

# Documentation Projet Processeur

## Pour l'utilisation :

Syntaxe du Code Assembleur :

UAL :

INSTRUCTION	SYNTAXE
ADD, SUB, AND, OR, XOR, SL, SR, MUL	OP R <sub>dest</sub> R <sub>src1</sub> R <sub>src2</sub>
ADDI, SUBI, ANDI, ORI, XORI, SLI, SRI, MULI	OP R <sub>dest</sub> R <sub>src1</sub> IM

À noter : SL et SR passent le résultat à 0 si la valeur dans R<sub>src2</sub> est supérieur à 2<sup>3</sup>

MEM,

Contrôle

INSTRUCTION	SYNTAXE
STR, LD	OP Reg SP Où SP est l'adresse Mémoire et Reg le registre utilisé

INSTRUCTION	SYNTAXE
JMP, CALL	OP label
JEQU, JNEQ, JSUP, JINF	OP R <sub>src1</sub> R <sub>src2</sub> label
RET	OP 🙌

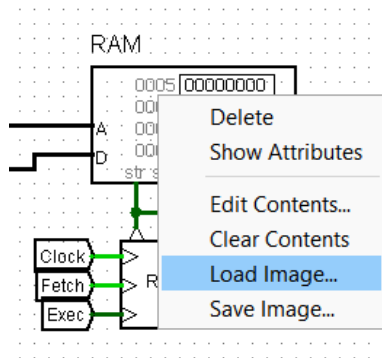
Pour déclarer un **label**, On fait un peu comme on veut (les espaces et les tabulations sont ignorés :

Label : INSTRUCTION
Label :   INSTRUCTION
Label   :
INSTRUCTION

Les **commentaires** : caractère d'échappement : '#'

Les commentaires peuvent se placer en fin de ligne ou seul sur une ligne

## Paramétrage du Compilateur (rapidement)



Il y a deux paramètres à entrer : le lien vers le fichier source (avec le code assembleur) , et le chemin relatif vers le fichier destination, respectivement test.txt et hexa.txt par défaut.

C'est hexa.txt qui doit être chargé dans la RAM du processeur Logisim (click droit puis Load Image comme montré ci-dessous)

## Fonctionnement du Compilateur

En deux mots :

Le compilateur parcourt l'ensemble du fichier d'entrée et dans un premier temps le 'nettoie' : Pour chaque ligne, il supprime les commentaires, supprime les caractères inutiles (N espaces ou Tabulation devient 1 espace), les lignes sans instructions, enregistre l'ensemble des labels déclarés avec le numéro de l'instruction à laquelle chacun mène.

Ensuite, il génère la trame binaire de chaque ligne, qu'il convertit en Hexadécimal pour que le fichier puisse être lu par Logisim.

## Trames IR

Celles-ci ont été réfléchies pour limiter la complexité du compilateur (on utilise jamais les bits 12-14 pour dire l'adresse d'un registre dans lequel on veut écrire, et l'adresse mémoire est toujours sur les bits 9-11. Pour plus de détails et de réflexions, se référer au document : [trad Assembleur.xlsx](#) , qui donne des informations (bien présentées) vis-à-vis des trames (entre autre).