# Rapport du labo FETCH

## Utilité du FETCH

Le FETCH est un composant d’un CPU permettant de déterminer l’adresse de la prochaine instruction en fonction de l’adresse courante. L’adresse courante est stockée est dans le registre « Program Counter (PC) ». Celui-ci étant normalement dans une banque de registre, nous l’avons mis juste en dehors pour simuler un comportement.

## Incrémentation de l’instruction

La première étape a consisté à gérer l’incrémentation de +NbreOctetInstruction, dans notre cas de +2 à chaque adresse. Nous avons donc simplement ajouté un additionneur +2 à l’adresse à pc\_present. Pour tester, nous avons simulé un clock et avons vérifié si chaque flan montant incrémentait de deux le registre PC (par ex, 0000 -> 0002 -> 0004 …). Nous avons ensuite compilé un léger programme et l’avons importé dans la ROM pour vérifier que l’adresse était modifiée à chaque coup de clock.

## Mécanisme de saut

### Détermination de fetch\_control\_bus

Pour déterminer si l’instruction courante est un saut ou non (bit en première position), nous avons vérifié si l’instruction dans la mémoire commençait par la série de bit des sauts conditionnels ou inconditionnels et pareil pour la détermination du type du saut. Les autres données étaient simplement mises directement dans le bus.

## Inconditionnel

Pour tester les sauts inconditionnels, nous avons créé un simple programme qui contient 2 blocs de codes. À la fin du premier se trouve une instruction « B ADDR\_SAUT\_2 » qui est à une adresse totalement arbitraire. Le saut fonctionne.  
  
Conditionnel

Pour tester les sauts, nous avons mis le CPSR à 0b0001, ce qui signifie que Z = 1 et que donc les deux opérandes sont égales nous permettant de tester le BEQ. Ceci fonctionne.

Par exemple, le bloc de code suivant :

Une image contenant texte

Description générée automatiquement

À l’instruction 5, le saut à l’adresse 0x6 est correctement fait.