ASR1, DM2
Un peu d'assembleur de votre processeur

Andreea Chis, Matthieu Gallet, Bogdan Pasca

4 décembre 2008

1 Intro technico-commerciale

Ce devoir est à faire en binôme et à rendre pour vendredi 19 décembre à 22h par mail.

Il faut commencer par récupérer le fichier `RISC2008v1.0.tgz` sur la page web de Bogdan Pasca [1], lire les README des répertoires asm, puis simu, et tester dans le simulateur les deux programmes objets fournis.

L'écran se compose de 60 lignes de 80 caractères, commence à l'adresse 0x2FFF, et se termine à l'adresse 0x3FFF. Chaque caractère est codé par un octet.

Vous êtes invités à vous inspirer du programme `deco.asm` fourni.

2 Le nez dans le programme d'assemblage

Modifiez le programme `asm.cc` pour y ajouter la génération de code pour chaque instruction. Aucune connaissance de C++ est nécessaire, un peu d'imitation suffira. Vous avez comme exemple la génération de code pour l'instruction **ldl**.

Vous allez rendre un fichier `asm.cc` qui fonctionne toujours, qui est capable de convertir le fichier deco.asm en deco.obj, de sorte que deco.obj marchera dans le simulateur fourni.

Au passage, la structure de ce programme est un *recursive descent parser*.

Bonus : le simulateur a été écrit sur un coin de table en période de surmenage. Tout bug original rapporté (on n'a pas laissé des erreurs dedans exprès, promis !) vaudra un nombre de points proportionnel à sa subtilité.

3 Coder

3.1 Écrire

Écrivez une routine `plot` qui affiche un caractère sur l'écran. Elle prendra x, y et le code du caractère respectivement dans r1, r2 et r3. Elle se permet de modifier ces trois variables. Elle ne vérifie pas que la position du caractère est dans l'écran ($0 \leq x \leq 79$, $0 \leq y \leq 59$), et a le droit de planter le système si ce n'est pas le cas.

1. http ://perso.ens-lyon.fr/bogdan.pasca/ASR1/

3.2 Texte

Écrivez une routine `plottext` qui affiche un texte sur l'écran. Elle prendra les coordonnées de départ pour l'affichage x, y, l'adresse du début du texte dans la mémoire, et le nombre de caractères à afficher dans les registres r1, r2, r3 et r4 respectivement.

3.3 Un peu de ... math !

On va s'occuper de l'évaluation d'un polynôme dans un point donné.

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$$

Écrivez une routine en langage assembleur pour afficher le polynôme. Les paramètres de cette routine seront le degré du polynôme (donné en r4) et l'adresse en mémoire où se trouve le coefficient pour a_0 (en r5). Les autres coefficients se trouveront à des adresses successives à partir de cette adresse.

Puis, écrivez une routine pour évaluer ce polynôme dans un point donné par le registre r6.

Bonus : affichez un message d'erreur si l'évaluation n'est pas possible. Cela veut dire que s'il y a un overflow, il faut afficher un message le disant.

3.4 Du joli boulot !

Écrivez un programme qui affiche les caractères de A a Z et les chiffres de 0..9, saute deux lignes, puis affiche « Joyeux Noël ». Vous devez rendre un fichier `noel.asm`.

Bonus : en utilisant votre routine `plot`, faites tomber des flocons de neige sur l'écran.