

1. Somme des n premiers entiers

Charger le circuit `micro1.circ`. En utilisant le circuit `micro1.circ`, écrire un programme qui charge dans le registre A une valeur n de votre choix puis affiche dans le registre B la somme des n premiers entiers $1+2+3+\dots+n$ en utilisant la formule $n*(n+1)/2$. Vous pouvez définir les instructions qu'il vous faut. Le programme nécessite 5 instructions.

Après avoir écrit le programme en assembleur, trouvez les codes hexa des instructions et rentrez les dans la mémoire programme du circuit puis testez votre programme.

Instruction	Code Héxa	Explication

2. Permutation avec seulement 2 registres

Charger le circuit `micro1.circ`. En utilisant le circuit `micro1.circ`, écrire un programme qui charge dans le registre A une valeur de votre choix et dans le registre B une autre valeur de votre choix puis permute ces deux valeurs. Vous pouvez définir les instructions qu'il vous faut. Le programme nécessite 5 instructions.

Après avoir écrit le programme en assembleur, trouvez les codes hexa des instructions et rentrez les dans la mémoire programme du circuit puis testez votre programme.

Instruction	Code Héxa	Explication

3. Puissance

Charger le circuit **micro4.circ**. Les étiquettes SAUTS et ALU sont juste des sortes raccourcis (appelés Tunnels dans Logisim, voir dossier Wiring) qui permettent d'alléger le schéma du circuit. Ces étiquettes relient les fils correspondant sans que cela apparaisse dans le circuit. Trouvez les codes héxa des instructions suivantes :

Instruction	Code Héxa	Explication
NOP		Ne fait rien
LOAD_A #valeur		$A \leftarrow \text{valeur}$
LOAD_B #valeur		$B \leftarrow \text{valeur}$
LOAD_C #valeur		$C \leftarrow \text{valeur}$
DEC_B		$B \leftarrow B - 1$
MUL_C_A		$C \leftarrow C * A$
CMP_B #valeur		Compare B et valeur et met à jour les indicateurs de l'UAL : si égalité Z passe à 1. Si $B < \text{valeur}$ N passe à 1.
JMP <label>		Saut incondtionnel à l'adresse indiquée par label
JMPZ <label>		Saut à l'adresse indiquée par label si le résultat de l'instruction précédente est nul (indicateur Z=1).

En utilisant le circuit **micro4.circ**, écrire un programme qui charge dans le registre A une valeur **x** de votre choix et dans le registre B une autre valeur **y** de votre choix puis calcule dans le registre C la valeur x^y (x puissance y). Vous utiliserez uniquement les instructions définies ci-dessus. Le programme nécessite 9 instructions. Après avoir écrit le programme en assembleur, traduire les instructions en langage machine et entrez les codes dans la mémoire programme du circuit puis testez votre programme.

Adresse	Instruction	Code Héxa	Explication
0000			
0001			
0002			
0003			
0004			
0005			
0006			
0007			
0008			

4. Suite de Fibonacci étendue

On considère la suite 0, 1, 2, 3, 6, 11, 20, 37, 68, 125, 230, 423, ... dans laquelle chaque élément est obtenu en additionnant les **3 éléments** précédents.

En utilisant le circuit micro3.circ, écrire un programme qui charge dans le registre A la valeur 0, dans le registre B la valeur 1 et dans le registre C la valeur **2** puis calcule les éléments de la suite ci-dessus à l'infini. Le programme nécessite 12 instructions. Après avoir écrit le programme en assembleur, traduire les instructions en langage machine et entrez les codes dans la mémoire programme du circuit puis testez votre programme.

Indice : pensez à utiliser la solution de l'exercice 2 (permutation à l'aide de 2 registres seulement).

Adresse	Instruction	Code Héxa	Explication
0000			
0001			
0002			
0003			
0004			
0005			
0006			
0007			
0008			
0009			
000A			
000B			