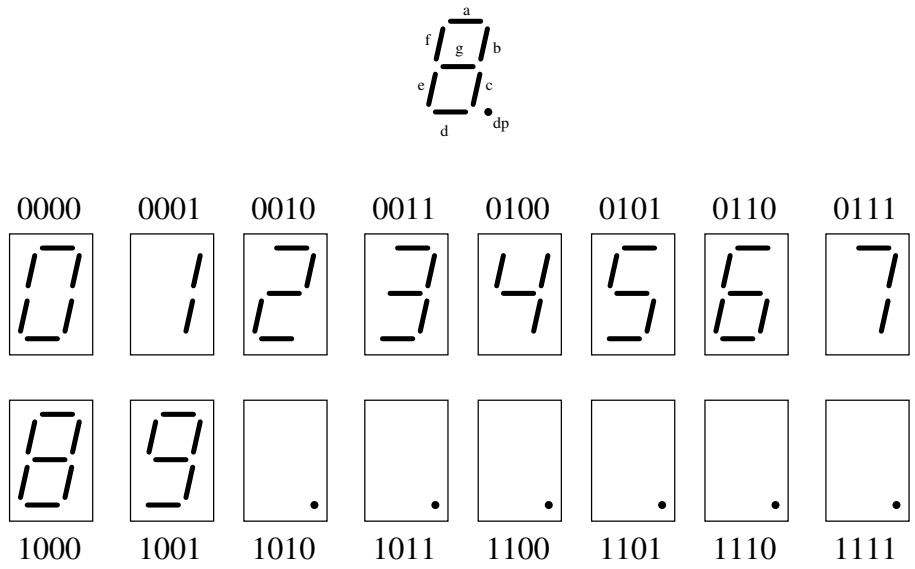
Présentation du TP : L'objectif du TP est de mettre en oeuvre la logique programmable et plus précisémment les FPGA. Cela se fait dans la continuité du module M1102 du semestre 1. Cependant, on décricra désormais les fonctions logiques à implanter par le langage VHDL plutôt que par des schémas logiques. Les TP seront basés sur une application de comptage de passage pour conduire à la description de fonctions combinatoires, séquentielles synchrones puis des machines d'état. Ensuite, il s'agira d'implanter un soft processeur à programmer en langage C . Pour les TP, on utilisera la carte basys2 de Digilent dotée d'un FPGA Xilinx XC3S250E-CP132 (Spartan3 starter kit board). On mémorisera bien cette référence de circuit pour régler les paramètres du logiciel ISE lors de la création des projets successifs. On prendra soin dès ce premier exercice à décrire les fonctions logiques sous forme de composants (component) interconnectés entre eux.
Dans les schémas, les nouveaux composants à programmer dans l'exercice en cours sont dessinés en rose. Les composants déjà programmés dans les exercices précédents ou fournis par l'enseignant sont dessinés en bleu. Le cadre rouge représente le circuit logique programmable. Les signaux entrant ou sortant de ce cadre sont donc les entrées/sorties du FPGA. En général, les noms des signaux sont imposés pour faciliter la correction des erreurs par l'enseignant. Les entrées sont reçues, et les sorties envoyées, sur les broches/bornes/pattes du circuit qui permettront les tests. Là encore, elles sont imposées par le matériel utilisé (carte Basys 2) ou par l'enseignant pour les mêmes raisons de facilité de correction.

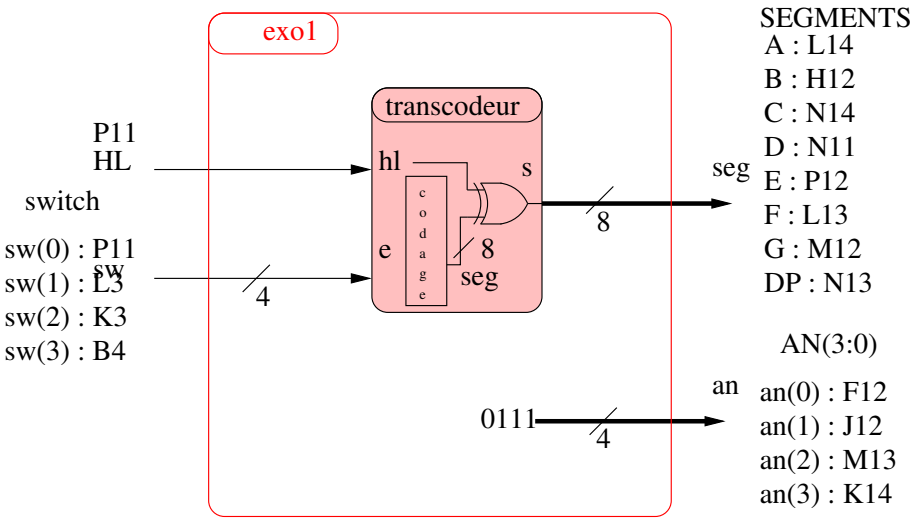
Exercice 1 : transcodeur BCD / 7 segments Le compteur de passage devra afficher sa valeur. Nous allons donc piloter les afficheurs 7 segments et, par là, revoir une manière au moins de coder une fonction logique combinatoire en VHDL.

Préparation : remplir un tableau de correspondance donnant, pour chaque valeur d'un nombre décimal 0 à 9, le code binaire dessinant le chiffre sur un afficheur 7 segments selon le modèle ci-dessous.



n	a	b	c	d	e	f	g	p
0000								
0001								
0010								
0011								
0100								
0101								
0110								
0111								
1000								
1001								
1010								
1011								
1100								
1101								
1110								
1111								

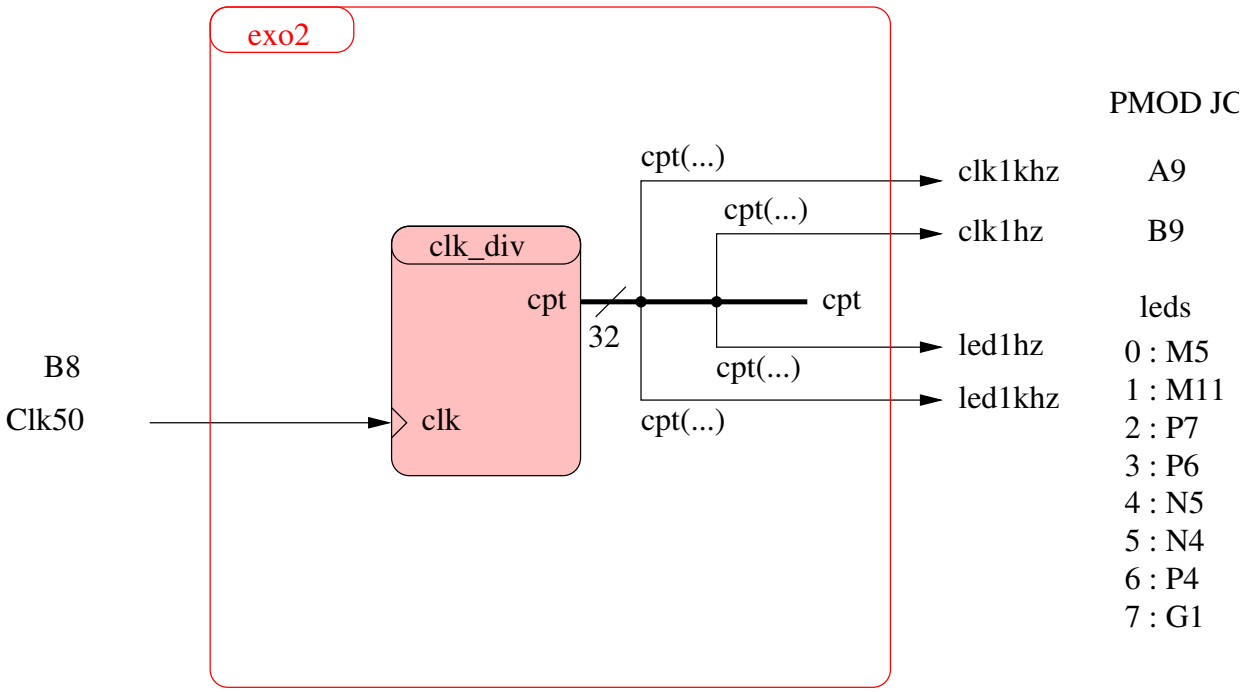
TP : Décrire le transcodage en 2 parties. Réaliser d'abord le codage en allumant les segments avec des '1' logiques. On génèrera donc un signal intermédiaire nommé ***seg*** dans le schéma, par exemple avec un style de programmation VHDL ***with select when***. Ensuite, on décrira une fonction inverseuse du signal ***seg*** commandée par le signal ***hl .*** Ce signal signifiant *high/low* a pour rôle de d'envoyer aux afficheurs soit des '0' soit des '1' pour les allumer, selon la manière dont sont connectés les segments (anode ou cathode commune). Un ***process*** combinatoire permettra de décrire cette fonction.



Exercice 2 : division de fréquence La carte Basys 2 est dotée d'un quartz délivrant un signal d'***horloge à 50MHz*** disponible sur la patte ***B8*** du FPGA. On souhaite disposer de signaux d'horloge de fréquence plus réduite (1Hz et 1Khz) pour les divers exercices à venir. Or, un compteur binaire a la propriété de diviser la fréquence d'horloge par des puissances de 2 si l'on observe les chronogrammes des différents bits produits par rapport à l'horloge. Ainsi, on aura une division par $2^{0+1} = 2$ pour le bit de rang 0 , $2^{1+1} = 4$ pour le bit de rang 1 , ... 2^{n+1}pour le bit rang n .

Préparation : Déterminer pour ce compteur binaire, les rangs des bits qui permettront d'approcher les deux fréquences souhaitées (1Hz et 1KHz).

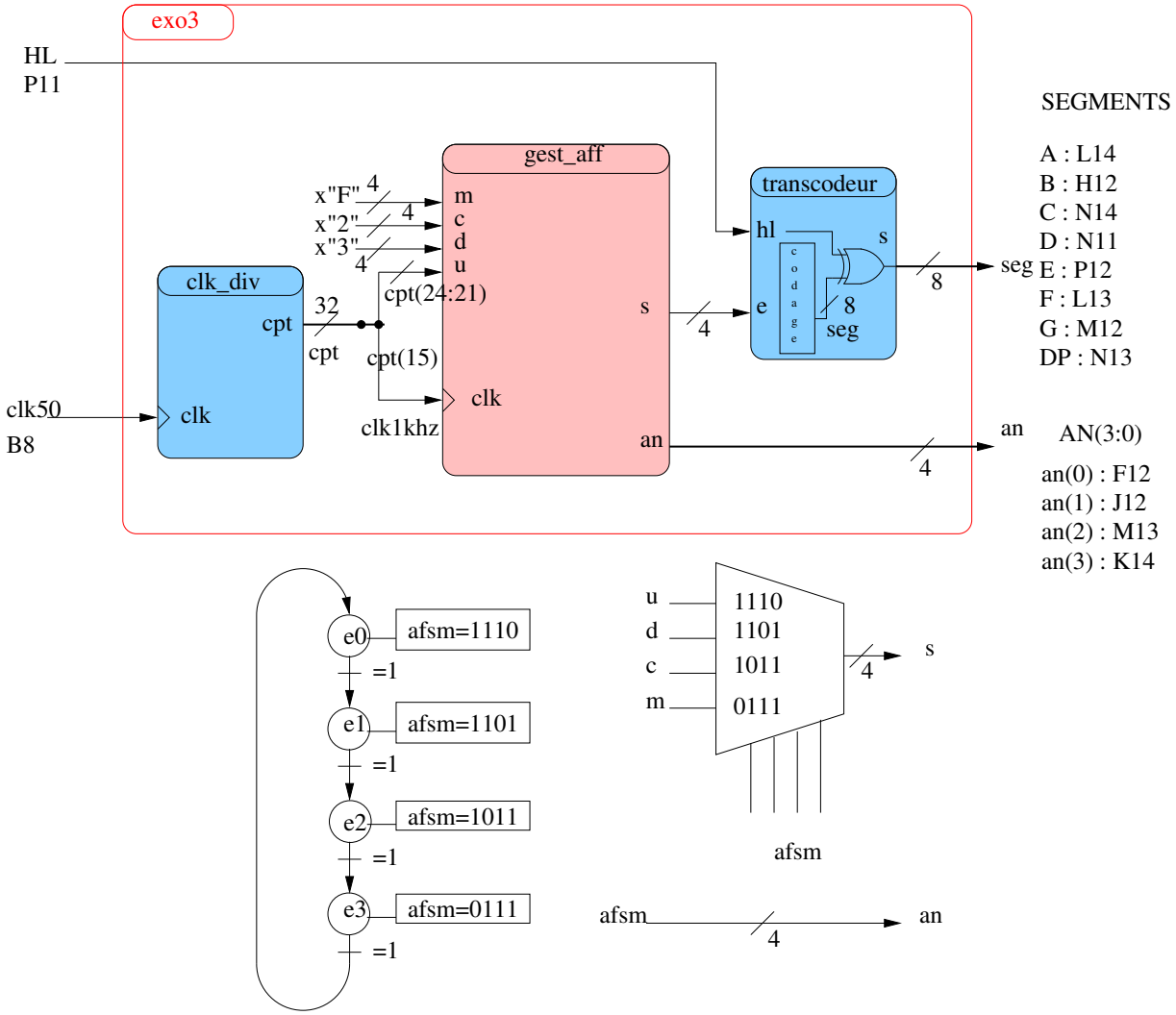
TP : Décrire en VHDL un simple compteur binaire et envoyer sur des bornes du FPGA les signaux d'horloge ainsi générés. On choisira deux bornes sur l'un des connecteurs PMOD (par exemple JC avec les bornes ***A9*** et ***B9***) pour pouvoir mesurer les fréquences et deux leds pour une visualisation (par exemple les leds 0 et 1 avec les bornes ***M5*** et ***M11***). Ici on a nommé ***clk_div*** le component qui divise la fréquence même si la fonction logique implantée est un compteur.



Exercice 3 : affichage 4 digits On souhaite maintenant étendre l'affichage sur les 4 digits de la carte Basys 2. Les 4 afficheurs 7 segments recevant tous simultanément les mêmes signaux pour les segments, leur fonctionnement repose donc sur le multiplexage des données, associé à une sélection de l'afficheur devant les exploiter. Un bit de sélection pour chaque afficheur permet de discriminer lequel d'entre eux sera effectivemment piloté. Gérer un affichage sur plusieurs digits consiste donc à mettre en place une machine d'état pilotant tour à tour chacun des 4 bits ***an*** et,

simultanément, mettre en oeuvre un multiplexeur envoyant de manière synchronisée la bonne donnée vers l'afficheur sélectionné.

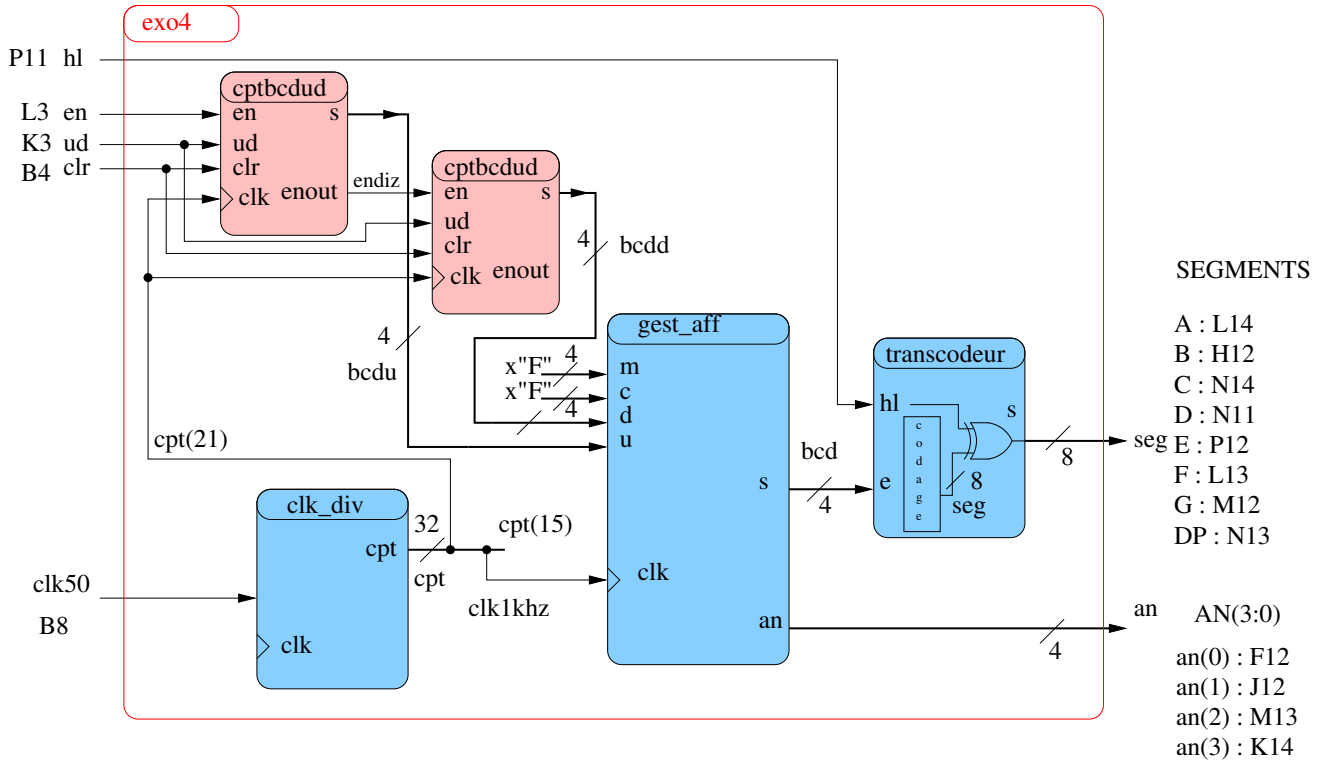
TP : implanter le schéma ci-dessous en exploitant les component des exercices précédents et en décrivant le component ***gest_ aff*** avec la machine d'état, l'envoi de ***afsm*** sur ***an*** et le multiplexeur représenté ci-dessous. ***afsm*** (action finite state machine) correspond aux actions menées par la machine d'état.



Exercice 4 : compteur/décompteur BCD cascadable Le compteur de passage nécessite de pouvoir compter ou décompter des passages. Par ailleurs, on souhaite afficher le nombre de passages en décimal sur 2 afficheurs . On doit donc décrire un compteur/décompteur BCD cascadable. Celui-ci pourra etre remis à zéro de manière asynchrone avec le signal ***clr***. Il sera validé (***en=1***) ou bloqué (***en=0***)par un signal ***en*** . Le signal ***ud*** permettra de choisir s'il doit compter (***ud=1***) ou décompter (***ud=0***). Il délivrera un nombre BCD sur 4 bits (signal ***s***) et un signal ***enout*** autorisant,ou non, le compteur de rang supérieur à évoluer.

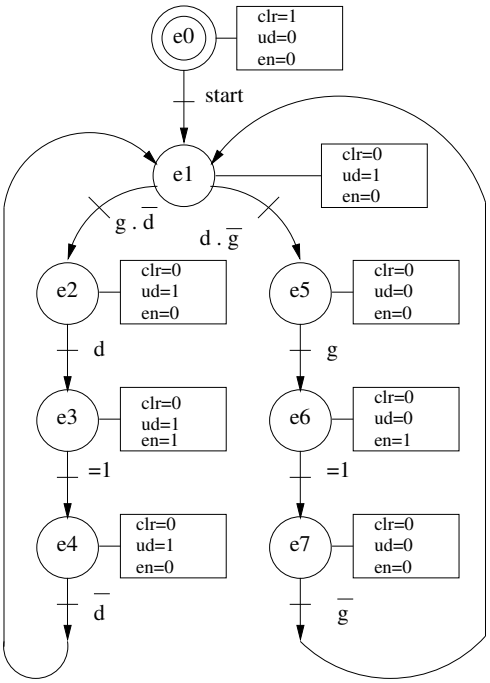
Préparation : Réfléchir au fonctionnement de ce compteur/décompteur BCD cascadable et proposer un algorithme sous la forme ***si alors*** pour chacun des signaux de sortie en fonction des entrées.

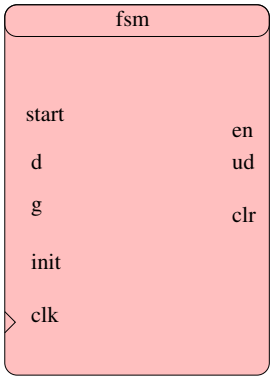
TP : Décrire en VHDL le compteur/décompteur. Implanter ensuite 2 compteurs cascadés et les connecter aux autres component précédents selon le schéma synoptique ci-dessous.

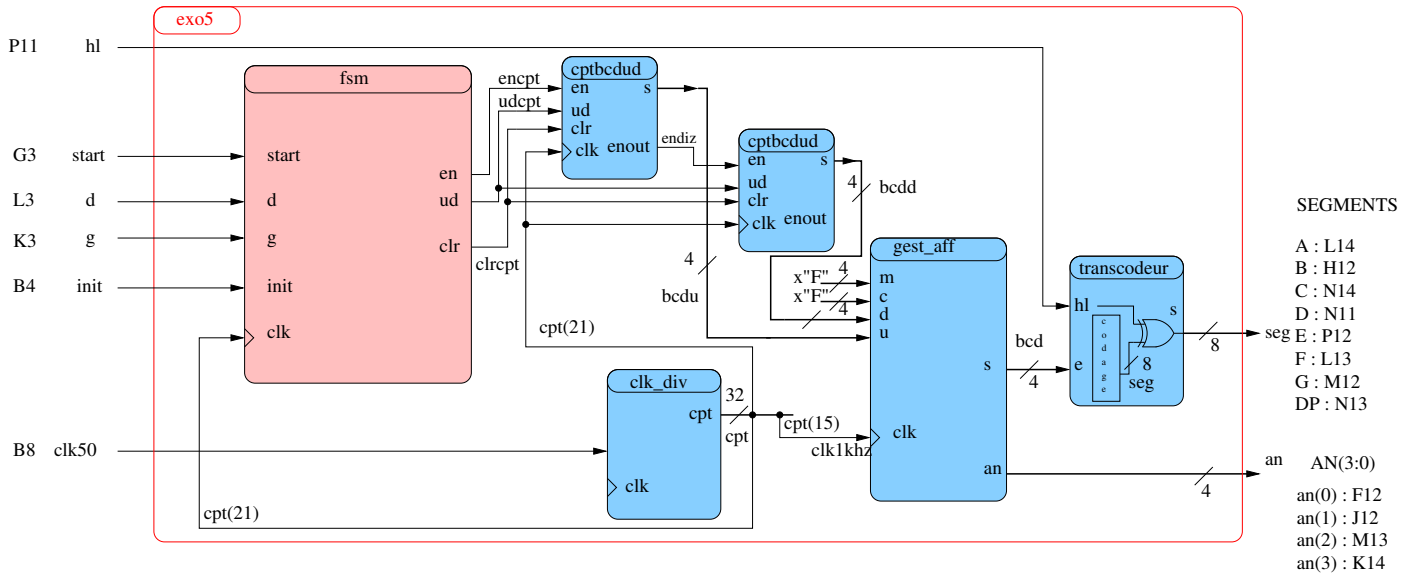


Exercice 5 : compteur de passage (machine d'état) Nous pouvons désormais mettre en place le compteur de passage complet. Il est constitué d'une partie opérative. C'est tout le dispositif de comptage et d'affichage préparé dans les exercices précédents. L'autre partie à développer est la partie commande. Une machine d'état va observer les signaux extérieurs issus des capteurs gauche et droite et va générer les signaux de commande des compteurs (***encpt***, ***udcpt***, ***clrcpt***).

TP : implanter la machine d'état dans le component ***fsm*** selon la représentation ci-dessous. Intégrer ensuite cette machine d'état dans l'application complète de comptage de passage selon le schéma suivant ci-dessous.







Exercice 6 : Processeur AVR ATMEGA 8 : une première implantation

Exercice 7 : Processeur AVR ATMEGA 8 : chenillard, compteur

Exercice 8 : Processeur AVR ATMEGA 8 : compteur de passage