

TME6 -- Modèle VHDL de l'étage DECOD (partie 1)



Contents

Objectifs	1
Exercices	1
Exercice 1 -- Modèle de REG	1
Question 1	1
Question 2	2

Objectifs

- Faire un premier modèle comportemental de l'étage DECOD .
- Tester ce modèle.

Exercices

Exercice 1 -- Modèle de REG

Vous allez dans un premier temps d'écrire le modèle comportemental du banc de registre REG qui va constituer un des composants de DECOD .

Question 1

Écrire le modèle VHDL du bloc REG .

Le bloc REG contient les 16 registres (dont PC) et contient également les 4 *flags* : C (la retenue), Z (résultat nul), N (résultat négatif) et V (dépassement de capacité). À chaque registre (32 bits) est associé un bit de validité, un unique bit de validité est associé aux 3 *flags* C, N et Z, un autre bit de validité est associé à V. Au reset tous les registres sont considérés comme valides même si ils ne contiennent alors aucune valeur pertinente. Quand un registre est identifié par DECOD comme constituant la destination d'une instruction celui-ci est invalidé. Quand un résultat produit par EXEC ou MEM est écrit dans REG , le registre de destination est validé. Il ne peut y avoir écriture dans un registre que si celui-ci est marqué comme invalide.

Il en est de même pour les *flags*, les instructions logiques écrivent dans les 3 *flags* C, N et Z, seules les instructions arithmétiques affectent le flag V.

MEM et EXEC peuvent produire simultanément un résultat, en cas de conflit (même registre de destination), l'écriture provenant de MEM doit être ignorée car nécessairement plus ancienne.

Le registre numéro 15 (PC) fait l'objet d'un traitement particulier : Il lui est associé un opérateur réalisant l'opération +4 et son contenu ainsi que sa validité sont directement visibles à l'interface de REG .

Voici récapitulé les ports de REG :

- 3 ports de lecture numérotés de 1 à 3,
- 2 ports d'écriture, le numéro 1 correspond à EXEC et est donc prioritaire,

- 2 ports d'invalidation car une instruction peut produire 2 résultats,
- 4 *flags* et leurs 2 bits de validité,
- PC, son bit de validité et sa commande de +4.

Fichier fourni :

- Interface du bloc REG : `reg_vide.vhdl`.

Question 2

Écrire un fichier *test bench* permettant de valider votre modèle.