

Architecture des microcontrôleurs

Introduction

Sommaire

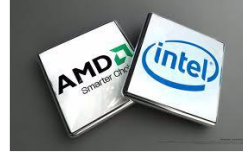
1. Principe des microcontrôleurs
2. Domaines d'application des microcontrôleurs
3. Principe des mémoires
4. Architectures VON NEUMANN et HARVARD
5. Architectures RISC et CISC
6. Anatomie d'un microcontrôleur typique

1. Principe des microcontrôleurs

- Un microcontrôleur (ou un microprocesseur) effectue **séquentiellement** des opérations indiquées dans un **programme**.
- Ces opérations sont appliquées sur des “mots” codés en **binaire**.
- Elles peuvent être de type **logique** (AND, OR, SHIFT, etc.) ou **arithmétique** (ADD, SUB, MULTIPLY, etc.).
- Elles sont exécutées les unes après les autres à **cadence** “élevée”.
- Des **périphériques** d’entrées / sorties permettent de **communiquer** avec d’autres composants (capteurs, afficheurs, etc.) et d’**interagir** avec le monde extérieur (lecture de clavier, de boutons, commande de moteurs, etc.)

Principaux composants

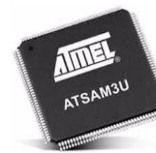
- Processeur (exécute les opérations)



- Mémoire (stocke les programmes et les données)



- Microcontrôleur (processeur + mémoires + périphériques d'entrées/sorties)



- Processeurs dédiés (Digital Signal Processor, etc.)



Taille des données

8 bits

16 bits

32 bits

64 bits

128 bits



Fréquence de fonctionnement

kHz

MHz

GHz



2. Domaines d'application

- Objets connectés



- Transports



- Multimédia



- Loisirs



- Médical

- Grand public



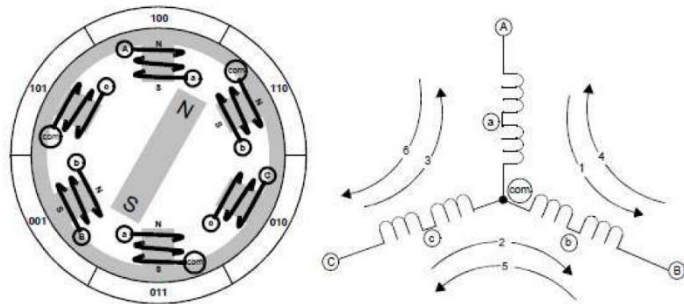
- Etc.



Un exemple: vélo électrique



moteur brushless (sans balais)

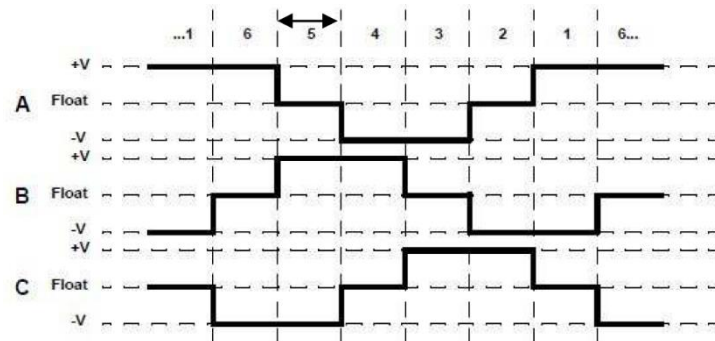


contrôleur moteur:

- récupère l'information des 3 capteurs de position moteur (capteurs à effet Hall)
- génère les signaux de commande des 3 enroulements
- contrôle l'intensité dans les enroulements afin de régler la vitesse de rotation (Pulse Width Modulation)



1 microcontrôleur





afficheur LCD
+
boutons



interface utilisateur: (Human Machine Interface)

- collecte l'information des boutons (marche/arrêt, niveau d'assistance, etc.)
- commande l'afficheur LCD (niveau d'assistance, tension batterie, vitesse, distance, courant consommé, etc.)
- dialogue avec le contrôleur moteur



1 microcontrôleur

3. Principe des mémoires

- Les mémoires sont indissociables des processeurs.

variables → mémoire “volatile” (RAM)

programmes → mémoire “non volatile” (mémoire Flash, disque dur, etc.)

- Une mémoire est un ensemble de **cellules** (ou **registres**).

Chaque cellule contient une (et une seule) donnée (**data**), généralement un octet (byte).

Chaque cellule est identifiée par un numéro unique: l'adresse (**address**).

adresses

données

0 (0x000)

12

1 (0x001)

2

2 (0x002)

128

3 (0x003)

99

255 (0xFF)

55

4095 (0xFFF)

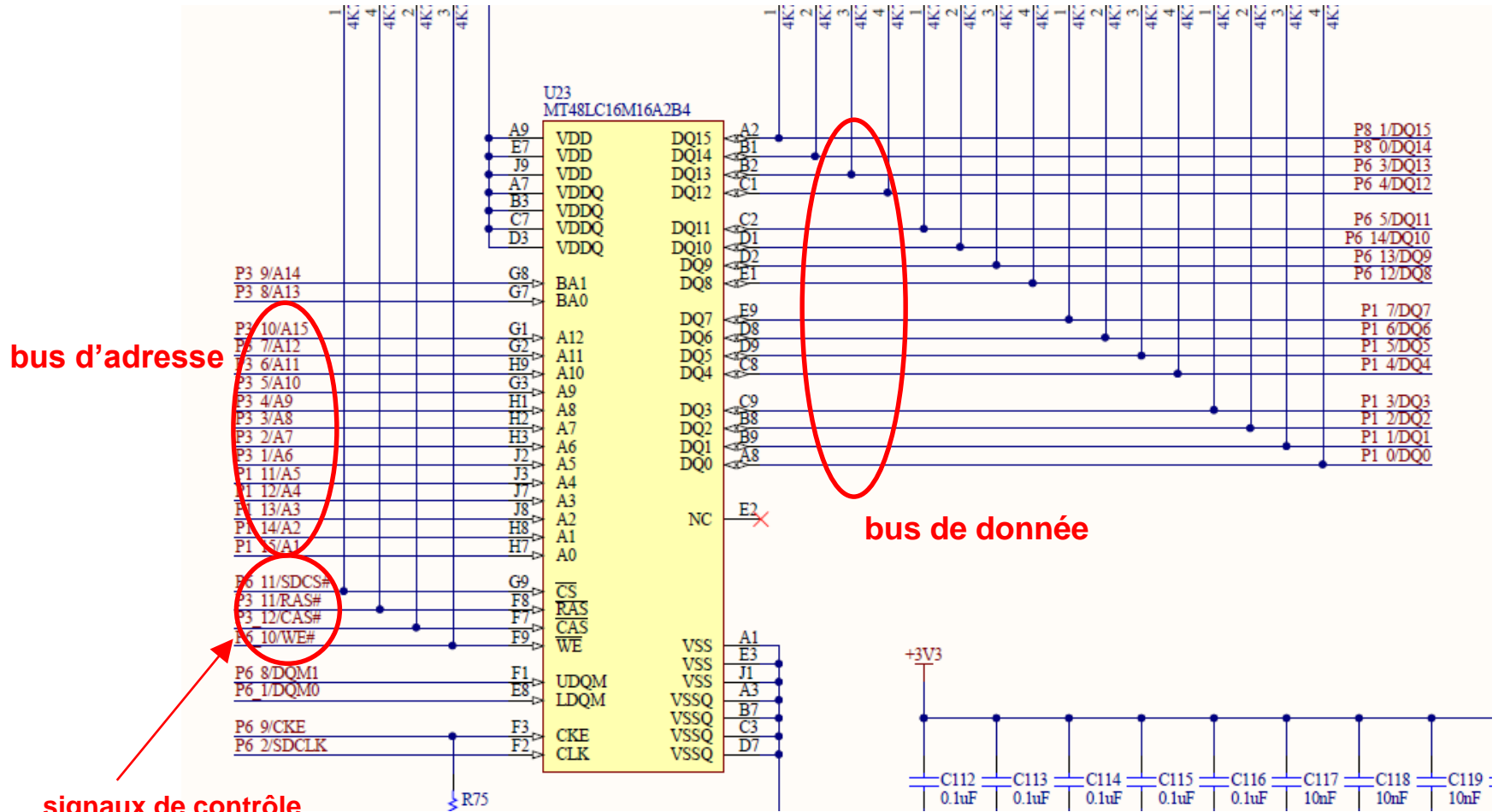
201

valeurs
quelconques

valeur maximale dépendante

- du nombre de bits utilisés pour coder l'adresse
- de la capacité physique de la mémoire

Exemple typique de mémoire:



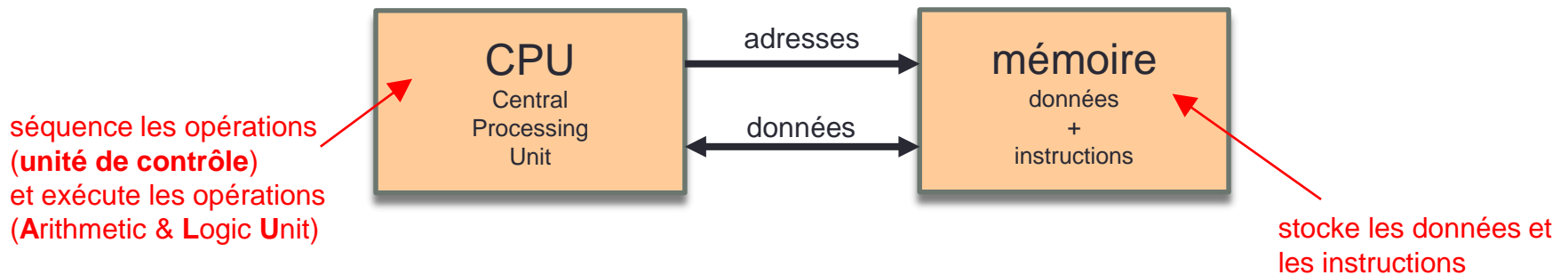
signaux de contrôle

- sélection du boîtier,
- lecture ou écriture, ...

4. Architectures VON NEUMANN et HARVARD

VON NEUMANN:

1 seule mémoire pour stocker les instructions et les données



Matériel simple

Données et instructions mélangées

Accès multiples en mémoire impossibles, pas de parallélisme

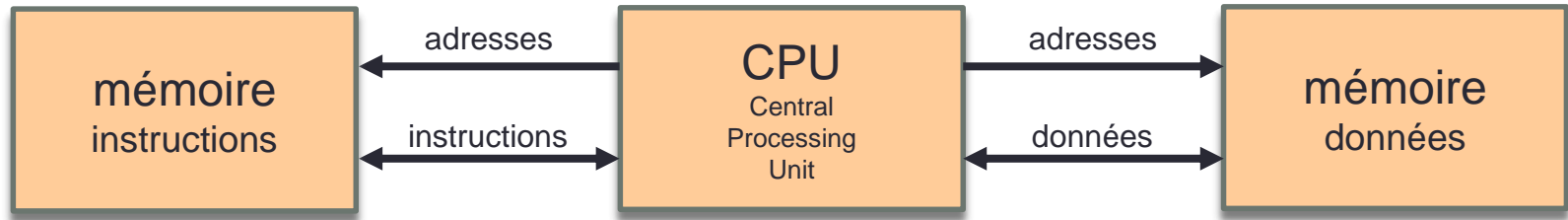
(temps d'exécution d'une instruction: plusieurs cycles d'horloge)

Structure obsolète

(John VON NEUMANN, 1903 – 1957)

HARVARD

Plusieurs mémoires, bus séparés pour les instructions et les données



Matériel plus complexe

Données et instructions séparées

Accès multiples en mémoire possibles, parallélisme possible

(temps d'exécution d'une instruction: possible en 1 cycle d'horloge)

Architecture typique des systèmes actuels

5. Architectures RISC et CISC

Tout processeur (ou microcontrôleur) est conçu afin de pouvoir exécuter une liste finie d'instructions (**jeu d'instructions**).

CISC:

Complex **I**nstruction **S**et **C**omputer

Jusque typiquement 200 instructions différentes, dont certaines très complexes
(même si dans la plupart des cas, ce sont toujours les mêmes instructions qui sont utilisées)

Silicium complexe, mais compilation de langages de haut niveau plus simple

Typiquement 3 à 10 cycles d'horloge par instruction

Exemple: Pentium (Intel ®)

RISC:

Reduced Instruction Set Computer

Typiquement moins de 100 instructions différentes, pour la plupart très simples

Silicium plus simple, mais compilation de langages de haut niveau plus complexe

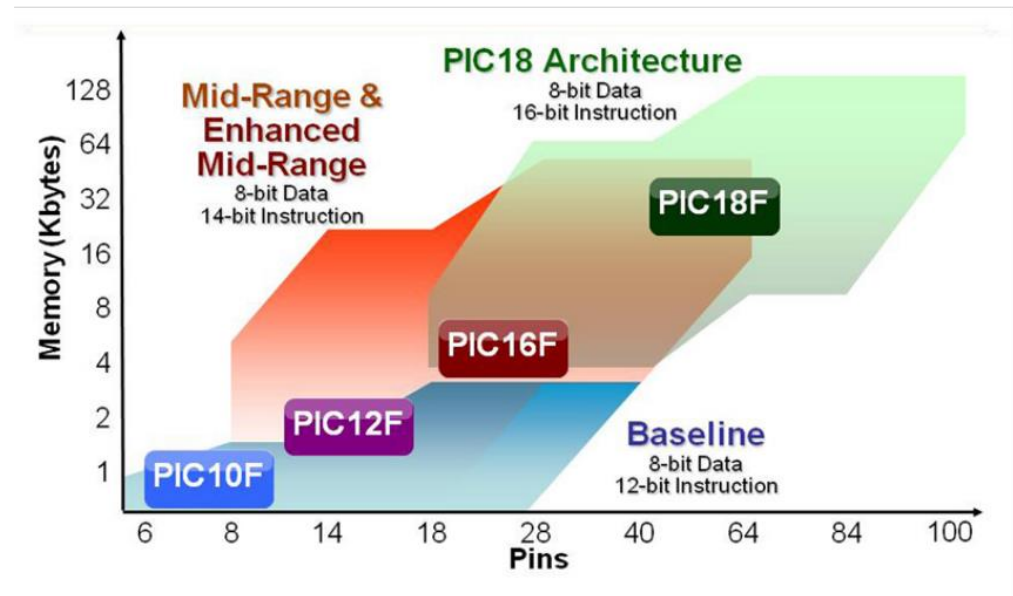
Typiquement 1 cycle d'horloge par instruction

Exemple: Microchip ®, Atmel ®

6. Anatomie d'un microcontrôleur typique

Pour ce cours:

- microcontrôleur **8 bits**
- architecture **HARVARD**
- jeu d'instructions **RISC**



mémoire programme
(Flash – non volatile)

mémoire données
(RAM – volatile)

CPU
(unité de contrôle
+ ALU)

entrées - sorties
(General
Purpose
Input
Output)

circuits d'horloge
(clock)

périphériques
(timers, convertisseur
analogique-numérique,
etc.)

