

TP 2

LOGISIM- Circuits combinatoires part. 2

Fichiers fournis : tp2_aluetu.circ, tp2_pgcdetu.circ

2.1 Quelques éléments du processeur LC-3

Dans la suite du cours, nous allons construire un petit processeur pédagogique, le LC-3. Nous prenons de l'avance dans ce TP en construisant quelques sous-circuits que nous assemblerons ensemble dans un prochain TP.

EXERCICE 1 ► ALU LC-3

Récupérez sur la page web du cours le fichier `tp2_aluetu.circ` et testez-le pour savoir ce qu'il fait. Remplissez les cases vides du tableau suivant avec des formules dépendant des entrées `Input1`, `Input2` et `Cst` :

$e_2/UseCst$	0	1
(00)		
(01)		
(10)		
(11)		

EXERCICE 2 ► NZP LC-3

Dans un nouvel onglet du fichier précédent (nommé NZP), créez un circuit qui prend une entrée 16 bits nommée `RES` considérée en complément à 2 sur 16 bits, et qui en sortie a un "Pin" 3 bits nommé NZP. Le bit de poids faible (`P`) est égal à 1 ssi $RES > 0$, le bit du milieu (`Z`) est égal à 1 ssi $RES = 0$, et le bit de poids fort est à 1 ssi $RES < 0$. Testez bien.

EXERCICE 3 ► Extensions de signe

D'après un des exercices de TD, l'extension de signe en complément à 2 se fait en dupliquant le bit de poids fort autant de fois que nécessaire. Créez dans un même fichier deux onglets différents :

- Dans un onglet appelé `Ext8vers16`, construisez un sous-circuit pour l'extension de signe d'un entier codé en complément à 2 sur 8 bits vers 16 bits. Testez le sous-circuit dans un onglet `Brouillon`.
- Comparez votre sous-circuit `Ext8vers16` avec le composant `BitExtender` de la librairie (dans `Wiring`).

2.2 Additionneurs

EXERCICE 4 ► Additionneurs à propagation simple de retenue

Dans un nouveau fichier :

- Réalisez l'additionneur 1 bit à retenue du cours et testez-le.
- Regardez dans la documentation comment fonctionne l'encapsulation (`Subcircuits`). Nommez l'additionneur 1 bit "FA1" et utilisez le pour réaliser un additionneur 4 bits.
- Utilisez l'additionneur 8 bits de la librairie (`Arithmetic->Adder`) avec des "constantes" (`Wiring->Constant`) en entrée de l'addition et un afficheur (`Probe`) 8 bits en sortie. On vérifiera que $(80)_{16} + (8C)_{16} = (00001100)_2$ (et 1 de retenue).
- En utilisant cet additionneur 8 bits (et les multiplexeurs de la librairie), réalisez un additionneur-soustracteur 8 bits, qui calcule $a - b$ ou $a + b$ suivant la valeur d'un bit de contrôle c . Vous n'avez pas le droit de dupliquer l'additionneur (vous pouvez vous reporter à l'exercice correspondant du cahier de TD).
- Réalisez un **ALU 8 bits** capable de faire une addition, une soustraction et un test d'égalité. L'opération sera choisie avec un signal qui vaut 00 pour une addition, 01 pour une soustraction et 10 pour un test d'égalité. On remarquera que c'est une modification mineure du circuit précédent.

EXERCICE 5 ► Additionneur à propagation rapide de retenue - sélection de retenue

L'inconvénient des additionneurs 8 bits en cascade est que chaque additionneur 1 bit doit attendre que sa retenue entrante soit disponible pour réaliser l'opération. Un additionneur 8 bits a donc un temps de traversée égal à 8 fois le temps de traversée d'un additionneur 1 bit. Un additionneur à sélection de retenue (*carry select*) peut être construit en utilisant le temps de traversée d'un additionneur 4 bits (utilisé pour additionner les 2×4 bits de poids faible) pour précalculer les deux résultats possibles de l'addition des 2×4 bits de poids forts (l'un avec une retenue entrante égale à 1, l'autre avec une retenue entrante nulle). Un multiplexeur est utilisé pour sélectionner le bon résultat lorsque la retenue entrante est finalement connue.

Réalisez un tel additionneur en utilisant l'additionneurs 4 bits de la question précédente.

EXERCICE 6 ► Additionneur à propagation rapide de retenue - anticipation de retenue

Nous avons vu précédemment l'additionneur à sélection de retenue. Ce circuit est rapide mais présente deux inconvénients :

- Le résultat d'un des deux additionneurs des bits de poids forts n'est pas utilisé,
- Deux multiplexeurs sont nécessaires pour le résultat de l'addition des bits de poids forts et pour la retenue sortante.

La surface utilisée est donc plus grande, d'où une plus grande consommation électrique et plus de chaleur produite.

Dans cet exercice nous allons réaliser un additionneur 8 bits à anticipation de retenue.

En vous inspirant du cours,

- Reprenez un additionneur complet 1 bit de l'exercice sur l'additionneur à propagation simple de retenue et ajoutez les sorties de génération et de propagation de retenue;
- Créez un additionneur 4 bits avec circuit d'anticipation de retenue;
- Créez un additionneur 8 bits couplant deux additionneurs 4 bits avec circuits d'anticipation de retenue, la retenue entrante de l'additionneur des bits de poids forts étant la sortie du circuit anticipateur de retenue de l'additionneur des bits de poids faibles;
- Comparez les temps de passage avec l'additionneur 8 bits à sélection de retenue.