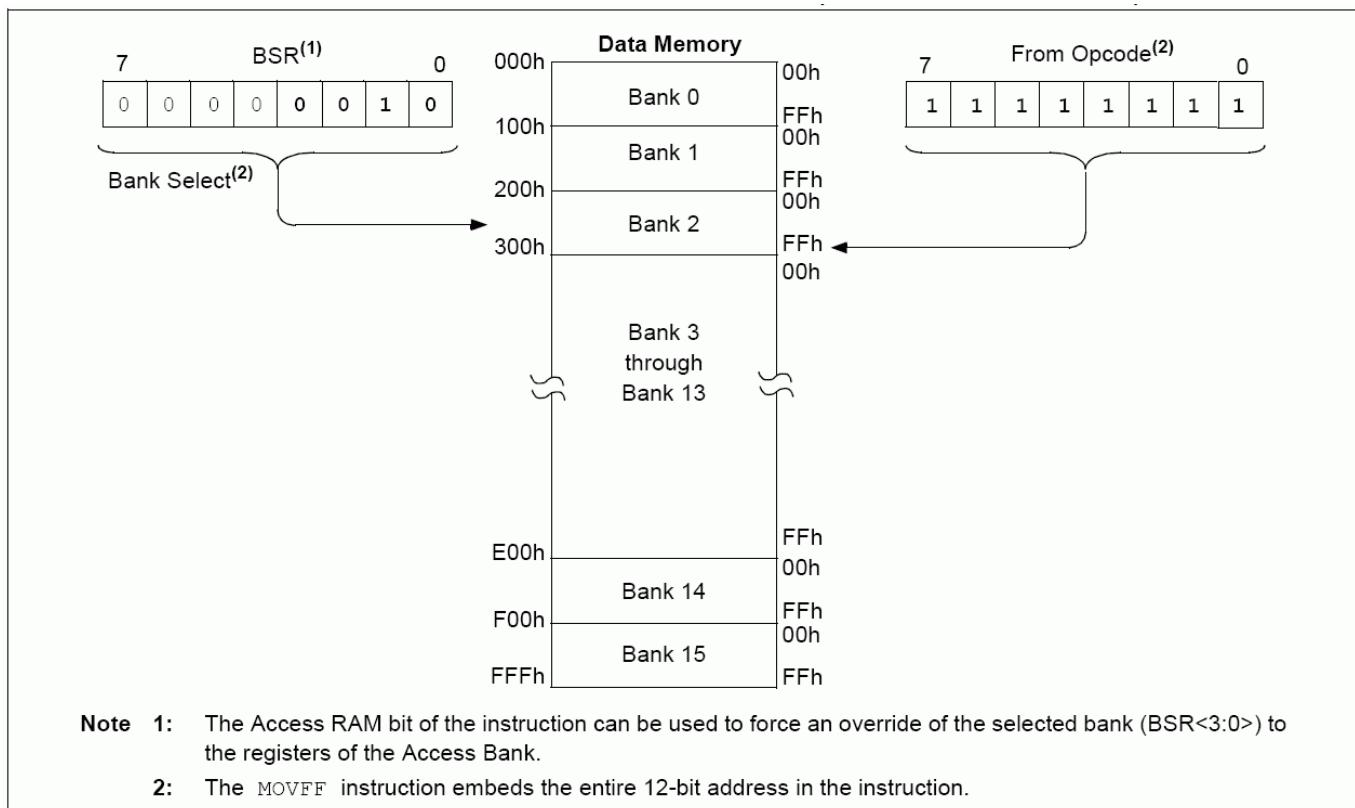
Exemple : **MOVLW k** qui place le « littéral k » dans le registre W
Codé **0000 0001 kkkk kkkk**

- L'adressage direct court ou long**

Cette fois ci l'instruction contient l'adresse de la donnée. On le réserve donc aux données élémentaires.

Exemple : **MOVF var, W** qui place la donnée « var » d'adresse « f » dans le registre W
Codé **0101 0001 ffff ffff**

On remarquera que l'adresse est sur 8 bits ; elle correspond en réalité à la partie basse de l'adresse dans la mémoire paginée (revoir plus haut)



Dans certains cas on peut utiliser ce mode d'adressage pour travailler sur toute la zone mémoire sans passer par la pagination (**mode direct long**)

Exemple : **MOVFF varS, varD** qui place la donnée « varS » d'adresse « s » à l'adresse « d » représentant « varD »

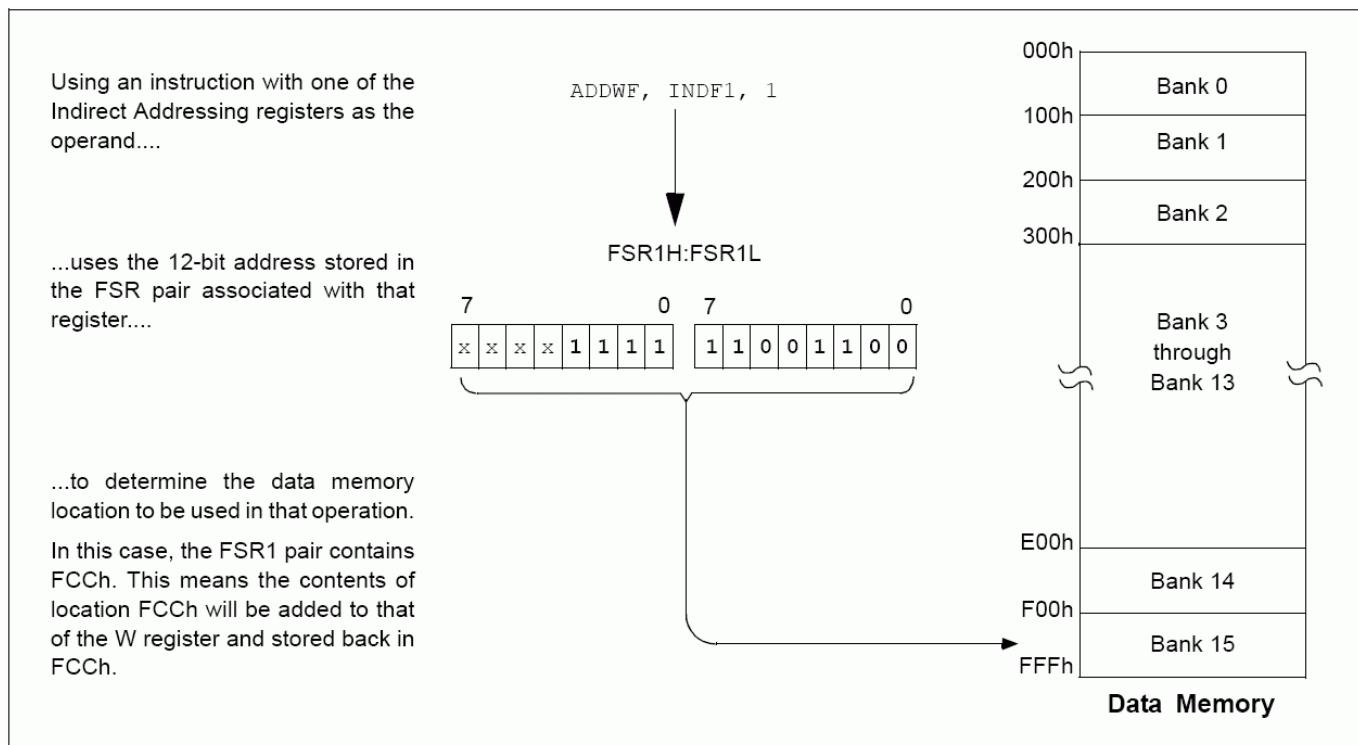
Codé **1100 ssss ssss ssss 1111 dddd dddd dddd**

On remarquera que les adresses sont sur 12 bits, que cette fois ci la longueur de l'instruction est de 32 bits et que la durée d'exécution sera de 2 cycles machines.

- **L'adressage indirect**

Cette fois ci traiter une donnée dont l'adresse est contenue dans un registre d'indexation INDFx. L'instruction ne connaît donc que ce registre. On le réserve pour parcourir les tableaux et autres zones mémoire.

Exemple : **MOVF INDF1, W** qui place la donnée d'adresse « FSR1 » (registre qui contient l'adresse) dans le registre W



Pour parcourir un ensemble de données on peut auto-incrémente ou auto-décrémenter le registre d'index.

Exemple : **MOVF POSTDEC1, W**

Dans ce cas l'instruction (en plus) décrémente « FSR1 » après l'affectation de W.

Pour parcourir un ensemble de données en exploitant des valeurs indicées comme tab(i) on peut recourir à l'instruction suivante qui calculera l'@ par la somme de FSR1 et W
FSR1 : @debut (tab) + W : indice (i)

Exemple : **MOVFF PLUSW1, var** fait : var = tab(i)

- **L'adressage relatif** est réservé aux instructions de sauts et de branchements.

Dans ce cas on ne donne pas l'adresse d'arrivée du branchement mais le nombre d'octets séparant l'endroit où on se trouve de celui où on veut aller.

LE JEU D'INSTRUCTIONS ASSEMBLEUR

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word		Status Affected	Notes
			MSb	Lsb		
BYTE-ORIENTED OPERATIONS						
ADDWF f, d, a	Add WREG and f	1	0010 01da ffff ffff	C, DC, Z, OV, N	1, 2	
ADDWFC f, d, a	Add WREG and Carry bit to f	1	0010 00da ffff ffff	C, DC, Z, OV, N	1, 2	
ANDWF f, d, a	AND WREG with f	1	0001 01da ffff ffff	Z, N	1, 2	
CLRF f, a	Clear f	1	0110 101a ffff ffff	Z	2	
COMF f, d, a	Complement f	1	0001 11da ffff ffff	Z, N	1, 2	
CPFSEQ f, a	Compare f with WREG, skip =	1 (2 or 3)	0110 001a ffff ffff	None	4	
CPFSGT f, a	Compare f with WREG, skip >	1 (2 or 3)	0110 010a ffff ffff	None	4	
CPFSLT f, a	Compare f with WREG, skip <	1 (2 or 3)	0110 000a ffff ffff	None	1, 2	
DECf f, d, a	Decrement f	1	0000 01da ffff ffff	C, DC, Z, OV, N	1, 2, 3, 4	
DECFSZ f, d, a	Decrement f, Skip if 0	1 (2 or 3)	0010 11da ffff ffff	None	1, 2, 3, 4	
DCFSNZ f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	0100 11da ffff ffff	None	1, 2	
INCf f, d, a	Increment f	1	0010 10da ffff ffff	C, DC, Z, OV, N	1, 2, 3, 4	
INCFSZ f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011 11da ffff ffff	None	4	
INFSNZ f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100 10da ffff ffff	None	1, 2	
IORWF f, d, a	Inclusive OR WREG with f	1	0001 00da ffff ffff	Z, N	1, 2	
MOVE f, d, a	Move f	1	0101 00da ffff ffff	Z, N	1	
MOVFF f _s , f _d	Move f _s (source) to 1st word f _d (destination) 2nd word	2	1100 ffff ffff ffff	None		
MOVWF f, a	Move WREG to f	1	0110 111a ffff ffff	None		
MULWF f, a	Multiply WREG with f	1	0000 001a ffff ffff	None	1, 2	
NEGF f, a	Negate f	1	0110 110a ffff ffff	C, DC, Z, OV, N		
RLCF f, d, a	Rotate Left f through Carry	1	0011 01da ffff ffff	C, Z, N	1, 2	
RLCNF f, d, a	Rotate Left f (No Carry)	1	0100 01da ffff ffff	Z, N		
RRCF f, d, a	Rotate Right f through Carry	1	0011 00da ffff ffff	C, Z, N		
RRNCF f, d, a	Rotate Right f (No Carry)	1	0100 00da ffff ffff	Z, N		
SETF f, a	Set f	1	0110 100a ffff ffff	None	1, 2	
SUBFWB f, d, a	Subtract f from WREG with borrow	1	0101 01da ffff ffff	C, DC, Z, OV, N		
SUBWF f, d, a	Subtract WREG from f	1	0101 11da ffff ffff	C, DC, Z, OV, N	1, 2	
SUBWFB f, d, a	Subtract WREG from f with borrow	1	0101 10da ffff ffff	C, DC, Z, OV, N		
SWAPF f, d, a	Swap nibbles in f	1	0011 10da ffff ffff	None	4	
TSTFSZ f, a	Test f, skip if 0	1 (2 or 3)	0110 011a ffff ffff	None	1, 2	
XORWF f, d, a	Exclusive OR WREG with f	1	0001 10da ffff ffff	Z, N		
BIT-ORIENTED OPERATIONS						
BCF f, b, a	Bit Clear f	1	1001 bbba ffff ffff	None	1, 2	
BSF f, b, a	Bit Set f	1	1000 bbba ffff ffff	None	1, 2	
BTFSC f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011 bbba ffff ffff	None	3, 4	
BTFSS f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010 bbba ffff ffff	None	3, 4	
BTG f, d, a	Bit Toggle f	1	0111 bbba ffff ffff	None	1, 2	

Notes : d → Direction =0 résultat vers WREG =1 résultat dans registre
 a → Access RAM =0 zone Access Bank =1 zone Data Memory
 bbb → Bit Adress exprime le rang du bit (0 – 7)

f, k ou n → @file, constant, @relative

écriture d'un nombre binaire

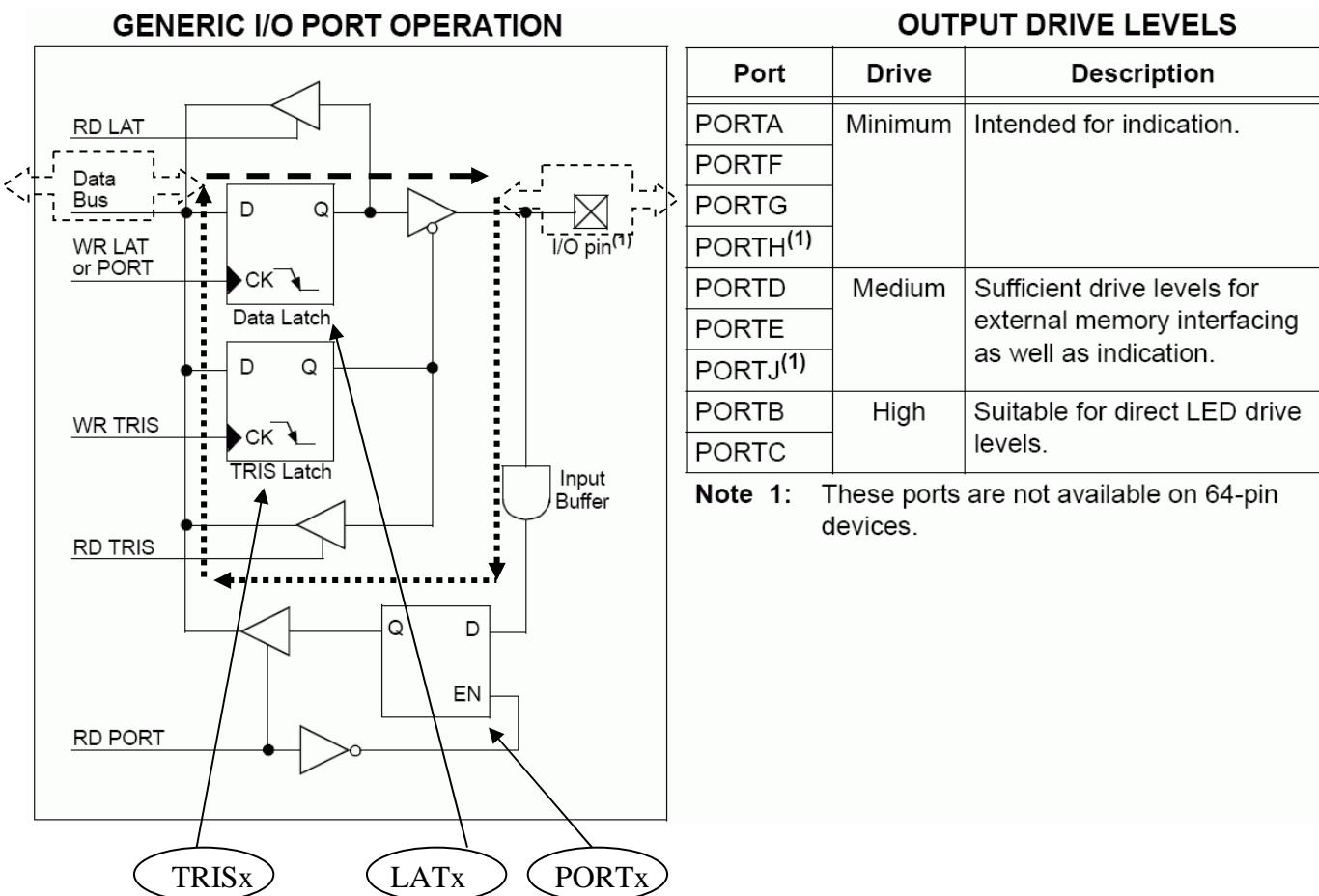
Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word		Status Affected	Notes
			MSb	Lsb		
CONTROL OPERATIONS						
BC n	Branch if Carry	1 (2)	1110 0010 nnnn nnnn	nnnn	None	
BN n	Branch if Negative	1 (2)	1110 0110 nnnn nnnn	nnnn	None	
BNC n	Branch if Not Carry	1 (2)	1110 0011 nnnn nnnn	nnnn	None	
BNN n	Branch if Not Negative	1 (2)	1110 0111 nnnn nnnn	nnnn	None	
BNOV n	Branch if Not Overflow	1 (2)	1110 0101 nnnn nnnn	nnnn	None	
BNZ n	Branch if Not Zero	1 (2)	1110 0001 nnnn nnnn	nnnn	None	
BOV n	Branch if Overflow	1 (2)	1110 0100 nnnn nnnn	nnnn	None	
BRA n	Branch Unconditionally	2	1101 0nnn nnnn nnnn	nnnn	None	
BZ n	Branch if Zero	1 (2)	1110 0000 nnnn nnnn	nnnn	None	
CALL n, s	Call subroutine 1st word 2nd word	2	1110 110s kkkk kkkk 1111 kkkk kkkk kkkk	kkkk	None	
CLRWDAT —	Clear Watchdog Timer	1	0000 0000 0000 0100	TO, PD		
DAW —	Decimal Adjust WREG	1	0000 0000 0000 0111	C		
GOTO n	Go to address 1st word 2nd word	2	1110 1111 kkkk kkkk 1111 kkkk kkkk kkkk	kkkk	None	
NOP —	No Operation	1	0000 0000 0000 0000	None		
NOP —	No Operation	1	1111 xxxx xxxx xxxx	xxxx	None	4
POP —	Pop top of return stack (TOS)	1	0000 0000 0000 0110	None		
PUSH —	Push top of return stack (TOS)	1	0000 0000 0000 0101	None		
RCALL n	Relative Call	2	1101 1nnn nnnn nnnn	nnnn	None	
RESET	Software device Reset	1	0000 0000 1111 1111	1111	All	
RETIE f	Return from interrupt enable	2	0000 0000 0001 000s	000s	GIE/GIEH, PEIE/GIEL	
RETLW k	Return with literal in WREG	2	0000 1100 kkkk kkkk	kkkk	None	
RETURN s	Return from Subroutine	2	0000 0000 0001 001s	001s	None	
SLEEP —	Go into Standby mode	1	0000 0000 0000 0011	0011	TO, PD	
LITERAL OPERATIONS						
ADDLW k	Add literal and WREG	1	0000 1111 kkkk kkkk	kkkk	C, DC, Z, OV, N	
ANDLW k	AND literal with WREG	1	0000 1011 kkkk kkkk	kkkk	Z, N	
IORLW k	Inclusive OR literal with WREG	1	0000 1001 kkkk kkkk	kkkk	Z, N	
LFSR f, k	Move literal (12-bit) 2nd word to FSR(f) 1st word	2	1110 1110 00ff kkkk 1111 0000 kkkk kkkk	kkkk	None	
MOVLB k	Move literal to BSR<3:0>	1	0000 0001 0000 kkkk	kkkk	None	
MOVLPW k	Move literal to WREG	1	0000 1110 kkkk kkkk	kkkk	None	
MULLW k	Multiply literal with WREG	1	0000 1101 kkkk kkkk	kkkk	None	
RETLW k	Return with literal in WREG	2	0000 1100 kkkk kkkk	kkkk	None	
SUBLW k	Subtract WREG from literal	1	0000 1000 kkkk kkkk	kkkk	C, DC, Z, OV, N	
XORLW k	Exclusive OR literal with WREG	1	0000 1010 kkkk kkkk	kkkk	Z, N	
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS						
TBLRD* —	Table Read	2	0000 0000 0000 1000	1000	None	
TBLRD*+	Table Read with post-increment		0000 0000 0000 1001	1001	None	
TBLRD*-	Table Read with post-decrement		0000 0000 0000 1010	1010	None	
TBLRD*+ —	Table Read with pre-increment		0000 0000 0000 1011	1011	None	
TBLWT* —	Table Write	2	0000 0000 0000 1100	1100	None	
TBLWT*+ —	Table Write with post-increment		0000 0000 0000 1101	1101	None	
TBLWT*-	Table Write with post-decrement		0000 0000 0000 1110	1110	None	
TBLWT*+ —	Table Write with pre-Increment		0000 0000 0000 1111	1111	None	

Notes

- (1) Lorsqu'un registre PORT est modifié en fonction de lui-même (par exemple MOVF PORTB, 1, 0) la valeur utilisée est celle réellement présente sur les lignes de port elles-mêmes.
- (2) Si l'instruction est exécutée sur le registre TMRO et lorsque l'utilisation de d = 1 est possible, le prédiviseur est remis à zéro.
- (3) Si le PC est modifié ou si un test conditionnel est vrai, l'instruction nécessite deux cycles. Le second cycle est exécuté comme un NOP.
- (4) Certaines instructions occupent deux mots. Le second mot de ces instructions est exécuté comme un NOP tant que le premier mot de l'instruction récupère les informations contenues dans ses 16 bits. Ceci permet de s'assurer que tous les emplacements de la mémoire de programme contiennent un code machine valide.

Les ports d'entrées/sortie parallèles

Schéma général d'une entrée/sortie parallèle



Ils sont au nombre de 7 pour notre famille (A..G)

Port A-F 8 bits bidirectionnels parfois partiellement disponible pour cause de lignes multifonctions
 Port G 5 bits bidirectionnels

3 registres permettent le contrôle :

PORTx pour la donnée

LATx tampon de sortie

TRISx pour définir le sens de fonctionnement ‘0’ pour sortie (‘1’ pour entrée)

Exemple du port A :

SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTA	RA7 ⁽¹⁾	RA6 ⁽¹⁾	RA5	RA4	RA3	RA2	RA1	RA0	59
LATA	LATA7 ⁽¹⁾	LATA6 ⁽¹⁾	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	58
TRISA	TRISA7 ⁽¹⁾	TRISA6 ⁽¹⁾	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	58
ANCON0 ⁽²⁾	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	57

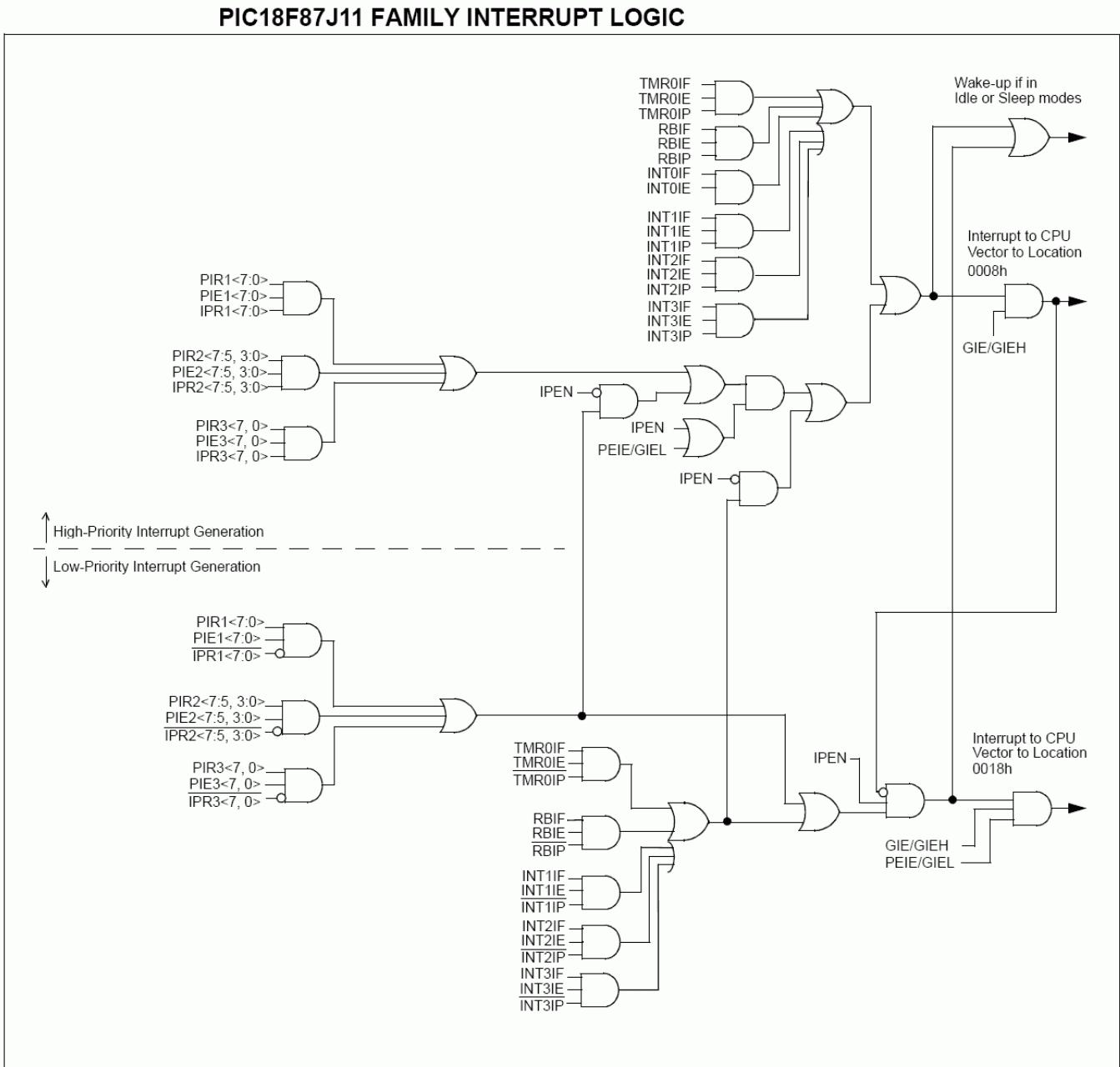
Legend: — = unimplemented, read as ‘0’. Shaded cells are not used by PORTA.

Note 1: Implemented only in specific oscillator modes (FOSC2 Configuration bit = 0); otherwise read as ‘0’.

2: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

Les sources d'interruptions sont nombreuses et toutes configurables indépendamment.

Schéma de principe des interruptions



Elles peuvent être hiérarchisées ou non (IPEN de RCON)

RCON: RESET CONTROL REGISTER

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	CM	RI	TO	PD	POR	BOR
bit 7	bit 0						

Le vecteur d'interruption est unique :

- @ 008 si non hiérarchisée
- @ 008 pour haute et @ 018 pour basse si hiérarchisée

il faut donc interroger INTCONx et PIRx pour déterminer la source d'interruption.

Chaque interruption doit être configurée INTCONx et PIRx, validée (dans PIEx), la priorité (dans IPRx).

INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF ⁽¹⁾
bit 7							bit 0

INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
bit 7							bit 0

INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
bit 7							bit 0

PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
MPPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MPPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MPPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
bit 7							bit 0

Même répartition pour PIRx, PIE1 et IPRx (x=2 et 3)

PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF
bit 7							bit 0

PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3

R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF
bit 7							bit 0

Le contrôle général des interruptions se fait par le bit GIE/GIEH (INTCON.7)

PIC18F87J11 FAMILY

PIC18F87J11 FAMILY

12.0 TIMER0 MODULE

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes
- Readable and writable registers
- Dedicated 8-bit, software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt-on-overflow

REGISTER 12-1: T0CON: TIMER0 CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit
 -n = Value at POR '1' = Bit is set U = Unimplemented bit, read as 'U'
 '0' = Bit is cleared x = Bit is unknown

The T0CON register (Register 12-1) controls all aspects of the module's operation, including the prescale selection. It is both readable and writable. A simplified block diagram of the Timer0 module in 8-bit mode is shown in Figure 12-1. Figure 12-2 shows a simplified block diagram of the Timer0 module in 16-bit mode.

12.1 Timer0 Operation

Timer0 can operate as either a timer or a counter. The mode is selected with the TOCS bit (T0CON<5>). In Timer mode (TOCS = 0), the module increments on every clock by default unless a different prescaler value is selected (see Section 12.3 "Prescaler"). If the TMRO register is written to, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMRO register.

The Counter mode is selected by setting the TOCS bit (= 1). In this mode, Timer0 increments either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (T0CON<4>); clearing this bit selects the rising edge. Restrictions on the external clock input are discussed below. An external clock source can be used to drive Timer0; however, it must meet certain requirements to ensure that the external clock can be synchronized with the

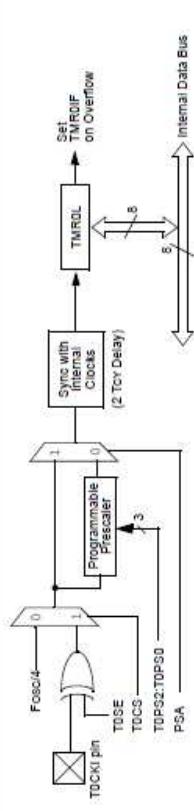
internal phase clock (TOSC). There is a delay between synchronization and the onset of incrementing the timer/counter.

12.2 Timer0 Reads and Writes in 16-Bit Mode

TMROH is not the actual high byte of Timer0 in 16-bit mode. It is actually a buffered version of the real high byte of Timer0 which is not directly readable nor writable (refer to Figure 12-2). TMROH is updated with the contents of the high byte of Timer0 during a read of TMROL. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte were valid, due to a rollover between successive reads of the high and low byte.

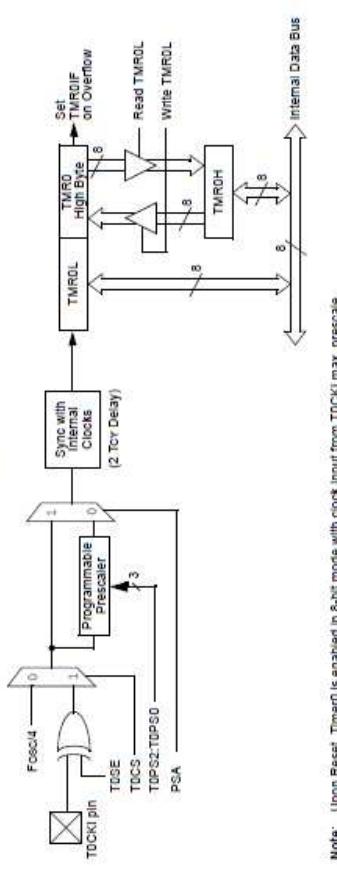
Similarly, a write to the high byte of Timer0 must also take place through the TMROh Buffer register. The high byte is updated with the contents of TMROH when a write occurs to TMROL. This allows all 16 bits of Timer0 to be updated at once.

FIGURE 12-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

FIGURE 12-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

PIC18F87J11 FAMILY

PIC18F87J11 FAMILY

13.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt on overflow
- Reset on ECCPx Special Event Trigger
- Device clock status flag (T1RUN)

A simplified block diagram of the Timer1 module is shown in Figure 13-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 13-2. The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation. Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead. Timer1 is controlled through the T1CON Control register (Register 13-1). It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

REGISTER 13-1: T1CON: TIMER1 CONTROL REGISTER(1)

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	bit 0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	
bit 7								

Legend:

R = Readable bit
 -n = Value at POR
 W = Writable bit
 '1' = Bit is set
 U = Unimplemented bit, read as '0'
 '0' = Bit is cleared
 x = Bit is unknown

bit 7 RD16: 16-Bit Read/Write Mode Enable bit

1 = Enables register read/write of Timer1 in one 16-bit operation
 0 = Enables register read/write of Timer1 in two 8-bit operations

T1RUN: Timer1 System Clock Status bit

1 = Device clock is derived from Timer1 oscillator
 0 = Device clock is derived from another source

T1CKPS1:T1CKPS0: Timer1 Input Clock Prescale Select bits

bit 5-4 1:1 = 1:8 Prescale value
 1:0 = 1:4 Prescale value
 0:1 = 1:2 Prescale value
 0:0 = 1:1 Prescale value

T1OSCEN: Timer1 Oscillator Enable bit

1 = Timer1 oscillator is enabled
 0 = Timer1 oscillator is shut off

The oscillator inverter and feedback resistor are turned off to eliminate power drain.

When TMR1CS = 1:
 T1SYNC: Timer1 External Clock Input Synchronization Select bit

bit 2 When TMR1CS = 0:
 1 = Do not synchronize external clock input
 0 = Synchronize external clock input

This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

TMR1CS: Timer1 Clock Source Select bit

bit 1 1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
 0 = Internal clock (FOSC/4)

TMR1ON: Timer1 On bit

1 = Enables Timer1

0 = Stops Timer1

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

13.1 Timer1 Operation

Timer1 can operate in one of these modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>). When TMR1CS is cleared (= 0), Timer1 increments on every internal instruction

FIGURE 13-1: TIMER1 BLOCK DIAGRAM

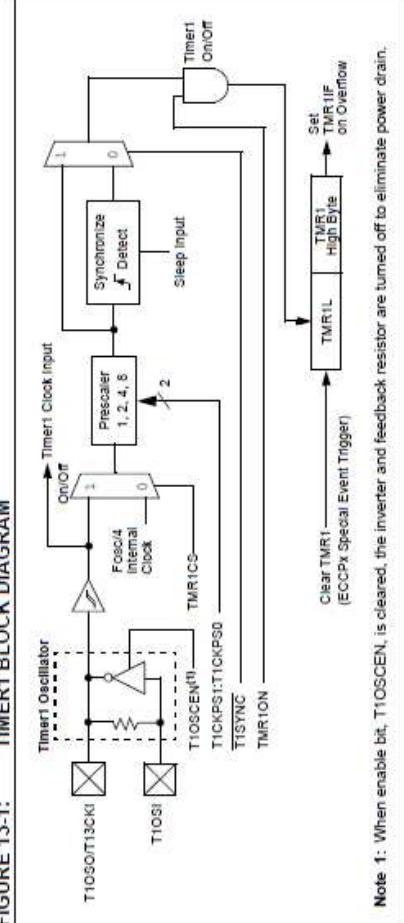
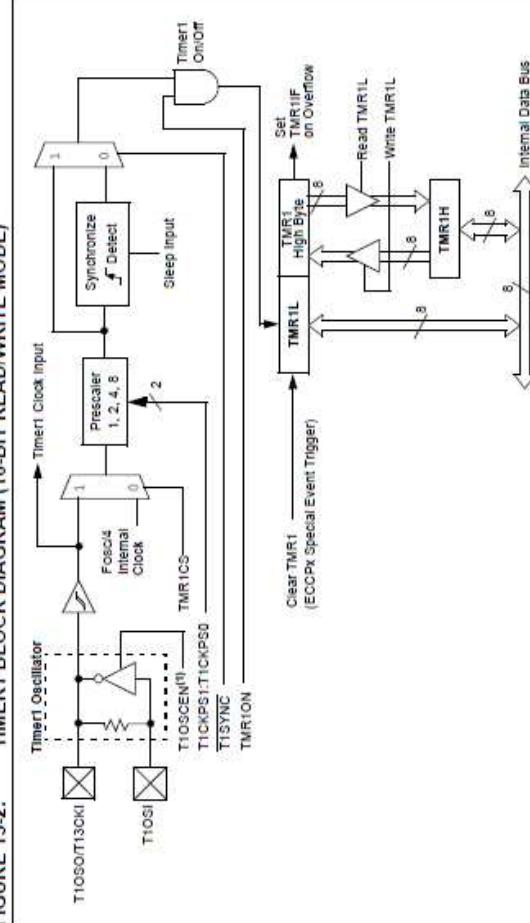


FIGURE 13-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



Note 1: When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

14.0 TIMER2 MODULE

The Timer2 module incorporates the following features:

- 8-Bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, and 1:16)
- Software programmable postscale (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP modules

In normal operation, TMR2 is incremented from 00h on each clock (FOSC4). A 4-bit counter/prescaler on the clock input gives direct input, divide-by-4, and divide-by-16 prescale options. These are selected by the prescaler control bits, T2CKPS1:T2CKPS0 (T2CON<1:0>). The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see Section 14.2 "Timer2 Interrupt").

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, while the PR2 register initializes at FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- any device Reset (Power-on Reset, MCLR Reset, Watchdog Timer Reset or Brown-out Reset)
- TMR2 is not cleared when T2CON is written.

REGISTER 14-1: T2CON: TIMER2 CONTROL REGISTER

bit 7	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2CKPS0	TMR2ON	T2CKPS1	T2CKPS0

Legend:

R = Readable bit
W = Writable bit
-n = Value at POR

U = Unimplemented bit, read as '0'
'1' = Bit is set
'0' = Bit is cleared

x = Bit is unknown

bit 7 Unimplemented: Read as '0'
bit 6-3 T2OUTPS3:T2OUTPS0: Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
0010 = 1:4 Postscale
0011 = 1:8 Postscale
0100 = 1:16 Postscale

•

1111 = 1:16 Postscale

TMR2ON: Timer2 On bit

1 = Timer2 is on

0 = Timer2 is off

T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

14.1 Timer2 Operation

The module is controlled through the T2CON register (Register 14-1) which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMF2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 14-1.

14.2 Timer2 Interrupt

The module is controlled through the T2CON register (Register 14-1) which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMF2ON (T2CON<2>), to minimize power consumption.

The module is controlled through the T2CON register (Register 14-1) which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMF2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 14-1.

14.3 Timer2 Output

The module is controlled through the T2CON register (Register 14-1) which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMF2ON (T2CON<2>), to minimize power consumption.

The module is controlled through the T2CON register (Register 14-1) which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMF2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 14-1.

TABLE 14-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMP1F	AD1F	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMP1IE	AD1IE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMP1IP	AD1IP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
TMR2 ⁽¹⁾	Timer2 Register								56
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2CKPS0	TMR2ON	T2CKPS1	T2CKPS0	56
PR2 ⁽¹⁾	Timer2 Period Register								56

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

PIC18F87J11 FAMILY

21.0 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has 11 inputs for the 64-pin devices and 15 for the 80-pin devices. This module allows conversion of an analog input signal to a corresponding 10-bit digital number.

The module has six registers:

- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

- A/D Port Configuration Register 2 (ANCON0)
- A/D Port Configuration Register 1 (ANCON1)
- A/D Result Registers (ADRESH and ADRESL)

The ADCON0 register, shown in Register 21-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 21-2, configures the A/D clock source, programmed acquisition time and justification.

REGISTER 21-1: ADCON0: A/D CONTROL REGISTER 0⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VCFG1	VCFG0	CHS3	CHS2	CHS1	GO/DONE
bit 7					bit 0

Legend:

- | | | |
|--------------------|------------------|------------------------------------|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared |
| x = Bit is unknown | | |
-
- | | |
|-------|------------------------------------|
| bit 7 | ADFM: A/D Result Format Select bit |
| | 1 = Right justified |
| | 0 = Left justified |
-
- | | |
|-------|---|
| bit 6 | ADCAL: A/D Calibration bit |
| | 1 = Calibration is performed on next A/D conversion |
| | 0 = Normal A/D Converter operation (no conversion is performed) |
-
- | | |
|---------|---|
| bit 5-3 | ACQ2:ACQ0: A/D Acquisition Time Select bits |
| | 111 = 20 TAD |
| | 110 = 16 TAD |
| | 101 = 12 TAD |
| | 100 = 8 TAD |
| | 011 = 6 TAD |
| | 010 = 4 TAD |
| | 001 = 2 TAD |
| | 000 = 0 TAD ⁽¹⁾ |
-
- | | |
|---------|---|
| bit 2-0 | ADCS2:ADCS0: A/D Conversion Clock Select bits |
| | 111 = FRC (clock derived from A/D RC oscillator) ⁽²⁾ |
| | 110 = FOSC/64 |
| | 101 = FOSC/16 |
| | 100 = FOSC/4 |
| | 011 = FRC (clock derived from A/D RC oscillator) ⁽²⁾ |
| | 010 = FOSC/32 |
| | 001 = FOSC/8 |
| | 000 = FOSC/2 |

- Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.
 2: If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

REGISTER 21-2: ADCON1: A/D CONTROL REGISTER 1⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCAL	ACQT2	ACQT1	ACQTO	ADCS2
bit 7					bit 0

REGISTER 21-1: ADCON0: A/D CONTROL REGISTER 0⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VCFG1	VCFG0	CHS3	CHS2	CHS1	GO/DONE
bit 7					bit 0

Legend:

- | | | |
|--------------------|------------------|------------------------------------|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared |
| x = Bit is unknown | | |
-
- | | |
|-----------------------------|---|
| bit 7-6 | VCFG1: Voltage Reference Configuration bit (V _{REF-} source) |
| 1 = V _{REF-} (AN2) | 0 = AVSS |
| bit | VCFG0: Voltage Reference Configuration bit (V _{REF+} source) |
| 1 = V _{REF+} (AN3) | 0 = AVDD |
-
- | | |
|--------------------------------|---------------------------------------|
| bit 5-2 | CHS3:CHS0: Analog Channel Select bits |
| 0000 = Channel 00 (AN0) | 0001 = Channel 01 (AN1) |
| 0010 = Channel 02 (AN2) | 0011 = Channel 03 (AN3) |
| 0100 = Channel 04 (AN4) | 0101 = Unused |
| 0110 = Channel 06 (AN6) | 0111 = Channel 07 (AN7) |
| 1000 = Channel 08 (AN8) | 1001 = Channel 09 (AN9) |
| 1010 = Channel 10 (AN10) | 1011 = Channel 11 (AN11) |
| 1100 = Channel 12 (AN12) (2,3) | 1101 = Channel 13 (AN13) (2,3) |
| 1110 = Channel 14 (AN14) (2,3) | 1111 = Channel 15 (AN15) (2,3) |
-
- | | |
|--------------------------------|------------------------------------|
| bit 1 | GO/DONE: A/D Conversion Status bit |
| When ADON = 1: | |
| 1 = A/D conversion in progress | |
| 0 = A/D idle | |

- Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.
 2: These channels are not implemented on 84-pin devices.
 3: Performing a conversion on unimplemented channels will return random values.

PIC18F87J11 FAMILY

PIC18F87J11 FAMILY

(Datasheet - page 294)

PIC18F87J11 FAMILY

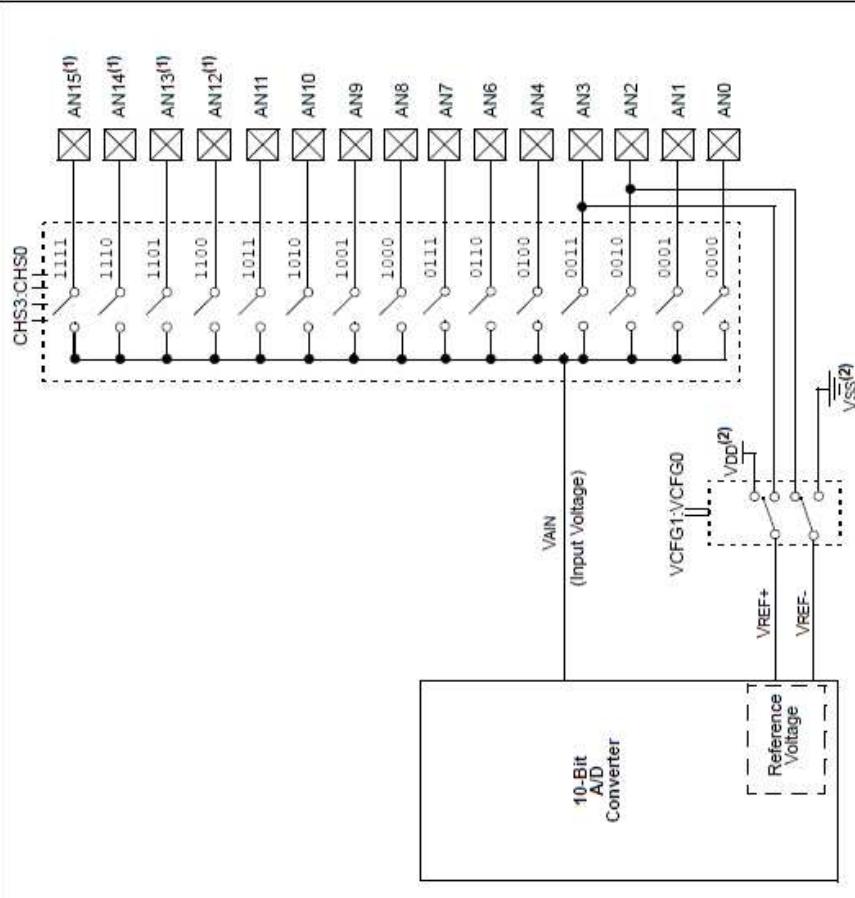
The analog reference voltage is software selectable to either the device's positive and negative supply voltage (V_{DD} and V_{SS}), or the voltage level on the RA3/AN3/V_{REF+} and RA2/AN2/V_{REF-} pins.

The A/D Converter has a unique feature of being able to operate while the device is in Sleep mode. To derive from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

Each port pin associated with the A/D Converter can be configured as an analog input or as a digital I/O. The ADRESH and ADRESL registers contain the result of

FIGURE 21-1: A/D BLOCK DIAGRAM



Note 1: Channels AN15 through AN12 are not available on 64-pin devices.
Note 2: I/O pins have diode protection to V_{DD} and V_{SS} .

the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH/ADRESL register pair, the GO/DONE bit (ADCON0<1>) is cleared and A/D Interrupt Flag bit, ADIF, is set.

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted. The value in the ADRESH/ADRESL register pair is not modified for a Power-on Reset. These registers will contain unknown data after a Power-on Reset.

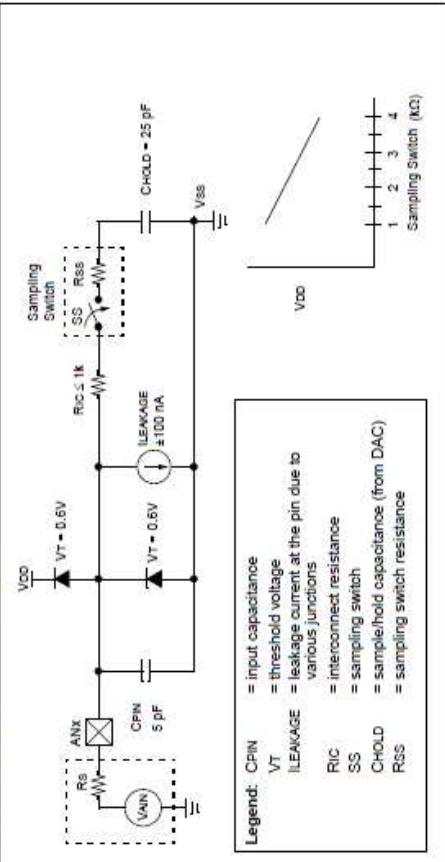
The block diagram of the A/D module is shown in Figure 21-1.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see Section 21.1 "A/D Acquisition Requirements". After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/DONE bit and the actual start of the conversion.

The following steps should be followed to do an A/D conversion:

- Configure the A/D module:
 - Configure required ADC pins as analog pins using ANCON0, ANCON1
 - Set voltage reference using ADCON0
 - Select A/D input channel (ADCON1)
 - Select A/D acquisition time (ADCON1)
 - Turn on A/D module (ADCON0)

FIGURE 21-2: ANALOG INPUT MODEL



PIC18F87J11 FAMILY

21.2 Selecting and Configuring Automatic Acquisition Time

The ADCON1 register allows the user to select an acquisition time that occurs each time the GO/DONE bit is set.

When the GO/DONE bit is set, sampling is stopped and a conversion begins. The user is responsible for ensuring the required acquisition time has passed between selecting the desired input channel and setting the GO/DONE bit. This occurs when the ACQT2:ACQT0 bits (ADCON1<5:3>) remain in their Reset state ('000') and is compatible with devices that do not offer programmable acquisition times.

If desired, the ACQT bits can be set to select a programmable acquisition time for the A/D module. When the GO/DONE bit is set, the A/D module continues to sample the input for the selected acquisition time, then automatically begins a conversion. Since the acquisition time is programmed, there may be no need to wait for an acquisition time between selecting a channel and setting the GO/DONE bit.

In either case, when the conversion is completed, the GO/DONE bit is cleared, the ADIF flag is set and the A/D begins sampling the currently selected channel again. If an acquisition time is programmed, there is nothing to indicate if the acquisition time has ended or if the conversion has begun.

21.3 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 11 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable.

There are seven possible options for TAD:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal RC Oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be as short as possible but greater than the minimum TAD (see parameter 130 in Table 27-30 for more information).

Table 21-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

TABLE 21-1: TAD VS. DEVICE OPERATING FREQUENCIES

AD Clock Source (TAD)	Maximum Device Frequency
ADC2:ADCS0	2.86 MHz
2 TOSC	0.00
4 TOSC	1.00
8 TOSC	0.01
16 TOSC	0.01
32 TOSC	0.10
64 TOSC	0.10
RC(2)	x:1

Note 1: The RC source has a typical TAD time of 4 μ s.

- 2: For device frequencies above 1 MHz, the device must be in Sleep mode for the entire conversion or the A/D accuracy may be out of specification.

21.4 Configuring Analog Port Pins

The ANCON0, ANCON1, TRISA, TRISF and TRISH registers control the operation of the A/D port pins. The port pins needed as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS3:CHS0 bits and the TRIS bits.

Note 1: When reading the PORT register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will be accurately converted.

2: Analog levels on any pin defined as a digital input may cause the digital input buffer to consume current out of the device's specification limits.

21.5 A/D Conversions

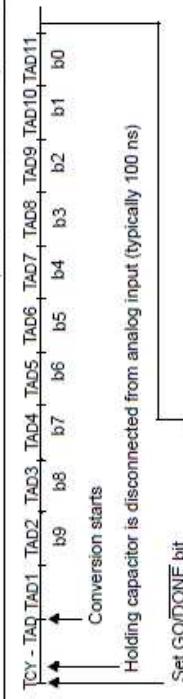
Figure 21-3 shows the operation of the A/D Converter after the GO/DONE bit has been set and the ACQT2:ACQT0 bits are cleared. A conversion is started after the following instruction to allow entry into Sleep mode before the conversion begins.

Figure 21-4 shows the operation of the A/D Converter after the GO/DONE bit has been set, the ACQT2:ACQT0 bits are set to '010' and selecting a 4 TAD acquisition time before the conversion starts. Clearing the GO/DONE bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. This means the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers).

After the A/D conversion is completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, acquisition on the selected channel is automatically started.

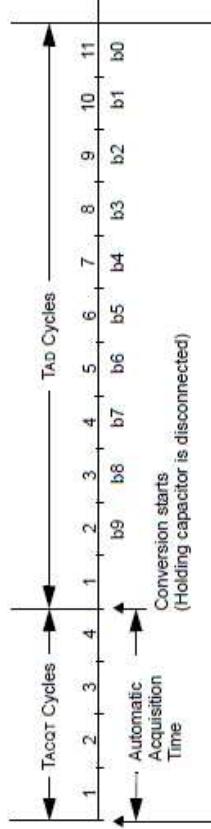
Note: The GO/DONE bit should NOT be set in the same instruction that turns on the A/D.

FIGURE 21-3: A/D CONVERSION TAD CYCLES (ACQT2:ACQT0 = 000, TACQ = 0)



Next Q4: ADRESH:ADRESL is loaded, GO/DONE bit is cleared, ADIF bit is set, holding capacitor is connected to analog input.

FIGURE 21-4: A/D CONVERSION TAD CYCLES (ACQT2:ACQT0 = 010, TACQ = 4 TAD)



Set GO/DONE bit (Holding capacitor continues acquiring input)
Next Q4: ADRESH:ADRESL is loaded, GO/DONE bit is cleared, ADIF bit is set, holding capacitor is disconnected (Conversion starts (Holding capacitor is disconnected))

FIGURE 21-1: TAD vs. Device Operating Frequencies

AD Clock Source (TAD)	Maximum Device Frequency
ADC2:ADCS0	2.86 MHz
2 TOSC	0.00
4 TOSC	0.10
8 TOSC	0.01
16 TOSC	0.01
32 TOSC	0.10
64 TOSC	0.10
RC(2)	x:1

PIC18F87J11 FAMILY

21.6 Use of the CCP2 Trigger

An A/D conversion can be started by the "Special Event Trigger" of the CCP2 module. This requires that the CCP2M3:CCP2M0 bits (CCP2CON<3:0>) be programmed as '1011' and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/DONE bit will be set starting the A/D acquisition and conversion, and the Timer1 (or Timer3) counter will be reset to zero. Timer1 (or Timer3) is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving ADRESH:ADRESL to the desired location). The appropriate analog input channel must be selected and the minimum acquisition period is either timed by the user, or an appropriate TACQ time is selected before the Special Event Trigger sets the GO/DONE bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), the Special Event Trigger will be ignored by the A/D module but will still reset the Timer1 (or Timer3) counter.

PIC18F87J11 FAMILY

20.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of two serial I/O modules. (Generically, the EUSART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN bus) systems.

All members of the PIC18F87J11 family are equipped with two independent EUSART modules, referred to as EUSART1 and EUSART2. They can be configured in the following modes:

- Asynchronous (full duplex) with:
 - Auto-wake-up on character reception
 - Auto-baud calibration
 - 12-bit Break character transmission
- Synchronous – Master (half duplex) with selectable clock polarity
- Synchronous – Slave (half duplex) with selectable clock polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions of PORTC (RC6/TX1/CK1 and RC7/RX1/DT1) and PORTG (RG1/TX2/CK2 and RG2/RX2/DT2), respectively. In order to configure these pins as an EUSART:

- For EUSART1:
 - bit SPEN (RCSTA1<7>) must be set (= 1)
 - bit TRISC<7> must be set (= 1)
 - bit TRISC<6> must be cleared (= 0) for Asynchronous and Synchronous Master modes
 - bit TRISC<6> must be set (= 1) for Synchronous Slave mode
- For EUSART2:
 - bit SPEN (RCSTA2<7>) must be set (= 1)
 - bit TRISG<2> must be set (= 1)
 - bit TRISG<1> must be cleared (= 0) for Asynchronous and Synchronous Master modes
 - bit TRISC<6> must be set (= 1) for Synchronous Slave mode

Note: The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of each Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These are detailed on the following pages in Register 20-1, Register 20-2 and Register 20-3, respectively.

Note: Throughout this section, references to register and bit names that may be associated with a specific EUSART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "RCSTAx" might refer to the Receive Status register for either EUSART1 or EUSART2.

PIC18F87J11 FAMILY

REGISTER 20-1: TXSTAX: TRANSMIT STATUS AND CONTROL REGISTER							
R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0	RC\$TAX: RECEIVE STATUS AND CONTROL REGISTER	
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							
							bit 0

Legend:
 R = Readable bit
 W = Writable bit
 U = Unimplemented bit, read as '0'
 '1' = Bit is cleared
 '0' = Bit is set
 x = Bit is unknown
 -n = Value at POR

bit 7	C\$SRC: Clock Source Select bit Asynchronous mode: Don't care.	bit 6	SPEI: Serial Port Enable bit Asynchronous mode: 1 = Serial port enabled (configures RXx/DTx and TXx/CLKx pins as serial port pins) 0 = Serial port disabled (held in Reset)	bit 5	TX9: 9-Bit Receive Enable bit Synchronous mode: 1 = Master mode (clock generated internally from BRG) 0 = Slave mode (clock from external source)	bit 4	bit 4	bit 3	bit 2	bit 1	bit 0
	Synchronous mode: 1 = Master mode (clock generated internally from BRG) 0 = Slave mode (clock from external source)		TX9: 9-Bit Receive Enable bit Synchronous mode: 1 = Selects 9-bit reception 0 = Selects 8-bit reception		TX9: 9-Bit Transmit Enable bit Synchronous mode: 1 = Selects 9-bit transmission 0 = Selects 8-bit transmission		TXEN: Transmit Enable bit(1) Synchronous mode: 1 = Transmit enabled 0 = Transmit disabled		SENDB: Send Break Character bit Synchronous mode: 1 = Sync Break on next transmission completed 0 = Sync Break transmission completed		TX9D: Transmit Shift Register Status bit Synchronous mode: 1 = High speed 0 = Low speed
	SYNC: USART Mode Select bit Asynchronous mode: 1 = Synchronous mode 0 = Asynchronous mode		SYNC: USART Mode Select bit Asynchronous mode: 1 = Send Sync Break on next transmission (cleared by hardware upon completion) 0 = Sync Break transmission completed		BRGH: High Baud Rate Select bit Asynchronous mode: 1 = High speed 0 = Low speed		TX9D: 9th bit of Transmit Data Synchronous mode: Unused in this mode.		TX9D: 9th bit of Transmit Data Can be address/data bit or a parity bit.		FERR: Framing Error bit Asynchronous mode 9-Bit (RX9 = 1): 1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
											OERR: Overrun Error bit Asynchronous mode 9-Bit (RX9 = 0): 1 = Overrun error (can be cleared by clearing bit CREN) 0 = No overrun error
											RX9D: 9th bit of Received Data This can be address/data bit or a parity bit and must be calculated by user firmware.

PIC18F87J11 FAMILY

REGISTER 20-2: RC\$TAX: RECEIVE STATUS AND CONTROL REGISTER							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-x
SPEN		RX9		SREN		OERR	
bit 7							

Legend:
 R = Readable bit
 W = Writable bit
 U = Unimplemented bit, read as '0'
 '1' = Bit is set
 '0' = Bit is cleared
 x = Bit is unknown
 -n = Value at POR

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SPEI: Serial Port Enable bit Asynchronous mode: 1 = Serial port enabled (configures RXx/DTx and TXx/CLKx pins as serial port pins) 0 = Serial port disabled (held in Reset)	TX9: 9-Bit Receive Enable bit Synchronous mode: 1 = Selects 9-bit reception 0 = Selects 8-bit reception	TX9: 9-Bit Transmit Enable bit Synchronous mode: 1 = Selects 9-bit transmission 0 = Selects 8-bit transmission	TXEN: Transmit Enable bit(1) Synchronous mode: 1 = Transmit enabled 0 = Transmit disabled	SENDB: Send Break Character bit Synchronous mode: 1 = Sync Break on next transmission completed 0 = Sync Break transmission completed	BRGH: High Baud Rate Select bit Asynchronous mode: 1 = High speed 0 = Low speed	TX9D: Transmit Shift Register Status bit Synchronous mode: 1 = High speed 0 = Low speed	FERR: Framing Error bit Asynchronous mode 9-Bit (RX9 = 1): 1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
							OERR: Overrun Error bit Asynchronous mode 9-Bit (RX9 = 0): 1 = Overrun error (can be cleared by clearing bit CREN) 0 = No overrun error
							RX9D: 9th bit of Received Data This can be address/data bit or a parity bit and must be calculated by user firmware.

REGISTER 20-3: BAUDCONx: BAUD RATE CONTROL REGISTER

	R-1	R/W-0	R/W-0	I/O-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE
bit 7						ABDEN

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set
U = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

bit 7 ABDOVF: Auto-Baud Acquisition Rollover Status bit

1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode
(must be cleared in software)

0 = No BRG rollover has occurred

bit 6 RCIDL: Receive Operation Idle Status bit

1 = Receive operation is Idle

0 = Receive operation is active

bit 5 RXDTP: Data/Receive Polarity Select bit

Asynchronous mode:

1 = Receive data (RXx) is inverted (active-low)

0 = Receive data (RXx) is not inverted (active-high)

Synchronous mode:

1 = Data (DTx) is inverted (active-low)

0 = Data (DTx) is not inverted (active-high)

bit 4 TXCKP: Synchronous Clock Polarity Select bit

Asynchronous mode:

1 = Idle state for transmit (TXx) is a low level

0 = Idle state for transmit (TXx) is a high level

Synchronous mode:

1 = Idle state for clock (CKx) is a high level

0 = Idle state for clock (CKx) is a low level

bit 3 BRG16: 16-Bit Baud Rate Register Enable bit

1 = 16-bit Baud Rate Generator – SPBRGHx and SPBRGx only (Compatible mode), SPBRGHx value ignored
0 = 8-bit Baud Rate Generator – SPBRGHx and SPBRGx only (Compatible mode), SPBRGHx value ignored

bit 2 Unimplemented: Read as '0'

bit 1 WUE: Wake-up Enable bit

Asynchronous mode:

1 = USART will continue to sample the RXx pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge

Synchronous mode:

Unused in this mode.

bit 0 ABDEN: Auto-Baud Detect Enable bit

Asynchronous mode:

1 = Enable baud rate measurement on the next character. Requires reception of a Sync field (55h); cleared in hardware upon completion.

Synchronous mode:

0 = Baud rate measurement disabled or completed
Unused in this mode.

20.1 Baud Rate Generator (BRG)

The SPBRGHx:SPBRGx register pair controls the period of a free-running timer. In Asynchronous mode, bits BRGH (TXSTA<2>) and BRG16 (BAUDCONx<3>) also control the baud rate. In Synchronous mode, BRGH is ignored. Table 20-1 shows the formula for computation of the baud rate for different USART modes which only apply in Master mode (internally generated clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRGHx:SPBRGx registers can be calculated using the formulas in Table 20-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 20-1. Typical baud rates and error values for the various Asynchronous modes are shown in Table 20-2. It may be advantageous to use present at the RXx pin.

TABLE 20-1: BAUD RATE FORMULAS

Configuration Bits			BRGEUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	Fosc/[64 (n + 1)]
0	0	1	8-bit/Asynchronous	Fosc/[16 (n + 1)]
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	
1	0	x	8-bit/Synchronous	Fosc/[4 (n + 1)]
1	1	x	16-bit/Synchronous	

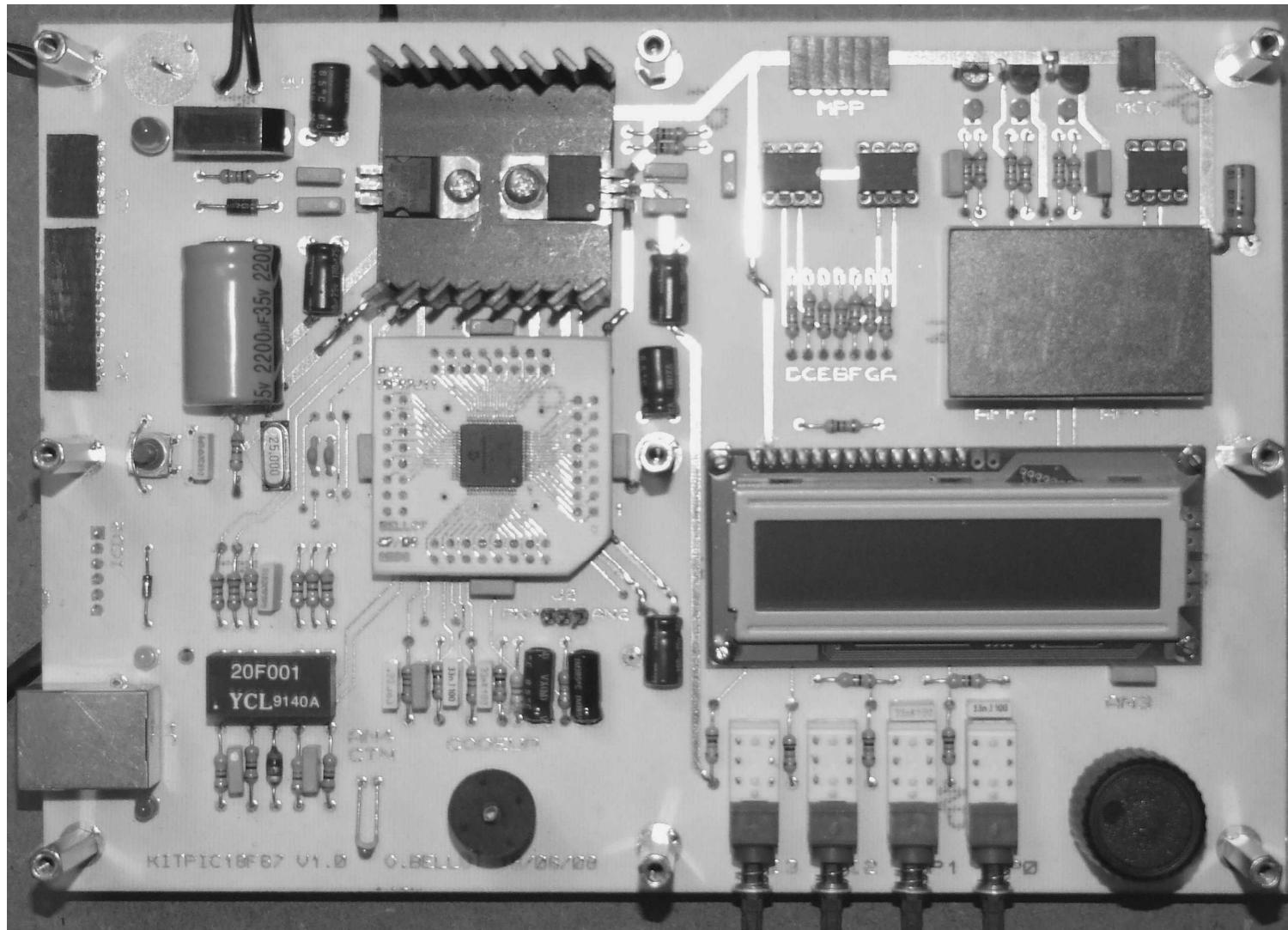
Legend: x = Don't care, n = value of SPBRGHx:SPBRGx register pair

TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2I

N° patte	NOM de patte	I / O	Actif 0 / 1	AFFECTATION KIT	FONCTION
				Affichage 7segments	
				Afficheur LCD	
				LEDs	
				Boutons	

N° patte	NOM de patte	I / O	Actif 0 / 1	AFFECTATION KIT	FONCTION
					Codeur
					CTN
					Potentiomètre
					Entrée Sortie ana.
					Interface M CC
					Interface M PàP
					BP RST
					Quartz
					Liaison série
					ICD2
					Alimentation

TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2I



localisez par un fléchage les différentes fonctions

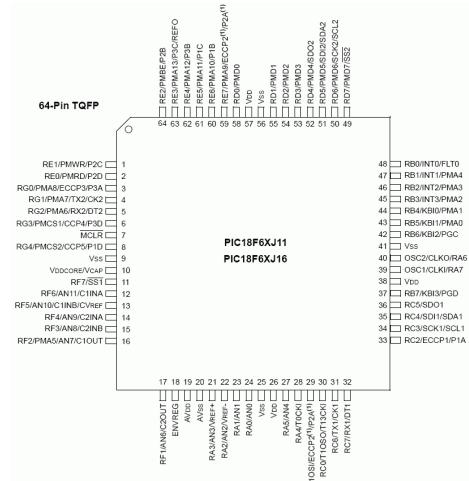
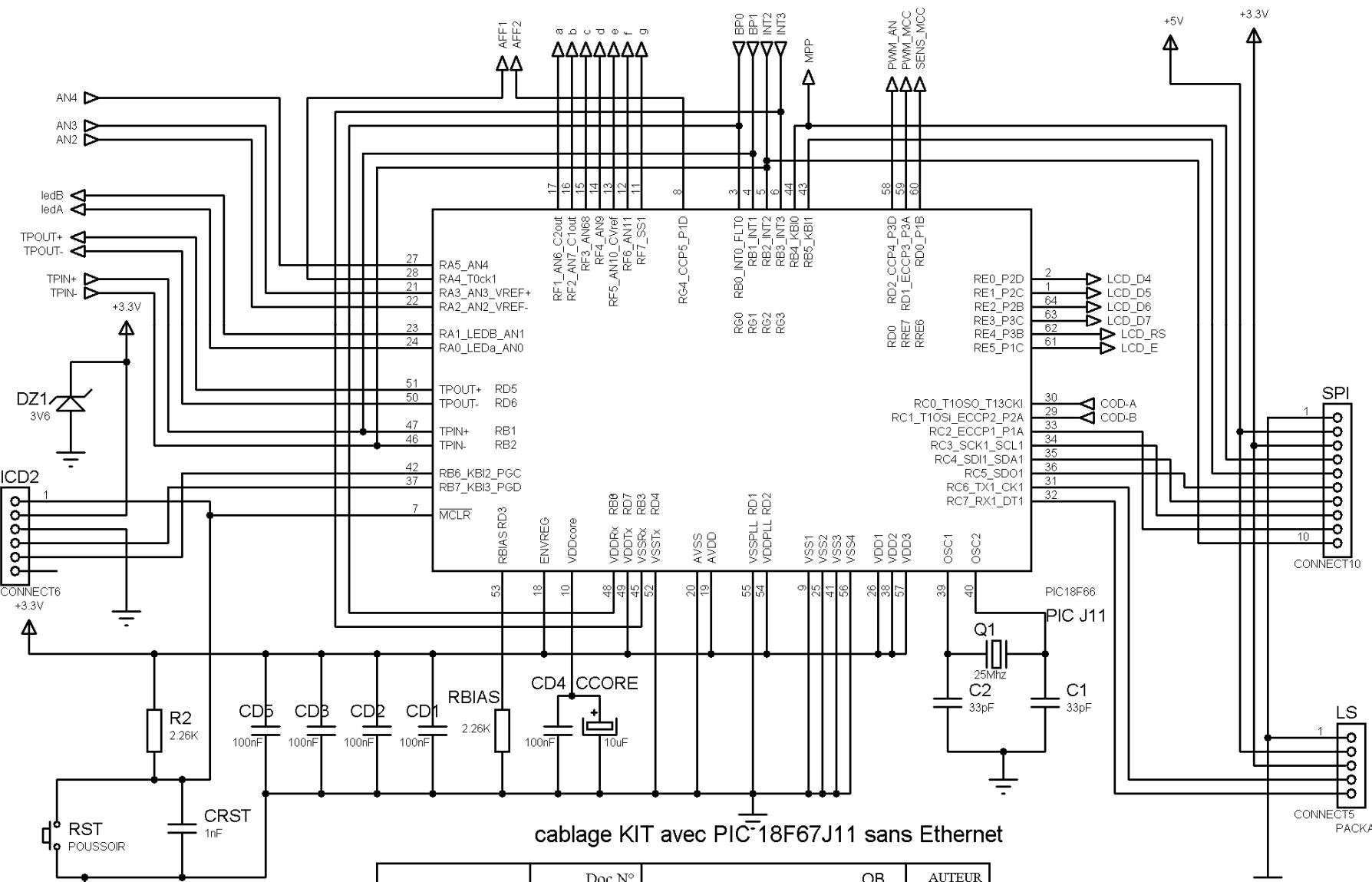


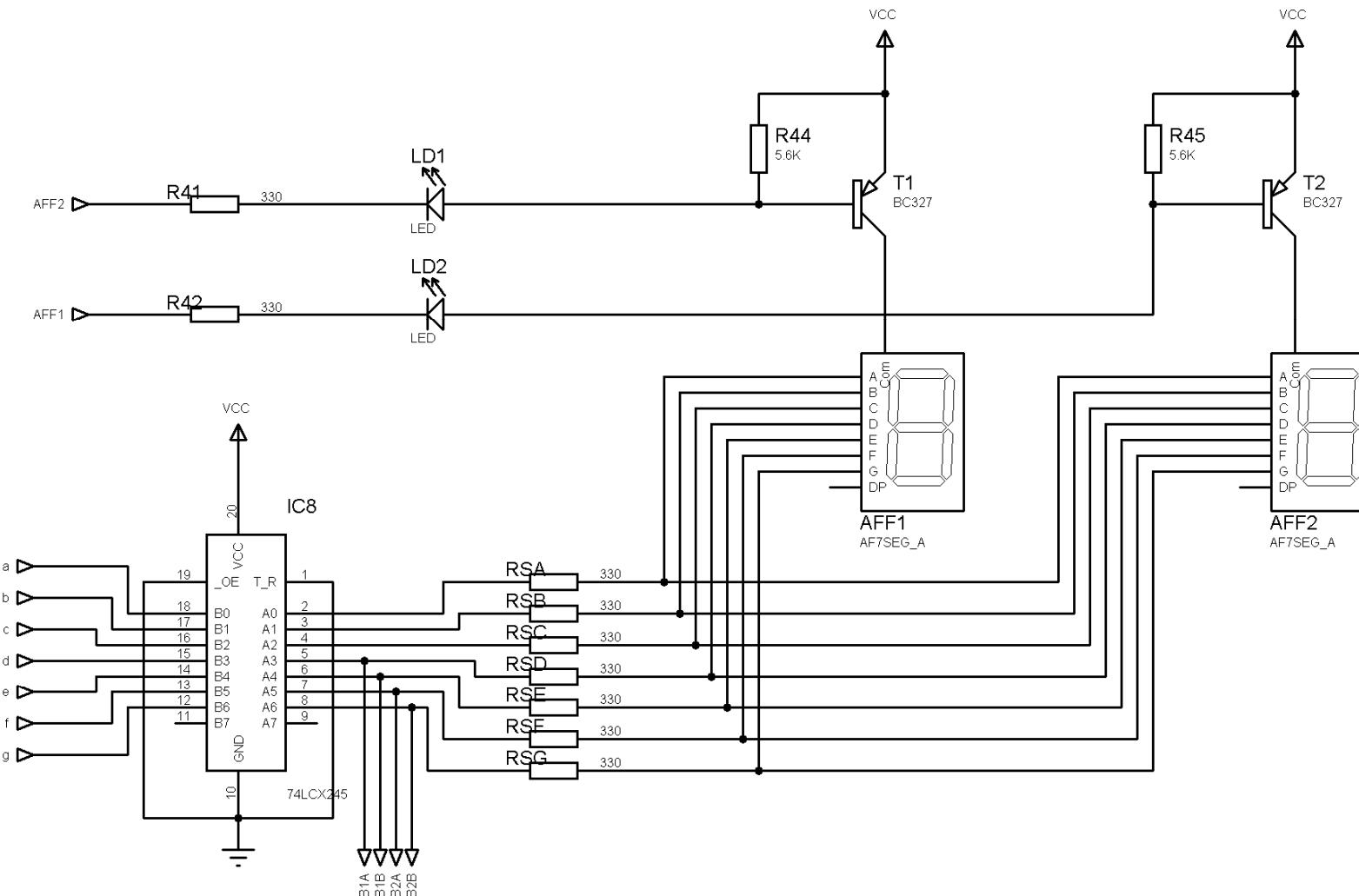
TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2I



cablage KIT avec PIC[®]18F67J11 sans Ethernet

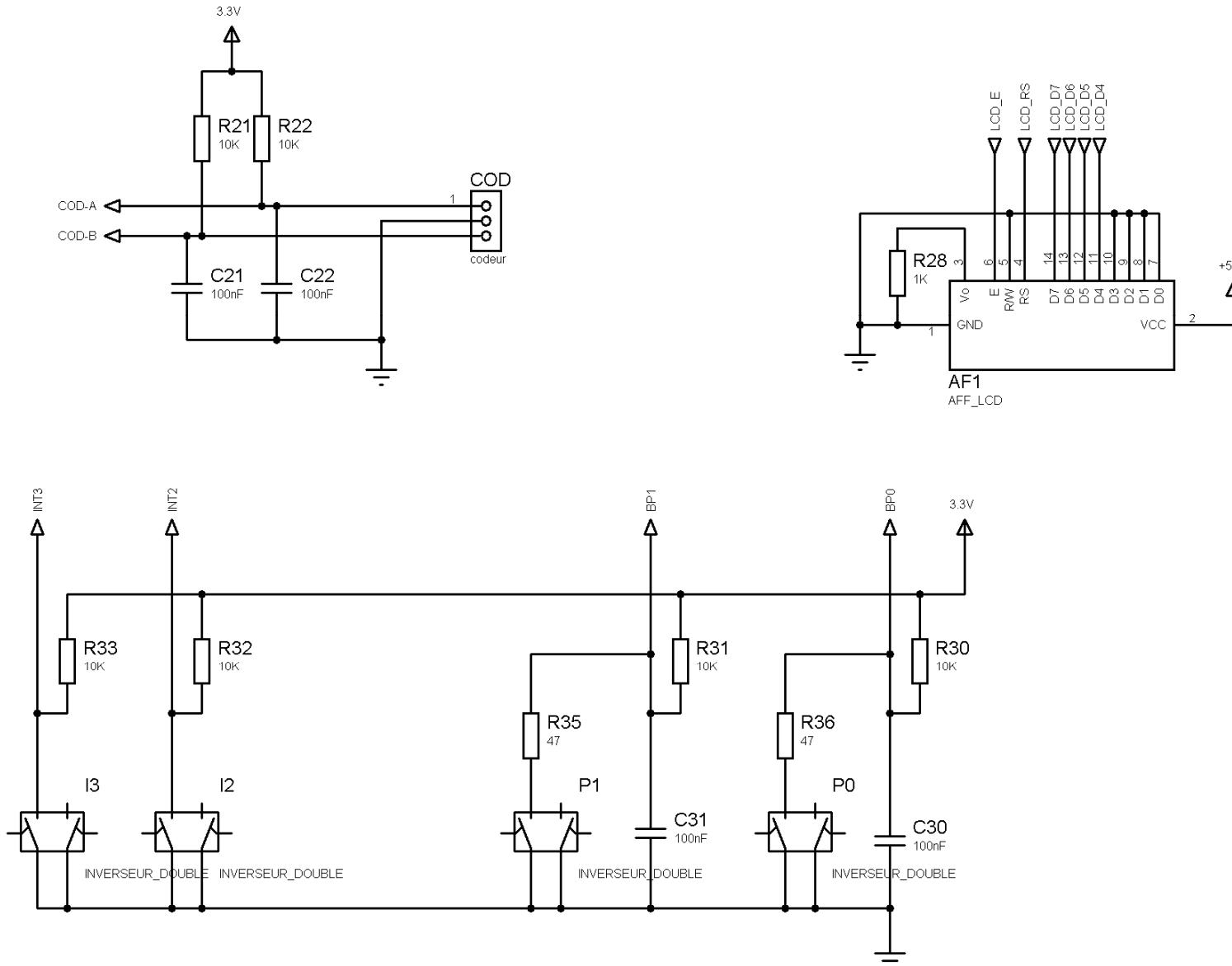
<i>GE2i</i> Nancy	Doc N°	OB		AUTEUR
	V20	Version Modification 23/03/09	Création 08/09/05	DATE
	processeur	kitpic18f67 kitpic18f67j/11v0.0.DSN	titre	TITRE

TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2I



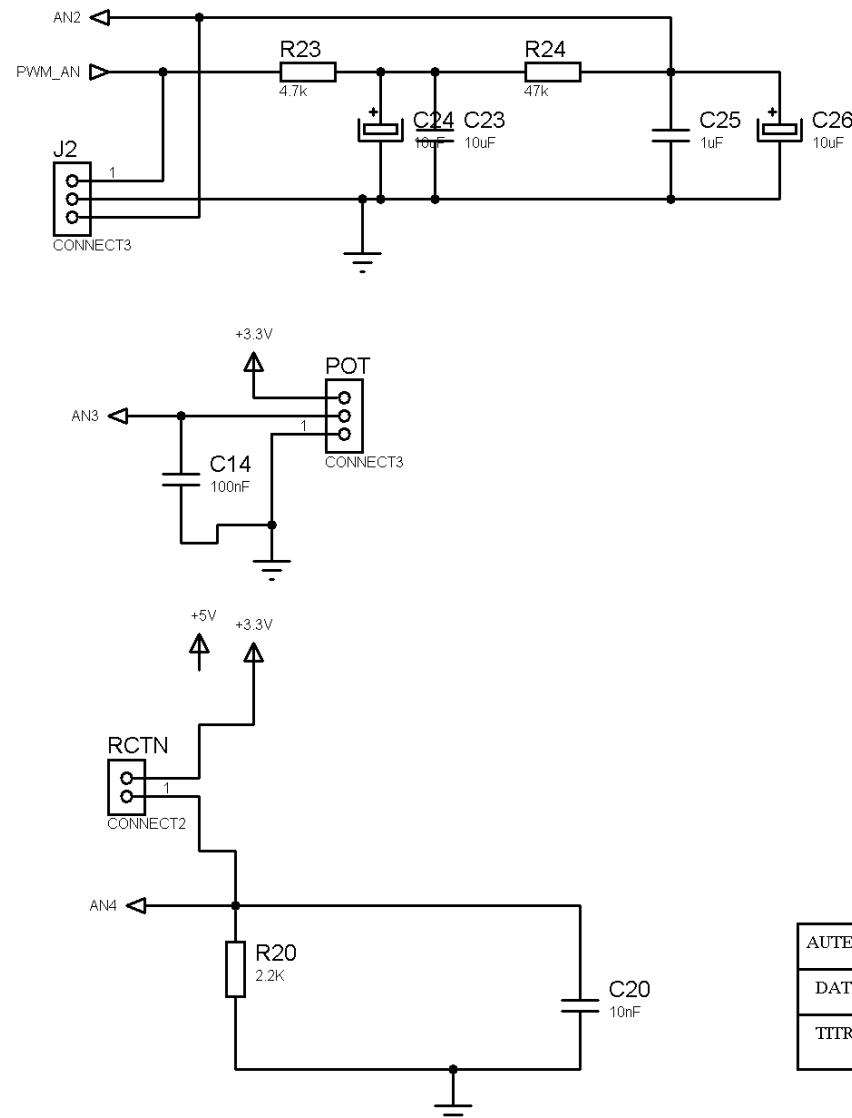
AUTEUR	Olivier Bellot	Doc N° 1	GE2i Nancy
DATE	Création 08/09/05	Modification 23/03/09	
TITRE	kitpic18F67J11 kitpic18f66j60v0.0.DSN	afficheurs 7 segments	

TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2I



AUTEUR	Olivier Belot	Doc N°	1	GE2i
DATE	Création 08/09/05	Modification	23/03/09	V20
TITRE	afficheur LCD, Inters, BP			

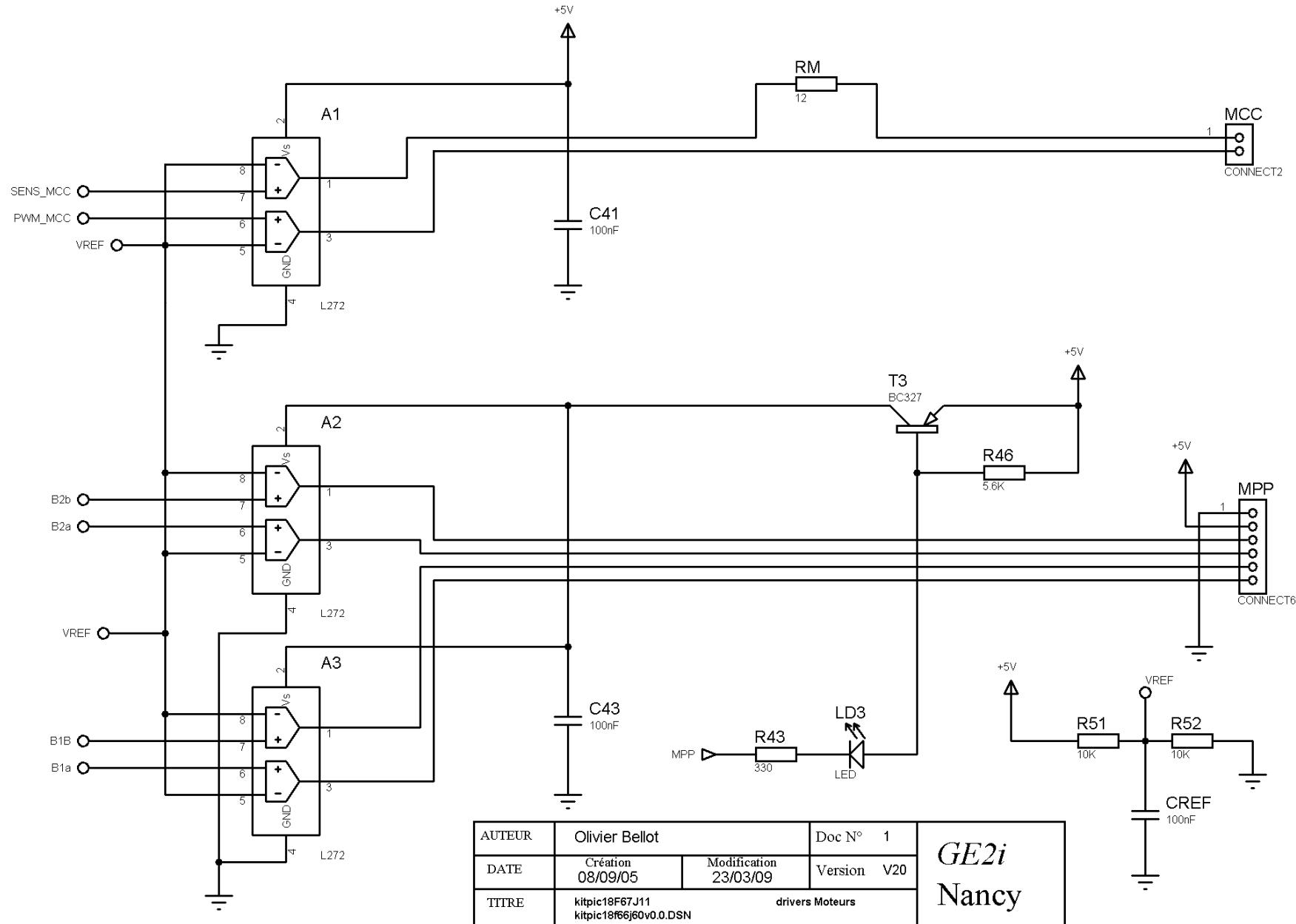
TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2I



AUTEUR	Olivier Bellot		Doc N°	1
DATE	Création 08/09/05	Modification 23/03/09	Version	V20
TITRE	kitpic18F67J11 kitpic18F66J60v0.0.DSN			analogique, CTN

GE2i
Nancy

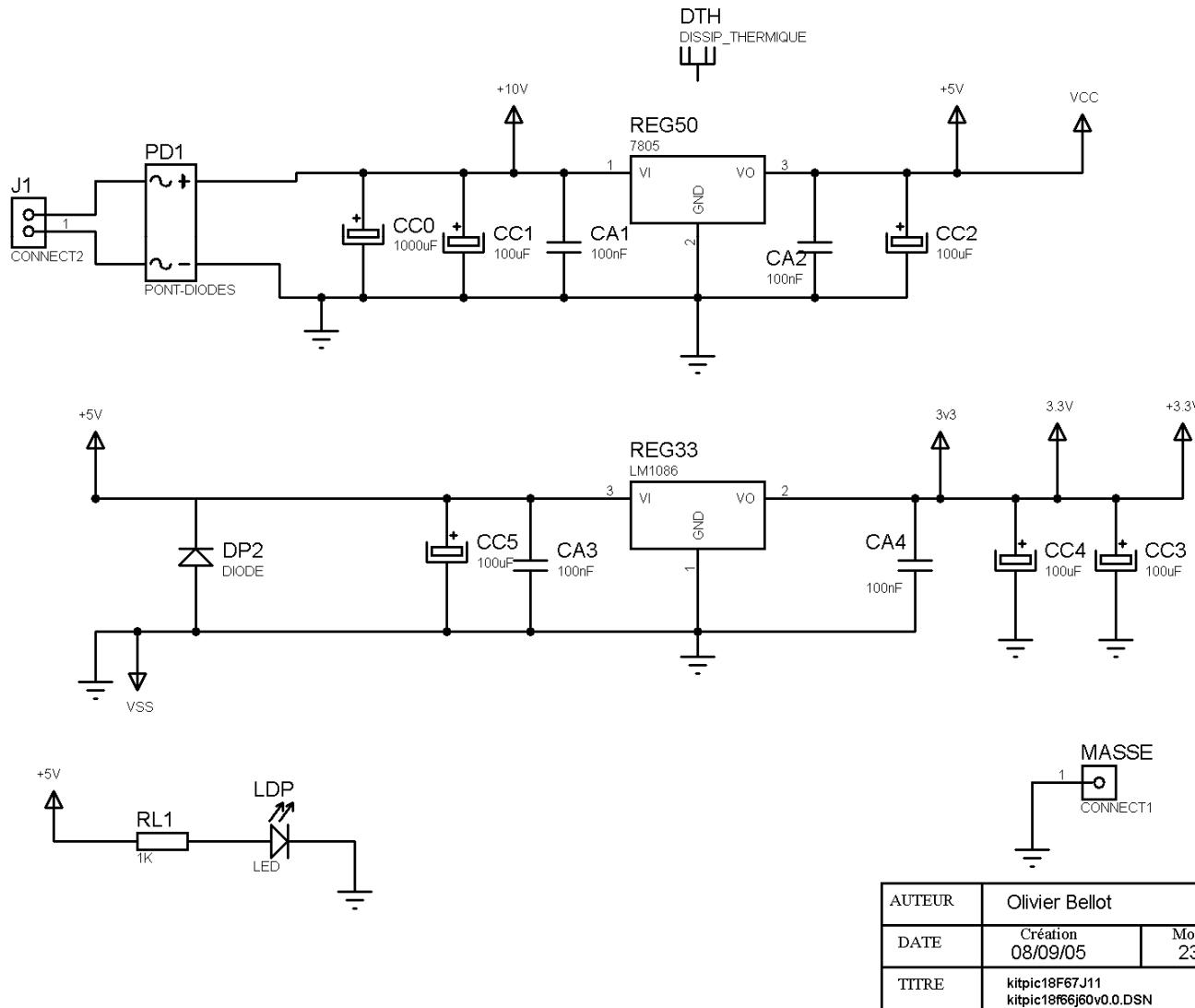
TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2I



AUTEUR	Olivier Bellot		Doc N°	1
DATE	Création 08/09/05	Modification 23/03/09	Version	V20
TITRE	kitpic18F67J11 kitpic18F66J60v0.0.DSN drivers Moteurs			

GE2i
Nancy

TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2I



AUTEUR	Olivier Bellot	Doc N° 1	GE2i Nancy
DATE	Création 08/09/05	Modification 23/03/09	
TITRE	kitpic18F67J11 kitpic18F66J60v0.0.DSN	Alimentations	

TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2I

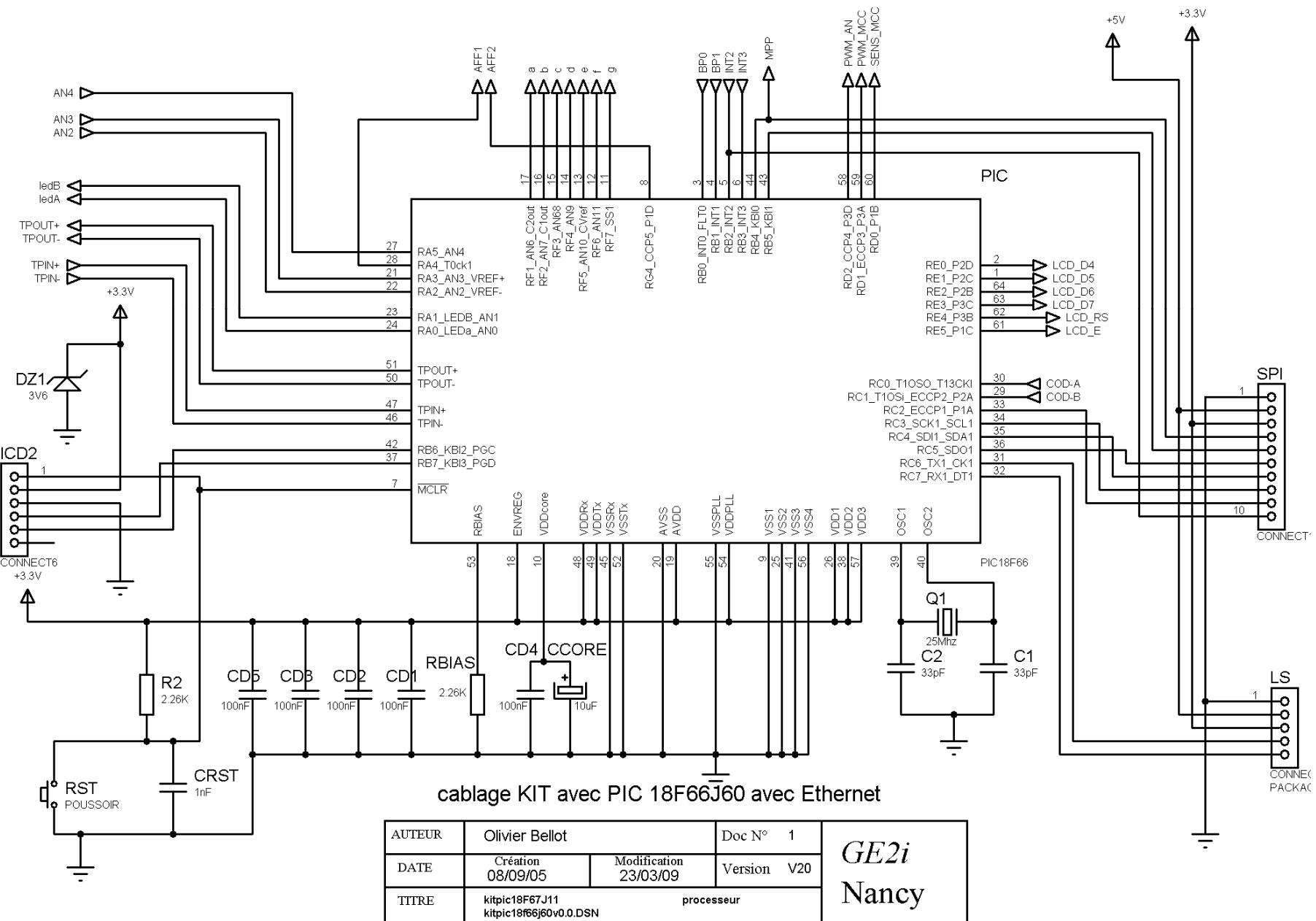
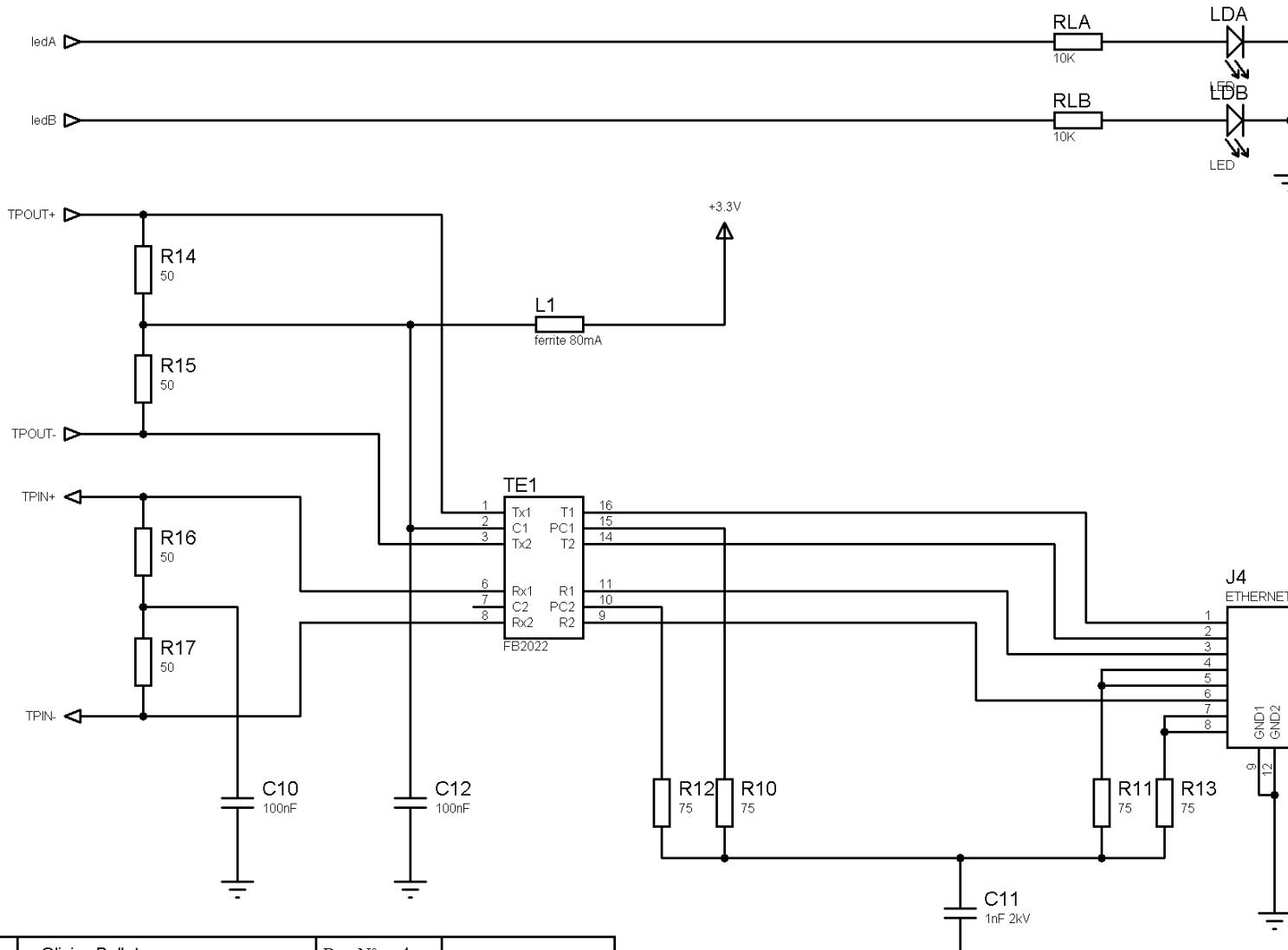


TABLE DE CABLAGE PIC ↔ COMPOSANTS PERIPHERIQUES DU KITPIC GE2i



AUTEUR	Olivier Bellot	Doc N°	1	<i>GE2i</i> Nancy
DATE	Création 08/09/05	Modification 23/03/09	Version V20	
TITRE	kitpic18F67J11 kitpic18F66J60v0.0.DSN	ethernet		