

Sujet 1 : Projet en autonomie de conception de circuit numérique

Pour ce projet, vous utiliserez une carte de type Digilent NEXYS4. Cette carte sera mise à votre disposition pendant la durée du projet.

Durée du projet : 9 séances de 4h00 plus 1 séance de soutenance/démonstration

Objectifs du projet :

- Développer un système permettant de faire la synthèse de l'expérience acquise lors des séances de travaux pratiques sur le langage VHDL et la synthèse de circuits numériques.
- Mettre en pratique de façon autonome les méthodes de développement, de débogage et de validation.
- Evaluer votre capacité à mener un projet à bien avec méthode et rigueur : de la rédaction du cahier des charges à la validation par le test.

Sujet du projet :

Vous devez concevoir et implanter une architecture numérique en utilisant la carte Digilent NEXYS4 mise à votre disposition. L'architecture numérique développée devra réaliser des fonctions que vous aurez vous-même définies avec votre encadrant. Ces fonctions devront **au minimum** vous « permettre » de mettre en œuvre :

- Une machine à états finis,
- Une interface utilisateur avec entrées et sorties :
 - o Exemples d'entrées : switches, boutons poussoirs, liaison série avec l'ordinateur, capteur...
 - o Exemples de sorties : LEDs simples, afficheurs 7 Segments, LEDs tricolores, signal sonore, liaison série avec l'ordinateur, écran VGA, écran texte OLED, actionneur...
- Le processeur minimal 8 bits à quatre instructions développé dans le module EN217 (optionnel pour l'année 2019-2020).

Chaque sujet de projet doit *a priori* être original pour chaque binôme. En cas de doublons, les architectures numériques devront être suffisamment différentes pour éviter tout doute quant aux auteurs du projet. Vous proposerez votre sujet à votre encadrant qui le validera, l'amendera, ou le refusera, en fonction (au moins) des critères suivants :

- Conformité aux exigences minimales,
- Niveau de difficulté technique (dans l'absolu et/ou par rapport à l'aisance du binôme),
- Originalité de l'application / de l'architecture,
- Faisabilité selon le matériel disponible.

L'application peut être un jeu, la simulation ou le contrôle d'un système. Son fonctionnement doit faire l'objet en fin de projet d'une démonstration fonctionnelle sur la carte Digilent NEXYS 4.

Remarque : Les cartes Digilent NEXYS4 seront mises à disposition de la promotion sur la durée du projet. De plus, un grand nombre de périphériques listés dans la suite de ce document sont disponibles au magasin du département Electronique. Vous pourrez le cas échéant les emprunter.

Déroulement du projet :

- 1) Rédaction du cahier des charges en précisant les objectifs intermédiaires et les dates prévues pour les atteindre,
- 2) **Soumission** du projet à votre encadrant pour validation. (Modalités de soumission à voir avec votre encadrant) **au minimum 3 jours** avant la première séance de projet, à savoir le **Dimanche 16 Février 2020 avant 20h00.**
- 3) Après accord de l'encadrant : **Définition de l'architecture**, (identification des blocs d'Electronique numériques nécessaires, découpage, choix des méthodes de vérification...), écriture des schémas hiérarchiques, définition des machines d'états...
- 4) **Réalisation** du projet. C'est à cette étape uniquement que s'effectuent les descriptions en VHDL, ce qui signifie que, quand vous décrivez un bloc, vous devez toujours être en mesure de fournir des documents écrits des 3 étapes précédentes et d'expliquer le positionnement de ce bloc dans l'architecture globale.
- 5) **Rédaction du rapport** (voir ci-dessous).
- 6) **Présentation** en binôme, courant Juin (selon emploi du temps).

Nous vous recommandons de fixer des objectifs hebdomadaires sur la durée du projet.

Il vous est évidemment interdit de «réutiliser» un projet intra-groupe, extra-groupe ou inter promotion. Si tel est votre besoin, cela doit être fait avec l'accord de votre encadrant, du binôme auteur et en citant vos sources.

Evaluation du projet :

Le projet sera évalué en tenant compte des paramètres suivants :

- Qualité de la démarche,
 - o Architecture,
 - o Style d'écriture VHDL : indentation, noms des identifiants (noms des fichiers, des projets, des entités, des architectures, des signaux), etc
 - o Rigueur des vérifications / simulations,
- Atteinte des objectifs annoncés,
- Clarté du discours, des explications, des questions au cours des séances,
- Rapport écrit,
- Soutenance,
- Investissement (sur l'ensemble du semestre, absentéisme inclus).

Le résultat de l'évaluation dépend principalement du travail effectué, des méthodes employées et de votre dextérité dans l'utilisation de vos connaissances et des compétences développées autour du thème des architectures d'électronique numérique. Choisir un projet basique n'est donc pas le meilleur choix stratégique puisque vous ne pourrez pas faire preuve d'un large éventail de capacités. A l'inverse, choisir un projet trop complexe ne vous permettra pas d'atteindre les objectifs minimums exigés, à savoir : architecture numérique fonctionnelle s'exécutant sur le circuit FPGA de la carte NEXYS 4 avec une organisation détaillée et une estimation des ressources assignées au sein du FPGA.

Rapport (les modalités de remise du rapport seront fournies par votre encadrant)

Le rapport rassemblera une description des éléments que vous avez conçus pour votre projet, ainsi que l'ensemble de votre démarche intellectuelle pour produire ce résultat. Il doit donc comprendre au moins :

- Le cahier des charges originel

- La stratégie d'architecture adoptée
 - o Où se fait quel calcul ? pourquoi ?
 - o Quel format pour les données ? pourquoi ?
 - o Quand et comment communiquer les données entre les différents modules ?
- Les schémas hiérarchiques
- Un commentaire des descriptions VHDL les plus intéressantes ou subtiles, une explication de leur fonctionnement.
- La stratégie de test pour la validation des modules (choix de l'environnement de test)
- Des chronogrammes significatifs commentés/annotés
- Une présentation et discussion des résultats de synthèse (ressources, fréquence...)

Conseils :

- Ne vous contentez pas de votre production technique, valorisez également votre démarche intellectuelle.
- Sauf avis contraire, il n'est pas utile d'imprimer des fichiers VHDL, envoyez-les plutôt par e-mail (pensez à faire un fichier zip). N'oubliez pas le fichier de contraintes.
- Un rapport trop long prend du temps à écrire (et à corriger), sachez doser le contenu, cela fait également partie de l'exercice.
- Une simulation n'est pas une illustration, explicitez comment la lire, pointez ce qui valide le résultat. Si elle doit être imprimée, évitez le fond noir.

Soutenance

La soutenance s'effectue devant une présentation de type *powerpoint* dans la salle projet, avec à disposition la carte NEXYS 4. Les objectifs sont :

- Montrer que vous pouvez présenter un projet technique de façon claire et concise.
- Montrer un chronogramme de simulation et le commenter.
- Démontrer le fonctionnement du projet sur la carte NEXYS4 (Quelque chose DOIT fonctionner)
- Discuter des ressources matérielles utilisées, justifier les performances technologiques (temps et occupation) du circuit synthétisé.

Les soutenances durent 20 minutes : 10mn de présentation et 10mn de questions. Chaque groupe dispose du temps de soutenance du groupe précédent pour se préparer et installer sa démonstration. Veillez donc à ne pas confondre l'heure de passage et l'heure de convocation. Veillez également à vous partager équitablement le temps de parole.

Les questions auront pour objectif principal :

- Vérification que le projet est bien un travail original (si le rapport ne le permet pas).
- Vérification que les membres du binôme maîtrisent le projet.
- Estimation du recul des membres du binôme sur la solution architecturale et sur la conduite du projet.

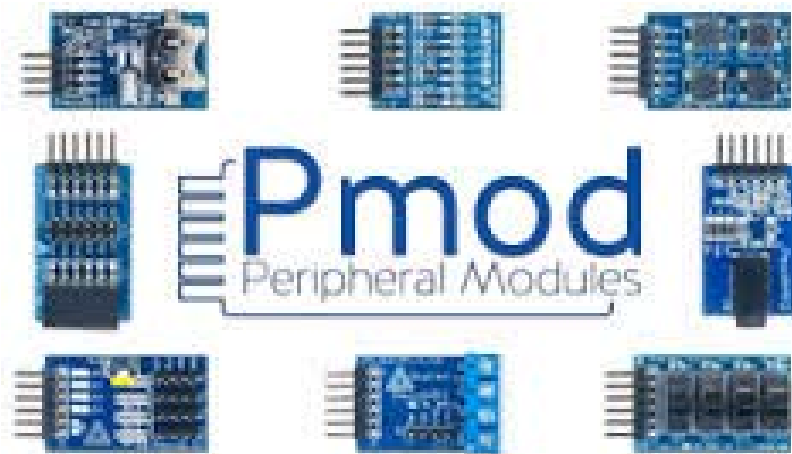
Conseils :

Gardez en tête que la soutenance sert à évaluer vos compétences plus que le projet rendu, montrez donc que vous savez commenter de manière pédagogique un schéma bloc, une machine d'états, une simulation, un rapport de synthèse...

Lors de la soutenance, vous n'aurez pas le temps de tout présenter. Faites des choix stratégiques sur ce que vous estimez le plus intéressant (techniquement subtil, permettant de mettre en avant vos compétences, bien adapté pour dire ce que l'examineur espère entendre...).

Faites (au moins) une répétition chronométrée.

Périphériques pour la carte NEXYS4



Nous disposons au sein du département électronique de nombreux périphériques qui peuvent s'interfacer avec la carte NEXYS 4. En effet, les ports d'E/S du FPGA sont repris sur 4 connecteurs femelles spécifiques sur la carte NEXYS 4. Ces connecteurs permettent l'utilisation de petits modules d'extensions optionnels (appelés "Pmod"). Une fois enfichées sur les connecteurs femelles de la platine "NEXYS4", ces modules vous permettront d'adjoindre de multiples possibilités et interfaces supplémentaires.

Modules Pmod Convertisseurs

Ces derniers vous permettront d'ajouter facilement et rapidement des convertisseurs "analogique/numérique" ou "numériques/analogique" à vos applications.

Modules Pmod Prototypage (USB/UART, PmodBB, RS232)

Ces derniers vous permettront d'ajouter des possibilités de prototypage et de raccordement rapide à vos applications.

Modules Pmod Mémoire (Cartes microSD)

Ces derniers vous permettront d'ajouter des possibilités de stockage mémoire additionnelles à vos applications.

Modules Pmod entrées/sorties (micro, jack audio, VGA)

Ces derniers vous permettront d'ajouter des possibilités d'entrées/sorties supplémentaires à votre application.

Modules Pmod radiofréquence (GPS, WIFI, 2.4GHz RF)

Ces derniers vous permettront d'ajouter des possibilités de communication radiofréquence sans fil à vos applications.

Modules Pmod leds / Afficheurs Oled

Ces derniers vous permettront d'ajouter des possibilités d'affichage par LEDs ou afficheurs OLED à vos applications.

Modules Pmod capteurs (lumière, température, pression, distance, capteur magnétique, gyroscope)

Ces derniers vous permettront d'ajouter des capteurs divers à vos applications.

Modules Pmod boutons / Switchs (KeyPad)

Ces derniers vous permettront d'ajouter des boutons-poussoirs, claviers, joystick, interrupteurs à vos applications.

Plus d'informations pour chacun de ces modules sont disponibles sur le site de Digilent :

<http://store.digilentinc.com/pmod-modules/>

Sujet 2 : Projet assisté de conception de circuit numérique

Pour ce projet, vous utiliserez une carte de type Digilent NEXYS4. Cette carte sera mise à votre disposition pendant la durée du projet.

Durée du projet : 9 séances de 4h00 plus 1 séance de soutenance/démonstration

Objectifs du projet :

- Développer un système permettant de faire la synthèse de l'expérience acquise lors des séances de travaux pratiques sur le langage VHDL et la synthèse de circuits numériques.
- Mettre en pratique les méthodes de développement, de débogage et de validation.
- Evaluer votre capacité à mener à terme un projet proposé et détaillé par l'enseignant.

Sujet du projet :

L'objectif final du projet assisté est de concevoir un lecteur de fichiers MP3. Pour ce faire, il est décomposé en trois parties complémentaires :

- Partie 1 : Développer l'interface utilisateur du lecteur de fichiers MP3 à l'aide de la carte FPGA NEXYS4.
- Partie 2 : Concevoir l'architecture de lecture, de modulation de type PWM et de transmission sur le sortie audio (Jack) de la carte FPGA NEXYS4.
- Partie 3 : Associer les deux contributions précédentes dans une architecture globale pour la lecture de fichiers MP3.

Partie 1 : Concevoir l'interface utilisateur du lecteur de fichiers MP3

L'architecture à concevoir, puis à prototyper sur une carte NEXYS4 devra permettre de gérer les interfaces d'un lecteur de fichiers MP3. Dans le détail, il faudra afficher sur 3 afficheurs 7-segments la valeur courante d'un compteur/décompteur de 1 à 599. L'affichage devra avoir la cadence suivante : unité 100 ms, dizaine 1s et centaine 10s. De plus, un nombre compris entre 1 et 9 plus un point devra être affiché sur un quatrième afficheur 7-segments. Le compteur associé devra pouvoir être incrémenté ou décrétementé par deux boutons poussoirs.

Par ailleurs, un codage sera affiché sur les 4 autres afficheurs 7-segments. Quatre valeurs seront nécessaires suivant les états de la machine d'états :

- [--] : ETAT INIT ou ETAT STOP
- ---] : ETAT PLAY-FWD
- [--- : ETAT PLAY-BWD
- ---- : ETAT PAUSE

Fonctionnement :

Le principe d'utilisation de la carte contenant l'architecture demandée est le suivant. Avant tout chose, il faut initialiser toutes les bascules de l'architecture en positionnant *RAZ* à 1, puis en le remettant à 0. Quatre des afficheurs doivent alors afficher **5.001** pour les fonctions de comptages/décomptages.

La fréquence de fonctionnement de l'horloge associée au projet est de 100 MHz. Cela signifie que des signaux d'activation sur les fronts montants de l'horloge (*Clock Enable*) sont nécessaires pour afficher de l'information à des fréquences exploitables par l'œil humain. Dans ce projet, deux signaux d'activation sont nécessaires :

- $CE_{\text{perception}}$: ce signal permet de cadencer les huit afficheurs 7 segments en tenant compte de la perception rétienne.
 \Rightarrow *Fréquence* $\sim 3 \text{ kHz}$
- $CE_{\text{affichage}}$: ce signal permet de gérer le défilement des nombres entre 1 et 599
 \Rightarrow *Fréquence* $\sim 10 \text{ Hz}$ (*100 ms*)

Consignes à respecter pour la définition de l'architecture :

L'entité devra respecter le schéma suivant au niveau du nom et du type des E/S :

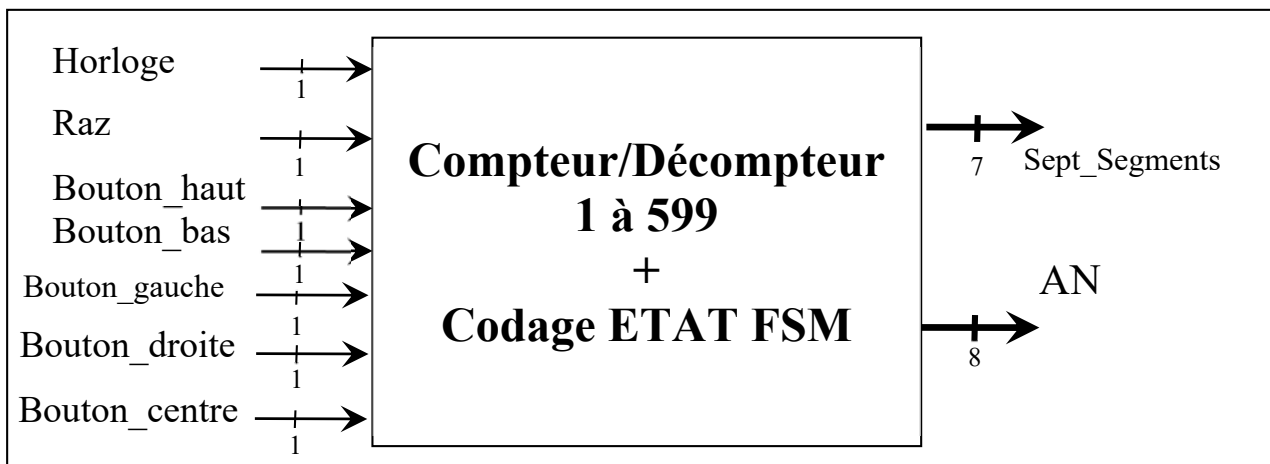


Figure 1 : entité de l'architecture

Entrées :

- ✓ Horloge : horloge du circuit fournie par un quartz à 100MHz,
- ✓ Raz : remise à zéro synchrone de l'architecture globale,
- ✓ Bouton_haut : entrée d'activation de la fonction de comptage 1 à 9
- ✓ Bouton_bas : entrée d'activation de la fonction de décomptage 1 à 9
- ✓ Bouton_gauche : entrée de commande de la machine d'états (B_LEFT)
- ✓ Bouton_droite : entrée de commande de la machine d'états (B_RIGHT)
- ✓ Bouton_centre : entrée de commande de la machine d'états (B_CENTER)

Sorties :

- ✓ Sept_Segments : affichage sur les 7-segments de la donnée courante du compteur et du codage (*Bus de 8 bits*).
- ✓ AN : indication de l'afficheur 7-segments courant, (*Bus de 8 bits*).

Afin de vous aider dans la réalisation de ce travail, nous vous donnons la découpe d'architecture. Cette dernière est constituée de 8 modules comme décrit dans la Figure 2.

Figure 2 : architecture globale du projet

Les 8 modules de l'architecture principale ont les fonctionnalités suivantes :

- ✓ **Gestion des signaux d'activation (*gestion_freq*)** : génération de deux signaux d'activation sur les fronts montants de l'horloge (*Clock Enable*). Les fréquences ont été définies préalablement à la page 1 du document.
- ✓ **Filtrage séquentiel associée aux boutons (*detec_impulsion*)** : synchronisation des 5 entrées associées aux 5 boutons poussoirs à l'aide de bascules. Utilisation du signal d'horloge pour la synchronisation.
- ✓ **Machine à états finis (*fsm_MP3*)** : machine à états finis du projet. Elle est spécifiée dans la suite de ce document.
- ✓ **Fonction compteur/décompteur (*cpt_1_599*)** : il s'agit du compteur/décompteur dont la valeur varie de 1 à 599 en fonction des signaux Play-Pause, Restart et Forward provenant de la FSM. Utilisation du signal $CE_{\text{affichage}}$ comme signal d'activation.
- ✓ **Fonction compteur/décompteur (*cpt_1_9*)** : il s'agit d'un compteur/décompteur dont la valeur varie de 1 à 9 en fonction des signaux Volume_UP et Volume_DW provenant de la FSM.
- ✓ **Transcodage pour afficheur sept segment (*Trans*)** : les valeurs issues des blocs compteur/décompteur représentent des nombres entiers codés sur n-bits. Ce bloc permet donc le transcodage de ces informations vers un format compatible avec un afficheur sept segments. Il permet aussi de spécifier le codage retenu sur les quatre autres afficheurs 7-segments.
- ✓ **Sélection de l'anode (*mod8*)** : les huit afficheurs sept-segments sont alimentés par huit anodes différentes qu'il faut sélectionner de façon cyclique. Ce bloc génère les signaux d'activation de ces anodes à l'aide d'un compteur modulo 8.
- ✓ **Module d'affichage 7 segments (*mux8*)** : commandé par le signal provenant du module mod8, ce multiplexeur sélectionne l'une des huit données à afficher sur les afficheurs 7 segments de manière synchrone avec le choix de l'anode. C'est un point important de l'architecture pour organiser correctement l'affichage.

Par ailleurs, l'architecture est entièrement synchrone. Ceci implique :

- les bascules, les registres et les compteurs sont commandés par la même horloge.
- la copie de l'entrée sur la sortie d'un bloc séquentiel sur le front montant de l'horloge est autorisée par un signal de type CE (Clock Enable).
- l'entrée synchrone de mise à 0 n'est utilisée qu'à l'initialisation du circuit. **Elle ne doit en aucun cas être utilisée pendant le fonctionnement normal du circuit.**

La machine à états finis est un bloc devant centraliser le contrôle de l'architecture. L'entité de cette dernière est décrite dans la Figure 3 :



Figure 3 : Entité de la FSM

Par ailleurs, le diagramme d'états associé à la machine d'états finis à concevoir est spécifié comme suit :

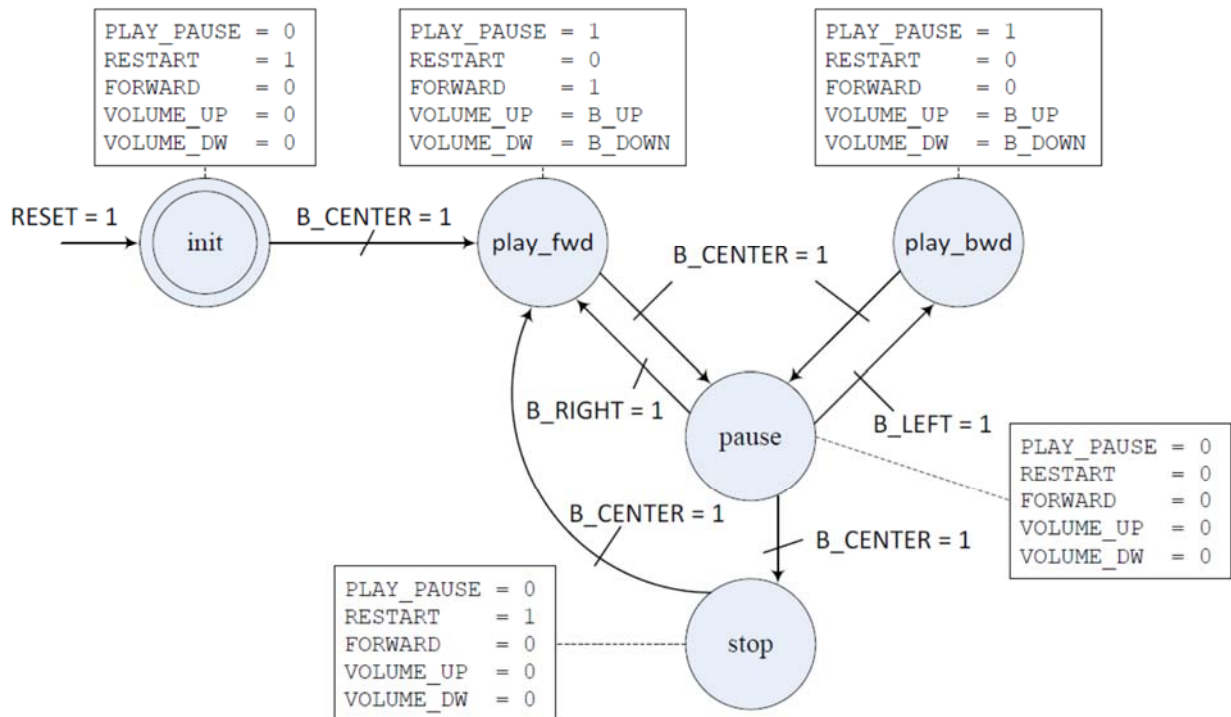


Figure 4 : Diagramme d'états de la FSM

Vous devez décrire en VHDL le couple entité/architecture respectant le schéma bloc et le diagramme d'états.

Consignes à respecter pour le déroulement du projet

Il est demandé de travailler de manière progressive avec des étapes intermédiaires devant être validées par votre encadrant :

- L'utilisation des outils ne doit débuter que lorsque le circuit est découpé en fonctions élémentaires (compteur, registre, décodeur, etc) et que les interconnexions entre les modules sont parfaitement comprises.
- Dans un deuxième temps, il faut se focaliser sur chacun des modules et les valider par des simulations comportementales à l'aide de testbenchs décrits en VHDL.

-c) Enfin, il faut associer les 7 blocs constituant l'architecture au sein d'une description VHDL. La fonctionnalité du module final devra également être simulée avant la synthèse, l'implémentation, la génération du bitstream et la configuration du circuit FPGA.

Partie 2 : Concevoir l'architecture de lecture, de modulation de type PWM et de transmission sur le sortie audio (Jack)

La sortie Jack de la carte Nexys 4 (nommée « Mono Audio Out ») est reliée au FPGA à travers un filtre de *Butterworth* discret (filtre passe bas du 4^{ème} ordre). Ce filtre de *Butterworth* est en charge de transformer le signal « binaire » sortant du FPGA (+Vcc/Gnd) en une tension « analogique » variant entre 0 et +5V (cf. la partie 15 du manuel de référence de la carte Nexys 4). Sous réserve d'une gestion temporelle adéquate, il est possible de reconstruire un signal audio à l'aide d'une seule sortie binaire et de ce filtre analogique.

Votre première tâche dans cette seconde partie est de concevoir le module situé en amont du filtre analogique. Pour ce faire, vous devrez transformer des informations numériques représentant l'amplitude du signal audio sous en un signal modulé suivant le format PWM (Pulse Width Modulation).

La modulation PWM :

La modulation PWM est un format de représentation de l'information basé sur la longueur d'une impulsion sur un créneau temporel borné (cf. la documentation de la carte, section 15.1 Pulse-Width Modulation).

L'exemple suivant décrit le comportement temporel du système lorsque **p** cycles d'horloge sont disponibles pour coder un échantillon audio. Si l'amplitude du signal audio est égale à **d**, alors la sortie de la modulation PWM doit être égale à 1 durant les premiers **d** cycles. Le reste de la période, soit (**p-d** cycles), la sortie doit être maintenue à la valeur 0.

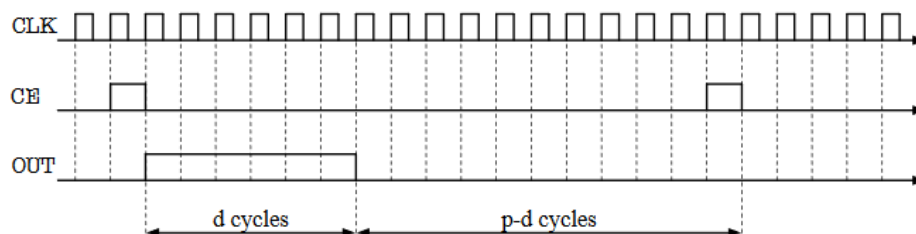


Figure 5 : Chronogramme de la gestion séquentielle du module PWM

Nous souhaitons gérer des séquences audio échantillonnées à **44100 Hz** sur la sortie audio de la carte FPGA. Afin de calculer la résolution du module PWM, il va falloir prendre en compte de la fréquence d'horloge présente au sein du circuit FPGA (100 MHz). En effet, cette dernière va limiter la dynamique des données utilisables.

*Tâche n°1 : Calculer la valeur maximale des échantillons audio (valeur de **p**) sachant que l'horloge présente au sein du FPGA est de 100 MHz.*

Maintenant que vous connaissez la dynamique d'entrée du module PWM, il ne vous reste plus qu'à le décrire en VHDL. La structure du module à concevoir est présenté dans la Figure 6.

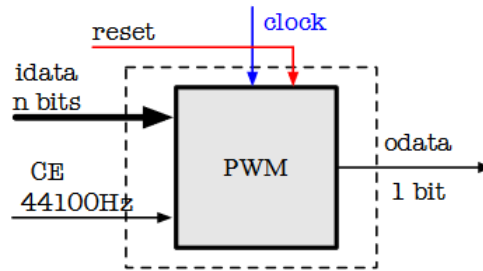


Figure 6 : Entité du module PWM

Tâche n°2 : Spécifier le nombre de processus puis décrire en VHDL le couple Entité/architecture

Tâche n°3 : Simuler le comportement de votre nouveau module VHDL.

Conception du lecteur d'échantillons audio :

Pour obtenir un son sur la sortie audio de la carte, il est nécessaire d'alimenter votre module PWM avec des échantillons audio. Vous allez donc devoir concevoir un système plus évolué. Le schéma du système est détaillé dans la Figure 7 :

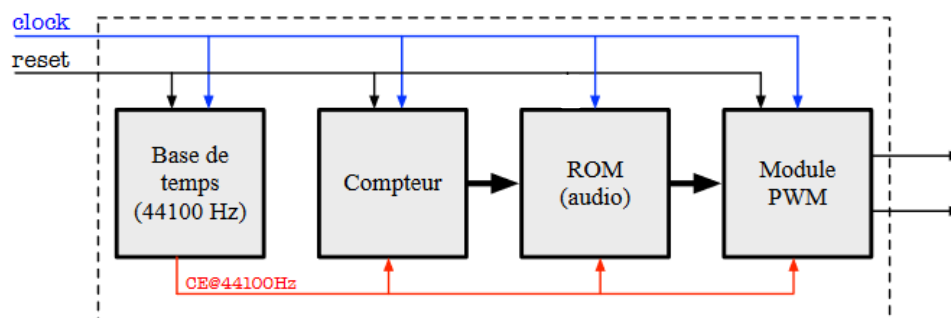


Figure 7 : architecture de génération d'échantillons audio

Les séquences audio que vous manipulerez ont été échantillonnées à 44100 Hz, cela implique que les échantillons audio doivent être envoyés au module PWM à cette cadence. En conséquence, dans ce système la base de temps correspond à un compteur de période égale à $1s/44100Hz$. Les échantillons audio seront stockés dans une mémoire de type ROM. Cette mémoire sera fournie. Sa profondeur sera de 44100 données et chaque donnée sera codée sur 11 bits. Le compteur d'adresses situé en amont de la mémoire ROM permettra de parcourir l'ensemble des données disponibles dans cette dernière au rythme de la base de temps.

Attention : les données présentes dans la mémoire ROM sont comprises entre $[-1024$ et $1023]$. Elles sont donc au format *signed* (complément à 2) sur 11 bits. Vous devrez les transformer en des données comprises entre $[0$ et $4095]$, au format *unsigned* (binaire pur) sur 12 bits. Cette modification de la représentation devra être faite à l'aide d'un processus implicite supplémentaire dans le module PWM.

Tâche n°1 : Spécifier en VHDL le compteur d'adresses et la base de temps (gestion du CE)

Tâche n°2 : Demander à votre encadrant le module VHDL décrivant une mémoire de type ROM à votre encadrant(ROM_Sinus.vhd) puis associer ces 4 modules dans un 5^{ème}.

Tâche n°3 : Simulez le comportement du système pour valider son fonctionnement. Afin de vous aider dans cette tâche, vous prendrez le soin de visualiser les valeurs transitant entre le module ROM et le module PWM. Pour ce faire, vous utiliserez le mode d’affichage analogique dans le simulateur ISIM.

Implémentation et prototypage sur carte FPGA :

Maintenant que votre système est complètement décrit et fonctionnel, il ne vous reste plus qu’à le tester sur carte afin d’écouter le sinus de la ROM. Pour cela il est indispensable d’ajouter un fichier de contrainte XDC adéquat.

Chargement de fichiers audio :

Afin de rendre ce système un peu plus flexible, il est nécessaire de remplacer la mémoire de type ROM par une mémoire de type RAM. Toutefois pour permettre le chargement de séquences audio dans la mémoire RAM une fois le FPGA configuré, il est indispensable de charger les échantillons audio depuis un PC à l’aide d’une liaison série.

Le développement de la partie liaison série du système assurant cette fonctionnalité a été préalablement effectué. Vous devrez donc intégrer le module développé dans votre système. Le système que vous devez obtenir est schématisé dans la Figure 8.

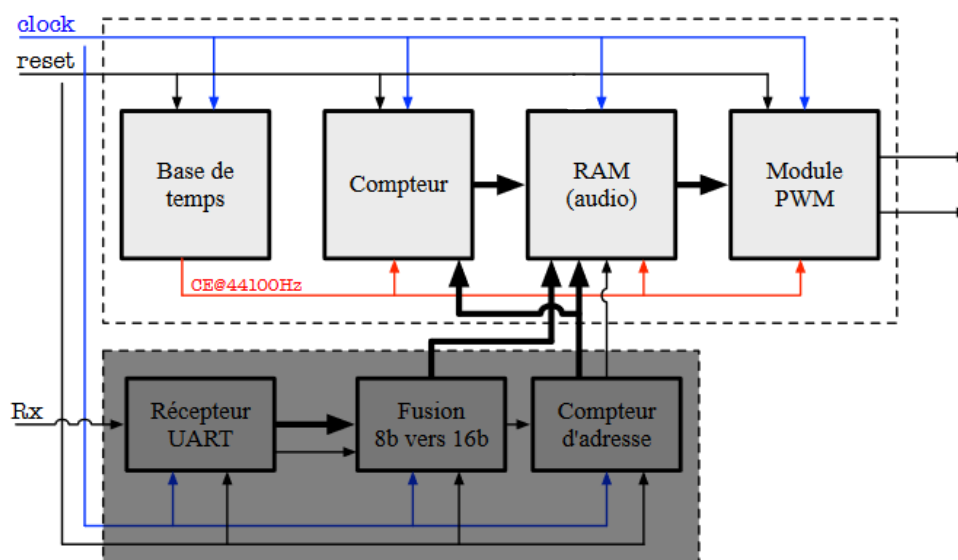


Figure 8 : intégration de la liaison série dans l’architecture

Des modifications sont nécessaires pour deux de vos modules :

- 1- Il faut transformer la ROM en RAM pour permettre l’écriture de données dans la mémoire (adresse d’écriture, bus, d’écriture et signal de validation d’écriture).
- 2- Le compteur d’adresses permettant la lecture des données dans la mémoire doit être borné par une valeur maximale qui dépend des données écrites préalablement.

Tâche n°1 : Demander à votre encadrant les modules VHDL décrivant la liaison série puis ajouter ces modules à votre projet. Vous ne devez pas modifier les modules fournis.

Tâche n°2 : Modifier le module compteur d'adresse et le module ROM comme demandé.

Tâche n°3 : Associer l'ensemble des modules dans un nouveau de plus haut niveau puis implémenter et prototyper l'architecture résultante. Modifier en conséquence le fichier de contrainte XDC.

Tâche n°4 : Prototyper sur circuit FPGA en utilisant un utilitaire de type Tera Term pour le chargement des fichiers audio. Demander à votre encadrant des exemples de fichiers audio.

Gestion du volume audio :

Une dernière fonctionnalité doit être développée. Il s'agit d'un module de gestion du volume. Ce module supplémentaire doit être positionné entre la mémoire de type RAM et le module PWM. Il est possible d'assurer simplement cette fonctionnalité. En effet, il suffit de diviser la valeur de l'échantillon provenant de la mémoire par une puissance de 2 selon une valeur de volume spécifiée (2^9 – valeur du volume).

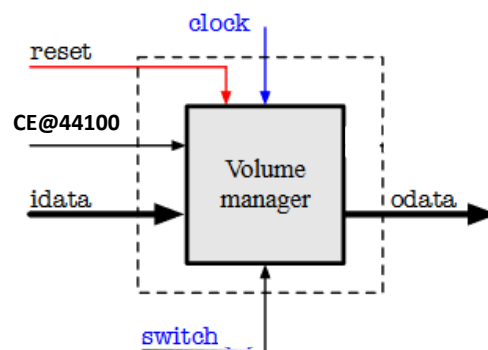


Figure 9 : Entité du module de gestion du volume

L'entrée switch permet de sélectionner une valeur atténuation du volume comprise entre 1 et 9.

Tâche n°1 : Spécifier en VHDL le module de gestion du son.

Tâche n°2 : Associer ce module à votre projet puis implémenter et prototyper l'architecture résultante. Modifier en conséquence le fichier de contrainte XDC.

Tâche n°3 : Prototyper sur circuit FPGA en utilisant un utilitaire de type Tera Term pour le chargement des fichiers audio. Demander à votre encadrant des exemples de fichiers audio.

Partie 3 : Associer les contributions des parties 1 et 2 dans une architecture globale pour la lecture de fichiers MP3

Le flux audio est pour le moment joué en continu. Vous allez maintenant développer une interface de contrôle qui doit permettre :

- ✓ De mettre en pause, arrêter, jouer en avant, jouer en arrière la séquence audio.
- ✓ D'augmenter et de diminuer le volume sonore
- ✓ D'afficher des informations sur les afficheurs 7 segments :
 - Les 4 afficheurs de gauche indiqueront l'état du lecteur (STOP, PAUSE, FWD, BWD) ;
 - Le premier des afficheurs de droite indiquera le volume sonore (1 => 9) ;
 - Les trois derniers représenteront le temps de lecture écoulé (2 chiffres pour les secondes et une pour les dixièmes). Cet affichage sera croissant lors d'une lecture FWD et décroissant lors d'une lecture BWD.

Pour une plus grande clarté du fonctionnement attendu, demandé à votre enseignant de vous faire une démonstration.

En l'état actuel de vos développements, les deux parties ne sont pas complètement compatibles. Par exemple, vous ne pouvez pas arrêter ou changer le sens de lecture de votre lecteur audio. Vous devez pour cela modifier le compteur d'adresses.

Tâche n°1 : Adapter votre lecteur audio afin de supporter les signaux de contrôle issus de la partie 1.

Tâche n°2 : Ecrire en VHDL le fichier de haut niveau assurant l'association des modules des deux parties. Modifier en conséquence le fichier de contrainte XDC.

Tâche n°3 : Synthétiser, Implémenter et prototyper sur circuit FPGA.

Evaluation du projet :

Le projet sera évalué en tenant compte des paramètres suivants :

- Qualité de la démarche,
 - o Architecture,
 - o Style d'écriture VHDL : indentation, noms des identifiants (noms des fichiers, des projets, des entités, des architectures, des signaux), *etc*
 - o Rigueur des vérifications / simulations,
- Atteinte des objectifs annoncés,
- Clarté du discours, des explications, des questions au cours des séances,
- Rapport écrit,
- Soutenance,
- Investissement (sur l'ensemble du semestre, absentéisme inclus).

Le résultat de l'évaluation dépend principalement du travail effectué, des méthodes employées et de votre dextérité dans l'utilisation de vos connaissances et des compétences développées autour du thème des architectures d'électronique numérique.

Rapport (les modalités de remise du rapport seront fournies par votre encadrant)

Le rapport rassemblera une description des éléments que vous avez conçus pour votre projet, ainsi que l'ensemble de votre démarche intellectuelle pour produire ce résultat. Il doit donc comprendre au moins :

- Le cahier des charges originel
- La stratégie d'architecture adoptée
 - o Où se fait quel calcul ? pourquoi ?
 - o Quel format pour les données ? pourquoi ?
 - o Quand et comment communiquer les données entre les différents modules ?
- Les schémas hiérarchiques
- Un commentaire des descriptions VHDL les plus intéressantes ou subtiles, une explication de leur fonctionnement.
- La stratégie de test pour la validation des modules (choix de l'environnement de test)
- Des chronogrammes significatifs commentés/annotés
- Une présentation et discussion des résultats de synthèse (ressources, fréquence...)

Conseils :

- Ne vous contentez pas de votre production technique, valorisez également votre démarche intellectuelle.
- Sauf avis contraire, il n'est pas utile d'imprimer des fichiers VHDL, envoyez-les plutôt par e-mail (pensez à faire un fichier zip). N'oubliez pas le fichier de contraintes.
- Un rapport trop long prend du temps à écrire (et à corriger), sachez doser le contenu, cela fait également partie de l'exercice.
- Une simulation n'est pas une illustration, explicitez comment la lire, pointez ce qui valide le résultat. Si elle doit être imprimée, évitez le fond noir.

Soutenance

La soutenance s'effectue devant une présentation de type *powerpoint* dans la salle projet, avec à disposition la carte NEXYS 4. Les objectifs sont :

- Montrer que vous pouvez présenter un projet technique de façon claire et concise.
- Montrer un chronogramme de simulation et le commenter.
- Démontrer le fonctionnement du projet sur la carte NEXYS4 (Quelque chose DOIT fonctionner)

- Discuter des ressources matérielles utilisées, justifier les performances technologiques (temps et occupation) du circuit synthétisé.

Les soutenances durent 20 minutes : 10mn de présentation et 10mn de questions. Chaque groupe dispose du temps de soutenance du groupe précédent pour se préparer et installer sa démonstration. Veillez donc à ne pas confondre l'heure de passage et l'heure de convocation. Veillez également à vous partager équitablement le temps de parole.

Les questions auront pour objectif principal :

- Vérification que le projet est bien un travail original (si le rapport ne le permet pas).
- Vérification que les membres du binôme maîtrisent le projet.
- Estimation du recul des membres du binôme sur la solution architecturale et sur la conduite du projet.

Conseils :

Gardez en tête que la soutenance sert à évaluer vos compétences plus que le projet rendu, montrez donc que vous savez commenter de manière pédagogique un schéma bloc, une machine d'états, une simulation, un rapport de synthèse...

Lors de la soutenance, vous n'aurez pas le temps de tout présenter. Faites des choix stratégiques sur ce que vous estimez le plus intéressant (techniquement subtil, permettant de mettre en avant vos compétences, bien adapté pour dire ce que l'examineur espère entendre...).

Faites (au moins) une répétition chronométrée.