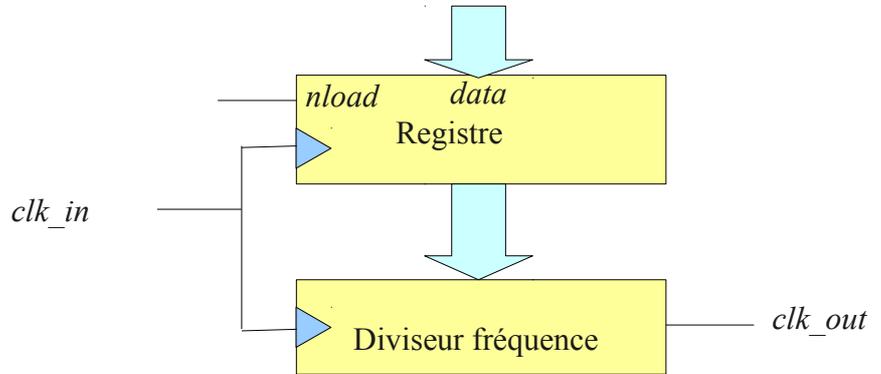


**MI41 - Final**

**1. VHDL(6)**

On souhaite réaliser un composant logique permettant de générer une division de fréquence programmable. Le facteur de division est une donnée 32 bits nommée *data*. Ce facteur de division est chargé dans un registre 32 bits à l'aide d'une commande de chargement active sur niveau bas appelée *nload*. Le composant prend en entrée une horloge de fréquence *f0* noté *clk\_in* et génère en sortie un signal d'horloge notée *clk\_out* de fréquence  $f=f0/data$ . L'horloge de sortie devra être au plus proche d'un rapport cyclique 1/2 (i.e. : le temps bas est aussi long que le temps haut). Si la valeur dans le registre est inférieure ou égale à 2 elle sera considérée comme égale à 2 (la fréquence sera divisée au moins par 2).

Schéma :



Donnez la description VHDL du composant décrit précédemment. Les deux sous composants (registre et diviseur fréquence) apparaîtront dans la même architecture sous la forme de 2 process.

**2. Graphe à état (7)**

On souhaite commander un dispositif à l'aide de 2 boutons poussoirs notés A et B. les commandes sont notées S1 et S0. Une commande de Forçage à 0 notée *nrst* met le dispositif de commande dans son état initial avec S1=S0=0.

De l'état initial, si B est activé, il ne se passe rien, S1 = S0 = 0.

Lorsque A est activé, S1 S0 = 01, si B est activé ensuite alors S1 S0 = 10, si au contraire c'est encore A qui est activé la commande de S1= S0 = 1. Dans les deux cas si B est activé ensuite, on revient dans l'état initial avec S1 = S0 = 0

L'activation d'un bouton (appui) n'est valide que s'il était désactivé auparavant. (ie : les appuis longs ne doivent permettre l'évolution que d'un seul état). A est toujours prioritaire sur B.

Le système de commande est un système synchrone, synchronisé par un signal d'horloge nommé *clk*. La valeur des boutons, n'est échantillonnée qu'au front montant de l'horloge. On suppose la fréquence d'horloge très rapide devant la durée d'un appui.



Il s'agit de réaliser un composant permettant de générer les commandes S1 et S0. Le séquenceur sera réalisé à l'aide d'une machine à états de Moore.

1. Représentez le graphe à états modélisant le comportement du séquenceur de commandes décrit.
2. Donnez la description vhdl du séquenceur de commande modélisé par le graphe à état précédemment établi
3. Synthétisez la description précédente en utilisant des portes logiques et des bascules D (avec reset et preset intégré). Vous donnerez le schéma complet.

**3. ASM (7)**

Les fonctions suivantes, suivent la convention d'appel ARM standard (ARM Architecture Procedure Call Standard [AAPCS]).

1. Ecrire une fonction qui détermine la plus grande valeur dans un tableau d'entiers A. La fonction renvoie l'adresse du plus grand nombre. Si le tableau contient plusieurs maxima, la fonction renverra l'adresse du premier maximum rencontré.

Prototype : `int * sdresse_max(int* A, int N); /* N : taille du tableau */`

2. Ecrire une fonction permettant de transformer une chaîne de caractère se terminant par zéro en un nombre binaire. La chaîne de caractères correspondant à un nombre positif. La fonction devra renvoyer la valeur binaire de ce nombre. On rappelle que pour le codage ascii, les nombres sont compris entre 0x30 (code du 0) et 0x39 (code du 9) dans l'ordre.

Prototype : `int atoi (char* chaine);`