Micro-architecture (UE S3)

Dossier de Projet

Serpentin 7-segment programmable (FPGA)

Etudiants :
Afizullah Rahmany
Romain Pereira

Enseignant :
M. Augé

25/10/2018

Table des matières

Préambule

Ce projet a été réalisé dans le cadre de nos études à l'ENSIIE (Ecole National Supérieur d'informatique pour l'industrie et l'entreprise) d'Evry.

Ce projet est l'aboustissement de nos cours en Micro-architecture.

Nous programmions sur un FPGA d'Altera (gamme Cyclone), en VHDL et à l'aide du logiciel Quartus.

L'objectif a été de réalisé un serpentin programmable, et affichable sur un 7-segment.

1 Introduction

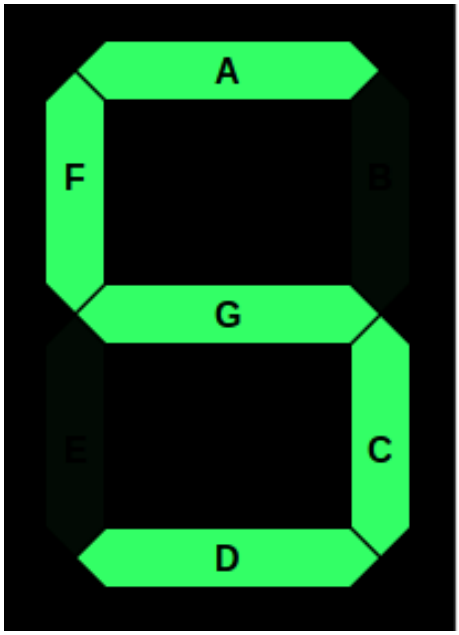
Vous cherchez un cadeau de noël pour vos enfants ou vos grands parents ?
Ne cherchez plus, voici **le Serpentin 3000**
Cette modeste plaquette electronique passionera les petits comme les grands enfants.

Amusez vous !
Modifier la vitesse et le type de défilement.

Exprimez votre créativité !
Grâce à l'interface 'Programmation 3000', vous pouvez créer votre propre serpentin.

Accessible Pour la modique somme de 29.99, offrez vous un

SERPENTIN 3000



2 Manuel utilisateur

3 Présentation générale

3.1 Fonctionnement

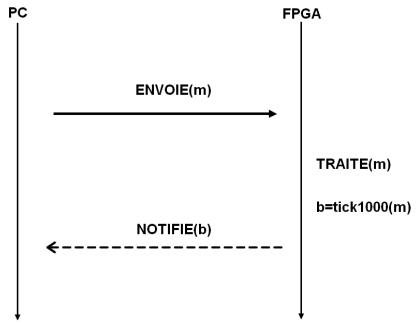


FIGURE 1 – MSC des actions utilisateur

m est un *message*. Il s'agit d'une suite de **44 bits** avec le format :

- **bits 43 à 24** Ce sont les bits du contrôle du BUS-IA (voir Bus-IA)
- **bits 23 à 0** Ce sont les bits du message

Un *message* doit avoir le format :

- **bits 23 à 21** Ce sont les bits coddant le *type* du *message*
- **bits 20 à 0** Ce sont les données du message, le format varie selon le *type*

3.2 Format des messages

Ci-dessous, voici les commandes supportées par le processeur, et le format des messages permettant d'executer ces commandes.

noop Ne rien faire

- **bits 23 à 21** : "000" (= (bit 23, bit 22, bit 21))
- **bits 20 à 0** : *non utilisées*

h-init(n) Génère un tick sur H100 tous les n coups d'horloge maître

- **bits 23 à 21** : "001"
- **bits 20 à 0** : *non utilisées*

h-check-ON() Demande au processeur d'envoyer au PC un message TICK1000 tous les 1000 tickets de H100.

- **bits 23 à 21** : "010"
- **bits 20 à 0** : *non utilisées*

h-check-OFF() Demande au processeur d'arrêter d'envoyer au PC un message TICK1000

- **bits 23 à 21** : "011"

- **bits 20 à 0** : *non utilisées*

clr() Indique au processeur d'afficher un serpentin vide sur le 7-segment (**n** à 32 et met à 0 toutes les *valeurs* programmées

- **bits 23 à 21** : "100"

- **bits 20 à 0** : *non utilisées*

set-N(n) Indique au processeur le nombre de **valeurs** à faire défiler en boucle sur le 7-segments. (**n** compris entre 1 et 32)

- **bits 23 à 21** : "101"

- **bits 20 à 6** : *non utilisées*

- **bits 5 à 0** : **n** coddé sur 6 bits ($2^6 = 64$)

set-val(i, v) Indique au processeur la i-ème valeur du 7-segments est **v**

- **bits 23 à 21** : "110"

- **bits 20 à 13** : *non utilisées*

- **bits 12 à 6** : **v** les 7 bits correspond aux 7 segments de l'afficheur (0 => éteinds, 1 => allumé)

- **bits 5 à 0** : **n** coddé sur 6 bits ($2^6 = 64$)

4 Description matérielle

5 Présentation générale

5.1 Schéma général

5.2 Description des blocs

6 Implémentation du bloc ? ? ? ?