

Rendu TP 1 Informatique embarquée

Equipe :

- Rayane Bait
- Guilhem Mizrahi

Exercice 1 :

Le codage des instructions utilise le schéma suivant

- Chaque instruction est codée sur 32 bits
- Opération sur 3 bits correspond à
 - opération arithmétique
 - opération arithmétique immédiate
 - opération de contrôle de logique
 - load word
 - save word
- La première opérande est toujours un registre code sur 6 bits (64 registres numérotés de 0 à 63)
- La deuxième opérande est toujours un registre codé sur 6 bits
- La troisième opérande peut être un registre sur 6 bits ou une constante sur 12 bits
- Fonction sur 3 bits correspond à l'indication de si l'opération est ADD, SUB, MUL...

On note que le total de bits utilisé est $3 + 6 + 6 + 12 + 3 = 30$. De plus, grâce à l'utilisation de du code Fonction qui n'est pas utilisé lors des opérations Load et Save, il est possible d'utiliser ces bits supplémentaires pour la constante et ainsi adresser plus de mémoire. En l'occurrence, il est envisageable d'avoir des constantes avec 5 bits (17 bits au total) de plus et ainsi adresser $2^5 = 32$ fois plus de mémoire (130ko de mémoire).

Table de correspondance :

Instruction	Opération	Opérande 1	Opérande 2	Opérande 3	Function
ADD	001	reg (6 bits)	reg (6 bits)	reg (6 bits)	001
SUB	001	reg (6 bits)	reg (6 bits)	reg (6 bits)	010
MUL	001	reg (6 bits)	reg (6 bits)	reg (6 bits)	011
DIV	001	reg (6 bits)	reg (6 bits)	reg (6 bits)	100
AND	001	reg (6 bits)	reg (6 bits)	reg (6 bits)	101
OR	001	reg (6 bits)	reg (6 bits)	reg (6 bits)	110
XOR	001	reg (6 bits)	reg (6 bits)	reg (6 bits)	111
ADDi	010	reg (6 bits)	reg (6 bits)	cst (12 bits)	001
SUBi	010	reg (6 bits)	reg (6 bits)	cst (12 bits)	010
MULi	010	reg (6 bits)	reg (6 bits)	cst (12 bits)	011
DIVi	010	reg (6 bits)	reg (6 bits)	cst (12 bits)	100
ANDi	010	reg (6 bits)	reg (6 bits)	cst (12 bits)	101
ORi	010	reg (6 bits)	reg (6 bits)	cst (12 bits)	110
XORi	010	reg (6 bits)	reg (6 bits)	cst (12 bits)	111
BEQ	011	reg (6 bits)	reg (6 bits)	label (12 bits)	001
BNE	011	reg (6 bits)	reg (6 bits)	label (12 bits)	010
BLO	011	reg (6 bits)	reg (6 bits)	label (12 bits)	011
BGT	011	reg (6 bits)	reg (6 bits)	label (12 bits)	100
LW	100	reg (6 bits)	reg (6 bits)	cst (12 bits)	000
SW	101	reg (6 bits)	reg (6 bits)	cst (12 bits)	000

Exercice 2 :

Le programme stocke dans la mémoire à l'adresse 0 le résultat de la comparaison du contenu de \$1 et \$2. C'est à dire que si $\$1 == \2 alors le bit stocké en mémoire est 1 correspondant à True, sinon le bit stocké est 0 correspondant à False. Ici $\$1 = 0$ et $\$2 = 42$ donc le résultat stocké est False (0).