

Session S4

**Génie électrique
Génie informatique**

**Unité d'apprentissage par problèmes (APP)
Circuits logiques séquentiels**

**Faculté de génie
Université de Sherbrooke**

Hiver 2024

Dernières révisions apportées à l'unité d'APP à la suite des rapports d'intendants et aux bilans de session

- Ajout de la procédure de création de bloc design dans l'annexe pour un meilleur alignement avec le projet.
- Création d'une liste d'erreurs de code VHDL fréquentes causant une implémentation physique non-fonctionnelle
- Modification du problème 1 du procédural 1 pour faciliter l'introduction du plan de vérification et d'une machine à état fini.
- Modification du laboratoire : réarrangement de l'ordre de problèmes et clarification des étapes de conception.
- Modification du devoir : Passage d'une rétroaction individuelle sommative à une rétroaction formative en grand groupe
- Mise à jour des grilles d'évaluation pour le rapport et la validation
- Reformulation du dernier numéro de l'examen formatif.
- Création d'ateliers VHDL asynchrones

Document Guide_APP_log_seq_H2024.docx

Historique de révision

1.0	1er février 2019	1.1	5 février 2019	1.2	23 janvier 2020
1.4	21 janvier 2021	1.5	20 janvier 2022	1.6	13 mai 2022
1.7	19 janvier 2023	1.8	16 janvier 2024		

Auteurs : Sébastien Roy, Réjean Fontaine, Daniel Dalle, Julien Rossignol, Audrey Corbeil Therrien

Copyright © 2019-2024 Département de génie électrique et de génie informatique, Université de Sherbrooke.

Table des matières

Session S4.....	1
Circuits logiques séquentiels	1
Table des matières	3
Liste des figures.....	4
Liste des tableaux	4
Éléments de compétences visés	5
GEN420 - Mathématiques des circuits logiques 2 crédits	5
GEN430 - Circuits logiques 2 crédits	5
Qualités de l'ingénieur.....	6
Énoncé de la problématique.....	7
Connaissances nouvelles à acquérir.....	11
Connaissances déclaratives : QUOI.....	11
Connaissances procédurales : COMMENT.....	11
Connaissances conditionnelles : QUAND.....	12
Guide de lecture et références.....	12
Volume obligatoire	12
Documents d'accompagnement de la problématique	13
Activités liées à la problématique.....	14
Activités de la semaine 1	14
Activités de la semaine 2	14
Semaine 1 : Formation à la pratique procédurale 1	15
But de l'activité	15
Problèmes à résoudre.....	15
Semaine 1 : Formation à la pratique en laboratoire.....	20
But de l'activité	20
Description du laboratoire :	20
Étapes du travail à réaliser :	20
Semaine 1 : Formation à la pratique procédurale 2	23
Buts de l'activité.....	23
Problèmes à résoudre.....	23
Semaine 2 : Support en laboratoire.....	27
Livrables	27
Semaine 1 : Devoir à remettre en équipe.....	27
Semaine 2 : Validation.....	27
Semaine 2 : Remise du projet et rapport d'APP	28
Semaine 2 : Schéma de concept.....	31
Pondérations des évaluations	31
Grilles d'évaluation	32
Références.....	35

Liste des figures

Figure 1 : Architecture du prototype envisagé, identifiant tous les modules à réaliser.....	10
Figure 2 : Description du protocole SPI	16
Figure 3 : Esquisse d'un schéma-bloc menant à la résolution du problème.....	17
Figure 4 : Système d'affichage sélectionné par un bouton.....	18
Figure 5 : Séquence des affichages distingués par les points	19
Figure 6 : Résultats attendus du laboratoire.	22
Figure 7 : Automate de sélection sur un convoyeur	24
Figure 8: Circuit séquentiel	25
Figure 9: Ligne à délai	25

Liste des tableaux

Tableau 1 : Pondérations des évaluations	31
Tableau 2 : Grille d'évaluation de la validation	32
Tableau 3 : Grille d'évaluation du rapport	33
Tableau 4: Évaluation de la communication	34

Éléments de compétences visés

GEN420 - Mathématiques des circuits logiques

2 crédits

<https://www.usherbrooke.ca/admission/fiches-cours/GEN420/>

Cible(s) de formation :

S4-APP1 : Modéliser et résoudre un problème de logique combinatoire à l'aide de représentations mathématiques de l'information discrète et par la synthèse des équations booléennes.

S4-APP2 : : Modéliser et résoudre un problème de logique séquentiels à l'aide de représentations mathématiques de l'information discrète et par la synthèse des équations booléennes.

Modéliser l'information discrète et son évolution temporelle. Déterminer les structures de données et les algorithmes appropriés pour les mettre en œuvre.

GEN430 - Circuits logiques

2 crédits

<https://www.usherbrooke.ca/admission/fiches-cours/GEN430/>

Cible(s) de formation :

S4-APP1 : Concevoir et réaliser des systèmes numériques combinatoires à partir de spécifications.

S4-APP2 : Concevoir et réaliser des systèmes numériques séquentiels à partir de spécifications.

<https://www.usherbrooke.ca/admission/fiches-cours/GEN420/>

<https://www.usherbrooke.ca/admission/fiches-cours/GEN430/>

Note : GEN420 correspond aussi à GEN265 et GEN430 à GEN-225 (anciennes AP).

Qualités de l'ingénieur

Les qualités de l'ingénieur visées par cette unité d'APP sont les suivantes. D'autres qualités peuvent être présentes sans être visées ou évaluées dans cette unité d'APP.

Qualité	Libellé	Touchée	Évaluée
Q01	Connaissances en génie	✓	✓
Q02	Analyses de problèmes	✓	✓
Q03	Investigation		
Q04	Conception	✓	
Q05	Utilisation d'outil d'ingénierie	✓	✓
Q06	Travail individuel et en équipe		
Q07	Communication	✓	
Q08	Professionnalisme		
Q09	Impact du génie sur la société et l'environnement		
Q10	Déontologie et équité		
Q11	Économie et gestion de projets		
Q12	Apprentissage continu		

Pour une description détaillée des qualités et leur provenance, consultez le lien [BCAPG](#) sur le site de la Faculté de génie. Les modalités d'évaluation des compétences, les pondérations et grilles d'évaluation sont présentées à la fin de ce guide.

Énoncé de la problématique

Votre expérience avec votre détecteur de proximité capacitif de l'APP1 de la S4 a été un succès dans votre dernier show rock. Vous continuez dans cette direction en créant votre propre pédale d'effets de guitare en utilisant votre carte Zybo pour appliquer différents types d'effets sonores sur le signal audio provenant d'une guitare avant d'envoyer le signal modifié vers l'amplificateur.

1- Description sommaire

Le projet contient 3 sections principales soit les conversions A/N et N/A, l'application d'un effet sonore sur le signal numérique audio et l'extraction de paramètres de ce signal modifié (figure 1). La première section convertit le signal analogique provenant de l'entrée audio sur votre carte Zybo en un signal numérisé. Ce même bloc effectue aussi la conversion inverse du signal afin de le transmettre à la sortie audio. La seconde section modifie le signal d'entrée numérisé selon l'effet sonore sélectionné. La troisième section permet de mesurer certains paramètres du signal audio après leur modification. Ces paramètres seront affichés sur des afficheurs 7 segments.

Le projet est déjà entamé et il reste quatre modules à compléter : la sélection de l'effet sonore à appliquer (M8), le décodeur du lien de communication I2S (M1), la mesure de la période du signal audio (M5) et la mesure de la puissance du signal audio (M6).

2- Module de commande (M8)

Le module de commande (M8) gère les différentes configurations du système. Il permet entre autres de sélectionner l'effet sonore désiré en activant les 2 bits de sélection d'un multiplexeur. Actuellement, la fonction de sélection de l'effet sonore est réalisée de façon combinatoire à partir des interrupteurs SW2 et SW3 sur la carte Zybo, ce qui est peu pratique. Vous allez modifier le module M8 pour que la sélection de la distorsion à appliquer sur le signal audio se fasse plutôt de manière incrémentale/décroissante et cyclique en appuyant à répétition sur les boutons BTN0 et BTN1 pour faire défiler dans les deux sens le type de distorsion désiré (style up/down). Les détails sur ce fonctionnement sont fournis dans l'annexe.

3- Décodeur I2S (M1)

Le signal est numérisé par un convertisseur analogique-numérique (CODEC SSM2603 de la compagnie *Analog Devices*) sur la carte Zybo. Ce convertisseur transmet la valeur des échantillons au FPGA par protocole de communication **série I2S**. Vous devez concevoir le module qui permettra de détecter une transmission, enregistrer les bits représentant la valeur de l'échantillon et transmettre cette valeur sur un bus parallèle aux prochains modules.

Les échantillons sont codés en **compléments 2** sur une plage de **24 bits à virgule fixe qui couvre un intervalle normalisé de -1 à 1**. Le **signal audio est stéréo**, c'est-à-dire qu'il y a un canal pour le côté gauche et le côté droit. Sur chacun de ces canaux, le **signal est échantillonné à une fréquence de 48 kHz**. Le protocole I2S transmet en alternance les valeurs des échantillons des canaux gauche et droit depuis le MSB jusqu'au LSB.

4- Mesure du signal (M5 et M6)

On demande d'extraire 2 caractéristiques des signaux, soit la période du signal audio (M5) et sa puissance (M6). La mesure de la fréquence doit être résiliente au bruit et son affichage doit se rafraîchir à chaque période. Pour la puissance, une valeur proportionnelle à la puissance du signal audio à un facteur constant près devra être calculé. Afin d'éviter les débordements, le calcul de la puissance inclura un **facteur d'oubli de $\frac{31}{32}$** . Des détails sur l'implémentation mathématique de ces deux modules sont fournis dans l'annexe.

5- Méthodologie

Afin de s'assurer que les modules demandés respectent les **intentions de conception** pour notre pédale de guitare, vous devez commencer par l'élaboration d'un **plan de vérification**. Certaines **spécifications fonctionnelles** devront être précisées afin de réaliser des tests significatifs dans le banc d'essai de simulation VHDL de haut niveau du design complet.

La logique combinatoire ne suffira pas à la conception de certains modules : il faudra procéder à la conception de machine à états finis selon une méthode rigoureuse. Au cœur de chaque machine à états finis se trouve un registre d'état, qui prend généralement la forme d'un élément de stockage constitué de bascules D. Les machines à réaliser peuvent être de type Moore ou de type Mealy et l'encodage des états est libre.

Plusieurs des opérations séquentielles devront s'effectuer à une cadence inférieure à l'horloge système qui vous sera fournie. Ceci sera réalisé par le biais de bascules sensibles à la fois aux transitions de l'horloge système et au niveau d'un signal de type « strobe » qui activera l'entrée « enable » du circuit visé à la fréquence d'opération souhaitée. Ce mécanisme permet la coexistence de circuits fonctionnant à plusieurs cadences différentes à partir d'une seule horloge système, de manière à éviter tout aléa ou perte de synchronisme.

Vous trouverez dans l'annexe la description du circuit initial, des précisions sur les méthodes de synchronisation avec les horloges et sur les ressources de plusieurs circuits utiles, de même qu'une description du banc d'essai initial. Afin de faciliter la compréhension du système et faciliter la réutilisation des modules, chacun des éléments de circuit synchrone (exemple : compteur, registre à décalage ...) entourant une machine à états finis sera décrit dans un fichier VHDL distinct (unité de design). Ceci facilitera grandement le dépistage des erreurs lors du développement.

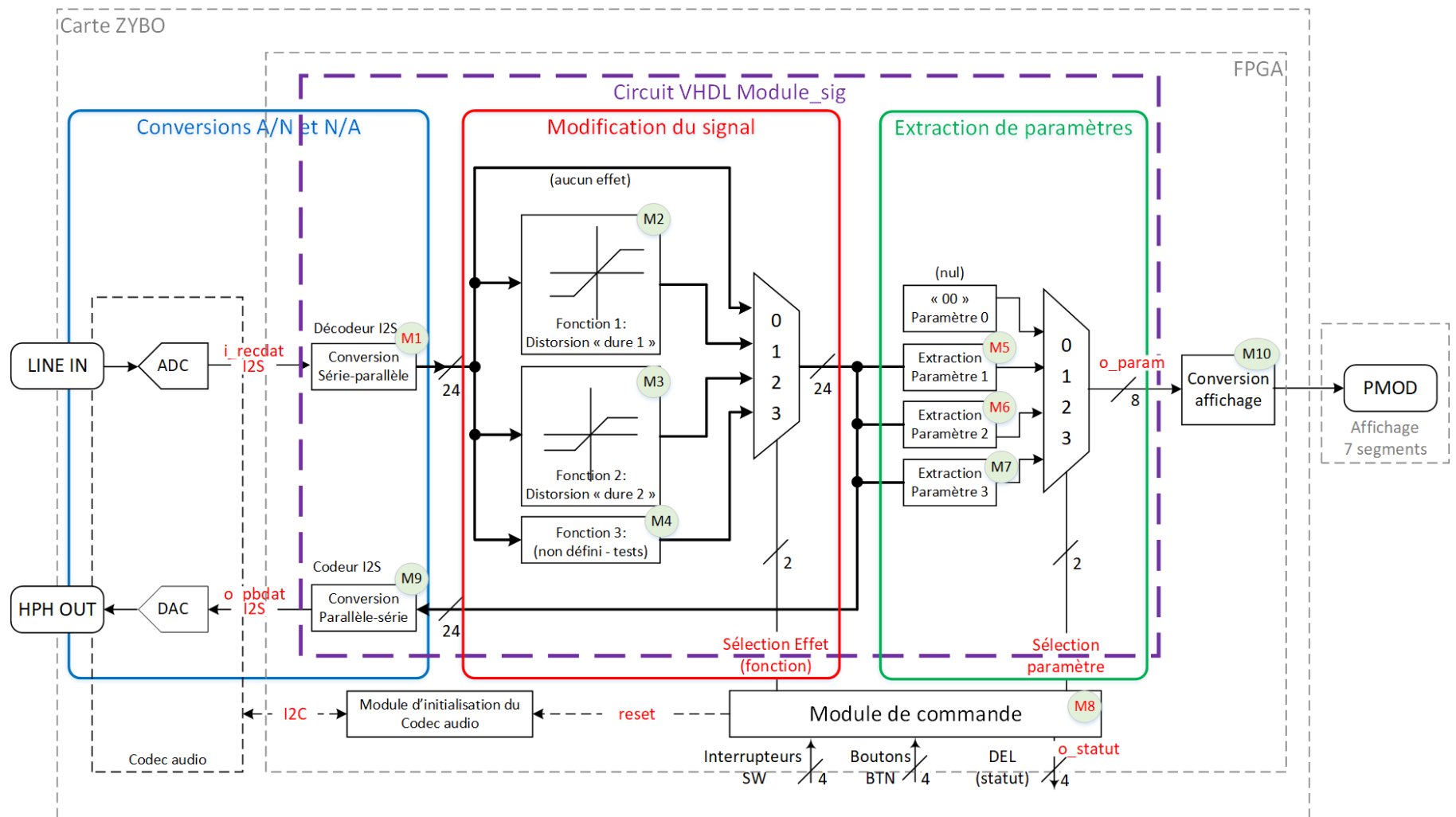


Figure 1 : Architecture du prototype envisagé, identifiant tous les modules à réaliser

Connaissances nouvelles à acquérir

Connaissances déclaratives : QUOI

Sciences fondamentales

Mathématiques

Mathématiques discrètes : représentation de l'information

Diagrammes d'états, machines à états finis (MEF)

Sciences de l'ingénierie

- Bascules élémentaires, D, R-S, T (de l'anglais *flip-flop*).
- Équations caractéristiques des bascules.
- Représentations et simulation en VHDL des bascules
- Structure de base des machines à états finis (Moore, Mealy).
- Analyse de machines à états finis.
- Description des MEF : narrative, diagramme d'états, VHDL, chronogramme.
- Conception des machines à états finis, codage des états.
- Analyse et production de chronogrammes.
- Rédaction de codes VHDL lisibles utilisant les portes de base.
- Circuits séquentiels standards : registres (de l'anglais *register*), registres à décalage (*shift registers*), compteurs (*counters*), circuits programmables (*programmable devices*).
- Méthodes d'implémentation avec des bibliothèques IP.
- Caractéristiques temporelles des circuits séquentiels.
- Analyse des contraintes de délais temporels et de fréquences d'horloges.

Connaissances procédurales : COMMENT

- Écrire les équations caractéristiques de bascules élémentaires.
- Mettre en œuvre les différentes étapes de conception d'une MEF synchrone à partir de sa description fonctionnelle.
- Mettre en œuvre des circuits séquentiels standards. Circuits et VHDL.
- Calculer les délais de propagation dans un circuit séquentiel. Analyser les contraintes.
- Valider le comportement logique d'un circuit séquentiel à l'aide d'un simulateur logique.
- Réaliser un circuit logique séquentiel sur une matrice de portes logiques programmables.

- Utiliser des appareils de mesure de base pour mettre au point un circuit logique séquentiel.

Connaissances conditionnelles : QUAND

- Utiliser des circuits combinatoires et séquentiels pour résoudre un problème de logique.
- Utiliser des circuits séquentiels standards et de librairie pour résoudre un problème de logique séquentielle.
- Choisir une structure de MEF de type Mealy ou de type Moore.
- Choisir un style de description VHDL pour des MEF,
- Déterminer les tests requis pour valider un système.
- Élaborer un banc d'essai VHDL.

Guide de lecture et références

Volume obligatoire

John F. Wakerly Digital Design, Principles and Practices, 4th edition, Prentice Hall, 2006, 895 p. ; disponible en version Kindle/électronique
(visionneur logiciel gratuit Windows/Apple/Android)

Séquence d'étude suggérée (lectures non exclusives) :

Pour les petits exercices :

- Wakerly : 7 à 7.2.7

Pour le procédural 1 :

- Wakerly : 7.3 à 7.4.5, 7.5, 7.12 (ou 7.11 selon l'édition)
- Extrait Functional Verification Coverage Measurement and Analysis (site web)
- Annexe du guide étudiant portant sur la vérification de circuits à logique séquentielle

Pour le laboratoire :

- Faire les ateliers VHDL disponibles sur le site web (incluant les exercices à compléter)

Pour le procédural 2 :

- Wakerly : 8.4 à 8.4.6

Pour maîtriser les concepts pertinents aux apprentissages de cette unité d'APP, il est essentiel de les mettre en pratique en résolvant les problèmes suggérés en exercices et en formation procédurale. Plusieurs exercices devraient être faits **avant** le procédural pour un apprentissage efficace en procédural.

Références utiles :

- Site de l'APP de circuits séquentiels de la session
- Site de la compagnie *Xilinx* : www.xilinx.com
- Documentation intégrée au logiciel *Vivado* ou sur le site web de la compagnie Xilinx.
- Site web du fabricant de composants logiques Texas Instruments.

À partir de ce site, sélectionner une famille de circuits, par exemple : ALS ou FCT

<http://focus.ti.com/docs/logic/logichomepage.jhtml>

Documents d'accompagnement de la problématique

Disponibles sur la page web de l'unité.

Activités liées à la problématique

Activités de la semaine 1

- 1^{ère} rencontre de tutorat ;
- Étude personnelle des sujets issus des objectifs d'étude du tutorat ;
- Exercices suggérés (série 1)
- Formation à la pratique procédurale 1
- Formation à la pratique en laboratoire
- Exercices suggérés (série 2)
- **(Livvable) Remise du devoir**
- Consultation/Atelier sur les machines à états finis
- Formation à la pratique procédurale 2

Activités de la semaine 2

- Support en laboratoire
- **(Livvable) Validation pratique de la solution ;**
- **(Livvable) Remise du projet final et du rapport d'APP;**
- 2^{ième} rencontre de tutorat
- Évaluation formative
- Étude personnelle et exercices
- Évaluation sommative

Semaine 1 : Formation à la pratique procédurale 1

But de l'activité

Mettre en pratique les procédures requises pour :

- Élaborer un plan de vérification préliminaire ;
- Mettre en œuvre les étapes de conception d'une MEF ;
- Dessiner le diagramme d'états ou le chronogramme de la machine à partir de la description narrative de son fonctionnement ;
- Décrire les équations d'excitation ;
- Dessiner la table de transition et la table des états ;

Avant le procédural 1, faire les exercices de la série 1 disponibles sur le site web.

Problèmes à résoudre

P1.1 Circuits séquentiels

On demande de faire la conception haut-niveau du module de contrôle d'une machine attrape-peluche (*claw crane machine*). Il s'agit d'une machine d'arcade très populaire qui débute lorsqu'un joueur insère un jeton de jeu. Le joueur peut alors contrôler le mouvement horizontal de la grue avec un *joystick*. Sur pression d'un bouton, la pince descend et se referme, saisissant idéalement un ou des objets. Une fois la pince refermée, la grue remonte et retourne au point de départ, où la pince ouvre laisse tomber le ou les objets dans la chute. Le joueur récupère les objets dans la chute et le jeu termine.

- Élaborer un plan de vérification pour valider le fonctionnement de cette machine attrape-peluche.
- Déterminer les signaux d'entrée et sortie du système de contrôle de ce jeu.
- Dessinez le diagramme d'états modélisant le comportement de la séquence de jeu de la machine attrape-peluche.

P1.2 Circuits séquentiels

Un circuit séquentiel possède 2 bascules (de type D, dont les sorties sont désignées A et B), 2 entrées X et Y, et une sortie globale Z.

Les équations d'excitation sont : $D_A = X'Y + XA$ $D_B = X'B + XA$

L'équation de sortie globale est : $Z = XB$

- Quelle est la structure du circuit, Moore ou Mealy ? Pourquoi ?
- Dessiner le circuit,
- Écrire la table des états et le diagramme d'états.

P1.3 Circuits séquentiels (circuit à décalage)

Le module de configuration d'un circuit complexe a besoin d'une série de 128 bits choisis par l'utilisateur. Vous devez concevoir le sous-module qui reçoit cette séquence sur un port série à 3 fils suivant le protocole SPI. Ce protocole exige une horloge, un signal de sélection ("chip select") et un signal de données. Après avoir amorcé la conception d'une machine à 128 états ou même plus, vous réalisez qu'il doit y avoir une meilleure stratégie.

(a) Concevez le sous-module en combinant une MEF avec au maximum 4 états et d'autres circuits complexes (ex : additionneur) pour réaliser le sous-module. Assumez qu'il n'y aura jamais d'erreurs de la part du transmetteur.

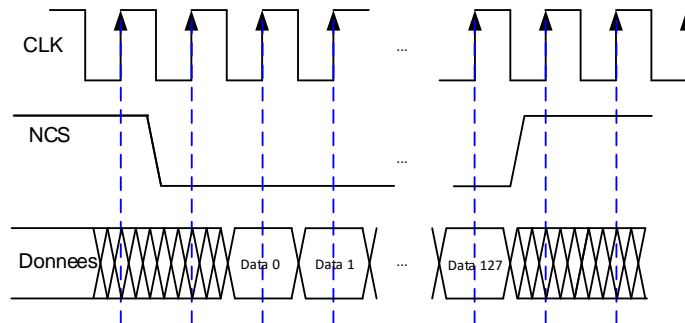


Figure 2 : Description du protocole SPI

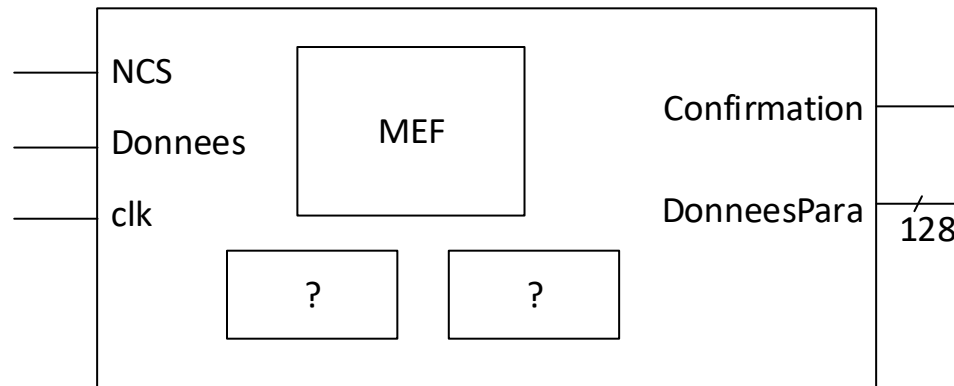


Figure 3 : Esquisse d'un schéma-bloc menant à la résolution du problème

(b) Votre circuit se modifie-t-il facilement pour le changer en récepteur 1024 bits?

P1.4 Conception d'un circuit de sélection d'un bus de données

Sur une carte possédant un FPGA (telle que Zybo), pour afficher 4 variables disponibles sur des bus distincts, on dispose d'un seul afficheur à 2 digits de sept segments et de 4 DEL. Pour choisir la variable à afficher, on dispose d'un bouton de commande. Il faut concevoir un module unique pour afficher une des 4 variables sur l'afficheur 7 segments en utilisant un bouton BTN0 qui permet de sélectionner en séquence la variable à afficher. À l'initialisation, l'afficheur présente la variable 1 et quand on appuie sur le bouton une fois, l'afficheur présente la variable suivante, la variable 2 et ainsi de suite. La variable 1 est réaffichée à la suite de la variable 4 à l'occurrence de la commande du bouton. Une impulsion d'une durée d'une période d'horloge est générée lorsque le bouton est appuyé.

L'organisation haut niveau du circuit est illustrée par la figure suivante :

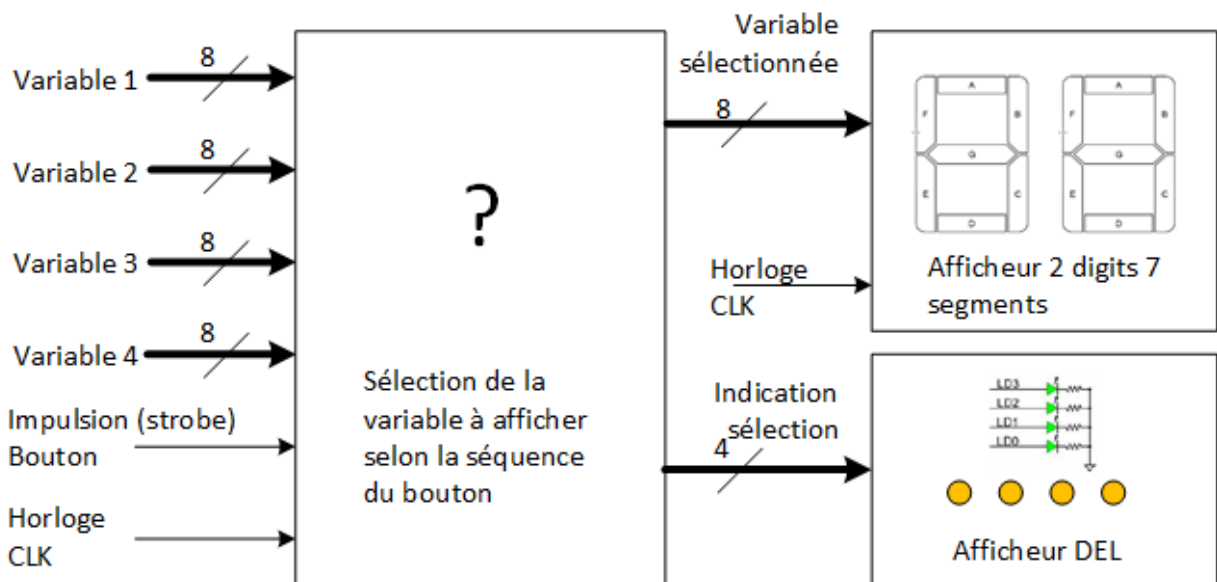


Figure 4 : Système d'affichage sélectionné par un bouton

Quatre bits supplémentaires indiqueront sur l'afficheur à diodes DEL quelle est la variable sélectionnée en encodage binaire selon la figure suivante.

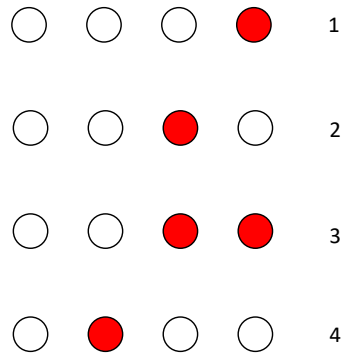


Figure 5 : Séquence des affichages distingués par les points

Question :

Proposez une conception architecturale de haut niveau pour cet automate fini. Cette conception combinera des circuits de moyennes intégration contrôlés par une machine à état fini.

- (a) Identifier les entrées et sorties du module
- (b) Identifier les circuits de moyenne intégration nécessaires pour la fonction demandée
- (c) Identifier les signaux de contrôle en entrée et en sortie de la machine à état fini.
- (d) Réaliser le diagramme d'état de la machine à état fini
- (e) De quel type de machine à état s'agit-il?

Semaine 1 : Formation à la pratique en laboratoire

Avant le laboratoire faire les ateliers VHDL disponibles sur le site web.

But de l'activité

Le but de cette activité est de mener à terme un projet de logique séquentielle, de sa conception jusqu'à la validation de son fonctionnement sur un simulateur via des outils de conception basés sur VHDL. Les développements effectués dans ce laboratoire sont directement utilisés dans la réalisation de la problématique.

Description du laboratoire :

Ouvrez le projet de la problématique sur le site web et ouvrez le schéma bloc dans l'onglet « IP integrator ». Vivado n'enregistre pas le positionnement des modules lors de l'enregistrement des schémas blocs. Pour avoir un positionnement qui ressemble au schéma de la problématique, vous pouvez lancer le script `schema.tcl` qui se trouve dans le dossier du projet vivado (onglet *Tools*, option *Run Tcl Script* dans vivado). Ce circuit constitue la fondation de la problématique abordée. Il contient les fonctions nécessaires pour réaliser les conversions sur le CODEC et visualiser des données sur les afficheurs 7 segments. **Le travail à réaliser consiste à faire la conception des machine à état finis de sélecteurs d'effet sonore (M8) et du décodeur I2S (module M1) et de l'implémenter en VHDL en utilisant le formalisme vu dans les ateliers.** Une implémentation d'une MEF est déjà présente dans le projet, mais celle-ci ne respecte pas le formalisme pour coder des MEF en VHDL. Vous pouvez regarder la MEF du codeur (module M9) pour voir un exemple de MEF respectant ce formalisme.

Étapes du travail à réaliser :

LI.1 Conception de la fonction de sélection de l'effet sonore

La sélection de l'effet sonore s'effectue actuellement par les interrupteurs SW3 et SW2 et un circuit combinatoire. Vous devez modifier ce comportement pour plutôt avoir une sélection cyclique incrémentale (BTN0) et décrémentationale (BTN1). Il est **TRÈS IMPORTANT** de commencer la conception par l'élaboration d'un schéma-bloc et du diagramme d'état **AVANT** d'aller écrire le code. Vous réduirez votre temps de débogage de 80% ou plus!

L1.2 Conception haut niveau de la fonction de lecture I2S

À partir du numéro 3 du procédural 1 et de la figure 7, réalisez la conception haut niveau de la désérialisation des canaux droit et gauche du port I2S. Le **schéma bloc** doit inclure, comme éléments fonctionnels, des circuits de moyenne intégration comme le registre à décalage et le compteur, en plus d'une petite machine à états finis servant à contrôler le flot d'information. **Il est important de réaliser cette conception par vous-même autant que possible. C'est la partie la plus difficile de la conception de MEF.** Quand vous aurez réalisé votre schéma-bloc, comparez-la avec le schéma bloc du décodeur dans le projet Vivado de la problématique. Ensuite, dessiner le **diagramme d'état de la MEF** contrôlant les circuits de moyenne intégrations permettant de faire cette fonction de lecture. Ne commencer pas à écrire de code avant d'avoir un diagramme d'état!

Comme illustré sur le chronogramme de la figure 7, annoté en rouge pour canal gauche (en vert pour canal droite), la machine à états du décodeur doit d'abord détecter les fronts descendants (ou montants pour droite) de *i_LRC* (1), puis détecter un à un les 24 bits présentés en série sur *i_dat* à chaque front montant de *d_ac_bclk* (2). Quand les 24 bits d'un canal ont été reçus, ils sont placés dans le vecteur *o_dat_left*(23 :0) (3).

La même séquence doit être déployée pour le canal de droite (annoté en vert 1, 2, 3).

Un signal de « *Strobe* » est généré (4) à partir de l'horloge *i_BCLK* pour indiquer aux modules subséquents la présence d'une donnée valide sur les deux canaux simultanément.

L1.3 Réalisation de la MEF capable de contrôler le flot d'information en VHDL

Notez que le projet distribué comporte déjà une MEF basé sur un compteur. L'objectif est de réaliser une MEF au sens formel (diagramme d'états et son code VHDL associé).

À partir de votre diagramme d'état, réalisez la MEF capable de lire les signaux provenant du CODEC et d'emmagasiner ces données sur un bus parallèle. **Référez vous au tutoriel Bloc design et aux ateliers disponibles sur le site web pour démarrer votre code.**

L1.4 Simulation avec le banc de test :

Le projet fourni contient un banc de test capable de générer des signaux variés tiré d'un fichier. Lisez le code du banc de test et les informations dans l'annexe pour identifier la séquence d'échantillons qui devraient être lus sur le vecteur à la sortie de votre module et réaliser le test.

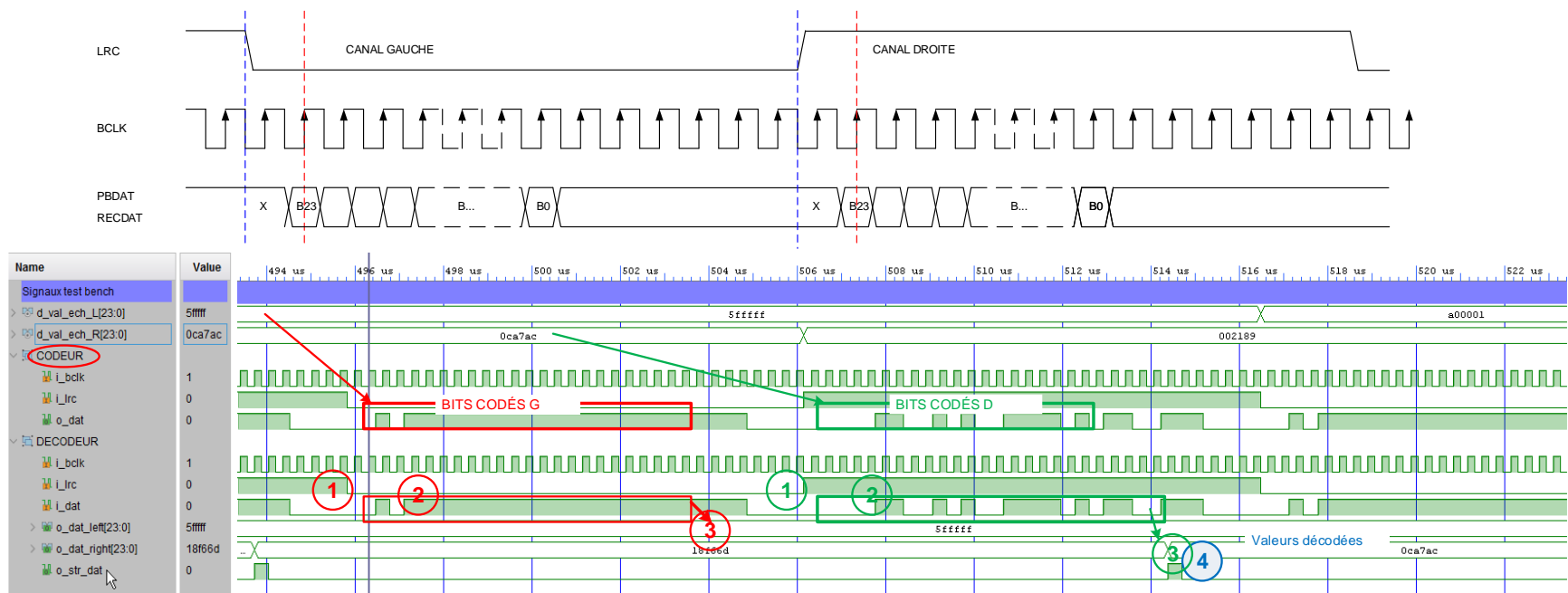


Figure 6 : Résultats attendus du laboratoire.

Semaine 1 : Formation à la pratique procédurale 2

Buts de l'activité

Mettre en pratique les procédures requises pour :

- Faire un plan de vérification;
- Mettre en œuvre les étapes de conception d'une MEF ;
- Dessiner le circuit et écrire le code VHDL d'une MEF;
- Décrire les équations d'excitation ;
- Assurer une bonne synchronisation de circuits séquentiels;
- Calculer les temps de propagation

Problèmes à résoudre

La résolution des problèmes suivants vous permettra de mettre en pratique les connaissances acquises durant votre étude et développer vos compétences en logique séquentielle.

NOTE : Avant le procédural 2, faire les exercices de la série 2 disponibles sur le site web.

P2.0 Discussion des plans de vérifications soumis

Une sélection des plans de vérification soumis pour le devoir seront présentés suivi d'une discussion sur les objectifs, la philosophie et les bonnes pratiques de la vérification. Voir la section **Livrables** pour des détails sur le devoir.

P2.1 Conception d'un circuit à partir d'un chronogramme

On considère la spécification suivante :

Une machine produit des pièces sur un convoyeur. Lorsqu'une pièce est détectée, un signal X est généré par un capteur. Ce signal constitue une entrée pour un automate synchrone à développer.

Un opérateur fait l'inspection visuelle du convoyeur et quand il actionne un bouton D , les trois pièces qui arrivent, incluant celle présente au moment de l'action du bouton D , seront retirées d'un seul coup du convoyeur. Pour cela, l'automate qui reçoit le signal du bouton et le signal X génère un signal Y qui active le retrait lorsque la troisième pièce est détectée. Une fois cette action accomplie, on revient à l'état initial.

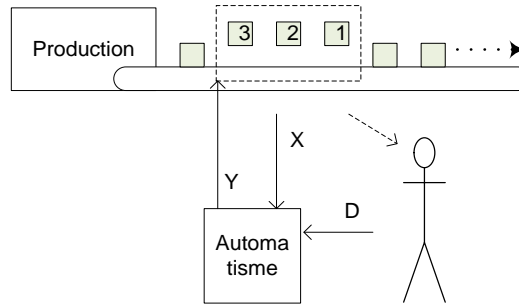


Figure 7 : Automate de sélection sur un convoyeur

Une horloge synchronise l'automate. Les signaux X et D ne sont reconnus qu'au front de l'horloge et ont une durée suffisante pour éviter les problèmes de métastabilité. L'impulsion du signal X ne dépasse pas la durée d'une période pour éviter un double comptage. Le signal D n'a d'effet que lorsque simultanément à X sur un front d'horloge.

- Proposer un chronogramme pour un scénario de test qui traduit et illustre cette spécification.
- Proposer, par une démarche complète, un circuit qui réalise cet automate et faites-en une description VHDL.

P2.2 Strobes de synchronisation

Pour la mise en œuvre de l'exercice précédent, on considère maintenant la réalisation d'un système équivalent, mais en considérant une horloge beaucoup plus rapide. Les signaux d'entrée, qui indiquent un événement, durent largement plus longtemps qu'une période.

Proposer une solution avec des impulsions comprenant des « *strokes* » de synchronisation (voir section sur ce sujet dans l'annexe).

P2.3 Circuits séquentiels (compteur BCD)

Dessinez un compteur BCD utilisant un registre de 4 bits, un module additionneur de 4 bits et une porte AND à 2 entrées.

P2.4 Circuits séquentiels (fréquence d'horloge)

On considère le circuit séquentiel suivant :

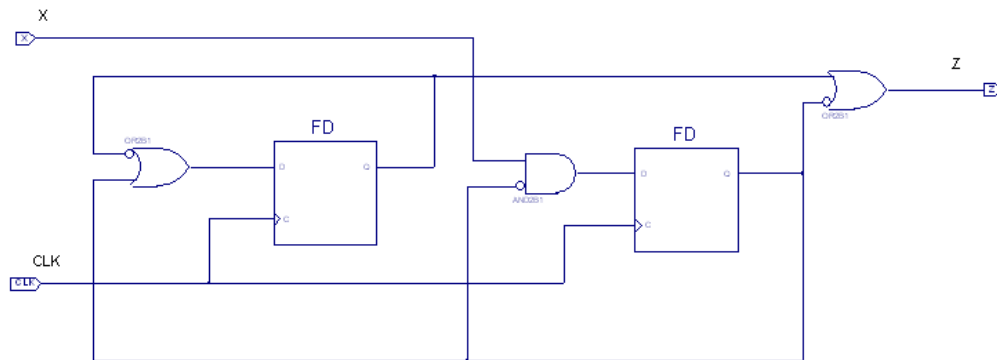


Figure 8: Circuit séquentiel

Les bascules D sont des 74HCT74. Calculer la fréquence d'horloge maximale admissible pour ce circuit. Les paramètres des composants sont :

- $t_{f\text{pd}}$ de 35 à 44 ns
- $t_{\text{su}} = 15$ ns
- $t_{\text{h}} = 3$ ns
- t_{w} (L ou H) = 23 ns
- Portes AND et OR : $t_{\text{pLH}} = t_{\text{pHL}} = 5$ ns
- Porte INV : $t_{\text{pLH}} = t_{\text{pHL}} = 3$ ns

P 2.5 Exercice sur les FIFO

Soit une ligne à délais à 4 étages :

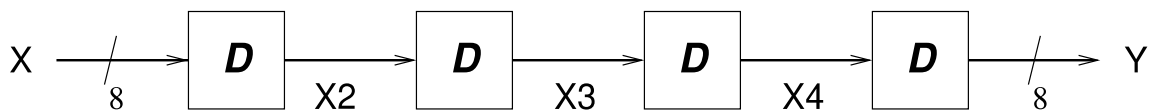


Figure 9: Ligne à délai

Les éléments de délai D induisent un délai d'un coup d'horloge entre leur entrée et leur sortie. À chaque coup d'horloge, une nouvelle donnée est présentée à l'entrée X . Au coup d'horloge suivant, cette donnée passe à $X2$, puis $X3$ et ainsi de suite pour émerger à la sortie Y 4 coups d'horloge après son arrivée à l'entrée. Ainsi, Y constitue une copie du flux de données en X , mais avec 4 coups d'horloge de retard. Cette structure en pipeline constitue un exemple de structure primitive de type FIFO (first in, first out) puisque les données sortent dans le même ordre qu'elles sont entrées.

- (a) Comment décririez-vous pareille structure en VHDL ?
- (b) Considérez à présent que les écritures à gauche et les lectures à droite sont désynchronisées.
 - a) Si on souhaite conserver la propriété FIFO, que se passerait-il au niveau conceptuel lorsqu'il y a plus d'écritures que de lectures ?
 - b) Et lorsqu'il y a plus de lectures que d'écritures ?

On souhaite maintenant réaliser une structure efficace de type FIFO désynchronisée. La structure fonctionne selon le principe d'une file d'attente qui peut être plus ou moins longue, mais qui a une longueur totale maximale qui ne peut être dépassée. Plutôt que de continuellement déplacer les données de gauche à droite, on écrit les données dans un tampon mémoire et on incrémente des indices ou des pointeurs. On commencera par créer un tableau de mémoire en VHDL de largeur 8 bits et de longueur maximale 16 :

```
type tabtyp is array (natural range <>) of std_logic_vector(7 downto 0) ;
signal tableau : tabtyp (0 to 15) := (others=>(others=>'0')) ;
```

À partir de ce tableau, on peut créer une structure de type FIFO sous la forme d'un tampon circulaire. Ce concept est détaillé dans l'annexe.

- c) Pourquoi pensez-vous qu'on désigne une telle structure « tampon circulaire » ?
- d) Y a-t-il un avantage au niveau matériel à avoir une taille de tampon qui soit une puissance de 2 ? Expliquez.
- e) Quel avantage voyez-vous à cette approche versus une approche qui serait plus directement inspirée de la ligne à délai ?

P2.6 Période de retour réflexif sur les concepts, les liens (entre les concepts et avec la problématique).

Pour conclure cette période de formation procédurale, réalisez une partie de schéma de concept pertinent aux sujets abordés en précisant les liens entre ces concepts.

Semaine 2 : Support en laboratoire

Une période au laboratoire est prévue à l'horaire pour offrir une possibilité de support et de consultation pour la solution de la problématique.

Livrables

Semaine 1 : Devoir à remettre en équipe

À remettre avant jeudi de la première semaine 18h

Vous devez faire un plan de vérification pour la machine M5 (mesure de période du signal). Ce plan contiendra les spécifications fonctionnelles, les objectifs de vérification, les conditions particulières (si pertinent), un sommaire des tests et des résultats attendus. Votre plan de vérification devrait pouvoir servir à n'importe quelle autre équipe. Ce document de format PDF d'au plus une page sera déposé sur le site de dépôt des travaux dans le dossier *APP2-Seq-Devoir*. Vous devez nommer ce document sous la forme « cip1-cip2.pdf ». Cette première itération du plan de vérification fera l'objet d'une évaluation formative en grand groupe au début du procédural 2. Vous pourrez alors le bonifier avant la validation et le joindre avec votre rapport pour une évaluation sommative.

Semaine 2 : Validation

Mardi de la deuxième semaine

Le but cette activité est de valider la solution à la problématique que vous avez développée à partir d'une démonstration pratique avec un signal généré et des simulations utilisant le banc de test que vous avez développé à partir de votre plan de vérification. Pour cette validation, vous allez réaliser, en équipe, une présentation de 4 minutes maximum, durant laquelle les deux étudiants doivent parler. Vous serez en présence d'un des tuteurs et vous pouvez utiliser des supports visuels (images, powerpoint, simulation) pour démontrez l'atteinte des spécifications. Votre présentation doit comprendre les 4 éléments suivants :

1. Démarche d'analyse. Vous devez décrire, verbalement et à l'aide d'un schéma-bloc (schéma haut niveau et diagramme d'état), les entrées/sorties et le comportement de la MEF du **module M5**. Par la suite, vous devez expliquer sommairement comment le code VHDL répond aux spécifications. (*temps suggéré 1-1.5 minutes*)
2. Plan de vérification. Expliquez votre plan de vérification couvrant la MEF présentée au point 1. Ce plan contiendra les objectifs de vérification, les conditions particulières, un sommaire des tests et les résultats attendus. (*temps suggéré ~30 secondes*)
3. Démonstration du fonctionnement du prototype avec des ondes sinusoïdales de fréquences pertinentes en entrée. (*temps suggéré ~30 secondes*)
4. Exécution et visualisation des simulations à partir du banc de test. Vous devez très brièvement montrer votre banc de test en VHDL et les résultats obtenus par simulation dans Vivado. Concluez sur la couverture (entière, incomplète et/ou avec des erreurs) et sur les résultats de tests. (*temps suggéré 1-1.5 minutes*)

Semaine 2 : Remise du projet et rapport d'APP

À remettre le mercredi de la deuxième semaine avant 9h

Vous devez faire le téléversement de votre projet Xilinx sur le site WEB de dépôt des travaux dans le dossier *APP2-Seq-Projet*. Cette partie comporte les étapes suivantes :

1. Renommer le fichier .bit d'implantation de votre solution par le nom *cip1-cip2.bit*
2. Pour limiter la dimension du projet archivé, vérifiez la dimension des fichiers .wdb créés pour les simulations qui peuvent devenir très volumineux. Éventuellement, faites une réinitialisation de la simulation avant d'archiver pour réduire la dimension.
3. Générer une archive .zip (dans Vivado : menu file/project/Archive..) ou comprimer le répertoire du projet en suivant une démarche équivalente. Le fichier téléversé est un fichier « zippé » comprenant tous les fichiers sources nécessaires à l'implémentation et la simulation et le fichier .bit .
4. Le fichier zippé doit avoir le nom sous le format suivant *cip1-cip2.zip* où *cip_i* correspondent à votre cip.

Le rapport d'APP doit être remis **par équipe de 2**. Le rapport portera sur la MEF du décodeur (M1) et le module du calcul de la puissance (M6). Pour le module M1, inclure :

- Description du fonctionnement du décodeur avec court texte approprié se basant sur le schéma-bloc fourni (il n'est pas nécessaire de l'inclure dans le rapport);
- Diagramme d'états de la MEF;

Pour le module M6, inclure :

- Schéma-bloc du système;
- Description sommaire du fonctionnement;

Pour les deux modules, inclure :

- Plan de vérification
- Validation via une simulation avec le banc de test fourni. Vous devez démontrer le fonctionnement de vos modules au travers de simulations et expliquez en quoi vous répondez aux spécifications. Faites bon usage des curseurs, formatage des données et agrandissements pour appuyer vos propos.

Pour toutes vos explications, considérez que votre lecteur ne connaît pas votre démarche ni pourquoi un tel résultat démontre le bon fonctionnement – vous devez l'expliquez brièvement.

Un total de 4 pages est permis, excluant la page titre, la table des matières, l'introduction, la conclusion et les références. Nous ne consulterons pas le dépôt de vos fichiers Xilinx (sauf dans des cas particuliers) ce qui signifie que votre rapport doit être complet en lui-même. Le rapport doit suivre le guide de rédaction remis en S1.

Le rapport d'APP doit être rédigé (texte et figures) avec un logiciel de traitement de texte. Il sera téléversé **en format PDF** sur le site web de dépôt des travaux dans le dossier **APP2-Seq-Rapport**. Les textes et/ou figures illisibles seront sanctionnés. Le fichier sera nommé *cip1-cip2.pdf* où *cip* correspond au CIP de chacun des membres de l'équipe. Notez la présence du signe « - » entre les deux CIP. Le système n'acceptera pas le dépôt si le CIP de la personne faisant le dépôt n'apparaît pas dans le nom de fichier. La même note sera attribuée aux 2 membres de l'équipe. Tout retard sur le dépôt électronique entraînera une pénalité de 20% par jour.

Le rapport sera corrigé une seule fois et une seule note sera attribuée aux aspects de l'ingénierie, en fonction du Tableau 3. Cependant, la qualité de la communication influence grandement la note attribuée aux aspects de l'ingénierie. Ainsi, des phrases mal construites ou pouvant porter à interprétation, des nombres sans unité ou des graphiques incomplets sont autant d'exemples qui nuisent à la compréhension technique et font perdre des points sur cet aspect. Soyez donc précis et concis. La qualité de la communication sera évaluée selon le Tableau 4 et 15% pourront être retranchés à la note du rapport (GEN420 et GEN430) en fonction de la grille de correction. L'atteinte du niveau *cible* dans la grille assure de ne pas perdre de points, mais l'atteinte d'un niveau inférieur au niveau *cible* fait automatiquement perdre des points.

L'évaluation du rapport d'APP contribue à l'évaluation des éléments de compétence de l'unité. Il s'agit donc d'une évaluation sommative, c'est-à-dire que le résultat de l'évaluation sera consigné au dossier scolaire et utilisé dans le calcul de la note finale. Toutefois, pour permettre à l'étudiante ou l'étudiant d'apprendre de ses erreurs, son rapport corrigé lui sera remis, avec une grille lui permettant d'apprécier son niveau de compétence.

Il est important de noter qu'il est de la responsabilité de l'étudiante ou étudiant signataire du rapport de s'assurer de l'exactitude de la valeur de chaque élément de solution et de la qualité et de l'uniformité de l'ensemble du contenu de son rapport. N'oubliez pas que, lors de l'évaluation sommative (examen à la fin de l'unité), vous allez être évalués **de façon individuelle** sur les compétences mises en œuvre pour l'élaboration de ce rapport. Vous êtes donc réputé pouvoir résoudre de façon individuelle l'ensemble de la problématique de même que tout problème relié aux connaissances nouvelles à acquérir durant cette unité.

Les travaux et documents soumis pour évaluation doivent être une production originale de l'étudiant. Nous faisons appel à votre intégrité intellectuelle pour réaliser un travail original exempt de plagiat, incluant le plagiat entre collègues de travail. Le non-respect des règlements de l'Université concernant l'intégrité intellectuelle peut entraîner des sanctions académiques ou disciplinaires sévères.

Semaine 2 : Schéma de concept

En préparation du tutorat de clôture, il est suggéré d'apporter un schéma de concept en rapport avec la question suivante :

Quels sont les concepts nécessaires pour résoudre une problématique de logique séquentielle ?

Pondérations des évaluations

Tableau 1 : Pondérations des évaluations

Activité	Élément de compétence	Rapport d'APP	Validation au laboratoire	Examen Sommatif	Totaux
GEN420_2	(APP2) Modéliser l'information discrète et son évolution temporelle. Déterminer les structures de données et les algorithmes appropriés pour les mettre en œuvre.	45		135	180
		22	23		
GEN430_2	(APP2) Concevoir et réaliser des systèmes numériques combinatoires à partir de spécifications.	45		135	180
		19	26		

Les notes du rapport et de la validation (qui inclut le devoir) apparaîtront comme additionnées en une seule note sur les grilles de résultats de la session.

Note : GEN420-2 correspond aussi à GEN265-2 et GEN430-2 à GEN-225-2 (anciennes AP).

Grilles d'évaluation

Tableau 2 : Grille d'évaluation de la validation

La validation concerne principalement le module M5

	Activité	Validation	Validation	Validation	Validation
	Compétence	GEN420_2	GEN420_2	GEN430_2	GEN430_2
	Critère	Analyse globale d'un problème de logique séquentielle	Analyse d'une machine à états finis	Validation du circuit du module par simulation et vérification des fonctionnalités du module sous test	Réalisation d'un plan de vérification
Niveaux	Pondération	12,00	11,00	12,00	14,00
Excellent	100,00%	Modélise l'information discrète séquentielle en employant une représentation adéquate complète et structurée.	Applique une méthode valide d'analyse et de représentation de la machine à états finis.	Valide le fonctionnement des circuits du FPGA grâce à une simulation globale complète réalisée correctement avec les outils et les méthodes qui conviennent.	Réalise un plan de vérification avec une couverture complète. L'application du plan démontre l'atteinte des spécifications.
Cible	80,00%	Modélise adéquatement l'information discrète séquentielle, mais emploie une représentation peu structurée.	Applique la méthode avec une démarche cohérente et un bon résultat, mais des erreurs mineures.	Valide le fonctionnement des circuits du FPGA grâce à une simulation globale, mais présente des lacunes mineures.	Réalise un plan de vérification avec des lacunes mineures. L'application du plan démontre un fonctionnement adéquat.
Seuil	50,00%	Modélise partiellement l'information discrète séquentielle.	Applique la méthode avec une démarche partielle. Applique la méthode avec plusieurs erreurs.	Valide le fonctionnement des circuits du FPGA grâce à une simulation partiellement valide.	Réalise un plan de vérification minimal pour les principaux cas à analyser. L'application du plan est partielle et ne démontre un fonctionnement suffisant.
Non satisfaisant	25,00%	Modélise mal l'information discrète séquentielle. Des lacunes majeures sont présentes.	Applique la méthode, mais avec des erreurs importantes et la démarche est déficiente.	Valide le fonctionnement des circuits du FPGA grâce à une simulation avec lacunes majeures.	Réalise un plan de vérification insuffisant. L'application du plan est incohérente et ne démontre pas le fonctionnement.
Non initié	0,00%	Incapacité à modéliser l'information discrète séquentielle.	Incapacité à appliquer une méthode valide d'analyse et de représentation de la machine à états finis.	Incapacité à valider le fonctionnement des circuits du FPGA par une simulation globale.	Incapacité à réaliser et appliquer un plan de vérification.

Tableau 3 : Grille d'évaluation du rapport

Le rapport concerne principalement les modules M1 et M6 ainsi que le fonctionnement global

	Activité	<i>rapport</i>	<i>rapport</i>	<i>rapport</i>	<i>rapport</i>
	Compétence	GEN420_2	GEN420_2	GEN430_2	GEN430_2
	Critère	Analyse globale d'un problème de logique séquentielle	Génération d'une solution à un problème de logique séquentielle: registres d'états, transitions, équations logiques.	Génération d'une solution à un problème de logique séquentielle: registres d'états, transitions, équations logiques.	Validation du circuit du module par simulation et vérification des fonctionnalités du module sous test
Niveaux	Pondération	11,00	11,00	11,00	8,00
Excellent	100,00%	Modélise l'information discrète séquentielle en employant une représentation adéquate complète et structurée.	Génère une solution valide à un problème de logique séquentielle.	Applique et développe une méthode valide, cohérente, complète et formelle d'implémentation des modules en VHDL.	Réalise un plan de vérification avec une couverture complète. L'application du plan démontre l'atteinte des spécifications.
Cible	80,00%	Modélise adéquatement l'information discrète séquentielle, mais emploie une représentation peu structurée.	Génère une solution à un problème de logique séquentielle avec une démarche cohérente et un bon résultat, mais des erreurs mineures.	Applique une méthode valide, cohérente et formelle d'implémentation des modules en VHDL, présentant des problèmes mineurs.	Réalise un plan de vérification avec des lacunes mineures. L'application du plan démontre un fonctionnement adéquat.
Seuil	50,00%	Modélise partiellement l'information discrète séquentielle.	Génère une solution à un problème de logique séquentielle avec une démarche cohérente mais incomplète ou qui comporte plusieurs erreurs	Applique une méthode d'implémentation VHDL des modules présentant des problèmes de cohérence ou de formalisme.	Réalise un plan de vérification minimal pour les principaux cas à analyser. L'application du plan est partielle et ne démontre un fonctionnement suffisant.
Non satisfaisant	25,00%	Modélise mal l'information discrète séquentielle. Des lacunes majeures sont présentes.	Génère une solution à un problème de logique séquentielle, mais avec des erreurs importantes et la démarche est déficiente.	N'applique pas une méthode valide pour l'implémentation VHDL des modules (mais des éléments de connaissance sont démontrés).	Réalise un plan de vérification insuffisant. L'application du plan est incohérente et ne démontre pas le fonctionnement.
Non initié	0,00%	Incapacité à modéliser l'information discrète séquentielle.	Incapacité à générer une solution à un problème de logique séquentielle : registres d'états, transitions, équations logiques.	Incapacité à réaliser l'implémentation VHDL des modules.	Incapacité à réaliser et appliquer un plan de vérification.

Tableau 4: Évaluation de la communication

Actu Qualité	communication_rapport		communication_rapport		communication_rapport		communication_rapport	
	Q7E	Organisation	Q7E	Présentation	Q7E	Références	Q7E	Graphiques
Rubrique	Organiser l'information		Présenter de l'information		Recourir à des références		Rédiger dans une langue de qualité	
Critère	Organiser l'information		Présenter de l'information		Recourir à des références		Rédiger dans une langue de qualité	
Excellent	L'étudiante ou l'étudiant organise efficacement l'information pour en faciliter la compréhension chez son auditoire. Les transitions entre les sections témoignent d'une compréhension approfondie du sujet.		L'étudiante ou l'étudiant présente de l'information pertinente et complète au regard des objectifs de la communication. Le sujet est bien délimité et est présenté avec concision.		L'étudiante ou l'étudiant recourt à des références pertinentes et variées pour appuyer ses propos. Les citations et paraphrases présentées sont toutes pertinentes.		L'étudiante ou l'étudiant rédige des phrases complètes et bien structurées. Il/elle respecte les règles grammaticales élémentaires, en plus d'utiliser une terminologie et un vocabulaire parfaitement appropriés.	
Cible	L'étudiante ou l'étudiant organise adéquatement l'information pour en permettre la compréhension chez son auditoire. La qualité des transitions entre les sections pourrait toutefois être améliorée.		L'étudiante ou l'étudiant présente de l'information pertinente et complète au regard des objectifs de la communication.		L'étudiante ou l'étudiant recourt à des références pour appuyer ses propos. Les citations et paraphrases présentées sont généralement pertinentes.		L'étudiante ou l'étudiant rédige généralement des phrases complètes et bien structurées. Il/elle respecte les règles grammaticales élémentaires, en plus d'utiliser une terminologie et un vocabulaire généralement appropriés.	
Seuil	L'étudiante ou l'étudiant organise minimalement l'information pour en permettre la compréhension. Les transitions entre les sections sont soit abruptes, soit inefficaces.		L'étudiante ou l'étudiant présente de l'information pertinente au regard des objectifs de la communication, mais des éléments importants sont soit mal présentés, soit absents.		L'étudiante ou l'étudiant recourt à des références pertinentes, mais peu variées. Les citations et paraphrases présentées sont soit trop longues, soit déformées/inexactes.		L'étudiante ou l'étudiant fait des erreurs d'orthographe, de syntaxe et de grammaire, mais celles-ci ne nuisent pas de façon importante à la compréhension de son texte. Il/elle utilise une terminologie et un vocabulaire minimalement appropriés.	
Non satisfaisant	L'étudiante ou l'étudiant n'est pas en mesure d'organiser l'information pour en permettre la compréhension. Il n'y a aucune transition entre les sections.		L'étudiante ou l'étudiant éprouve de la difficulté à sélectionner l'information pertinente à présenter au regard des objectifs de la communication.		L'étudiante ou l'étudiant recourt à trop peu de références et/ou abuse de citations et de paraphrases.		L'étudiante ou l'étudiant éprouve de la difficulté à rédiger des phrases complètes et bien structurées, de même qu'à respecter les règles grammaticales élémentaires, ce qui nuit à la compréhension de son texte. Il/elle n'utilise pas une terminologie et un vocabulaire appropriés.	
Non initié	Totalelement non satisfaisant.		Totalelement non satisfaisant.		Absence complète de référence.		Totalelement non satisfaisant.	
							non initié	

Références

Manuels de référence

- Charles Roth Jr. , Lizy John ; 2017 ; Digital Systems Design Using VHDL, 3rd Edition(3rd ed.); ISBN-13: 978-1-305-63514-2, ISBN-10: 1-305-63514-0 : <https://www.nelsonbrain.com/shop/isbn/9781305635142>
- John F. Wakerly, «Digital Design, Principles & Practices» 4ième édition, Pearson Prentice Hall. 2006, ISBN 0-13-186389-4, 895 pages.

Annexe au guide :

- Annexe Description et spécifications du circuit « APP logique séquentielle », Site WEB de l'APP circuits logiques séquentiels.

Guides de référence de la carte Zybo-z7

- Manuel technique : <https://reference.digilentinc.com/reference/programmable-logic/zybo-z7/reference-manual>
- Schéma : https://reference.digilentinc.com/_media/reference/programmable-logic/zybo-z7/zybo_z7_sch-public.pdf

Notices techniques

- Notice « datasheet » du CODEC « Analog Devices SSM2603 Audio Codec » : <https://www.analog.com/media/en/technical-documentation/data-sheets/ssm2603.pdf>
- Site de la compagnie Xilinx : www.xilinx.com
- Familles de circuits logiques de Texas Instruments : <http://focus.ti.com/docs/logic/logichomepage.jhtml>