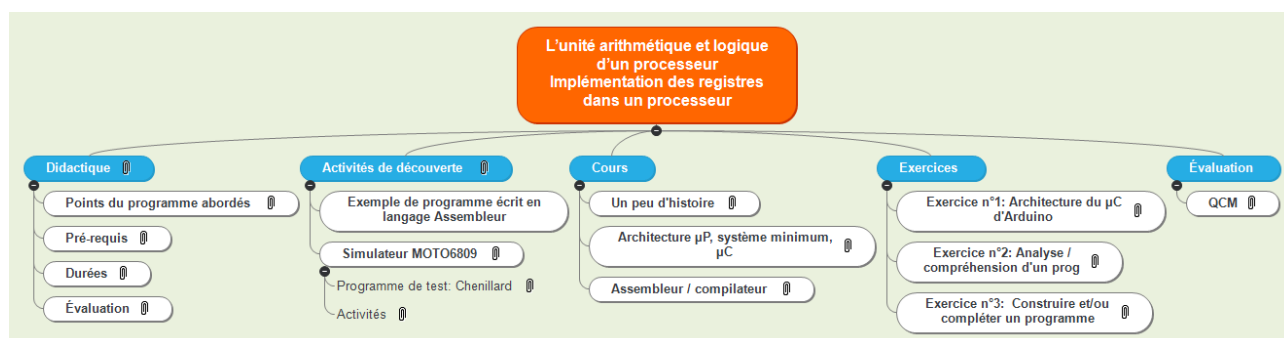


# L'unité arithmétique et logique d'un processeur

## Implémentation des registres dans un processeur



<b>Didactique</b> .....	<b>2</b>
Points du programme abordés .....	2
Pré-requis.....	2
Durées.....	2
Évaluation.....	2
<b>Activités de découverte</b> .....	<b>2</b>
Exemple de programme écrit en langage Assembleur .....	2
Simulateur MOTO6809 .....	2
Programme de test: Chenillard.....	2
Activités .....	3
<b>Cours</b> .....	<b>3</b>
Un peu d'histoire.....	3
Architecture µP, système minimum, µC.....	3
Assembleur / compilateur.....	3
<b>Exercices</b> .....	<b>3</b>
Exercice n°1: Architecture du µC d'Arduino .....	3
Exercice n°2: Analyse / compréhension d'un prog .....	3
Exercice n°3: Construire et/ou compléter un programme .....	3
<b>Évaluation</b> .....	<b>4</b>
QCM.....	4

## 1. Didactique

[objectifs\\_et\\_referentiel.pdf](#)

### 1.1. Points du programme abordés

#### 1. Histoire de l'informatique :

- Situer dans le temps les principaux événements de l'histoire de l'informatique et leurs protagonistes.

#### 2. Architectures matérielles et systèmes d'exploitation :

- Distinguer les rôles et les caractéristiques des différents constituants d'une machine.

- Dérouler l'exécution d'une séquence d'instructions simples du type langage machine.

#### 3. Langages et programmation :

- Repérer, dans un nouveau langage de programmation les traits communs et les traits particuliers à ce langage.

### 1.2. Pré-requis

✓ **Représentation des données:** Écriture d'un entier positif dans une base  $b \geq 2$

◆ Passer de la représentation d'une base dans une autre.

◆

✓ **Traitement de données en tables :** Tableau indexé, tableau donné en compréhension

◆ Lire et modifier les éléments d'un tableau grâce à leurs index.

◆ Itérer sur les éléments d'un tableau

### 1.3. Durées

Activités de découverte: 30mn

Cours: 1h

Exercices: 1h45

Synthèse: 15mn

Evaluation: 30mn

### 1.4. Évaluation

À voir... QCM ?

## 2. Activités de découverte

[Activité de découverte - Elève.pdf](#)

[Activité de découverte - Profs.pdf](#)

### 2.1. Exemple de programme écrit en langage Assembleur

### 2.2. Simulateur MOTO6809

Programme de test: Chenillard (exemple fourni)

[MOTO6809.exe](#)

#### 2.2.1. Programme de test: Chenillard

[chenillard.as9](#)

### 2.2.2. Activités

- ✓ Exécuter le programme et observer son déroulement (changer la vitesse d'exécution)
- ✓ Modifier une valeur et observer son effet
- ✓ Dans quelle mémoire se trouve le programme ?
- ✓ Justifier la taille du programme
- ✓ Calculer la taille de la mémoire RAM
- ✓ ....
- ✓

## 3. Cours

### 3.1. Un peu d'histoire

A faire...

Un peu d'histoire sur les microprocesseurs, ...

### 3.2. Architecture $\mu$ P, système minimum, $\mu$ C

[cours.pdf](#)

### 3.3. Assembleur / compilateur

- Complément de cours ?
  - Rôle de l'assembleur, mécanismes, fichier S19
  - Assembleur vs compilateur

## 4. Exercices

### 4.1. Exercice n°1: Architecture du $\mu$ C d'Arduino

- Reconnaître les différents éléments d'un système minimum à  $\mu$ P dans un  $\mu$ C
- Synthèse: différence entre  $\mu$ P et  $\mu$ C
- Comparaison  $\mu$ P Arduino,  $\mu$ P Intel

[Exercice1 - Comparaison structures  \$\mu\$ P et  \$\mu\$ C.pdf](#)

### 4.2. Exercice n°2: Analyse / compréhension d'un prog

Dans le simulateur MOTO6809:

- ✓ Analyse / compréhension d'un programme donné (recherche du nombre maximum d'une table)
- ✓ Que fait le programme ?
- ✓ Compilation du programme: étapes de compilation, fichier S19

Pour aller plus loin:

- ✓ À partir du jeu d'instructions, calculer le nombre de cycles du prog. analysé
- ✓ Connaissant la fréquence d'horloge du processeur (ou temps d'un cycle), déduire le temps d'exécution du programme.
- ✓ Calculer le temps d'exécution du programme

[exercice\\_1.as9](#)

[Exercice2 - Assembleur.pdf](#)

### 4.3. Exercice n°3: Construire et/ou compléter un programme

A l'aide d'une liste d'instructions données, élaborer un programme en assembleur 6809 et le tester dans le simulateur

## 5. Évaluation

### 5.1. QCM

A faire...