DEVIS GÉNÉRAL

Description du projet

ELE748-01
Architecture de système ordiné

Action à prendre : Approbation du projet.

À l'intention de : M. Simon Pichette

Préparé par : Kevin Parent Legault & Jonathan Lapointe

2015-06-14

Montréal, Québec

Table des matières

Description du système

Dans le but d'amorcer la première phase du projet 1 qui consiste à réaliser un système complet à l'aide de la technologie des FPGA SOC, on a à sélectionner un système que l'on pourra intégrer et ainsi appliquer certaines notions essentielles telles que l'intégration d'accélérateur matériel à un concept. Le présent devis vise à faire accepter notre projet et ainsi valider que notre stratégie de conception, pour ce projet, soit valide. En globalité, le système que l'on désire implanter sur la plaquette de développement TERASIC DE1 contenant une microplaquette Altera Cyclone V SOC consiste essentiellement à réaliser un synthétiseur audio. Ce synthétiseur pourra être contrôlé au moyen d'un système d'interface utilisateur ce qui rendra le système interactif.

Le système sera apte à générer huit notes musicales situées dans l'octave moyenne soit de 220 Hz (la 3e octave) à 440Hz (la 4e octave). De plus, le synthétiseur audio sera capable de générer plusieurs sons d'instrument notamment un piano, une guitare, une flûte et un saxophone. L'option d'enregistré de nouveau son d'instrument pour ainsi pouvoir les intégrer en tant que nouveaux instruments disponibles sera également disponible.

La réalisation sera fragmentée en plusieurs parties pour ainsi assurer une conception efficace. Dans ce cas, nous fragmenterons la conception et nous fixerons des points de contrôle afin d'obtenir un rendu réaliste et qui correspond aux objectifs d'étape tel que décrit dans le diagramme qui suit.

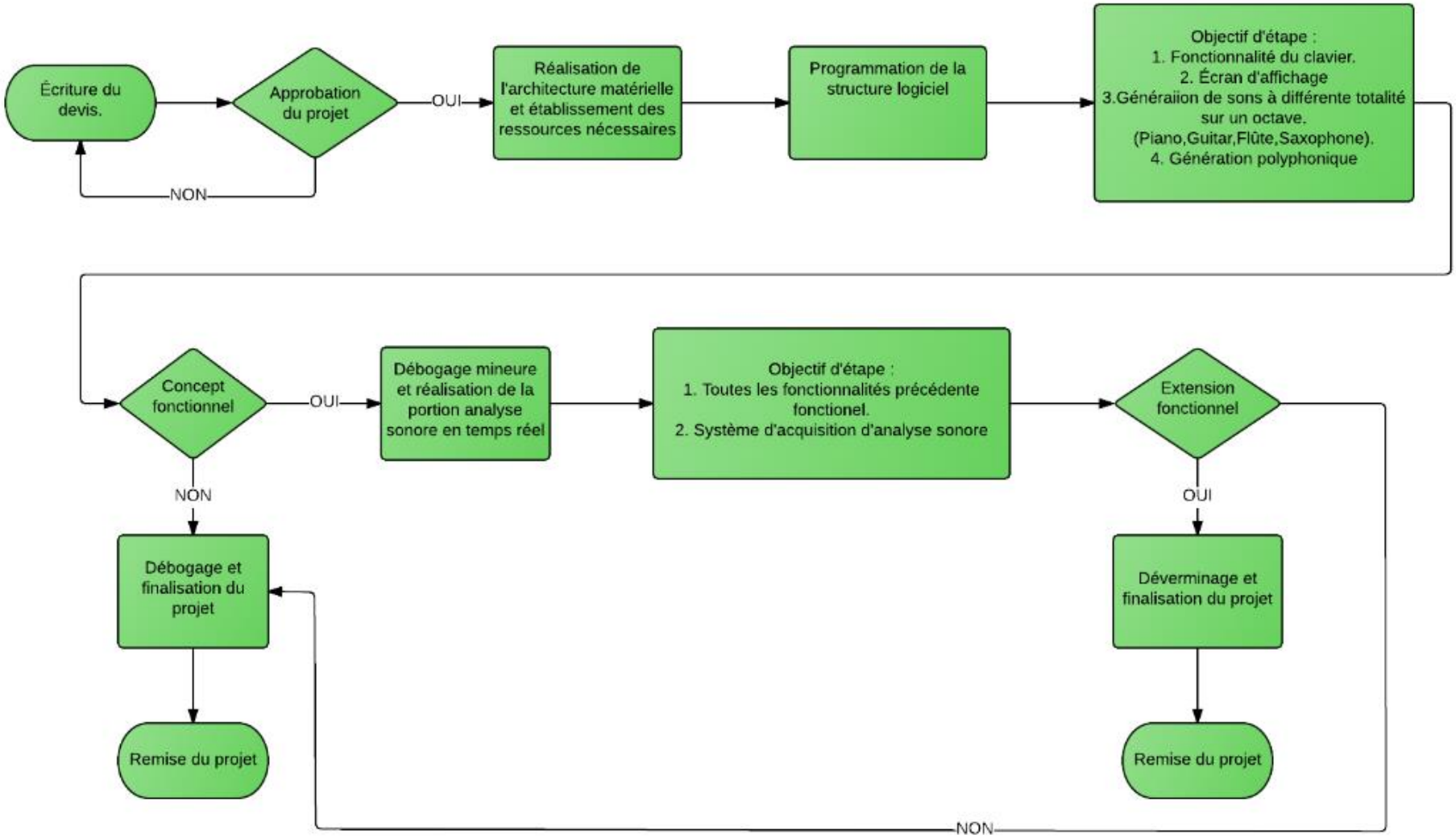

Figure 1 Diagramme de conception

Volet implémentation physique

Cette section vise à spécifier les technologies choisies afin de réaliser le système et ainsi atteindre les objectifs. L'architecture choisie est une architecture de base incluant plusieurs périphériques internes et externes.

Architecture physique privilégiée.

Les schémas de la figure 2 décrivent bien l'ensemble des composantes qui formera notre design. Les modules IP core seront hautement privilégiés. De cette façon, on s'assure que notre architecture physique sera optimale et robuste.

-Le processeur

Nous utiliserons l'IP cores NIOSII/s afin de réaliser notre projet. Ce module sera utilisé en tant que CPU de notre système

-La mémoire

Les ressources n'étant pas vraiment limité et étant donné que nous utiliserons l'affichage graphique seulement pour afficher les touches du piano, on va allouer la mémoire on microplaquette de256Ko pour la portion vidéo et la SDRAM de 64MB pour la portion système et autre périphérique.

-L'accélérateur matériel et codec audio

Dans l'optique d'obtenir une précision qui sera représentative des sons que l'on souhaite générer, on n'aura aucun autre choix que d'utiliser le FPU déjà intégré afin de procéder aux différents calculs reliés notamment à la génération de l'enveloppe de notre signal, de même qu'à l'analyse du signal. En second lieu, nous ferons la conception d'un accélérateur matériel qui fera la même opération que ce que l'on demandait au FPU. Pour ce qui est la portion acquisition et transmission de signaux audio, le module IP core codec audioWM8731 sera intégré à notre architecture. De cette façon, on pourra acquérir des données audio pour en faire l'analyse de même qu'envoyer des données audio.

-Interface graphique

L'interface graphique sera exactement le même que pour le laboratoire 2. En effet, ayant plusieurs semaines à travailler sur le laboratoire 2, on croit que cette plate-forme sera idéale en ce qui trait à l'affichage de même que l'interaction humain-machine.

-Interface utilisateur

Le même module IP core que pour la souris PS2 sera utilisé. Soit le PS2 Controller de la librairie d'altera. Partant du même principe que la souris, il n'y aura aucune modification à faire au niveau matériel mis à part du fait que l'on devra spécifiiez, cette fois-ci, que le périphérique externe utilisé sera un clavier. Le clavier exercera la fonction d'interface usager. L'utilisateur pourra notamment utiliser les touches prédéfinies du clavier pour activer des sons à la fréquence sonore associée. Ex. (A, Bmin, C, D…)

Volet implémentation logiciel

Dans cette portion, on procédera à l'intégration, à l'analyse et à l'interaction des tous les systèmes de façon à atteindre les objectifs d'étape précédemment illustré.

L'algorithme pour la synthèse sonore

Pour faciliter la compréhension de ces concept de synthèse audio phonique, on a procédé à différente simulation sur matlab permettant ainsi de bien comprendre l'étendu du projet et la bonne méthode à utiliser afin de synthétisé un son.

Notre concept abordera l'algorithme de même que les bases de la synthèse FM. En effet, afin de générer un son, on procédera à la modulation en phase de même qu'en amplitude d'un signal de référence. Dès lors, en capturant et/ou en ayant des données stockées en mémoire sur les amplitudes de chacune des harmoniques de notre signal, nous serons en mesure de régénérer l'enveloppe de notre signal en sommant l'ensemble des harmoniques. Les codes matlab suivants, disponible dans les librairies de matlab, on permit l'analyse du concept. De ce fait, on en est venu à la conclusion, que ce serait la meilleure méthode afin d'analysé notre signal audio de même que transmettre un son de synthèse.

- *Script Matlab pour l'analyse sonore*

```
function analyze(file)
% Matlab function analyze(file)
% plots the waveform and power spectrum of a wav sound file.
% For example, type analyze('piano.wav') at the Matlab prompt.
%
% Mark R. Petersen, U. of Colorado Boulder Applied Math Dept, Feb 2004


[y, Fs] = wavread(file);      % y is sound data, Fs is sample frequency.
t = (1:length(y))/Fs;         % time

ind = find(t>0.1 & t<0.12);   % set time duration for waveform plot
figure; subplot(1,2,1)
plot(t(ind),y(ind))
axis tight
title(['Waveform of ' file])

N = 2^12;                     % number of points to analyze
c = fft(y(1:N))/N;            % compute fft of sound data
p = 2*abs( c(2:N/2));         % compute power at each frequency
f = (1:N/2-1)*Fs/N;           % frequency corresponding to p

subplot(1,2,2)
semilogy(f,p)
axis([0 4000 10^-4 1])
title(['Power Spectrum of ' file])
```

Figure 2 Script matlab pour analyse sonore

Toutes les fonctions ont été testé et celle-ci donnent des résultats très convaincants. De plus, grâce à ces outils d'analyse on a la possibilité de récolté les données d'amplitude des harmoniques pour ainsi les stockés en mémoire et régénérer les signaux. Voici les résultats d'analyse sonore de quelques instruments.

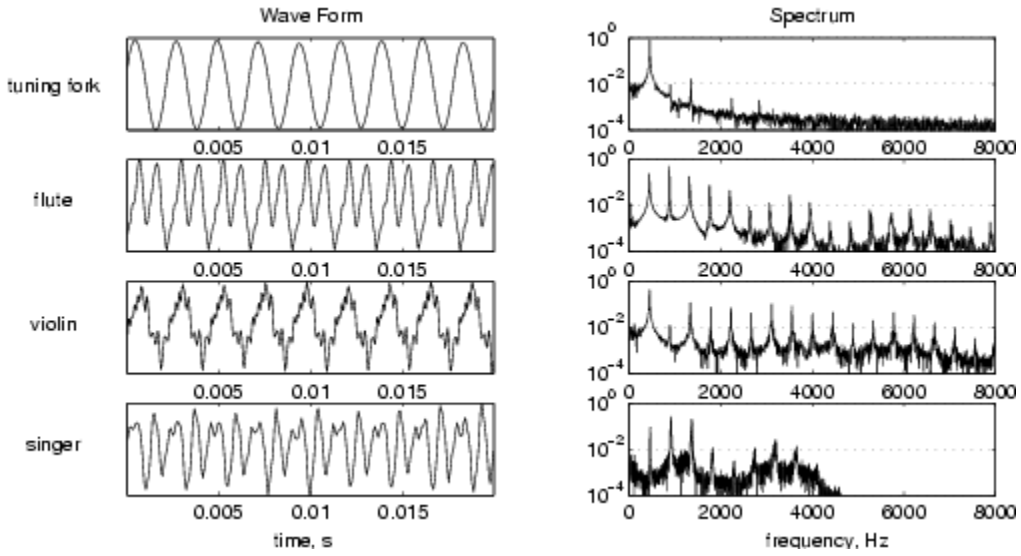

Figure 3 Réponse fréquentielle et temporel d'un son

- Script matlab pour la Synthèse FM

Tel qu'on peut le constater à la figure 4, la fonction appel en boucle p(n)*Cos (2pi*n*f*t) ce qui permet d'exécuté la sommation de toute les n harmoniques du signal à générer. La partie complexe du signal soit jsin(2pi*n*f*t) est négligé pour ainsi récupéré que la partie réel.

```
function synthesize(file,f,d,p)
% Matlab function synthesize(file,f,d,p)
% creates a .wav audio file of a sound where the fundamental frequency
% and amplitudes(power) of the harmonics may be specified.
%
% file is a string which is the name of the .wav file.
% f is the fundamental frequency in Hz
% d is the duration in seconds
% p is a length n vector of amplitudes
%
% For example, synthesize('test.wav', 220, 3, [1 .8 .1 .04])
% makes a 3 second sample at 220 Hz with the harmonics shown.
%
% Mark R. Petersen, U. of Colorado Boulder Applied Math Dept, Feb 2004

Fs=22050; nbits=8;              % frequency and bit rate of wav file

t = linspace(1/Fs, d, d*Fs);    % time
y = zeros(1,Fs*d);              % initialize sound data
for n=1:length(p);
  y = y + p(n)*cos(2*pi*n*f*t); % sythesize waveform
end
y = .5*y/max(y);                % normalize.  Coefficent controls volume.
wavwrite( y, Fs, nbits, file)
```

Figure 4 Script synthèse fm

Méthodologie privilégié pour l'intégration de la polyphonie

À cette étape, on tient en compte que la génération de signaux audio sur plusieurs tonalités est fonctionnelle. À cet effet, la stratégie, ici, sera basée sur le même principe que la superposition de signaux analogiques utilisant un op amp pour la réalisation du circuit. En effet, la polyphonie deviendra possible grâce à plusieurs buffers(8) qui pourront contenir la signature de chacun des signaux que l'on souhaite générer. De ce fait, on additionnera les résultantes des signaux sélectionnés ensemble, mais à des fréquences de cadencement différente tout dépendamment de la note que l'on souhaite superposé. Le résultat final devrait, en théorie, être la superposition de tous les signaux audio sélectionnés.