

Formation Automatique et Informatique Industrielle

Master 1 S2

Matière : Systèmes Embarqués et Systèmes
Temps Réel SE-STR

Par : ATOUI Hamza

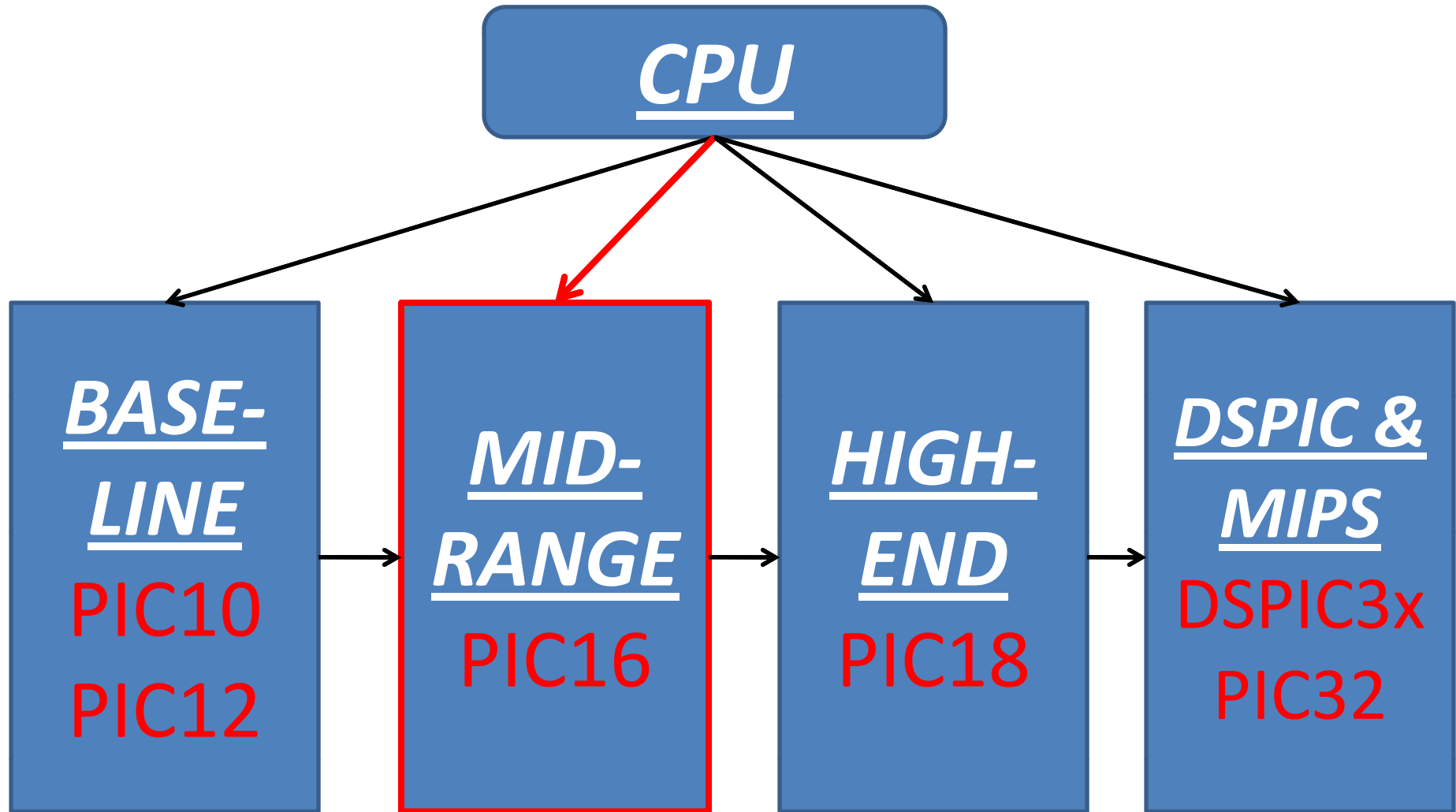
Plan du cours –partie 1-

- MICROCHIP TECHNOLOGY.
- Les μ P et μ C de MICROCHIP.
- La série 16Fxxx.
- Le μ P MID-RANGE de la série 16Fxxx.
 - **Architecture interne.**
 - **Fréquence d'horloge, cycle machine et pipeline de MID-RANGE.**
 - **Modes d'adressage.**
 - **Organisation de l'espace DATA/IO.**

MICROCHIP TECHNOLOGY

- **MICROCHIP TECHNOLOGY**, ou tout simplement **MICROCHIP**, est un fabricant de semi-conducteurs fondé en 1989, à partir d'une division de **General Instrument**.
- MICROCHIP produit les microcontrôleurs PIC, des composants radiofréquences, des composants de gestion des batteries, des interfaces, des composants analogiques, etc.
 - Source : Wikipédia

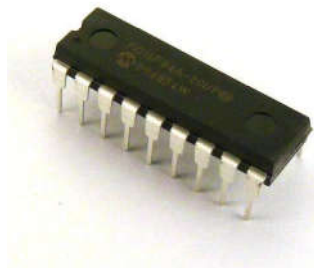
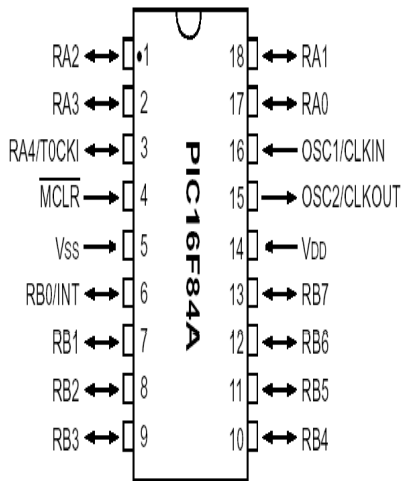
Les μ P et μ C de MICROCHIP



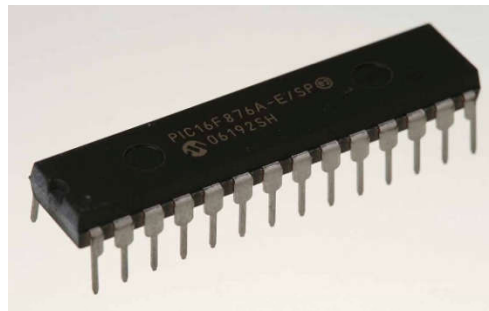
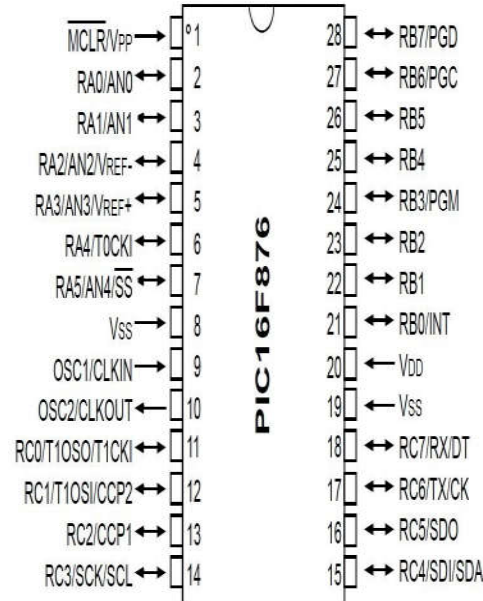
Pourquoi cette évolution ?

La série 16Fxxx

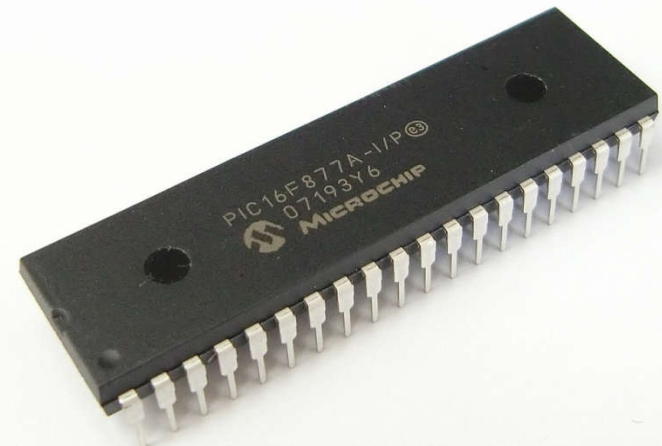
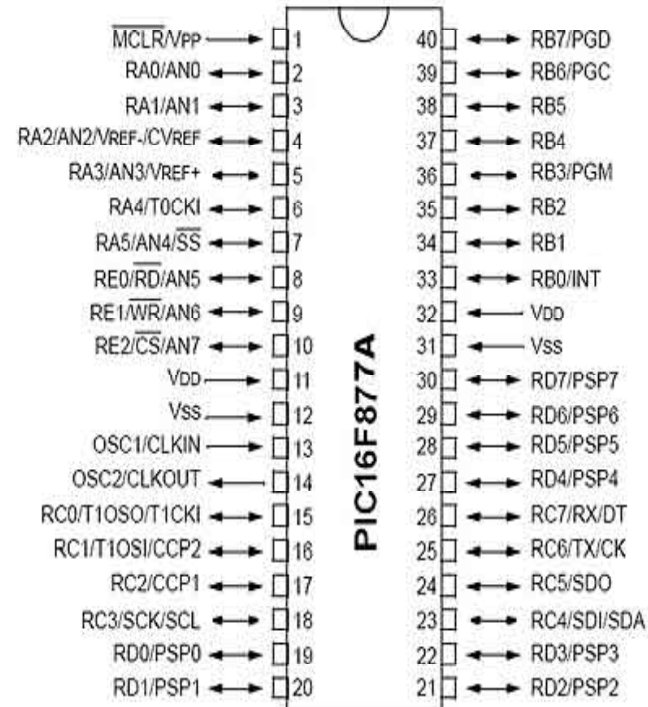
PIC16F84A 18-pins



PIC16F876A 28-pins



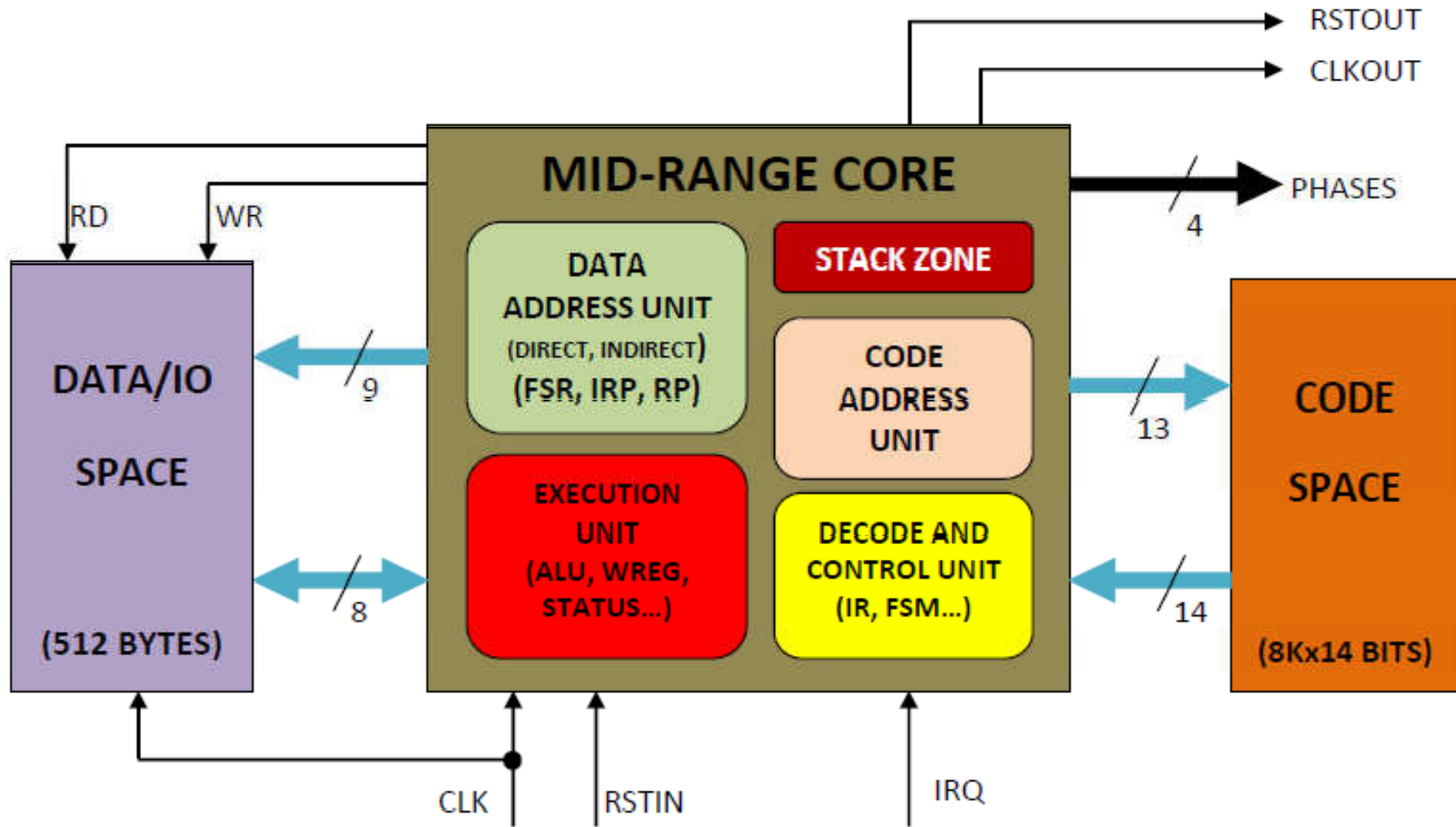
PIC16F877A 40-pins



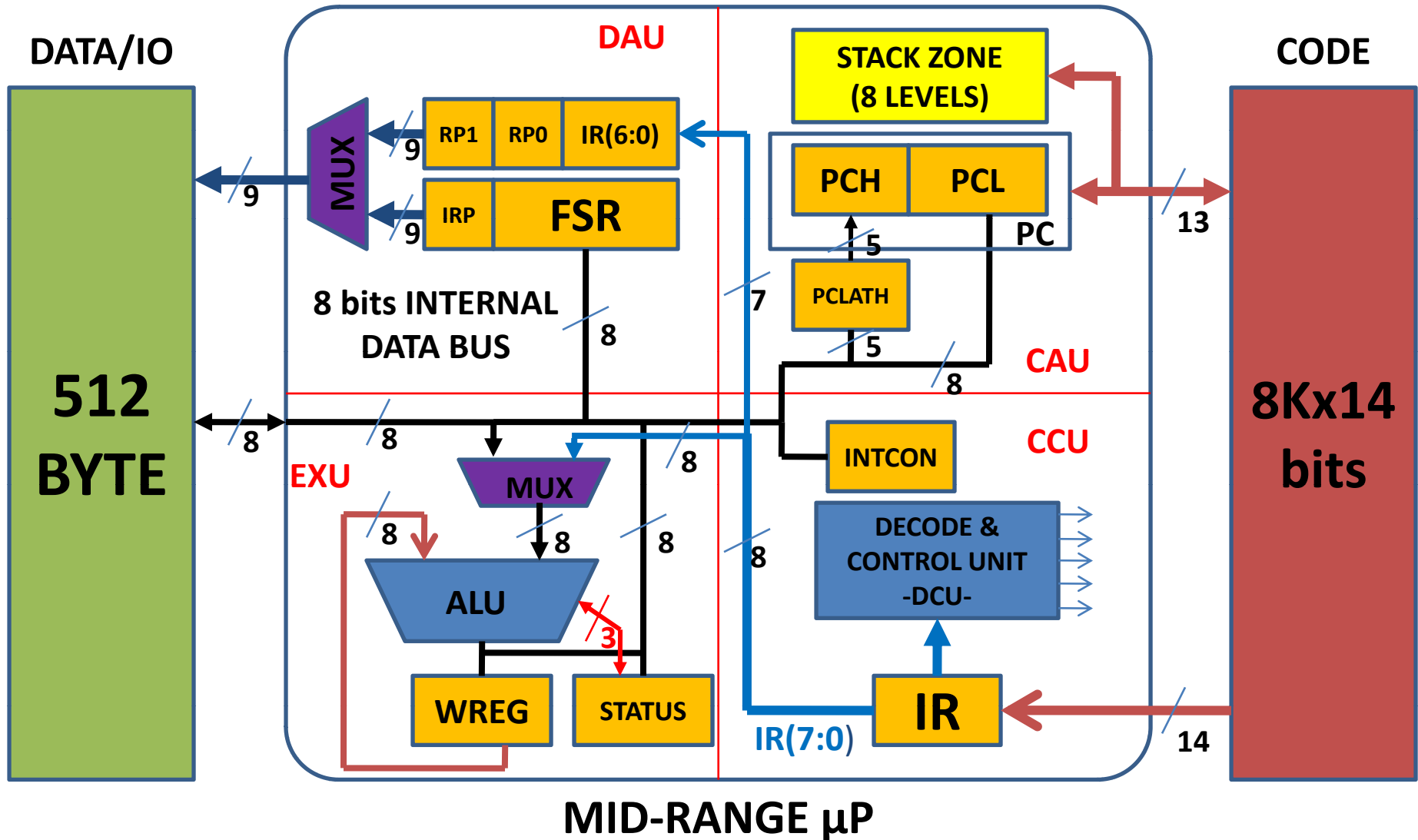
La série 16Fxxx

- Les différences entre ces μ C sont :
 - Le nombre de pattes, donc nombre des ports I/O.
 - Le nombre de pattes analogiques/Numériques.
 - La taille des espaces DATA, CODE et EEPROM.
 - Le nombre de périphériques (ADC, UART, TIMERS...).
- **Mais, tous sont équipés par le MID-RANGE comme microprocesseur (CPU).**

Architecture interne



Architecture interne

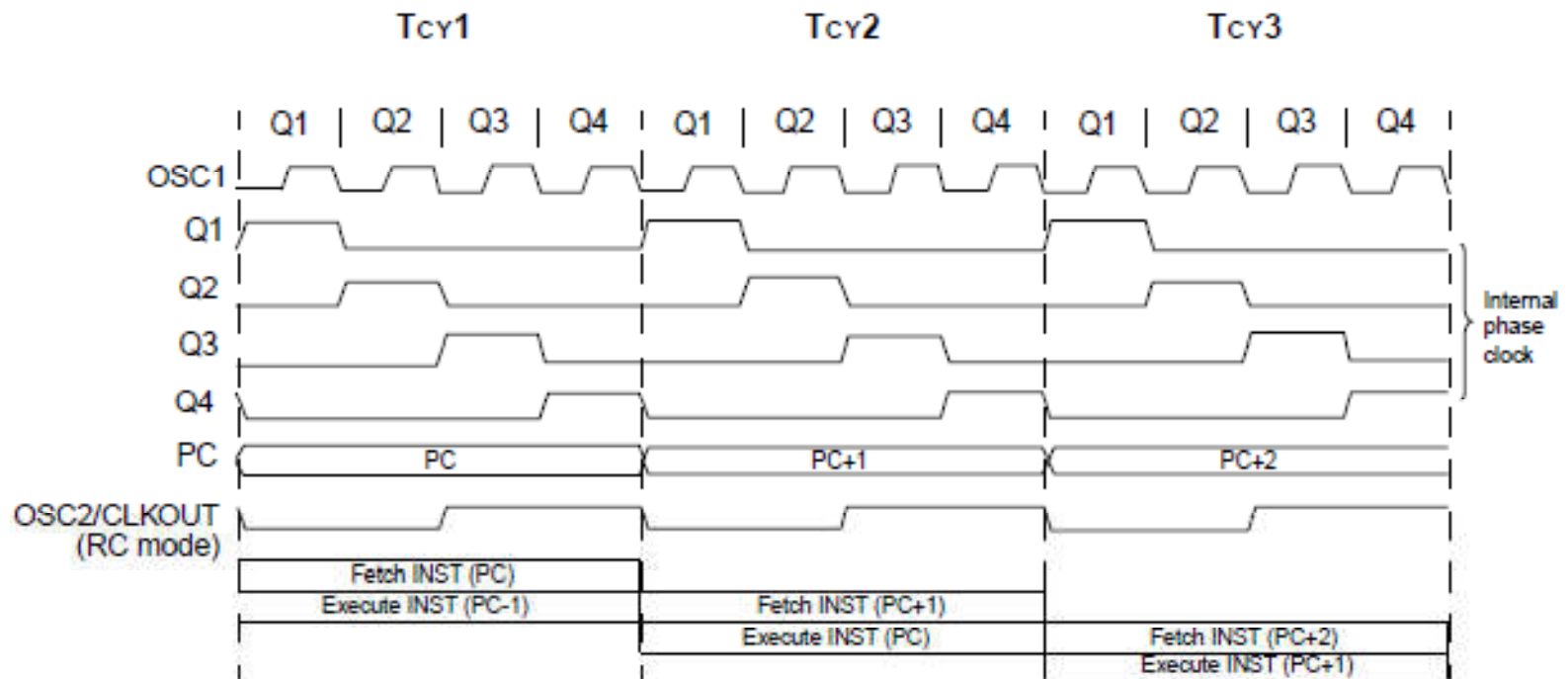


Architecture interne

- **EXECUTION UNIT (EXU)** : est composée de registre WREG, l'ALU et le registre STATUS.
- **DATA ADDRESSING UNIT (DAU)** : est composée de registre FSR plus 3 bits RP0, RP1 et IRP.
- **CODE ADDRESSING UNIT (CAU)** : est constituée d'une pile de 8 niveaux (STACK ZONE), le registre compteur ordinal (PC), et le registre PCLATH.
- **COMMAND & CONTROL UNIT (CCU)**: est formée de registre d'instruction (IR), de registre d'interruption (INTCON) et l'unité la plus compliquée dans notre μP (DCU), cette dernière génère les différents signaux pour exécuter une instruction.

Fréquence d'horloge, cycle machine et pipeline de MID-RANGE

- La **fréquence** de l'horloge d'entrée du MID-RANGE est divisée par 4 pour générer 4 phases sont (**Q1,Q2,Q3,Q4**), donc le **cycle machine** est 4 fois plus long que la période d'horloge.

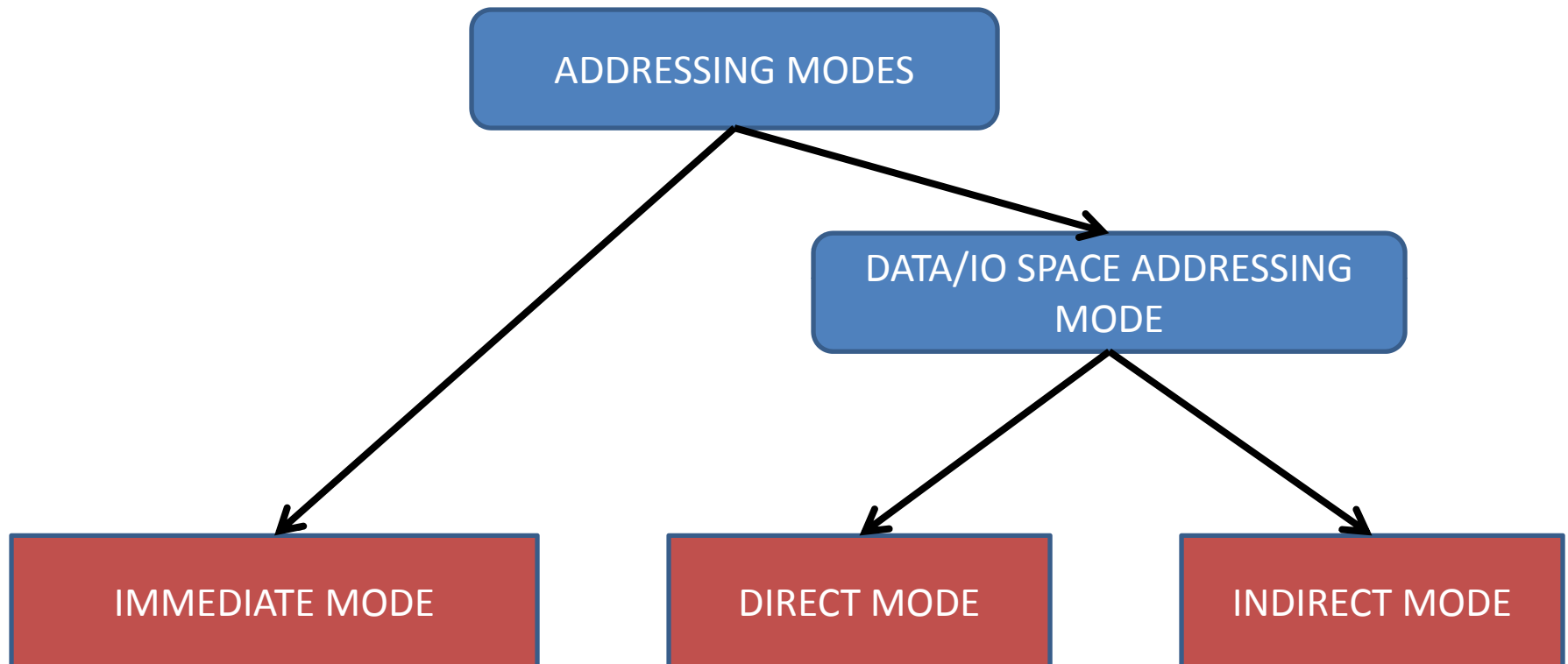


Fréquence d'horloge, cycle machine et pipeline de MID-RANGE

- D'après la figure précédente le MID-RANGE utilise un pipeline à 2 étages (**FETCH/EXECUTION** ↔ **recherche de l'instruction suivante/exécution de l'instruction en cours**).
- Dans l'étage **FETCH** : le MID-RANGE incrémente le PC à la fin de Q4.
- Dans l'étage **EXECUTION** : le MID-RANGE fait 4 phases:
 - **Q1** : pour décoder d'instruction.
 - **Q2** : pour lire les opérandes (case mémoire ou constante) s'il existe.
 - **Q3** : pour réaliser l'opération demander dans l'instruction.
 - **Q4** : pour ranger le résultat vers la destination (case mémoire ou WREG).

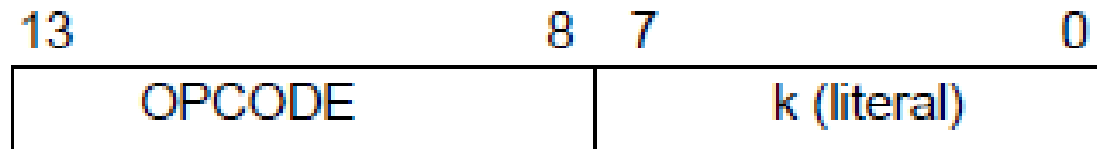
Modes d'adressage

- Le MID-RANGE utilise 3 modes d'adressage sont :



IMMEDIAT MODE

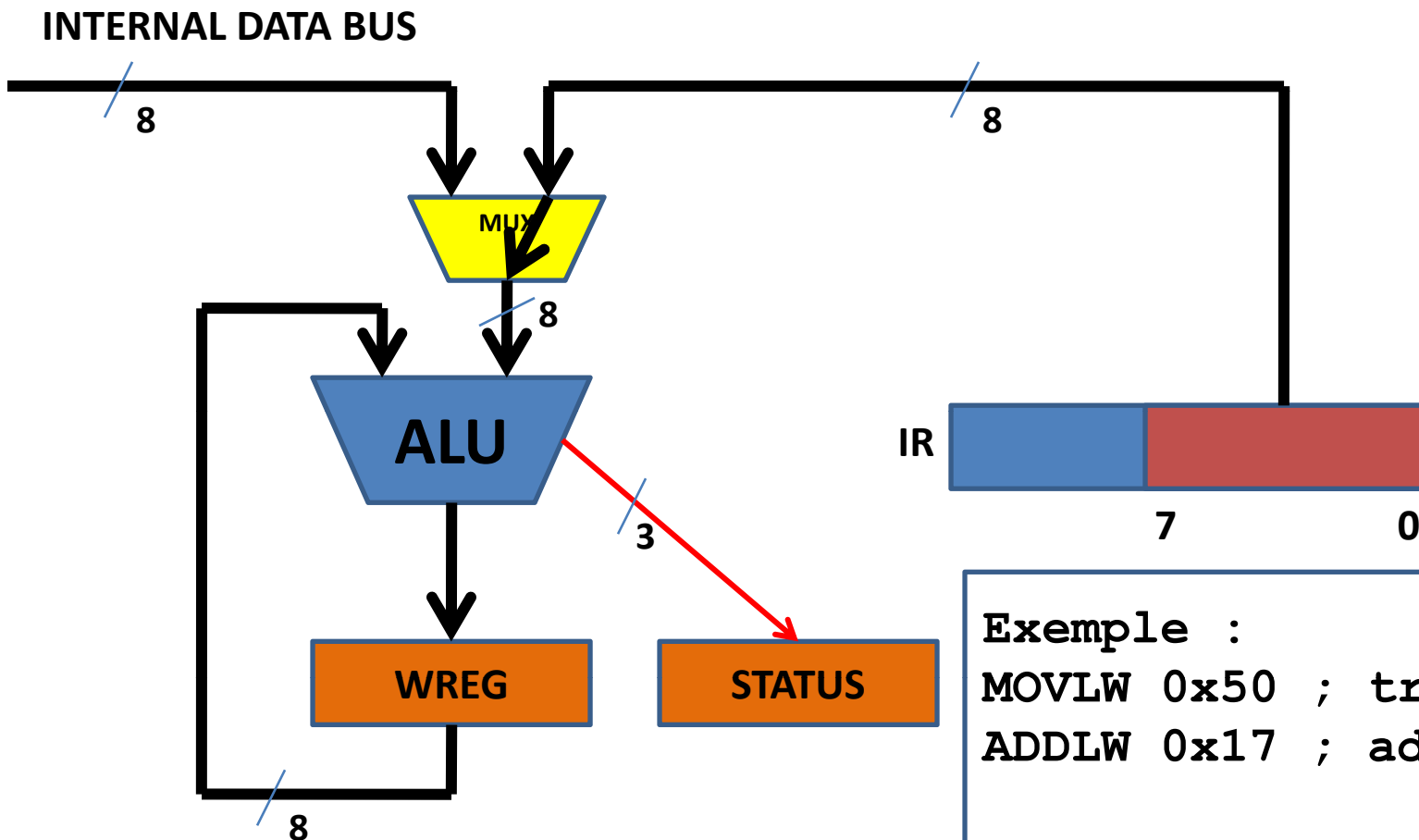
- Ce mode fait la manipulation des valeurs immédiates avec le WREG (transfert, opération A/L)
 - $WREG \leftarrow K$ (constante sur 8bits)
 - $WREG \leftarrow K (op) WREG$ (op : opération A/L).
 - Le format de l'instruction est le suivant:



k = 8-bit literal (immediate) value

IMMEDIAT MODE

- L'unité d'exécution devienne comme suit:



DATA/IO SPACE ADDRESSING MODE

- Ce mode en général fait la manipulation de l'espace DATA/IO avec le WREG (transfert, opération A/L)
- Ce mode comprend le mode direct et le mode indirect par le registre FSR.
- Le format de l'instruction pour ces modes est:



d = 0 for destination W

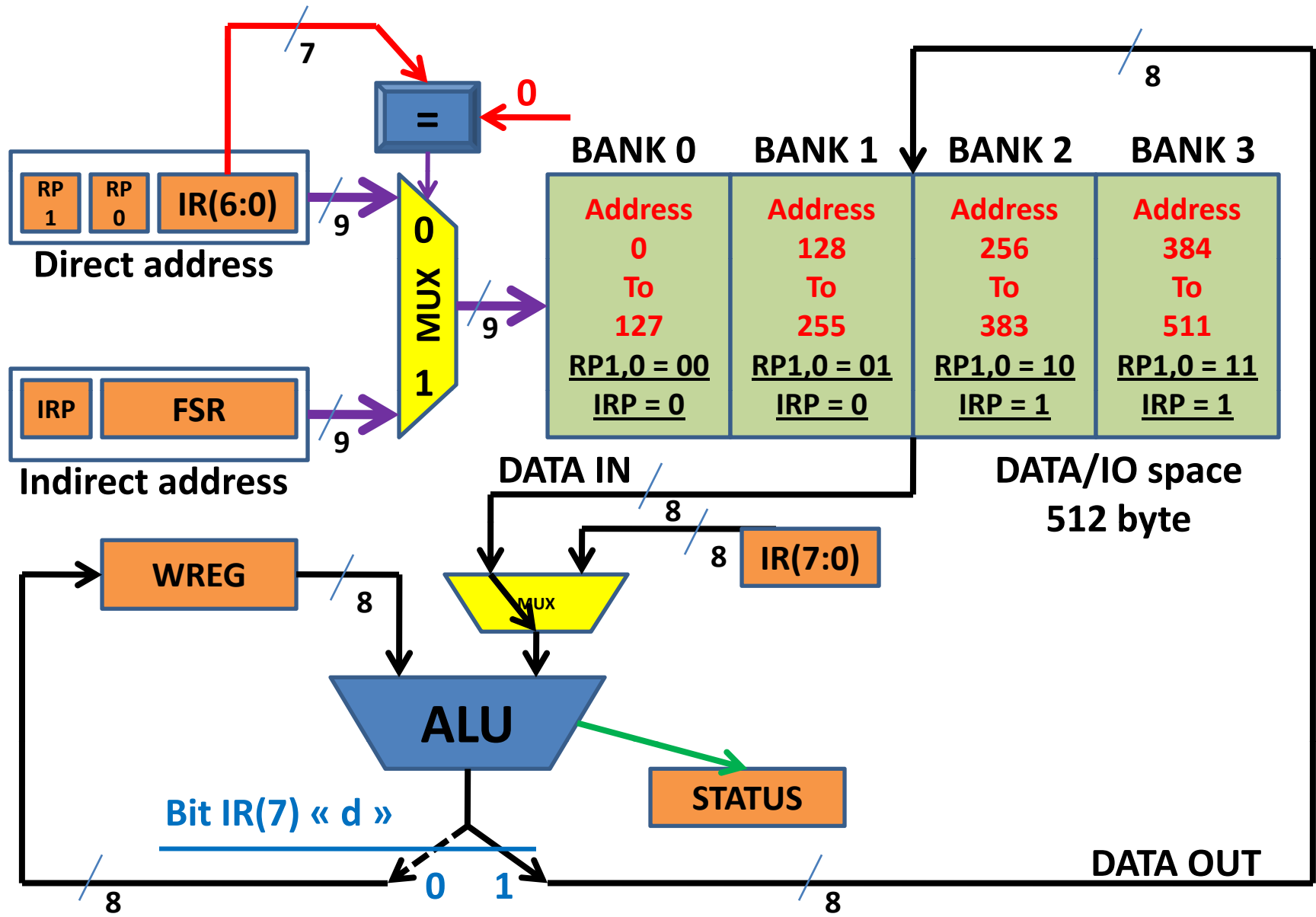
d = 1 for destination f

f = 7-bit file register address

- Vous remarquez que l'adresse d'une case mémoire est sur 7 bits, et notre unité d'adressage DATA/IO du MID-RANGE sortir 9 bits, donc comment faire pour les 2 derniers bits??? (notre instruction présente une limite d'accès de 128 octets seulement)
- Et aussi, on a le même format d'instruction pour les deux modes, donc comment différencier entre ces deux modes???

DATA/IO SPACE ADDRESSING MODE

- Dans ce mode l'unité d'exécution devienne comme suit:



DATA/IO SPACE ADDRESSING MODE

- D'après la figure précédente :
 - Dans le mode d'adressage direct, on peut considérer le DATA/IO SPACE comme une barrette de RAM à 4 circuits intégrés, dont chaque circuit a une taille de 128 octets.
 - Dans le mode d'adressage indirect, on peut considérer le DATA/IO SPACE comme une barrette de RAM à 2 circuits intégrés, dont chaque circuit a une taille de 256 octets.
 - Le mécanisme de pointage vers le DATA/IO SPACE est :

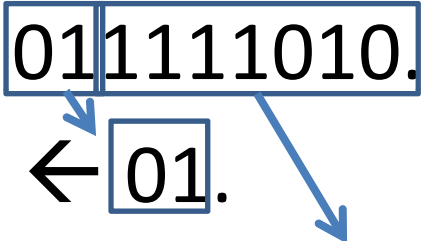
```
IF IR(6:0) = « 0000000 » THEN
    MEM ADDRESS ← [IRP:FSR]; -INDIRECT-
ELSE
    MEM ADDRESS ← [RP1:RP0:IR(6:0)]; -DIRECT-
END IF
```

Exemple

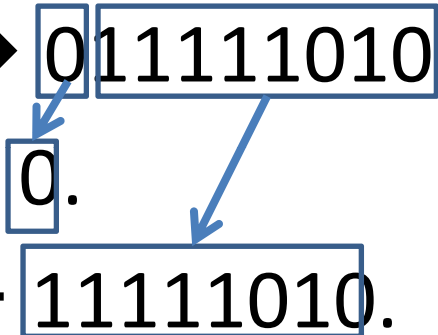
- Transférer le contenu de la case mémoire d'adresse 250 vers le WREG par le mode direct et indirect.

Solution de l'exemple

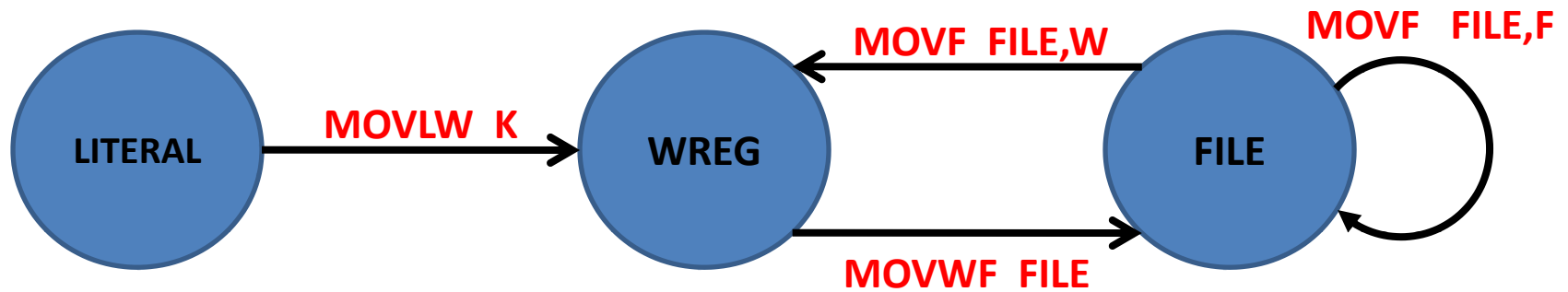
DIRECT MODE

- L'adresse 250 existe dans la BANK1
- $250 \rightarrow 011111010.$ 
- $RP(1,0) \leftarrow 01.$
- $ADR\ FILE \leftarrow 1111010.$
- **$W \leftarrow MEM(0x7A)$**

INDIRECT MODE

- L'adresse 250 existe dans la première partie 256 octets.
- $250 \rightarrow 011111010$ 
- $IRP \leftarrow 0.$
- $FSR \leftarrow 11111010.$
- **$W \leftarrow MEM(0x00)$**

GRAPHE D'ETAT DES MODES D'ADRESSAGES AU MOMENT DU TRANSFERT DE DONNEES



Organisation de l'espace DATA/IO

IRP	0		0		1		1	
RP1,0	00		01		10		11	
0x000	INDF	0x080	INDF	0x100	INDF	0x180	INDF	
0x002	PCL	0x082	PCL	0x102	PCL	0x182	PCL	
0x003	STATUS	0x083	STATUS	0x103	STATUS	0x183	STATUS	
0x004	FSR	0x084	FSR	0x104	FSR	0x184	FSR	
0x00A	PCLATH	0x08A	PCLATH	0x10A	PCLATH	0x18A	PCLATH	
0x00B	INTCON	0x08B	INTCON	0x10B	INTCON	0x18B	INTCON	
0x07F		0x0FF		0x17F		0x1FF		
BANK 0		BANK 1		BANK 2		BANK 3		