

TD3 : Programmation en assembleur 8051 (1/2)

Exercice 1 :

Écrire un programme qui calcule l'addition et la soustraction de deux données stockées dans R0 et R1. Les résultats sont stockés dans les registres R2 et R3 ($R2=R0+R1$ et $R3=R0-R1$). Les pins P1.0 et P1.1 sont reliées à des LED qui vont jouer le rôle d'indicateurs de dépassement de format. La première LED pour l'addition et l'autre pour la soustraction. Développer deux versions de ce programme pour les 2 types de codage :

1. Version 1 : Codage non-signé.
2. Version 2 : Codage signé.

Exercice 2 :

Écrire un programme qui va trouver le PGDC (Plus Grand Diviseur Commun) de deux entiers non-signés stockés dans R0 et R1. Le PGDC trouvé est stocké dans R2.

Exercice 3 :

Écrire un programme qui réalise un multiplexeur 2×1 . Les deux entrées sont P0 et P1 et la sortie est P2. L'entrée adresse (ou de sélection) est la pin P3.0. Le multiplexeur fonctionne comme suit : si P3.0=1 alors P2=P1, sinon P2=P0.

Exercice 4 :

Écrire un programme qui réalise un décodeur 3×8 (3 bits d'adresse). L'entrée adresse est codée avec les 3 pins de poids faibles de P0 (du MSB=P0.2 au LSB=P0.0). Les sorties (actives à 1) sont les 8 bits du port P1 (du P1.7 au P1.0). Son fonctionnement est comme suit : selon la valeur de l'entrée adresse = X en active seulement la sortie d'indice P1.X (les autres sont inactives). Par exemple si X=0, on active seulement P1.0.

Exercice 5 :

Écrire un programme qui calcule l'addition de données lues via le port P0 et P1 et transfère le résultat au port P2. Ce programme réalise ce traitement à l'infini.

Exercice 6 :

Écrire un programme qui compare 2 données signées qui sont stockées dans R0 et R1. Une LED est reliée à P1.0 qui joue le rôle d'indicateur de comparaison. Si $R0 \geq R1$, la LED est allumée ($P1.0 = 1$) sinon elle est éteinte ($P1.0 = 0$). On peut définir un symbole LED=P1.0 avec la directive BIT.

Exercice 7 :

Écrire un programme qui compte le nombre de 1 dans une donnée stockée dans le registre R0 (exemple : R0=88H). Ce nombre trouvé est stocké dans R1 (pour l'exemple : R1=2).

Exercice 8 :

Écrire un programme qui cherche tous les diviseurs d'un entier non-signé défini par "Nbre EQU 50;" (le nombre 50 est un exemple). Les diviseurs sont stockés dans la RAM interne à partir de l'adresse 20H.