

---

# Architecture des ordinateurs

# TD n° 1

---

## *Circuits combinatoires élémentaires*

---

### **Exercice 1. Installation de Digital (Digital)**

#### **Q- 1.1 Installation du logiciel**

Télécharger et Installer Digital : <https://github.com/hneemann/Digital>

Remarque : il sera judicieux d'installer Digital qu'une seule fois en le plaçant quelque part dans votre dossier Mon\_dossier\_reseau\_UPVD.

### **Exercice 2. Circuits élémentaires de sélection**

Dans cet exercice, nous utiliseront uniquement des portes logiques à 2 entrées.

#### **Q- 2.1 Décodeur 2->4**

Réasiler puis tester un décodeur 2 vers 4.

#### **Q- 2.2 Décodeur 3->8**

Réasiler puis tester un décodeur 3 vers 8.

#### **Q- 2.3 MUX à 2 entrées**

Réasiler puis tester un multiplexeur à 2.

#### **Q- 2.4 MUX à 8 entrées récursif**

Réasiler puis tester un multiplexeur à 8 récursif (à l'aide des multiplexeurs à 2 entrées).

#### **Q- 2.5 MUX à 8 entrées non-récursif**

Réasiler puis tester un multiplexeur à 8 non-récursif.

#### **Q- 2.6 Analyse MUX**

Comparer les 2 multiplexeurs à 8 entrées précédemment réaliser (nombre de portes logiques et chemin critique).

### **Exercice 3. Circuits élémentaires arithmétiques**

Dans cet exercice, nous utiliseront uniquement des portes logiques à 2 entrées.

#### **Q- 3.1 ADD 1 bit**

Réaliser puis tester un additionneur complet 1 bit.

#### **Q- 3.2 ADD 8 bits**

Réaliser puis tester un additionneur complet 8 bits à l'aide de l'additionneur complet 1 bit.

**Q- 3.3 ADDPG 1 bit**

Réaliser puis tester un additionneur avec indication de propagation (P) et de génération (G) de retenue.

**Q- 3.4 CC 8 bits**

Réaliser puis tester un circuit de calcul des retenues 8 bits. Le circuit prend en entrées les  $P[7..0]$ ,  $G[7..0]$ , la retenue initiale  $C_0$  et renvoie les 9 retenues  $C[8..0]$ .

**Q- 3.5 ADDPG 8bits**

En réutilisant, les deux circuits précédents réaliser un additionneur 8 bits avec calcul des retenues anticipé.

**Q- 3.6 Analyse**

Comparer les deux additionneurs 8 bits (nombres de portes et chemin critiques).

**Exercice 4. Circuits élémentaires supplémentaires**

Dans cet exercice, nous utiliseront uniquement des portes logiques à 2 entrées.

**Q- 4.1 Un “barrel shifter” 8 bits**

Un “barrel shifter” est un circuit à décalage. Il prends en entrées une valeur codé sur 8 bits  $e[7..0]$  et un nombre de décalage codé sur 3 bits  $d[2..0]$ . En sortie le circuit donne la valeur d’entrée décalée de  $d$  bits à gauche ( $s[7..0]$ ).  $d$  étant la valeur codé sur les bits  $d[2..0]$ .

Construire puis tester le circuit d’un “barrel shifter” 8 bits.

**Exercice 5. Unité Arithmétique et Logique****Q- 5.1 Circuit logique XOR binaire 8 bits**

Réaliser puis tester le circuit XOR binaire 8 bits qui prends en entrées deux entrées  $a$  et  $b$  sur 8 bits et place sur la sortie  $s$  8 bits le *xor* binaire de  $a$  et de  $b$ .

**Q- 5.2 Circuit logique OR binaire 8 bits**

Réaliser puis tester le circuit OR binaire 8 bits qui prends en entrées deux entrées  $a$  et  $b$  sur 8 bits et place sur la sortie  $s$  8 bits le *or* binaire de  $a$  et de  $b$ .

**Q- 5.3 Circuit logique AND binaire 8 bits**

Réaliser puis tester le circuit AND binaire 8 bits qui prends en entrées deux entrées  $a$  et  $b$  sur 8 bits et place sur la sortie  $s$  8 bits le *and* binaire de  $a$  et de  $b$ .

**Q- 5.4 Unité Arithmétique et Logique**

Réaliser une Unité Arithmétique et Logique 8 bits capable de produire les résultats de 5 opérations : ADD, SUB, XOR, OR et AND.