

L'unité de contrôle d'un processeur lit des instructions contenues dans la mémoire et ordonne à l'unité arithmétique et logique de les exécuter.

Comment l'Homme communique avec la machine ?

1 Langage machine

À retenir

Un processeur est un composant électronique : il n'interprète que des signaux électriques.

Un processeur ne peut exécuter que des instructions basiques :

- **opérations arithmétiques** : « additionne la valeur contenue dans le registre R1 et le nombre 789 et range le résultat dans le registre R0 »
- **transfert de données entre les registres et la mémoire vive** : « prendre la valeur située à l'adresse mémoire 487 et la placer dans la registre R2 »
- **rupture de séquence** : « saute de l'instruction 2 à l'instruction 5 »

2 Langage assembleur

2.1 Un langage intermédiaire

À retenir

Pour faciliter la vie des informaticiens, on remplace les code binaires par des **symboles mnémoniques**.

ADD, MOV, SUB...

2.2 Découverte d'un simulateur

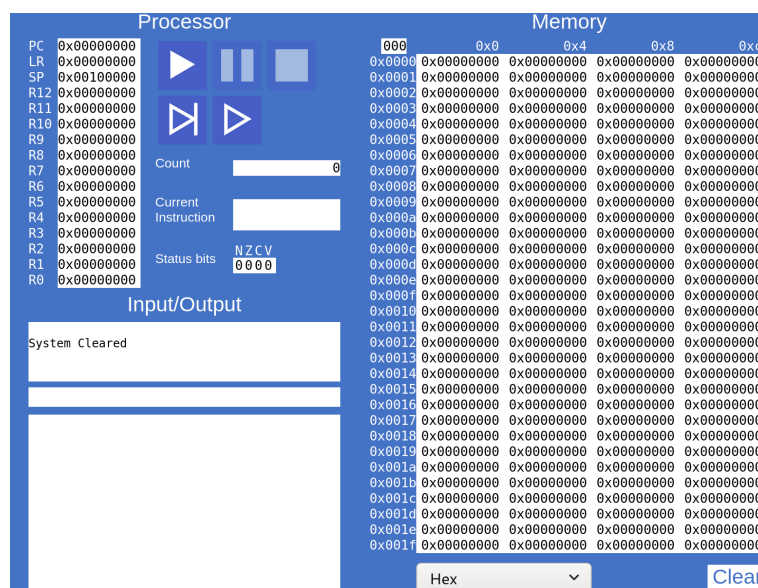


FIGURE 1 – Simulateur 32-bit ARM

2.3 Opérations arithmétiques

```
1 ADD R0,R1,R2
```

Code 1 – Ajoute la valeur du registre R2 à celle de R1 puis place le résultat dans R0.

```
1 ADD R1,R1,#10
```

Code 2 – Ajoute l'entier 10 à celle du registre R1 puis place le résultat dans R1.

2.4 Transfert de données

```
1 MOV R0, #10
```

Code 3 – Place la valeur 10 dans R0.

```
1 LDR R0,16
```

Code 4 – **LoaDR**egister : charge dans R0 la valeur située à l'**adresse** 16 de la mémoire.

```
1 STR R0,20
```

Code 5 – **SToreR**egister : stocke la valeur de R0 dans l'espace mémoire situé à l'**adresse** 20.

2.5 Rupture de séquence

```
1 MOV R0, #10
2 MOV R1, #10
3 // Compare les valeurs de R0 et R1
4 CMP R0, R1
5 // Si les valeurs sont égales, saute au label
6 BEQ labelegal
7 MOV R2, R0
8 HALT
9 labelegal:
10 STR R0, mavaleur
11 HALT
12 mavaleur: 5
```

Code 6 – Les // permettent de commenter le code.