Module Circuits numériques et microcontrôleurs

Supports de cours et TPs autorisés. Durée : 1H30

Exercice 1 :

Répondre au QCM de l'annexe.

Exercice 2 :

A- Soit la mémoire interne d'un microcontrôleur à base d'un processeur Nios II, organisée comme indiqué dans le tableau suivant :

Adresse mémoire :	i	i+1	i+2	i+3	i+4	I+5	i+6	I+7	i+8
Contenu	82h	E3h	42h	06h	CAh	0Bh	0Bh	10h	30h

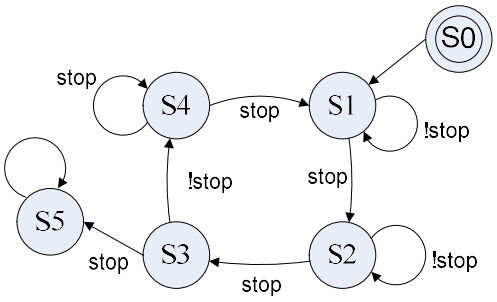
.

1) Une variable *int X* est implantée à partir de l'adresse *i+2*. Indiquez dans ce cas la valeur de X en décimal.
2) Une variable *short Y* est implantée à l'adresse *i*. Indiquez sa valeur en décimal.

B- Considérons une variable *char Z*. Nous souhaitons modifier cette variable de la manière suivante : mettre à 1 les bits 3, 4 et 5 et mettre à 0 les bits 6 et 7 de cette variable *Z*. Les bits 0 à 2 de *Z* restent inchangés. Proposer l'expression en langage C pour effectuer cette modification sur Z.

C- Considérons le graphe de l'automate suivant :

1) Indiquer si l'automate est déterministe. Justifiez la réponse.
2) Indiquez si cet automate est vivant. Justifiez la réponse.



Exercice 3 :

On supposera que les fonctions **detect_touche()** et **decode_touche(…)** vous sont données.

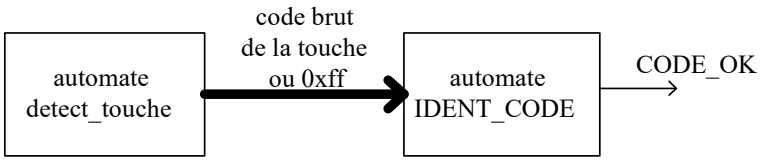
char detect_touche( ); */* détecte une transition "appui sur une touche" et retourne le code position dans la matrice clavier, dans le cas contraire : retourne 0xFF */*
char decode_touche(char c); */* conversion d'un code "position dans la matrice clavier" en une valeur : binaire si la touche est un chiffre et code ascii si * ou # */*

On considère dans cette partie que la gestion et le décodage du clavier est réalisé par logiciel par les deux fonctions ci-dessus. La fonction **detect_touche( )** met en œuvre l'automate de détection d'appui sur une touche.

On souhaite utiliser une carte à base d'un microcontrôleur pour réaliser un système d'identification d'un code secret entré au clavier en décimal. L'utilisateur devra entrer un code sur 3 chiffres suivis de la touche # pour

valider son code. Le bon code (472) est enregistré dans l'EPROM de la carte microcontrôleur dans un tableau de constantes (SECRET[]).

Lorsque le code composé par l'utilisateur est bon, un signal sortant **CODE_OK** (sur un pin d'entrée/sortie) du microcontrôleur sera positionné à 1 puis repassera à 0. Il n'y a pas de contrainte de temps sur le niveau logique à 1 de **CODE_OK**.



1) Donner l'automate **IDENT_CODE** permettant de lire les 3 digits suivis de la touche #, de vérifier que le code entré est bon et de le signaler avec **CODE_OK**. S'il y a une fausse manipulation ou si la comparaison avec le code secret a échoué, alors on positionnera l'automate dans son état initial.

2) Sans donner les détails de codage en C de la fonction IDENT_CODE(…), préciser les paramètres passés à cette fonction non bloquante.

3) Donner le corps (en C) du programme principal en précisant essentiellement comment sont appelées les fonctions **detect_touche, decode_touche, IDENT_CODE** et les variables intermédiaires nécessaires.

Exercice 4 :

On dispose d'un chronogramme d'échange de données sur un bus I2C composé d'un microcontrôleur (type 80C592) et de plusieurs périphériques esclaves, tous reliés sur ce bus I2C. Le microcontrôleur est le maître du bus. Les signaux SDA et SCL sont implantés sur les pattes P1.7 et P1.6 du microcontrôleur 80C592.

1) Expliquez la signification de la terminologie "périphérique esclave" pour le bus I2C.
2) Précisez les conditions électriques sur les pattes 6 et 7 du port P1 pour pouvoir implanter un bus I2C.

Le chronogramme de la figure de l'annexe (**voir dernière page**) donne les signaux SCL et SDA du bus I2C dans le temps.

3) Identifiez sur ce chronogramme les séquences START et STOP du protocole I2C. **(Répondre en utilisant le chronogramme de l'annexe qu'il faudra rendre avec votre copie)**
4) Identifiez sur ce chronogramme les Adresses des esclaves sollicités et précisez l'action demandée sur les périphériques (Lecture ou Écriture).
5) Précisez si les esclaves sollicités sont présents sur le bus. Justifiez votre réponse.

Exercice 5 :

Les applications à base de microprocesseur nécessitent des bus d'interfaces pour connecter des cartes d'entrées/sorties additionnelles. On considère une carte à base d'un processeur Nios II. On souhaite adjoindre à la carte un circuit de Conversion Numérique->Analogique. Ce CNA permet de sortir une tension comprise **entre 0** et **4** volts à partir d'une donnée numérique sur 8 bits.

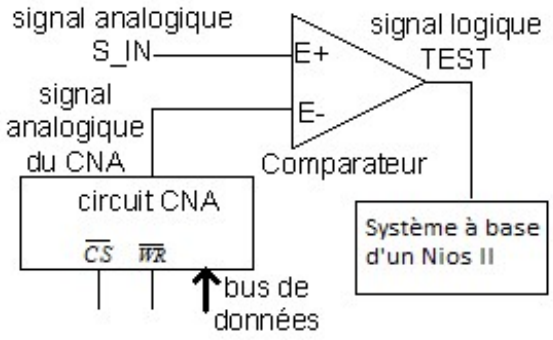
1) Donner la résolution en sortie du CNA (la plus petite tension pouvant être générée).

La carte microcontrôleur dispose d'un certain nombre de circuits annexes occuppant le plan mémoire externe du microprocesseur. Ce plan d'organisation indique une zone mémoire libre à l'adresse 00007FFFH.

2) Donner la déclaration de l'adresse du CNA pour une implantation en langage C avec un processeur Nios II.

On souhaite réaliser un convertisseur analogique/numérique par logiciel en utilisant le CNA. On réalise pour cela le montage ci-dessous.

Le principe de la conversion analogique/numérique consiste à contrôler le circuit CNA de sorte que sa tension analogique soit comparée avec le signal d'entrée (qu'on souhaite convertir en numérique) et on arrête la conversion lorsque ces deux signaux analogiques sont proches. La dernière valeur numérique envoyé dans le CNA correspondra à la tension analogique d'entrée sur S_IN. Le signal **TEST** du comparateur sera à 1 lorsque (E_+ **-** E-)≥ 0 sinon le signal TEST sera à 0 (ce signal TEST pourra être câblé sur un pin d'entrée/sortie d'un FPGA implantant un system Nios II).



On souhaite écrire une fonction CONV, en langage C, permettant de convertir le signal analogique **S_IN** en une valeur numérique. L'algorithme de conversion sera similaire à celui utilisé dans les convertisseurs analogiques/numériques à comptage.

3) Proposer la mise en œuvre de cette fonction CONV et indiquer le type de la valeur retournée.

Exercice 6 :

On souhaite réaliser un système permettant de calculer le PGCD (Plus Grand Diviseur Commun) de deux nombres entiers positifs, dont l'algorithme est le suivant :

```
Entrer A et B; /* deux entiers dont on souhaite calculer le PGCD*/

Tantque A!=B faire
        Si (A>B) alors
                A=A-B;
            Sinon
                Si (A<B) alors
                        B=B-A;
                Finsi;
        Finsi;
Fin_tantque;
S= A;  /* S résultat : le plus grand diviseur commun des données entrées */
```

1) Déduire, à partir de l'algorithme, une architecture de chemins de données permettant de mettre en œuvre un tel algorithme. Mettre en évidence tous les signaux de contrôle de cette architecture et les signaux de sorties ainsi que les signaux destinés à l'automate de contrôle.

2) Proposer une synchronisation des échanges entre le système et son interface avec l'extérieur (signal de début et fin de calcul). Représentez, sous la forme d'un timing simple, le déroulement des échanges.

3) Donner l'automate de contrôle optimisé du système en prenant en compte la synchronisation des Entrées/Sorties. Vous ne devez donner que le graphe de l'automate (en clair).

On constate que lorsqu'on présente en entrée du système deux données X et Y tel que : **X=q*Y+r**
Le système va itérer q fois de suite sur la condition A>B (ou bien A<B) de l'algorithme précédent. Nous supposons disposer d'un opérateur noté : MOD. Cet opérateur est le MODULO :

X MOD Y = r **--->** r représente le reste de la division de X par Y.

4) Proposer une modification de l'algorithme initial en supposant disposer d'un opérateur MODULO.

ANNEXE

Réponse à l'exercice 1 :

Répondre aux questions suivantes :

	OUI	NON
A) Une transmission série RS232 permet d'envoyer des données sous forme de 8 bits simultanément ?		
B) Un convertisseur Analogique->Numérique est un amplificateur analogique dont le gain est commandé par une valeur numérique ?		
C) A partir d'une description sous forme d'un graphe, il est impossible de la réaliser sous forme d'une description en VHDL ?		
D) Un circuit programmable est une structure matérielle qui permet d'implanter des bus bidirectionnels ?		
E) Le processeur Nios II est un circuit ASIC avec un bus d'adresses sur 32 bits ?		
F) Le processeur Nios II a une architecture organisée en BIG ENDIAN ?		
G) Un programme écrit en C est séparé en deux parties : partie calcul et partie Entrées/Sorties. Ce même programme n'est pas portable d'un microprocesseur vers un autre car les deux parties doivent être réécrites ?		

Réponse à l'exercice 4 :