

ARC2 – TP 7

Prune Forget, Léo Noël-Baron & Thierry Sampaio

16/03/2016

Ce TP propose de gérer un clavier d'entrée sur le BihaNios.

Connexions au bus

On choisit d'utiliser pour l'entrée les interrupteurs `SW[14..7]`, que l'on doit rediriger sur le `Bus[7..0]` si l'adresse demandée est `0x8000`. L'adressage se fait donc en conjuguant `Lec` et `MA[31]`, comme exposé en figure 1.

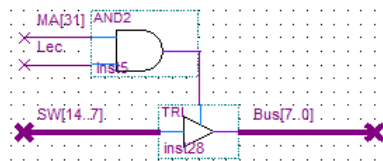


FIGURE 1 – Interface sans synchronisation

Programme de test

Voici le programme utilisé pour tester l'interface de base :

```
1 .equ AddLec 0x8000
2 debut:
3     addi r6,r0,0 ; Initialisation
4 while:
5     ; Lecture de l'entrée clavier dans r9
6     ldw r9,AddLec(r0)
7     andi r9,r9,0xFF ; Masque pour prendre seulement 8 bits
8     ; Somme et affichage
9     add r6,r6,r9
10    stw r6,1024(r0)
11    br while
12
13 .end debut
```

En exécutant le programme pas à pas, on peut basculer l'entrée à 0 une fois qu'elle a été interprétée. En continu, cela devient impossible et les nombres s'additionnent anarchiquement.

Interface avec bit d'état

On réalise un bit d'état au moyen de deux portes NOR (R en haut, S en bas) ; le bouton NKEY[2] doit le mettre à 1 et il doit être lu dans Bus[0] à l'adresse 0x8001, c'est-à-dire quand $\text{Lec} \sim \text{MA}[15] \sim \text{MA}[0]$. Il doit être remis à 0 soit manuellement par le bouton NKEY[0], soit après lecture de l'entrée (on reprend donc le signal de lecture précédent, en tenant compte cette fois de MA[0]). On obtient le circuit en Figure 2.

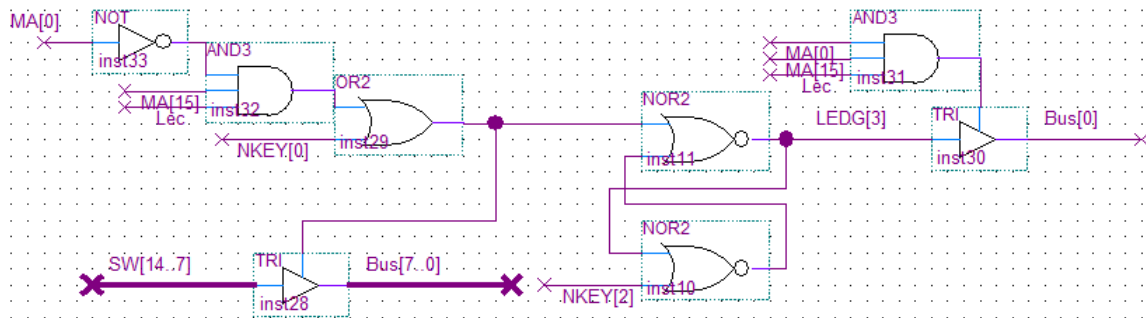


FIGURE 2 – Interface avec synchronisation

On a maintenant un bit d'état activé manuellement pour signaler qu'une entrée est prête, qui se désactive lorsque l'entrée a été lue. Ceci permet de modifier le programme précédent pour y rajouter une attente active sur ce bit d'état :

```

1 .equ AddLec 0x8000
2 .equ State 0x8001 ; Bit d'état
3 debut:
4     addi r6,r0,0
5 while:
6     ; On commence par charger le bit d'état
7     ldw r9,State(r0)
8     andi r9,r9,0x1 ; Masque sur le bit de poids faible
9     ; S'il n'y a rien à lire, on boucle directement
10    beq r9,r0,while
11    ; Sinon, on lit l'entrée et on l'interprète comme avant
12    ldw r9,AddLec(r0)
13    andi r9,r9,0xFF
14    add r6,r6,r9
15    stw r6,1024(r0)
16    br while
17
18 .end debut

```

Le programme se comporte maintenant comme on le souhaite ; chaque entrée n'est lue qu'une seule fois, lorsque la frappe est signalée.