# Mémento pour la Conception d'un Processeur 8 bits avec Logisim-evolution

## Objectif
Ce document fournit un guide pratique pour que les étudiants conçoivent un processeur 8 bits simple en utilisant Logisim-evolution. Le but est de permettre une compréhension des principes de base de la conception des processeurs.

## Pré-requis
- Logisim-evolution installé sur vos ordinateurs.

- Une compréhension de base des circuits logiques (portes AND, OR, NOT, etc.).

- Connaissance de la structure des processeurs (ALU, registre, unité de contrôle, etc.).

## Architecture du Processeur 8 bits

## 1. Unité Arithmétique et Logique (UAL)
- Une UAL capable de réaliser les opérations suivantes :

  - Addition

  - Soustraction

  - ET logique (AND)

  - OU logique (OR)

  - Décalage à gauche (SHL)

  - Décalage à droite (SHR)

## 2. Registres
- Deux registres principaux (A et B) de 8 bits chacun.

- Un registre de sortie (OUT) de 8 bits.

## 3. Unité de Contrôle
- Une unité de contrôle avec un décodeur pour interpréter les instructions.

- Un ensemble d’instructions de base (chargement, stockage, addition, soustraction, etc.).

## 4. Mémoire
- Une mémoire de 256 octets (8 bits par case mémoire).

### 5. Bus de Données
- Un bus de données de 8 bits reliant les différentes unités.

## Guide de Construction Pas-à-Pas

### 1. Création des Registres
- Utilisez les bascules D (flip-flops D) pour créer les registres A, B et OUT.

### 2. Conception de l'UAL
- Construisez les portes logiques nécessaires pour chaque opération.

- Utilisez un multiplexeur pour sélectionner l'opération active.

### 3. Mise en Place de l'Unité de Contrôle
- Ajoutez un décodeur pour interpréter les instructions.

- Connectez les signaux de contrôle aux registres et à l'UAL.

### 4. Ajout de la Mémoire
- Utilisez la mémoire RAM de Logisim-evolution et configurez-la pour 256 octets.

### 5. Connexion du Bus de Données
- Reliez les registres, l'UAL et la mémoire à travers un bus de données de 8 bits.

### Instructions Supportées
- LDA (Load A) : Charger une valeur dans le registre A.

- STA (Store A) : Stocker la valeur de A en mémoire.

- ADD : Ajouter la valeur de B à A.

- SUB : Soustraire la valeur de B de A.

- AND : Effectuer un ET logique entre A et B.

- OR : Effectuer un OU logique entre A et B.

- JMP : Sauter à une adresse spécifique.

### Exemple de Programme
LDA 0x10

ADD 0x11

STA 0x12

## Conseils

- Concevoir chaque composant séparément avant de les connecter.

- Faites-les tester chaque unité (UAL, registres, mémoire) séparément avant l’intégration finale.

## Extensions possibles

- Ajouter de nouvelles instructions (MUL, DIV).

- Augmenter la taille des registres pour un processeur 16 bits.

- Implémenter une unité de gestion d'interruptions.