

# Architecture des systèmes numériques – Assembleur AQA

## Plan de cours

Halim Djerroud

révision 1.0

### Introduction

Le module **Architecture des systèmes numériques – Assembleur AQA** est organisé en **7 séances**, réparties en **5 cours magistraux/TD (2h)** et **2 travaux pratiques (3h)**. L'objectif est de donner aux étudiants une compréhension progressive du fonctionnement interne d'un ordinateur : du **codage de l'information**, aux **circuits logiques**, jusqu'à l'**exécution de programmes en assembleur AQA**.

L'évaluation pourra comporter :

- un **contrôle continu** (questions rapides en cours/TD),
- un **TP noté**,
- un **devoir écrit (DE)** de synthèse sur les notions vues.

### Objectifs

- Comprendre la représentation de l'information (nombres, caractères).
- Concevoir et analyser des circuits logiques simples.
- Identifier la structure d'un processeur (UC, UAL, registres, mémoire).
- Comprendre le cycle d'instruction *fetch-decode-execute*.
- Écrire et exécuter des programmes en assembleur AQA.
- Relier le code assembleur à des notions de haut niveau (boucles, conditions).

### Répartition du volume horaire

- **5 cours/TD de 2h**
- **2 travaux pratiques de 3h**
- Total : **20h**

### Bibliographie indicative

- C. Hamacher, Z. Vranesic, S. Zaky, *Computer Organization and Embedded Systems*, McGraw-Hill.
- Documentation AQA : [https://qkzk.xyz/docs/nsi/cours\\_premiere/architecture/5\\_aqa/6\\_assembleur\\_aqa\\_intro/](https://qkzk.xyz/docs/nsi/cours_premiere/architecture/5_aqa/6_assembleur_aqa_intro/) , <https://www.peterhigginson.co.uk/AQA/>
- Documentation Logisim : <http://www.cburch.com/logisim/>.

## Cours 1 : Représentation de l'information

### — Cours 2h / TD 2h

1. Numération binaire et hexadécimale.
2. Codage des entiers (non signé, complément à deux).
3. Codage des caractères (ASCII).
4. Exercices de conversion et d'opérations arithmétiques.

## Cours 2 : Logique combinatoire

### — Cours 2h

1. Portes logiques, tables de vérité.
2. Circuits combinatoires : additionneur, multiplexeur.
3. Simplification logique (Karnaugh).
4. Introduction à Logisim.

## TP1 : Circuits logiques et ALU (Logisim)

### — TP 3h

1. Prise en main de Logisim.
2. Réalisation d'un additionneur 4 bits.
3. Extension en mini-ALU (ADD, SUB, AND, OR).

## Cours 3 : Logique séquentielle et registres

### — Cours 2h

1. Bascules et registres.
2. Notion d'horloge et de mémoire.
3. Exemple : registre accumulateur.

## Cours 4 : Structure d'un processeur et cycle d'instruction

### — Cours 2h

1. Architecture de von Neumann.
2. Cycle *fetch* – *decode* – *execute*.
3. Jeu d'instructions AQA (LDA, STA, ADD, SUB, BRA, BRZ, BRP, INP, OUT, HLT).

## TP2 : Premiers programmes en assembleur AQA

### — TP 3h

1. Prise en main du simulateur AQA.
2. Écriture de programmes simples : somme, boucle de comptage, conditions.
3. Comparaison avec un pseudo-code équivalent (C).