

I-2 MICROCONTROLLEUR

Les fabricants des microprocesseurs ont aussi mis sur le marché des circuits intégrés appelés microcontrôleurs. Ces derniers représentent des systèmes à microprocesseurs complets intégrés dans un même chip. Le rôle principal des microcontrôleurs est la réduction des tailles des systèmes pour être facilement intégrés dans des chaînes de contrôle. Du moment que la plupart des composants sont intégrés dans un même boîtier, ceci réduit considérablement le temps mis pour l'exécution d'une instruction. La figure 1 définit d'une façon générale la constitution d'un microcontrôleur.

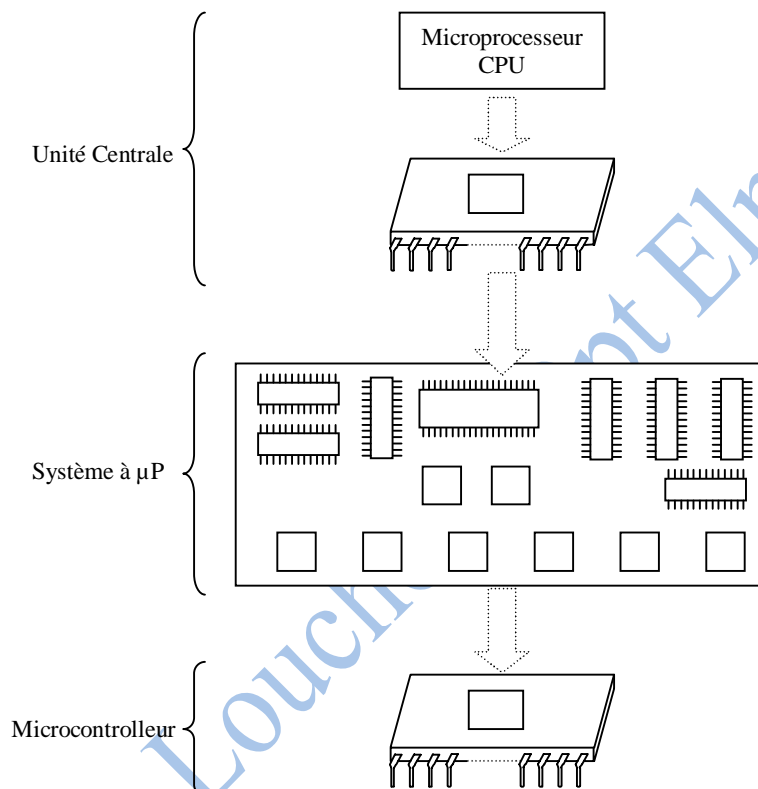


Figure-1. Cette figure montre le principe d'obtention d'un microcontrôleur qui n'est autre qu'un système à microprocesseur intégré.

i-2-1 Exemple de microcontrôleur de la famille mcs51

En y intégrant encore d'énormes potentiels de calcul et de gestion (timers, UART, gestionnaire d'interruptions, mémoires), le μC 8051, proposé par la société INTEL comme suite logique du 8048, est devenu un vrai microsystème. Son dérivé le 8751 est doté d'une EPROM interne. Sauf pour des cas particuliers (large espace nécessaire pour mémoires et E/S), il peut être implanté seul dans un environnement de contrôle plus flexible, moins coûteux et beaucoup moins encombrant.

On parle encore du 8xC51; la version CHMOS du 8x51. La technologie CHMOS combine les

avantages des versions HMOS (haute vitesse et densité) et CMOS (faible puissance) et offre la possibilité de fonctionner en deux modes de consommation réduite sélectionnés par logiciel: *mise en veille* (oscillateur actif mais isolé de l'unité centrale, le μC revient au mode normal par une interruption), *mise en sommeil* (oscillateur bloqué, débloqué par un 'RESET'). Les points qui suivent décrivent en bref les caractéristiques fonctionnelles principales du 8x51, et les points forts qui ont justifié son choix:

- Il dispose de 4 *ports bidirectionnels* d'E/S de 8 bits notés: P0, P1, P2, et P3 adressables en octets ou au niveau du bit, à chacun correspond un registre de la RAM interne portant le même nom et accessible par la même adresse. Seul le port P0 est bidirectionnel (les autres sont considérés comme quasi-bidirectionnels), il a une très grande impédance en entrée, cette particularité est expliquée par le fait que ses lignes sont à drain ouvert; en sortie il doit être équipé de résistances de rappel. Les ports P0, P2, et P3 possèdent des fonctions secondaires.
- Il dispose de deux *timers* internes notés T0 et T1, ce sont des compteurs/temporisateurs 16 bits avec une variété de modes sélectionnés par logiciel. Si le timer compte les cycles de l'oscillateur interne, il est en mode *temporisateur*. S'il est entrain de compter des événements externes (transitions sur une broche désignée T0 ou T1), il est en mode *compteur*. Le choix d'un mode quelconque, la mise en marche ou en arrêt et l'initialisation des timers sont faits par logiciel en validant certains bits des registres TCON et TMOD.
- Il autorise 5 sources d'*interruptions* vectorisées, ce sont les composantes matérielles pour un système temps réel. Lorsque le processeur reconnaît une interruption, son compteur ordinal (PC) est changé vers une adresse spécifique à l'interruption en question, cette adresse pointe un espace réservé de 8 octets (3 octets pour l'interruption particulière RESET) suffisant pour sauvegarder le contexte interrompu et exécuter une instruction de saut vers la routine de service. Une fois l'interruption servie, le PC est restauré et le μC continue l'exécution de la tâche interrompue. L'autorisation des interruptions et la définition des priorités sont configurables par logiciel en validant certains bits des registres IE, IP, et TCON.
- Il possède un *UART* (émetteur récepteur asynchrone universel) pour la communication série qui peut fonctionner en 'full duplex'. Il exige un circuit externe tel que le MAX232 pour l'adaptation de niveaux avec la norme RS232. L'horloge de communication (baud rate generator) est activée par le timer interne T1. les registres SCON et PCON servent à configurer la transmission série (définition du baud rate et du format de la donnée transmise).
- Une particularité des microcontrôleurs apparaît dans la *notion de mémoire*: il y a une séparation physique entre espaces 'code' et 'donnée' d'une part et espaces 'interne' et 'externe' d'autre part. ce qui permet avec 16 lignes d'adresses d'accéder deux espaces mémoire de 64Koctets chacun; un pour les données et l'autre pour le programme.

I-2-2 Microcontrôleur de microchip PIC

Les microcontrôleurs PIC de Microchip ont une architecture RISC " Reduced Instruction Set Computer". Plus on réduit le nombre d'instructions, plus facile et plus rapide en est le décodage, et plus vite le composant fonctionne. Ce type d'architecture connu dans le domaine de l'informatique comme étant une architecture Harvard. L'objectif principal de cette architecture est l'augmentation de la vitesse de travail d'un système. La séparation entre mémoire de donnée et mémoire de programme permet aussi une nette amélioration de la vitesse d'exécution d'un programme.

I-2-2-1 Structure de base d'un PIC

Pour avoir une idée claire sur l'architecture interne des PIC; prenons l'exemple du PIC16F84 qui appartient à la famille des Pic à 8 bits. Comme il a été défini dans les paragraphes précédents un microcontrôleur est un système à microprocesseur complet intégrant dans le même boîtier l'ensemble des périphériques. La figure 13 montre l'architecture interne simplifiée du PIC 16F84.

La séparation des deux mémoires de données et de programme permet de réserver une longueur aux mots instruction indépendamment de celle réservée à la donnée. Dans le cas des PIC, la longueur de la combinaison numérique codant l'instruction permet leur classification dans trois grandes familles:

- La famille Base-line où l'instruction est codée sur 12 bits
- La famille Mid-Range où l'instruction est codée sur 14 bits
- La famille High-End où l'instruction est codée sur 16 bits

Au sein de la même famille on trouve une large variété de microcontrôleurs qui diffèrent les uns des autres par leur architecture interne. Le tableau ci-dessous résume les caractéristiques des PIC de la famille Mid-Range que nous rencontrons le plus sur le marché national.

Espace mémoire

Comme mentionné précédemment, on distingue 2 zones distinctes, la zone mémoire des données et la zone mémoire programme. Chacune de ces mémoires possède son propre bus. Le tableau ci-dessous résume les caractéristiques de quelques microcontrôleurs.

PIC	FLASH	RAM	EEPROM	I/O	A/D	Port //	Port série
16F84	1K	68	64	13			
16F870	2K	128	64	22	5	NON	USART
16F871	2K	128	64	33	8	PSP	USART
16F872	2K	128	64	22	5	NON	MSSP
16F873	4K	192	128	22	5	NON	USART/MSSP
16F874	4K	192	128	33	8	PSP	USART/MSSP
16F876	8K	368	256	22	5	NON	USART/MSSP
16F877	8K	368	256	33	8	PSP	USART/MSSP

Mémoire programme

Le PC (compteur programme) contient 13 bits permettant d'adresser 8K x 14 bits (taille d'une instruction). Sur le 16F84, seul les 1024 premiers mots sont utilisés et matérialisés par de la mémoire EEPROM. L'espace programme s'étale donc de 0000h à 03FFh tel qu'il est montré sur la figure 2. Le vecteur RESET se trouve en 0000h. L'unique vecteur lié aux autres interruptions se trouve en 0004h. (Un registre de contrôle et d'état des interruptions permet de connaître la source de l'interruption survenue (port, timer ...)).

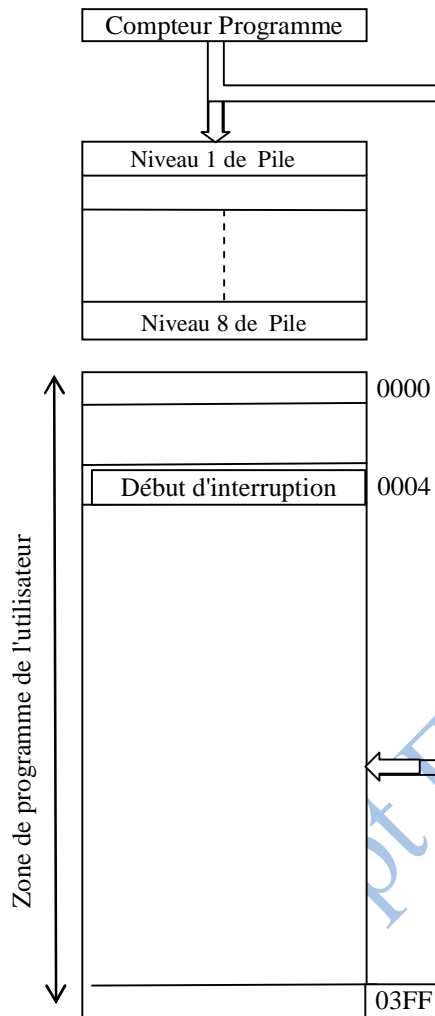


Figure 2 : Mémoire programme

Mémoire des données (RAM)

La mémoire des données est vue par le constructeur comme un fichier (FILE). Elle s'étale de 00h à FFh comme illustré par la figure 3. Une majorité de cet espace est inutilisé. On distingue 2 zones distinctes :

- les registres à usage général (GPR) qui constituent les 36 octets de RAM du 16F84
- les registres à usage spécial (SFR) qui sont en fait constitués du registre d'état du noyau du processeur (CORE) et des registres de contrôle et d'état des périphériques.

	BANK 0	BANK 1	
00h			80h
01h	TMR ₀	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORT A	TRIS A	85h
06h	PORT B	TRIS B	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch			8Ch
	Registre à usage général		
4Fh			CFh
	Non Implémentée	Non Implémentée	
7Fh			FFh

Figure 3 : Mémoire des données

Configuration externe du PIC 16F 84

Le PIC 16F84 se présente dans un boîtier DIL de 18 broches. Les 18 broches sont réparties suivant leurs fonctions comme il est illustré par la figure 4.

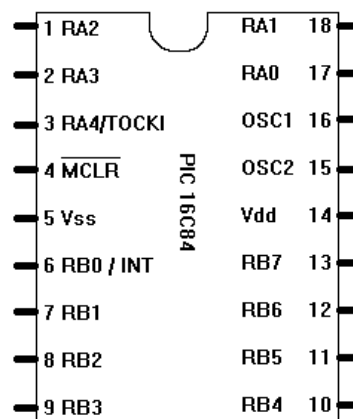


Figure 4 : Brochage du PIC 16F84

Architecture interne du PIC 16F84

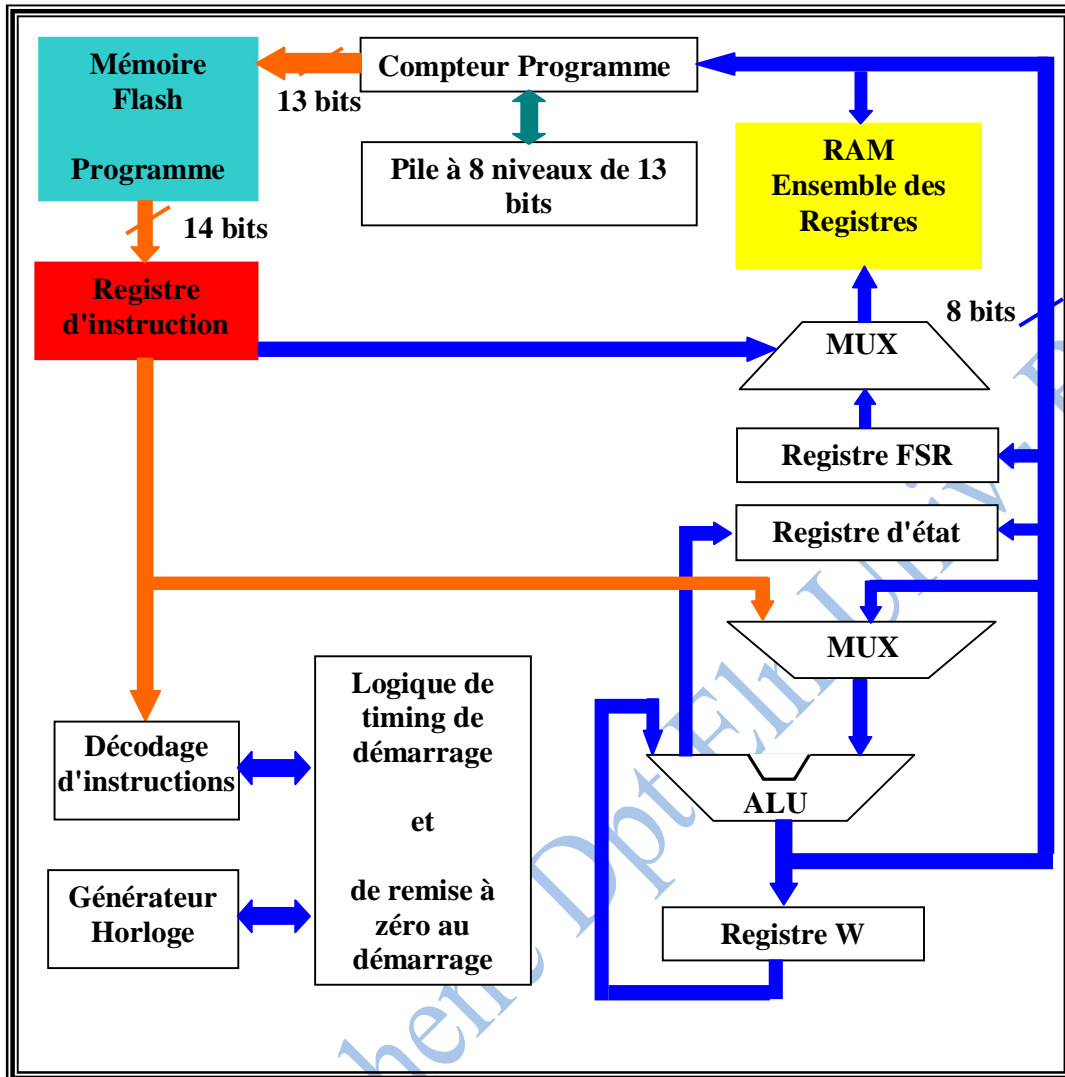


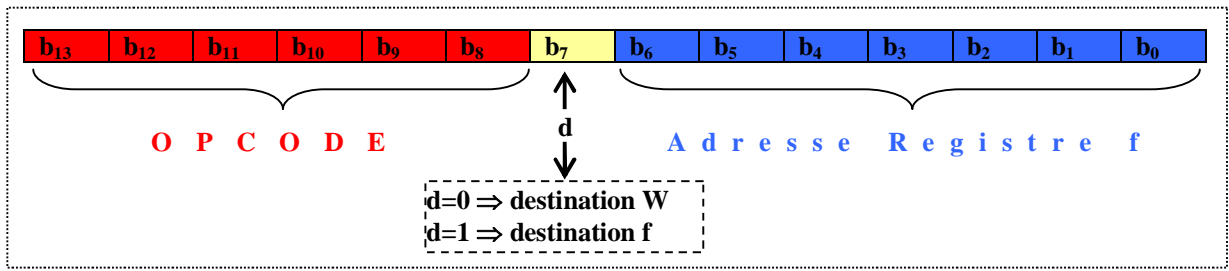
Figure 5 Structure interne du PIC 16F84

Format des instructions

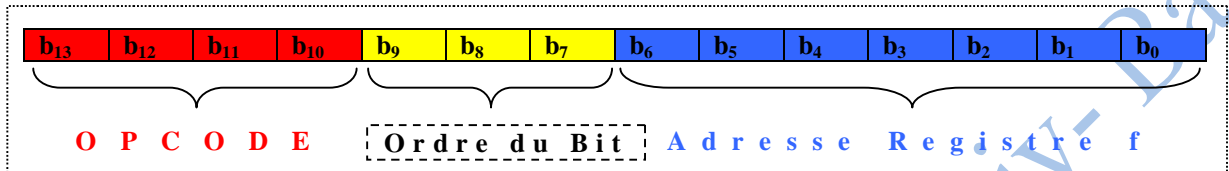
Le format de l'instruction correspond à un mot sous forme d'une combinaison numérique de 14 bits (b13—b0). Les bits les plus significatifs définissent l'opcode. L'opérande est défini par les bits les moins significatifs. La dimension en bits du champ réservé pour chacune des deux parties dépend de la nature de l'instruction. Les instructions peuvent être réparties comme suit:

- Le traitement se fait sur 8 bits : Les instructions orientées Bytes
- Le traitement se fait sur un bit : Les instructions orientées Bit
- Les instructions de contrôle et de branchement.

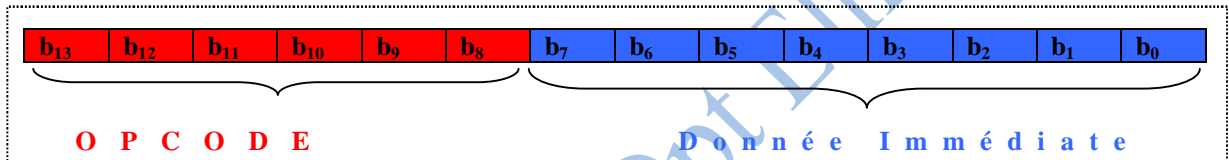
Les parties a b c et d de la figure 6 données ci dessous illustrent d'une façon claire le cas des trois points cités ci-dessus.



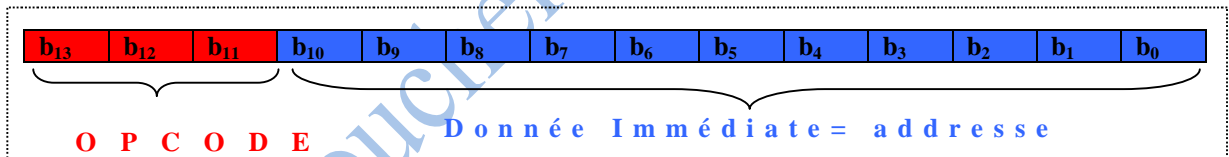
a: Opération sur byte: adressage registre



b: Opération sur bit: adressage registre



c: Opération sur la donnée immédiate



d: Opération d'appel ou de saut

Figure 6: Format d'une instruction

Exemples :

ADDWF 0x0C, 1 ; Ajouter au contenu du registre de travail W le contenu du registre de la zone mémoire RAM d'adresse 0x0C. Le résultat sera gardé dans le registre d'adresse 0x0C.

Cette instruction entre dans le cadre de la figure –a. Son codage correspond à cette même figure dont le contenu des champs est illustré par la figure 7.

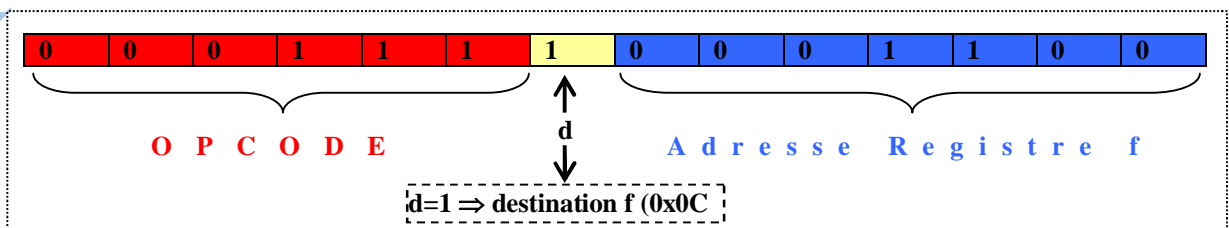


Figure 7 : format de l'instruction ADDWF 0x0C,1

PIC 16F84 Instruction Set

◆ Move or Define Data

MOVLW	k	Loads W with literal.
MOVF	f, d	Moves copy of selected register contents into W or f.
MOVWF	f	Moves copy of W contents into selected register

◆ Change Register Contents

CLRF	f	Clears selected register to 0.
CLRW		Clears W register to 0.
COMF	f, d	Complements selected register contents. Result in W or f.
DECF	f, d	Decrements selected register. Result in W or f.
INCF	f, d	Increments selected register. Result in W or f.
BCF	f, d	Clears selected bit in selected register to 0.
BSF	f, d	Sets selected bit in selected register to 1.
RLF	f, d	Rotates bits in selected register one position to the left. Bits rotate through carry flag. Result in W or f.
RRF	f, d	Rotates bits in selected register one position to the right. Bits rotate through carry flag. Result in W or f.
SWAPF	f, d	Exchanges MS and LS nibbles of selected register. Result in W or f.

◆ Control Program Flow

GOTO	k	Go to specified address.
CALL	k	Call subroutine at specified starting address.
RETURN		Return from subroutine.
RETLW k		Return from subroutine. Load W with literal.
RETFIE		Return from interrupt.

BTFSC	f, b	Tests specified bit in specified register, skip next instruction if clear.
BTFSS	f, b	Tests specified bit in specified register, skip next instruction if set (1).
DECFSZ	f, d	Decrements specified register, skips next instruction if register = 0. Result in W or f.
INCFSZ	f, d	Increments specified register, skips next instruction if register = 0. Result in W or f.

◆ Nothing

NOP Do nothing for one instruction cycle.

◆ Control Microcontroller

CLRWDT Clear watchdog timer (reset to 0). Also resets watchdog timer prescaler. Status bits TO and PD are set OPTION W contents (bit pattern) sent to option register. (prescaler ratio, RTC trigger edge, and RTC source)

SLEEP Puts microcontroller to sleep to reduce Power consumption. (wakeup via reset, WDT, or external input)

TRIS f W bit pattern determines port line input vs. output for elected port.

◆ Logic

ANDLWk • $W \& k \rightarrow W$.

ANDWFf, d $W \& f \rightarrow W$ or f.

IORLW k $W // k \rightarrow W$.

IORWF f, d $W // f \rightarrow W$ or f.

XORLW k $W \text{ xor } k \rightarrow W$.

XORWFf, d $W \text{ xor } f \rightarrow W$ or f.

◆ Arithmetic

ADDLWk $W + k \rightarrow W$.

ADDWFf, d $W + f \rightarrow W$ or f.

SUBLW k $k - W \rightarrow W$.

SUBWF f, d $f - W \rightarrow W$ or f.

Exemple d'application

A titre d'exemple d'application, nous citons les cartes à puces où le terme puce désigne le système à microcontrôleur intégré dans la carte. Les figures 8 et 9 montrent deux types de cartes à puce avec respectivement le schéma électronique de la puce intégré.

Les Silvers card :

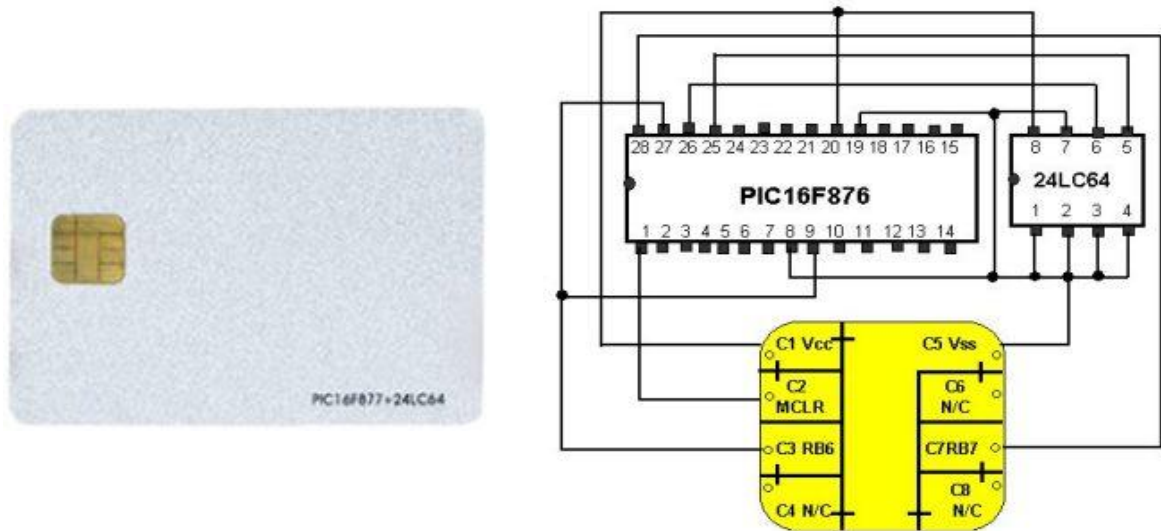


Figure-8 carte à puce argent et le schéma du système intégré

Les GOLDs card :

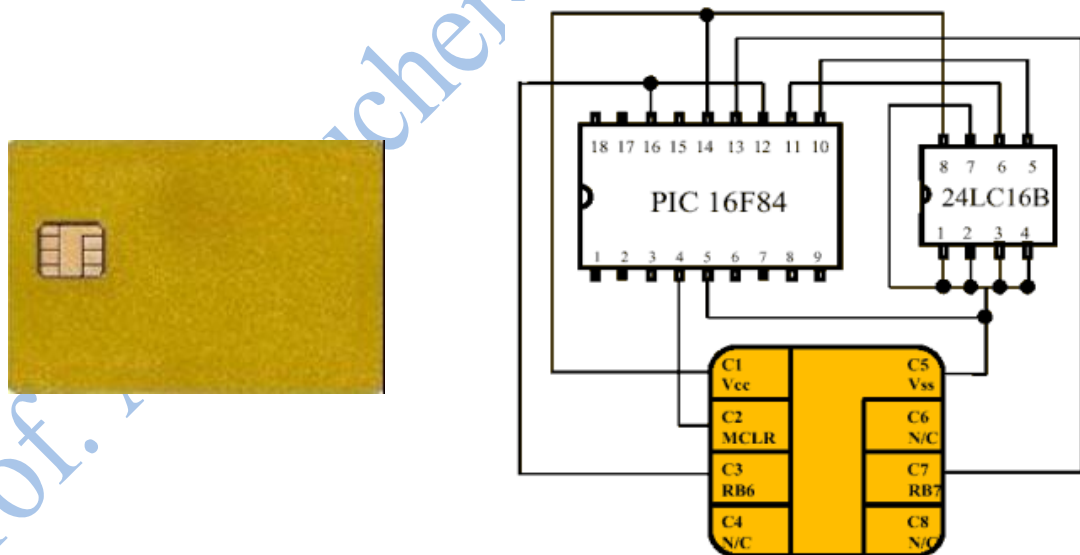


Figure- 9 carte à puce Or et le schéma du système intégré