.:: Architecture ::.

semaine n°1
Introduction et premières instructions

Introduction

L'architecture :

- Comprendre le fonctionnement du coeur systèmes informatiques.

  - par la programmation au plus bas niveau
  - par l'étude du microprocesseur et des composants de proximité
  - par l'étude d'une carte électronique contenant un système embarqué complet.

Introduction

Les systèmes informatiques généraux :

- Un système dont l'usage n'est pas dédié à une unique application
  - Ordinateur
  - PDA
  - Consoles (de plus en plus)

  ==> Ces systèmes doivent répondre à des besoins divers :

  - écouter de la musique : peu de mémoire, fréquence faible
  - afficher des monde virtuels 3d : beaucoup de mémoire, beaucoup de CPU

  => Car il doivent tout et rien faire, leur coût n'est pas optimisé.

Introduction

Les systèmes informatiques spécifiques :

- Un système dont l'usage est dédié à une seule application
  - Téléphones cellulaires
  - Consoles (entre les deux donc)
  - Voitures
  - Tv, machines à laver ...

  ==> Chacun ne rempli qu'une seule fonction : téléphoner, jouer, freiner/accélérer, afficher les programme, peser et laver le linge ...

  => Ces systèmes sont optimisés pour leur unique fonction, ils sont donc réalisés à moindre coûts

Introduction

Des exemples :

- Routage, connexion à Internet
  - Système de 1 à 100 Mhz, 64ko à 8Mo de mémoire
  - 100 euro
- Télécommande de TV
  - Système à quelques Khz, 1 à 2 Ko de mémoire
  - 10 euro
- Nabztag (Lapin Wifi)
  - Système à environ 200 Mhz, < 1Mo de mémoire
  - 70 euro
- Console de jeu Xbox
  - PIII 800, 64 Mo de mémoire, 8Go de HD
  - 150 euro (vente à perte)

Introduction – les couches

Les différentes couche d'une SI :

Programmation haut niveau : C, java...	Interface avec l'utilisateur
Programmation bas niveau : asm	
Couche Système d'exploitation	Interface avec les composants
Couche langage machine	
Couche micro-programmée	Les composants électroniques processeurs, mémoire, E/S
Couche physique	

Introduction

Niveau matériel :

- Utilisation de micro-processeur (exemple PIII, Athlon ...)
- Utilisation de micro-contrôleurs
  - Système électronique comprenant dans un même composant, en plus d'un processeur, de la mémoire, des E/S spécifiques...
  - Un micro-contrôleur permet donc de réaliser des systèmes avec peu de composants et donc à moindre coûts
  - Exemple : Ubicom IP2022 – 120MHz, 20Ko de RAM, 64Ko de Flash (HD). Applications : Point d'accès Wifi, routeurs, périphériques USB... prix : environ 15 euros
  - Exemple : DS80C410 (famille 8051) – 75 Mhz, 73 Ko RAM, 64Ko ROM, Ethernet 10/100M. prix : 5 euros

Introduction

Niveau matériel :

- Processeurs RISC / CISC

=> Le choix du bon composant est primordial dans la conception d'une solution économiquement viable.

- Dans un système informatique :
  - Un CPU (Unité de calcul, Registres (mémoire de proximitée))
  - Mémoire
    - Statique : cache
    - Non volatile : Bios
    - Dynamique : SDRAM, DDRAM
  - Périphériques d'entrée / sortie
  - Périphériques de stockage

Introduction

Niveau logiciel :

- L'optimisation du code permet l'utilisation de solutions matérielle de moindre coût ...
- L'optimisation passe par une maîtrise du fonctionnement interne du processeur et une bonne analyse algorithmique.
  - Exemple : optimisation assembleur dans les calculs de cryptographie.
  - ==> Utilisation dans certain cas du langage du processeur : LM ou Assembleur.
    - Assembleur : instructions de plus bas niveau incluant des directives de compilation – langage humain
    - Langage Machine : valeurs binaires codant les instructions et directement exécutables par le processeur.

.:: Architecture ::.

Le micro-contrôleur MCS251

Le MCS251

Famille des 8051 :

- Conçue par Intel mais existe aussi chez
  - Dallas, Cypress, Atmel, Siemens, Philips, Matra ...

  - Sans doute la famille la plus dépendue dans les systèmes nécessitant peu de performance avec les PIC. (électroménager, périphériques USB, industrie ...)

Le 80251 :

- La génération suivante ; malheureusement, elle ne connaît pas le succès du 8051.
- Est capable d'exécuter des programme 8051 mais possède en plus de nombreuses instructions supplémentaires.

Le MCS251

Le 80251 :

- 8 bits mais adresse 16Mo de mémoire (24 bits)
- 40 registres
- 16 Ko de mémoire OTPROM ou EPROM
- 1 Ko de RAM
- Bus externe de 18 bits (256 Ko)
- 32 lignes E/S

- Pipeline

Le MCS251

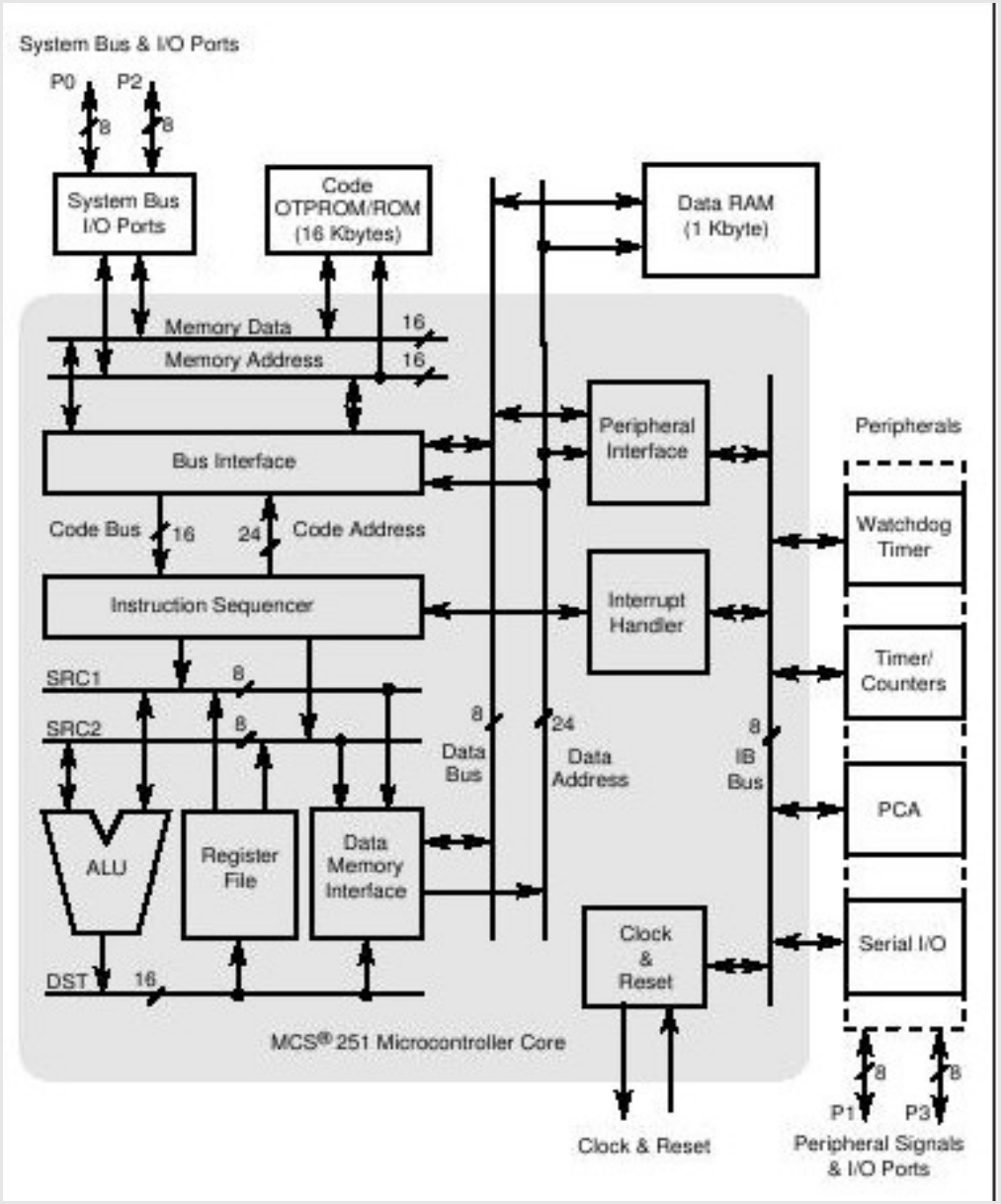
Le 80251 :

- Code bus 16 bits
- Code adresse 24 bits

- Source Data 8b
- Destination Data 16b

- Data adresse 24b
- Data value 8b

- Aussi :
  - Reset
  - Interruptions

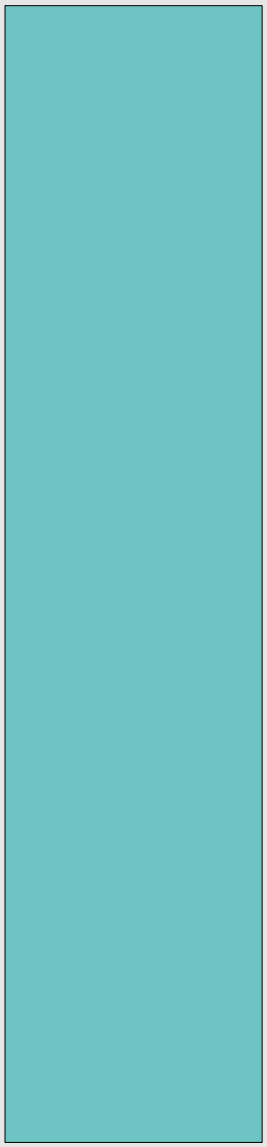


Le MCS251

La mémoire du 80251 :

- Mémoire sur 24 bits
  - Par convention, le premier octet est séparé des suivants par « : »

- Adresses 0 à 63 : Les registres
  - Proche du processeur
  - Accessible plus rapidement que la mémoire classique
  - Comme des variables prédéfinies dont le nom serait de la forme yRxx



Le MCS251- Registres

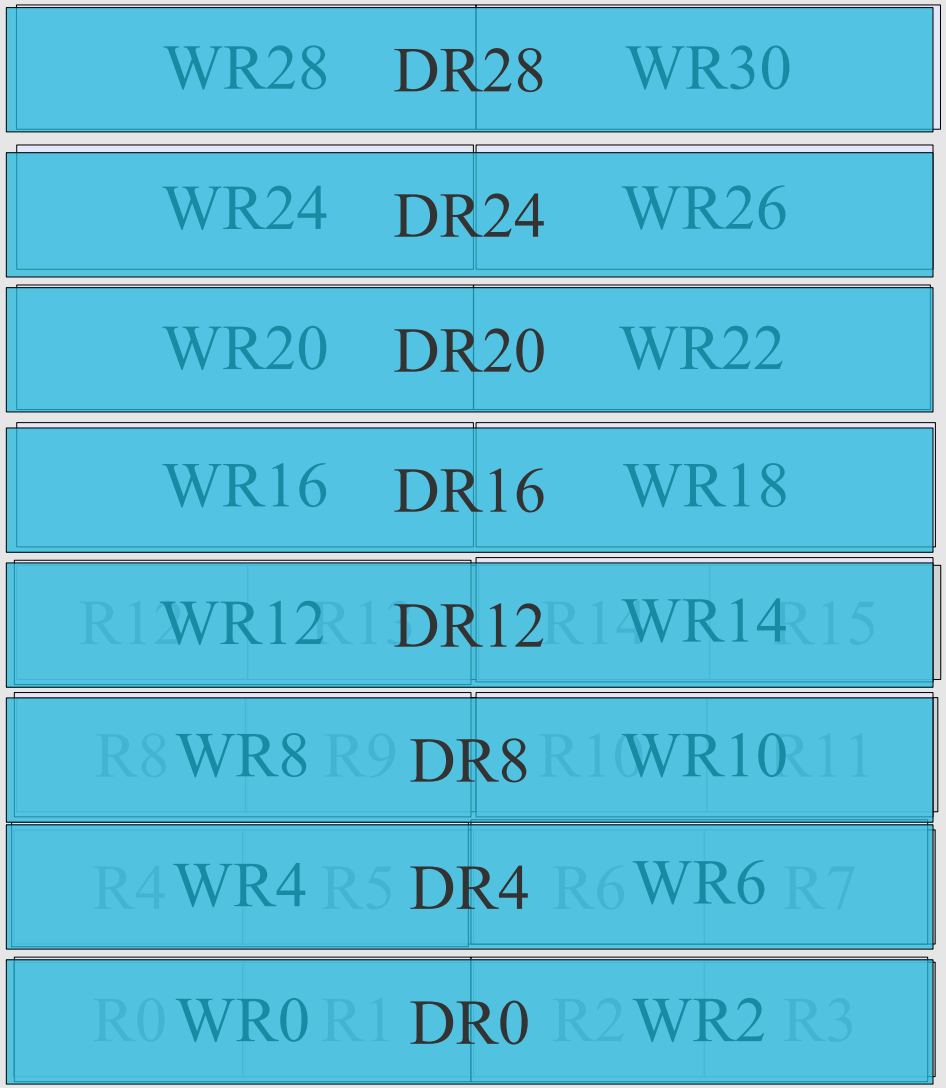
Les registres 80251 :

- Registres 8 bits R0 à R15

- Registres 16 bits WR0 à WR30

- Registres 32 bits DR0 à DR28

Le MCS251- Registres



Le MCS251-Registres

Registres spécifiques du 80251 :

- Registres prioritaires dans les instructions
  - R11 – appelé A comme Accumulateur
  - R10 – appelé B
- Registres de gestion de la pile
  - SP
- Registres d'accès à la mémoire étendue
  - DP

Registres de fonction spéciales SFR :

- 53 registres utilisés pour configurer le micro-contrôleur

Le MCS251-Instructions

Compatibilité avec le 8051 :

- Le 80251 peut être utilisé en mode source ou binaire
  - Le mode binaire privilégie la compatibilité 8051, les nouvelles instructions 80251 seront moins rapides
  - Le mode source privilégie les instructions 80251, les instructions 8051 seront moins rapides

Format des instructions :

- Sans opérande : *opération* exemple ret
- Une opérande : *opération destination* exemple dec R0
- Deux opérandes :

	opération	*destination, source*
exemple :	add	R0, #02

Le MCS251-Instructions

L'instruction MOV :

- MOV *Reg, #direct*
  - écrit dans le registre *Reg* la valeur *direct*

- Exemples :
  - Mov DR0, #3C8F4A1h

R0	R1	R2	R3
03	C8	F4	A1

DR0

  - Mov WR2,#6118h

R0	R1	R2	R3
03	C8	61	18

WR2

  - Mov R0,#20h

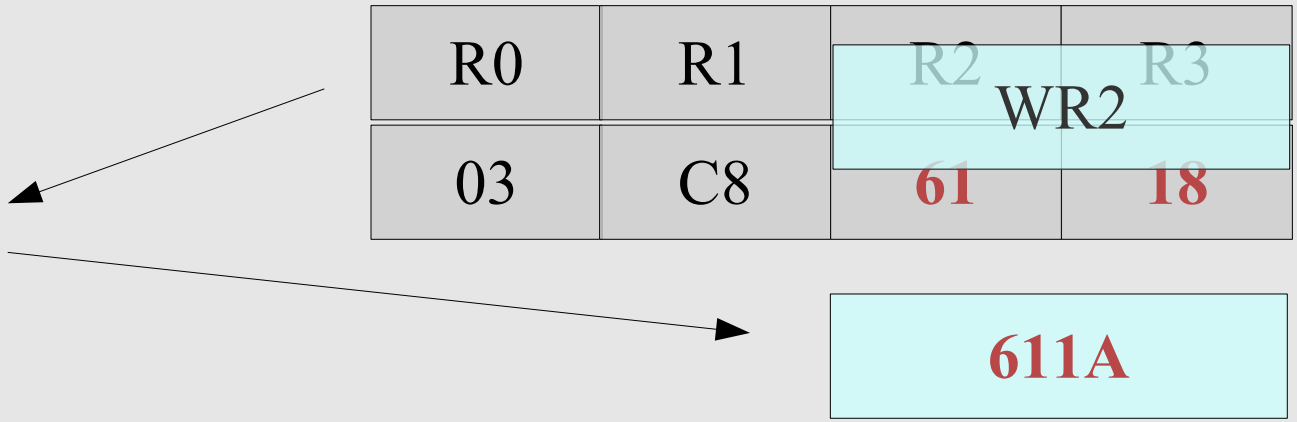
R0	R1	R2	R3
20	C8	61	18

Le MCS251-Instructions

L'instruction ADD :

- ADD *Reg, #direct*
  - Ajoute à la valeur du registre *Reg* la valeur *direct*

- Exemples :
  - ADD WR2, #2h



Le MCS251-Instructions

Utilisation de variables en mémoire :

- Possibilité de définir des variables mémoire dans une zone de programme spécifique : le segment de données
  - Mnémonique DB, DW, DD pour définir un espace mémoire initialisé.
    - exemple : **index DB 0**
      initialise une variable 8b (index) à 0
    - exemple : **tableau DW 0,1,2,3,4**
      décrit un tableau de 5 mots de 16b initialisé avec 0,1,2,3,4
  - Mnémonique DS pour définit un espace mémoire non initialisé.
    - exemple : **tableau2 DS100**
      réserve 100 octets pour un tableau non initialisé

Le MCS251-Instructions

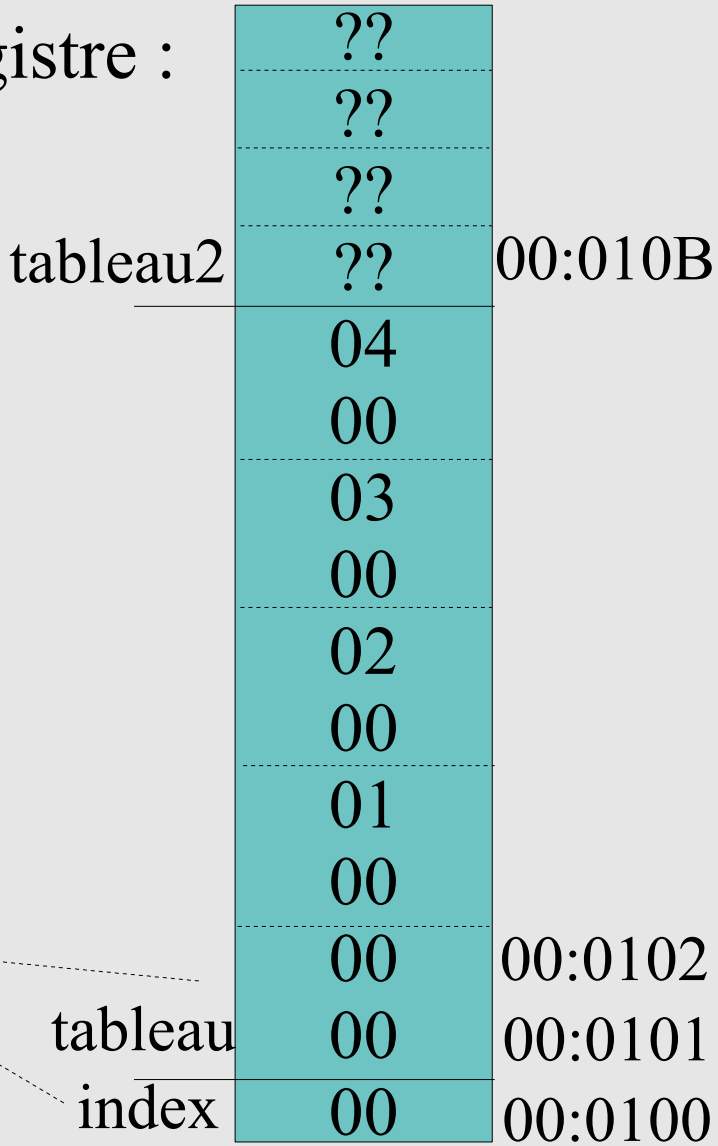
Utilisation de variables en mémoire :

- Charger une variable mémoire dans un registre :
  - MOV R0, index

R0	R1	R2	R3
00	C8	61	1A

  - MOV WR2, tableau

R0	R1	WR2	
03	C8	00	00

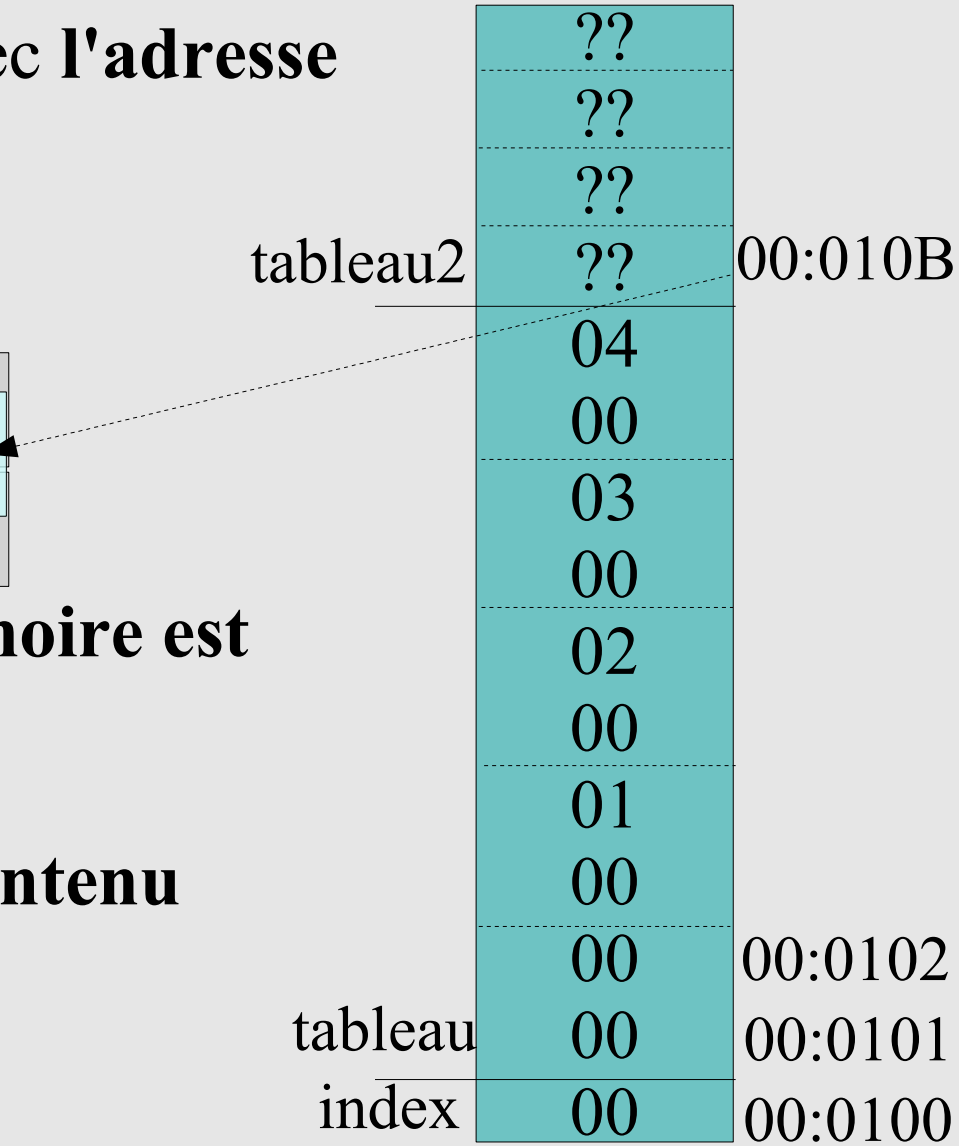


Le MCS251-Instructions

Utilisation de variables en mémoire :

- Charger une variable mémoire avec **l'adresse d'une étiquette :**
  - MOV WR2, #tableau2

R0	R1	R2	R3
		WR2	
03	C8	01	0B



L'adresse d'une variable mémoire est toujours sur 16 bits !!

Ne pas confondre adresse et contenu

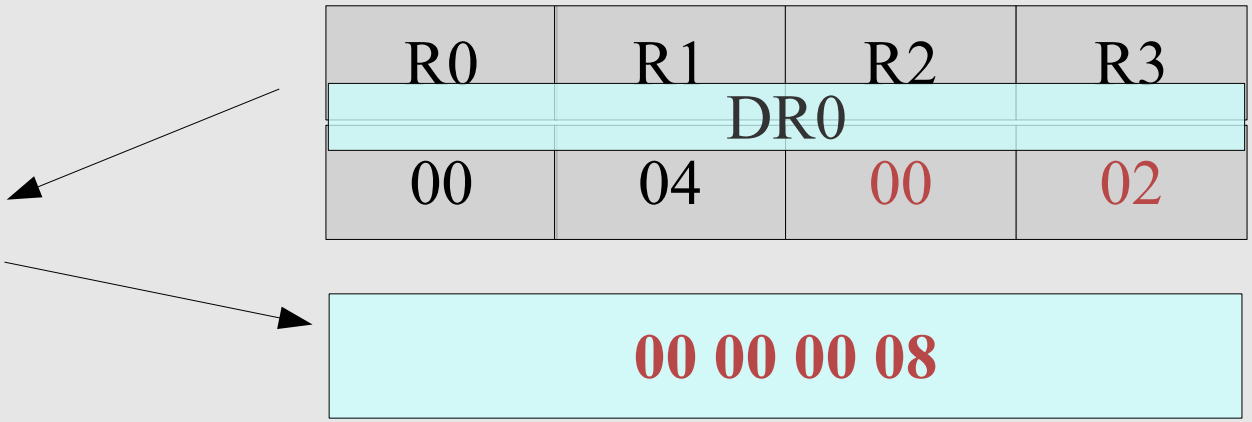
Le MCS251-Instructions

L'instruction MUL :

- MUL *Reg, Reg* *ou* MUL *Wreg, Wreg*
  - Multiplie la destination par la source
  - Le résultat est enregistré dans le registre de taille supérieure contenant la destination

Exemples :

- MUL WR2, WR0



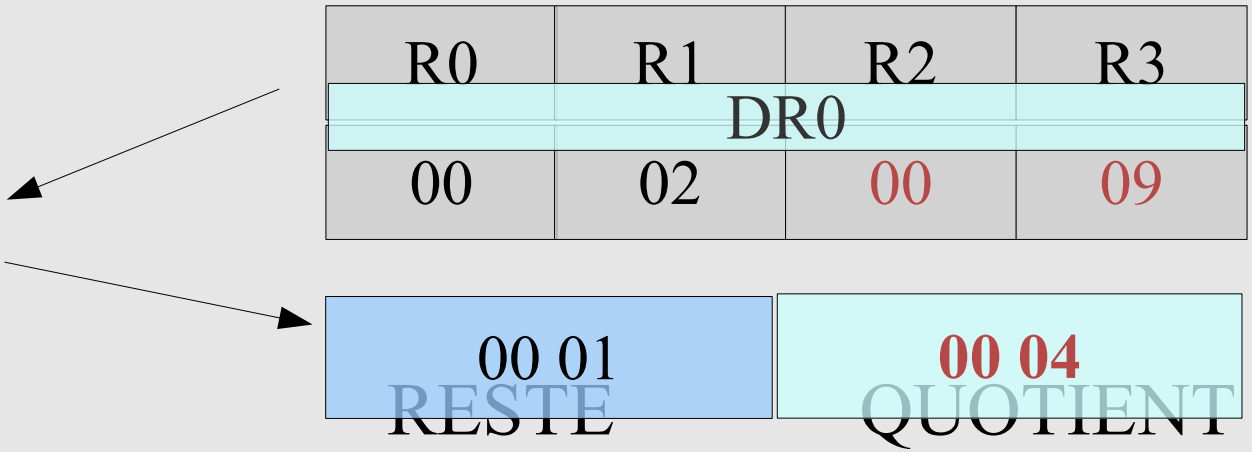
Le MCS251-Instructions

L'instruction DIV :

- DIV *Reg, Reg* *ou* DIV *Wreg, Wreg*
  - Divise la destination par la source
  - Le résultat est enregistré dans le registre de taille supérieure contenant la destination comme suit :
    - Les poids forts contiennent le RESTE
    - Les poids faibles contiennent le QUOTIENT

Exemples :

- DIV WR2, WR0



Le MCS251-Instructions

Exercice :

- Implemetez l'equation suivante (où x et y sont des variables en mémoire, x sur 8b et y sur 16b)

  y = 3*x + 5

- Quel est le temps d'exécution ?

- Quelles est la taille du programme ?

Le MCS251-Instructions

Exercice :

- Implemetez l'equation suivante (où x et y sont des variables en mémoire, x faisant 8b et y 16b)

```
y = 3*x + 5
x:  DB    5
Y: DS     2
```

	temps	taille
Mov R0,x	2	3
Mov R1,#3	2	3
Mul R0,R1	5	2
Add WR0,#5	3	4
Mov y,WR0	4	4
	---	---
	16c	16o

Le MCS251-Instructions

Exercice :

- Implemetez l'equation suivante avec x et y des variables sur 8b

  y = x % 8

- Quel est le temps d'exécution ?
- Quelles est la taille du programme ?

- Implemetez l'equation suivante avec x et y des variables sur 16b

  y = 3*x + 5

- Quel est le temps d'exécution ?
- Quelles est la taille du programme ?

Le MCS251-Instructions

Exercice :

- Implemetez l'equation suivante avec x et y des variables sur 8b

  y = x % 8

	taille	temps
Mov R0,x	3	2
Mov R1,#8	3	2
Div R0,R1	2	10
Mov y,R0	3	3
	---	---
	11	17

Autre solution

Mov R0,x	3	2
ANL R0,#7	3	2
Mov y,R0	3	3
	---	---
	9	7

Le MCS251-Instructions

Exercice :

- Implemetez l'equation suivante avec x et y des variables sur 16b

  y = 3*x + 5

```
x:  DW   5
Y: DS    2
```

		temps	taille
Mov	WR0,x	3	4
Mov	WR2,#3	2	4
Mul	WR0,WR2	11	2
Add	WR2,#5	3	4
Mov	y,WR2	4	4
		---	---
		23c	18o