Projet Microprocesseur

1 Méthodologie

1.1 Dates

- 3 novembre : présentation du projet ; formation des groupes
- 7 novembre : travail individuel, réflexions sur l'architecture
- 20 novembre : date limite pour le retour du rapport écrit sur le simulateur
- 27 novembre : date limite pour le retour du programme du simulateur
- 4 décembre : date limite pour le retour d'une version préliminaire des spécifications du microprocesseur
- 8 décembre : correction / remarques sur le simulateur de net-list et sur l'architecture du microprocesseur
- 15 décembre : travail individuel

L'idéal serait de rendre votre projet avant les vacances de noel...

1.2 Travail demandé

Le projet se décompose en deux parties : vous devrez d'abord réaliser un simulateur de netliste ; pour ensuite l'utiliser pour la simulation d'un processeur de votre conception, faisant tourner un programme de votre conception représentant la montre digitale du cours.

1.3 Méthode

Vous travaillez par groupe de 2 ou 3. Les documents à rendre doivent être typographiés (latex si possible).

2 Simulateur

2.1 Enjeux

Le but de ce projet est de réaliser un simulteur de circuit digital synchrone. Vous êtes libres de beaucoup de détails d'implémentation.

Il vous faut mettre au point un langage de description des net-list. Il doit intégrer les opérations logiques de base ainsi que le registre.

Réfléchissez sur les exemples simples ci-après aux différentes passes de votre simulateur : lecture de la *net-list*, simulation proprement dite.

Le simulateur peut être hiérarchique (possibilité de définir des blocs ou fonctions synchrones) ou à plat. La hiérarchie vous aidera considérablement pour la génération du projet microprocesseur.

Pour ce qui est du langage, vous pouvez inventer le vôtre ou vous immerger dans un langage existant. Gardez à l'esprit que vous aurez à l'utiliser pour un projet d'envergure, le microprocesseur. En particulier, celui-ci nécessitera, pour conserver des performances correctes, l'ajout d'une primitive RAM et/ou ROM.

La discussion de tous les aspects de votre simulateur est vivement encouragée. JBNote est à votre disposition.

2.2 Jeu de tests

Les circuits suivants (éventuellement traduits dans votre langage de *net-list*) doivent être simulés:

i. Full-Adder

$$\begin{aligned}(s,r) &= \text{FullAdd}(a,b,c)\,\{\\ s &= a \otimes b \otimes c\\ r &= (a \wedge b) \vee ((a \otimes b) \wedge c)\,\}\end{aligned}$$

ii. Moins unaire

$$\begin{aligned}y &= \text{minus}(x)\,\{\\ y &= x \otimes c\\ c &= Z(x \vee y)\,\}\end{aligned}$$

iii. Compteur modulo 2

$$\begin{aligned}(s,r) &= \text{cm2}(x)\,\{\\ s &= Z(x \otimes s)\\ r &= x \wedge s\,\}\end{aligned}$$

iv. Diviseur d'horloge

$$\begin{aligned}o &= \text{clk2}()\,\{\\ o &= Z(c)\\ c &= \sim Z(o)\,\}\end{aligned}$$

v. Compteur modulo 24 pour les heures de la montre : à vous de donner le schéma et la *net-list*.

2.3 Interface

Les entrées du simulateur sont :

i. la *net-list* du circuit

ii. le nombre n de cycles de simulation, ainsi que la définition de ces entrées sur n cycles.

La sortie du simulateur est :

i. la valeur des sorties calculées sur n cycles

2.4 Échéance & documents

Les documents à rendre sont composés d'un document écrit et du code source de votre simulateur.

Pour le document écrit, vous y ferez figurer :

1. Une spécification du langage de description de net-liste et des valeurs d'entrée
2. Un bref compte-rendu sur la mécanique interne de votre simulateur, et les difficultés rencontrées.
3. Pour chacun des 4 premiers circuits ci-dessus, un jeu de test minimal nécessaire et suffisant pour en établir la correction. Justifiez la réponse.
4. Le résultat de la simulation des circuits ci-dessus, sur au moins 16 cycles.

Pour le code source, il s'agit d'envoyer par mail (`jean-baptiste.note@ens.fr`) une tarball de votre application. Précisez par mail une target de *Makefile* ou un script à exécuter pour lancer l'ensemble des simulations et pouvoir récupérer les sorties. Précisez l'emplacement des fichiers de description des *net-list*.

Le document écrit est à remettre avant le 20 novembre 2004 ; le code source doit me parvenir avant le 27 novembre 2004.

3 Microprocesseur

3.1 Enjeux

Il s'agit de concevoir un microprocesseur capable d'exécuter un programme faisant fonctionner la montre digitale du cours.

Le travail se décompose en deux temps, une phase de spécification, une phase de réalisation à l'aide de votre simulateur de *net-list*.

3.2 Spécification

Cette phase englobe

i. la conception du jeu d'instruction de votre micro

ii. la conception de son architecture

iii. la simulation logicielle du code assembleur de votre specification

Au terme de cette étape, vous devez être capable de simuler le calcul de la montre du cours 1.

Outre les choix classiques sur l'architecture de votre microprocesseur (mémoire, taille des registres, etc...), vous devez déjà penser aux problèmes concrets d'entrée-sortie qui vont se poser : compter le temps, afficher l'heure.

Vous pouvez faire un processeur généraliste ou hautement spécialisé.

3.3 Réalisation

Il s'agit maintenant de définir le circuit réalisant votre microprocesseur, afin, in fine, de simuler votre code assembleur compilé par votre simulateur de *net-list*.

3.4 Échéances et documents

Ce projet donne lieu à deux échéances : avant le 4 décembre 2004, remise de la spécification de votre processeur. Le rapport écrit comprendra la spécification de votre jeu d'instructions et de l'architecture. Le programme comportera une simulation haut niveau du programme de l'horloge.

Le rapport final est à remettre avant les vacances de Noel. La partie écrite comprend la décomposition hiérarchique et la spécification de chaque bloc du microprocesseur. La partie programme doit comprendre :

1. Un assembleur pour votre processeur (voire un compilateur !)
2. Un script premettant de lancer la simulation de votre micro à partir du source du micro sous forme *net-liste*, et du source du programme de la montre.

4 Bibliographie

Vous pourrez trouver sur Internet de nombreuses spécification de jeux d'instruction dont vous pouvez vous inspirer pour votre processeur. Deux livres sont à votre disposition (passez me voir) : Digital Circuit Design de Niklaux Wirth, et Microcomputer Structures, d'Henry d'Angelo.