

MODÈLE D'ARCHITECTURE SÉQUENTIELLE - MODÈLE DE VON NEUMANN - EXERCICES

Exercice 1

- Déterminer : le symbole (américain), la table de vérité et la notation en algèbre de Boole de la porte **NON ET** ou **NAND** (qui est le contraire de la porte ET).
- Déterminer : le symbole (américain), la table de vérité et la notation en algèbre de Boole de la porte **NON OU** ou **NOR** (qui est le contraire de la porte OU).
- Déterminer : le symbole (américain), la table de vérité et la notation en algèbre de Boole de la porte **NON OU EXCLUSIF** ou **XNOR** (qui est le contraire de la porte XOR).
- Déterminer la table de vérité de l'expression suivante : $\text{NOT}((A \text{ AND } B) \text{ XOR } A)$.
- Vérifier les résultats précédents avec le calculateur accessible à l'adresse suivante : <https://sheabunge.github.io/boolcalc/>

Exercice 2 [se référer à la dernière page du cours sur l'assembleur ARM]

Expliquez les instructions suivantes :

- | | |
|--------------------|----------------|
| 1) ADD R0, R1, #42 | 4) CMP R4, #18 |
| 2) LDR R5, 98 | BGT 77 |
| 3) STR R0, 15 | 5) B 100 |

Exercice 3 [se référer à la dernière page du cours sur l'assembleur ARM]

Écrire les instructions en assembleur correspondant aux phrases suivantes :

- Additionne la valeur stockée dans le registre R0 et la valeur stockée dans le registre R1, le résultat est stocké dans le registre R5
- Place la valeur stockée à l'adresse mémoire 878 dans le registre R0
- Place le contenu du registre R0 en mémoire vive à l'adresse 124
- Si la valeur stockée dans le registre R0 est égale 42 alors la prochaine instruction à exécuter se situe à l'adresse mémoire 85, sinon la prochaine instruction à exécuter se situe en mémoire vive à l'adresse 478

Exercice 4 [se référer à la dernière page du cours sur l'assembleur ARM]

Des entiers sont stockés aux adresses 7,8,9,10 dans la mémoire vive. Écrire une suite d'instructions en assembleur qui additionne le contenu de ces mémoires et le stocke à l'adresse 11.

Pour cela, vous n'avez le droit d'utiliser que deux registres : R0 et R1

Exercice 5 [se référer à la dernière page du cours sur l'assembleur ARM]

Les cases mémoires 11 12 13 de la mémoire vive contiennent des entiers. Que fait le programme suivant :

100	102	104	106	108	110	112	114	116
LDR R0,11	CMP R0,#2	BEQ 112	LDR R2,13	STR R2,30	B 116	LDR R2,12	STR R2,30	HALT

Exercice 6

Voici un programme Python, et son équivalent en assembleur.

Après avoir analysé très attentivement le programme en assembleur ci-contre, établir une correspondance entre les lignes du programme en Python et les lignes du programme en assembleur.

À quoi sert la ligne "B endif" ?

À quoi correspondent les adresses mémoires 23, 75 et 30 ?

```

MOV R0, #4
STR R0, 30
MOV R0, #8
STR R0, 75
LDR R0, 30
CMP R0, #10
BNE else
MOV R0, #9
STR R0, 75
B endif
else:
LDR R0, 30
ADD R0, R0,
#1
STR R0, 30
endif:
MOV R0, #6
STR R0, 23
HALT

```

Exercice 7

Faire les exercices de cette page :

https://pixees.fr/informatiquelycee/n_site/nsi_prem_sim_cpu.html