.:: Architecture ::.

semaine n°5
Masques et Carte MSC251

Exo

• Indiquez quels registres contiendront les paramètres et code retour des fonctions suivantes :

```
int f1(char p1, int p2, char p3, long p4);

char f2(int p1, long p2, char p3, int p4);
```

Traduire en assembleur:

```
int min ( int v1, int v2 ) {
    if ( v1 < v2 )
        return v1;
    else
        return v2;
}
```

Exo

• Indiquez quels registres contiendront les paramètres et code retour des fonctions suivantes :

```
  int f1(char p1, int p2, char p3, long p4);
WR10   R0     WR2   R1       DR4
  char f2(int p1, long p2, char p3, int p4);
R11    WR0   DR4     R2       pile
```

Traduire en assembleur:

```
int min ( int v1, int v2 ) {          _min:
WR10   WR0   WR2
  if ( v1 < v2 )                      if1:    CMP   WR0, WR2
                                              JSGE  else1
     return v1;                       then1:  MOV   WR10, WR0
                                              JMP   endif1
  else                                else1:  MOV   WR10, WR2
     return v2;
                                      endif1: RET
}
```

Le MCS251-Masques

- Exos P 33- P35

Le MCS251-Masques

- Pourquoi les masques ...
  - Des opérations qui permettent de ne modifier qu'un unique bit d'un octet (par exemple)
  - Des opérations pour extraire des bits d'un octet
  - Des opérations pour positionner les bits dans un octet

Carte du MCS251

La carte du MCS251 comprend une carte d'extension sur laquelle nous trouvons :

- DES LEDS (8) --> L'écran
- DES INTERRUPTEURS (8) --> Le clavier

Ces éléments sont des périphériques accessibles au travers d'une adresse mémoire spécifique :

-> LES LED : FFFA00 => Le port P4
-> LES INTERRUPTEURS : FFF900 => Le port P5

L'écriture d'un bit à « 1 » à l'adresse des LEDs éteint la LED correspondante.

La lecture de l'octet à l'adresse des interrupteurs permet de lire leur état.

Carte du MCS251

Exercice : écrire une fonction assembleur qui permet de modifier l'état des LEDs en allumant celle pour lequel le bit est à « 1 »

1 – inverser les bits de la valeur
2 – écrire cette nouvelle valeur sur les LEDs

Prototype de la fonction :
void ecrire(long adresse, char valeur)

Exercice : écrire une fonction assembleur qui lit l'état des interrupteurs

Prototype :
char lire(long adresse)

Carte du MCS251

Exercice : écrire une fonction assembleur qui permet de modifier l'état des LEDs en allumant celle pour lequel le bit est à « 1 »

```
ecrire :    ; DR0  : adresse du port P4
            ; R4   : valeur
            XRL   R4,#0FFh
            MOV   @DR0,R4
            RET

lire :      ; DR0 : adresse du port P5
            MOV   R11,@DR0
            RET
```

Carte du MCS251

Exercice : Ecrire en C un programme qui fait clignoter la led correspondant au poids le plus faible. Les autres led restent éteintes. Vous utiliserez la fonction assembleur « ecrire » pour modifier l'état des leds.

Carte du MCS251

Exercice : Ecrire en C un programme qui fait fait clignoter la led correspondant au poids le plus faible. Les autres led restent éteintes. Vous utiliserez la fonction assembleur « ecrire » pour modifier l'état des leds.

```
#define     P4  0xFFFA00
main() {
     While(1) {
          ecrire(P4,0x01);        // allume la led
          ecrire(P4,0x00);        // éteint la led
     }
}
```

=> Problème n°1 : le programme sera trop rapide !

Carte du MCS251

Exercice : Ecrire une fonction d'attente

```
attente() {

}
```

Carte du MCS251

Exercice : Ecrire une fonction d'attente

```
attente() {
   for ( int i = 0 ; i < 20 ; i++ )
     for (int j=0 ; j < 32000 ; j++ );
}

main() {
      While(1) {
            ecrire(P4,0x01);          // allume la led
            attente();
            ecrire(P4,0x00);          // éteint la led
            attente();
      }
}
```

Carte du MCS251

Exercice : Ecrire une fonction qui permet de faire clignoter les 4 led de poids faible sans modifier les 4 led de poids fort ....

```
clignotte() {

}
```

Carte du MCS251

Exercice : Ecrire une fonction qui permet de faire clignoter les 4 led de poids faible sans modifier les 4 led de poids fort ....

```
unsigned char etatLed;

void ecrireLed(unsigned char v) {
    etatLed=v;
    ecrire(P4,v);
}

unsigned char lireLed(){
   return etatLed;
}

clignotte() {
   ecrireLed(lireLed() ^ 0x0F);
}
```

Carte du MCS251

Voir poly sur carte du MCS251