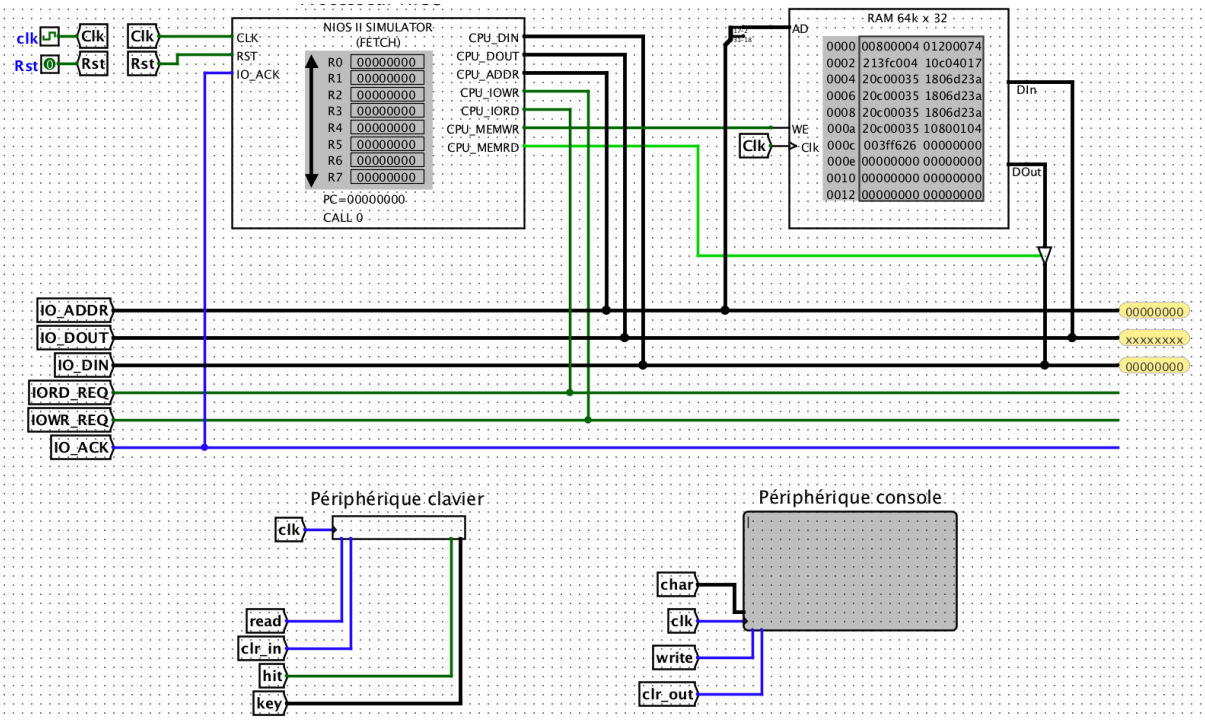
TP 5-6

Entrées/sorties physique

L'objectif de ce TP est de mettre en œuvre à l'aide de Logisim un système matériel/logiciel basé sur un processeur NIOS-II. Ce système devra permettre d'effectuer des opérations d'entrée/sorties simples (lecture de caractères sur un clavier virtuel, et affichage de texte sur une console virtuelle).

Récupérez dans votre espace de travail le répertoire **`/share/l3info/ILM/TP/TP5-6.`**

Ouvrez à l'aide de Logisim (Logisim se trouve dans le répertoire **`/share/l3info/ILM/TP`**) le circuit **`tp56.circ`** disponible dans ce répertoire **`TP5-6`**, celui-ci représenté ci-dessous, contient un composant NIOS-II, une mémoire RAM de 64ko, deux périphériques (clavier, console de sortie) et expose les différents signaux du bus d'entrée/sortie du processeur.



Au cours de ce TP, vous serez amené à concevoir la logique d'interface de ces deux coupleurs, et de la contrôler via des instructions du processeur NIOS-II. ***Il est conseillé de relire le TD4 avant d'attaquer le TP.***

Prise en main

Vous allez commencer par observer en mode pas à pas le fonctionnement du processeur NIOS-II pour l'exécution du programme stocké dans la mémoire. Pour ce faire, activez la combinaison CTRL-T qui permet de passer au top d'horloge suivant. Observez l'évolution du fonctionnement du programme (valeur de **`PC`**, instruction exécutée, valeur des registres **`R0-R31`**), et expliquez ce que fait le programme stocké en mémoire.

On souhaite modifier ce programme afin d'utiliser les instructions d'E/S du processeur. Pour ce faire, ouvrez le fichier **`tp5_1.s`** dans un éditeur de texte et analysez son contenu. Expliquez ce que

fait ce programme.

Ouvrez maintenant un terminal de commande ***cmd***, et placez-vous dans votre copie du répertoire **`TP56`**. Lancez le script **`nios-compiler.bat`** pour ajouter le compilateur au **`PATH`**. Celui-ci doit normalement contenir un fichier **`Makefile`**. A quoi correspondent les commandes utilisées dans le fichier **`Makefile`** ? Exécutez-le à l'aide de la commande **`make SRC=tp5_1`**. A l'issue de l'exécution, vous pourrez constater la présence d'un fichier d'extension **`tp5_1.bin`**, qui contient l'image binaire du programme assembleur **`tp5_1.s`**.

Vous chargerez ensuite ce fichier dans le composant RAM du circuit **`tp56.circ`**. Pour ce faire, il faut éditer le contenu de la mémoire (cliquez sur l'entrée *contenu* du menu *propriétés,* fenêtre du bas à gauche). Vous verrez s'ouvrir une fenêtre permettant d'éditer le contenu de la mémoire. Sélectionnez l'image à charger via le bouton *ouvrir* (Si le bouton *ouvrir* n'apparaît pas*,* prenez soin d'ouvrir cette fenêtre en plein écran*)*. Il faut ensuite réinitialiser l'exécution du processeur en forçant le signal **`rst`** à 1. Faites une exécution pas à pas, que constater vous ? Comment expliquez-vous que le processeur reste bloqué sur l'exécution de l'instruction **`stbio r3,(r2)`**?

Partie I : affichage

Réalisation d'un coupleur pour l'affichage de caractères

Le coupleur d'affichage utilisé dans ce système sera affecté à la plage d'adresses **`[0x80007040-0x80007043]`**. Le coupleur ne dispose que d'un port d'E/S qui n'est accessible qu'en écriture pour des mots de 32 bits (les autres opérations seront ignorées, mais vous veillerez à acquitter ces transactions afin d'éviter tout inter-blocage).

Port E/S	Fonction
`0`	31 … 16 15 14 13 12 11 10 9 8 **`clr`** 7 6 5 4 3 2 1 0 — **`ascii[6:0]`**

Les bits **`ascii`** contiennent le code ASCII du caractère à afficher, le bit **`clr`** permet d'effacer le contenu de la console.

Le coupleur pilote directement le périphérique de sortie (signaux **`char`**, **`write`**, **`clr_out`**), comme illustré sur le schéma de la page 1.

Travail à effectuer :

1. Réalisez le circuit de décodage d'adresse associé au coupleur de sortie, puis vérifiez son bon fonctionnement, en modifiant le fichier **`tp5_1.s`** afin de lui faire exécuter des instructions d'E/S sur cette plage d'adresses.
   Attention : n'oubliez pas de relancer l'assemblage (via le **`Makefile`**) et de recharger la nouvelle image dans la RAM du système après chaque modification du code assembleur !
2. Concevez la machine à état capable de reconnaître/acquitter les transactions d'E/S concernant ce coupleur, et de commander le périphérique de sortie.
3. Complétez le fichier **`tp5_1.s`** avec la mise en œuvre des fonctions **`putc`**, **`clear`** et **`print_string(char* s)`** définies ci-dessous. Utilisez ces fonctions pour valider l'acquittement des transactions, la bonne saisie à la console et son effacement (*clear*).
   - **`void putc(char c)`** : affiche le caractère **`c`** sur la console de sortie.
   - **`void clear()`** : efface le contenu de la console

- **`void print_string(char* s)`** : affiche la chaîne de caractères **`s`** sur la console jusqu'au caractère **`null.`**

Partie II : saisie

Mise en œuvre d'un coupleur pour la saisie de caractères

Le coupleur clavier utilisé dans ce système sera affecté à la plage d'adresses **`[0x8000FF00-0x8000FF07]`**. Attention l'interface avec le clavier fonctionne différemment du TD et n'utilise pas de protocole **`req/ack`**. En particulier le signal de lecture **`read`** ne doit rester actif que pendant un seul cycle, car sinon plusieurs caractères sont consommés.

Les ports d'E/S de ce coupleur ne sont accessibles qu'en lecture, on considèrera que les écritures sont sans effet. Vous veillerez cependant à acquitter les transactions d'écriture afin d'éviter les inter-blocages matériels (par exemple en cas d'accès à une adresse erronée).

Port E/S	Fonction
0	31 … 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 — **unused** (31…8) — **Data** (7…0)
1	31 … 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 — **hit** (0)

Le coupleur pilote directement le périphérique de sortie (signaux **`char`**, **`write`**, **`clr_in`**), comme illustré sur le schéma de la page 1.

Travail à effectuer :

4. Réalisez le circuit de décodage d'adresse associé au coupleur de sortie, puis vérifiez son bon fonctionnement.
5. Concevez la machine à état capable de reconnaître/acquitter les transactions d'E/S concernant ce coupleur, et de commander le périphérique d'entrée.
6. Complétez le fichier **`tp5_1.s`** avec la mise en œuvre de la fonctions **`getc`** et **`echo_string`** définie ci-dessous. Utilisez ces fonctions pour valider l'acquittement des transactions, la bonne lecture du clavier et l'effacement des caractères lus.
   - **`char getc()`** : lit un caractère au clavier, ne rend la main que lorsqu'un caractère est disponible (fonctionne par attente active).
   - **`void echo_string()`** : Répète les caractères saisis au clavier sur la console jusqu'à atteindre un retour de ligne.

   Bonus : écrire une fonction **`read_string(char* s, int maxcount)`** qui lit les caractères saisis au clavier et les écrit en mémoire **`s`** jusqu'à atteindre un retour de ligne ou le nombre maximum de caractères **`maxcount`**.