



UNIVERSITÉ
DE LORRAINE

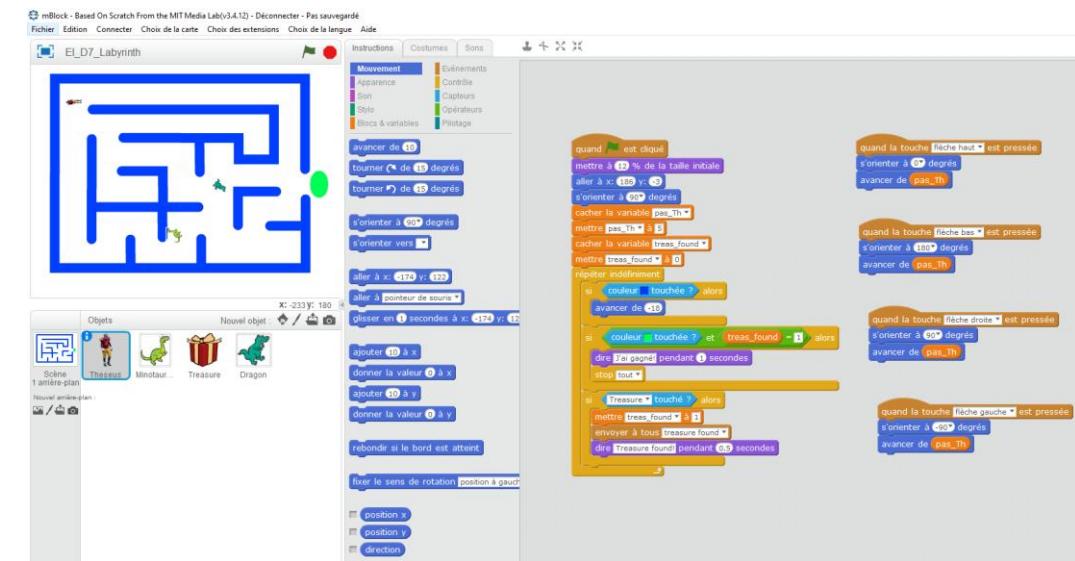
Informatique Industrielle

Stefan Bittner

stefan.bittner@univ-lorraine.fr

Contenu du cours

- Principes de l'informatique et de l'informatique industrielle
- Introduction à la programmation avec Scratch (interface graphique)
- Programmation des cartes Arduino et des robots



10h de cours: cours et exercices de programmation en Scratch

5x 4h de TP: exercices de programmation des robots

Votre expérience

- Programmation en général? Quels langages?
- Programmation avec Scratch ou mBlock?
- Programmation carte électronique (Arduino)?
- Programmation des robots?



Évaluation

- (1) Exercices de programmation des robots:
vos logiciels et un rapport écrit
précisions au début des TPs
 - (2) Contrôle terminal:
 - 2h pendant la dernière séance de TP (10/04/2026)
 - questions générales et exercices de programmation avec les robots
-
- Note finale: 50% exercices TP, 50% contrôle terminal
 - 2nd session (similaire au contrôle continu) le 25 juin 2026

Emargement

- Votre présence est obligatoire et nécessaire pour les TPs
- Signer la feuille d'émargement au début de chaque séance

En cas d'absence

- Justifier votre absence par un certificat médical ou similaire (à transmettre à la scolarité et l'enseignant)
- Si l'absence est justifiée, le temps de TP qui vous manque sera pris en compte pour la note

Objectifs d'apprentissage

- Connaître les techniques de programmation de base (variables, boucles, conditions, ...)
- Savoir programmer un système embarqué (carte Arduino)
- Concevoir un système de pilotage autonome
- Aimer les robots et l'informatique industrielle
- Être capable et motivé de réaliser vos propres projets

Informatique industrielle (1)

« **L'informatique industrielle** est une branche de l'informatique appliquée qui couvre l'ensemble des techniques de conception, d'analyse et de programmation de systèmes informatiques à vocation industrielle. »

Source: https://fr.wikipedia.org/wiki/Informatique_industrielle

Conception et programmation des systèmes « industriels » qui présentent typiquement

- microcontrôleur
- capteurs
- actionneurs

Ces systèmes peuvent

- acquérir, traiter et transmettre des informations
- interagir avec les humains ou leur environnement
- faire différentes tâches physiques



Robot de soudure

Source: en.wikipedia.org/wiki/Industrial_robot
Phasmatisnox - CC BY 3.0

Informatique industrielle (2)

Domaines d'application:

- Industrie
- Alarme
- Automobile
- Aviation
- Instrumentation
- Médicale
- Téléphonie mobile
- Terminaux de paiement pour carte bancaire
- Internet of Things (IOT)



Image fournie par Microchip

Différents aspects de l'informatique industrielle

- Programmation: programmer un microcontrôleur embarqué
- Electronique analogique: connecter une carte électronique avec capteurs, actionneurs, ...
- Physique appliquée: conception mécanique, optique, électronique, ..., des capteurs et actionneurs

Les différents systèmes programmables (1)

Les circuits spécialisés ou **ASIC** (Application Specific Integrated Circuit) :

- Les circuits spécialisés sont des circuits spécialisés dès leur conception pour une application donnée.
- Exemples : DSP (Digital Signal Processing), co-processeur arithmétique, processeur 3-D, contrôleur de bus, ...



Source: Texas Instruments

Source: Nvidia

Avantages

- Très rapide
- Consommation légère
- Optimisé pour une application

Inconvénients

- Faible modularité
- Possibilité d'évolution limitée
- Coût

Les différents systèmes programmables (2)

Les **systèmes en logique programmée** et/ou en logique programmable sont connus sous la désignation de **PLD** (programmable logic device, circuit logique programmable)

« *Un circuit logique programmable, ou réseau logique programmable, est un circuit intégré logique qui peut être reprogrammé après sa fabrication. Il est composé de nombreuses cellules logiques élémentaires pouvant être librement assemblé.* » (Source: Wikipédia)

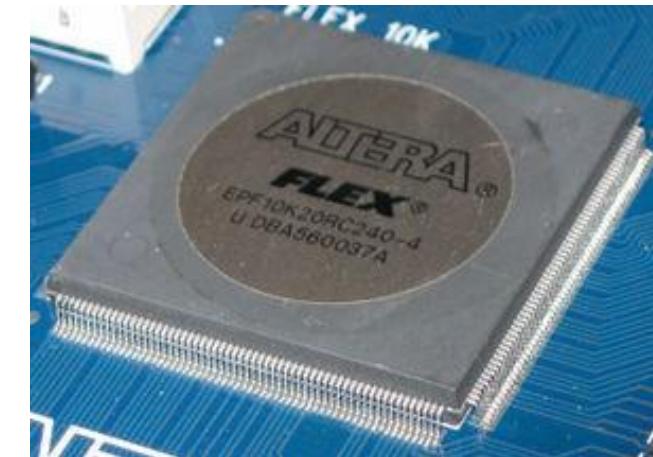
- **PAL** (programmable array logic, réseau logique programmable)
- **FPGA** (field-programmable gate array, réseau de portes programmables in-situ)

Avantages

- Forte modularité
- Rapidité

Inconvénients

- Programmation complexe
- Coûts de développement élevé



Source: Altera

Les différents systèmes programmables (3)

Les **systèmes micro-programmés**

- Les **microcontrôleurs** sont typiquement des systèmes micro-programmés.
- Un microcontrôleur est un :
« Circuit intégré comprenant essentiellement un microprocesseur, ses mémoires, et des éléments personnalisés selon l'application. » (Arrêté français du 14 septembre 1990 relatif à la terminologie des composants électroniques.)



Microcontrôleur Microchip
PIC16F690 en boîtier DIL20

Avantages

- Mise en œuvre simple
- Coûts de développement réduits

Inconvénients

- Plus lent

Composants d'un microcontrôleur

Les éléments typiques d'un microcontrôleur sont

- Microprocesseur
- Mémoire de données (RAM = random access memory)
→ volatile, contient les données lors de l'exécution d'un programme
- Mémoire programmable (EEPROM = Electrically Erasable Programmable Read-Only Memory)
→ non-volatile, contient le programme
- Ports de communication (USB, RS232, ethernet, Bluetooth ...)
- Entrées / sorties logiques
- Entrées / sorties analogiques avec ADC / DAC
→ capteurs, actionneurs, ...

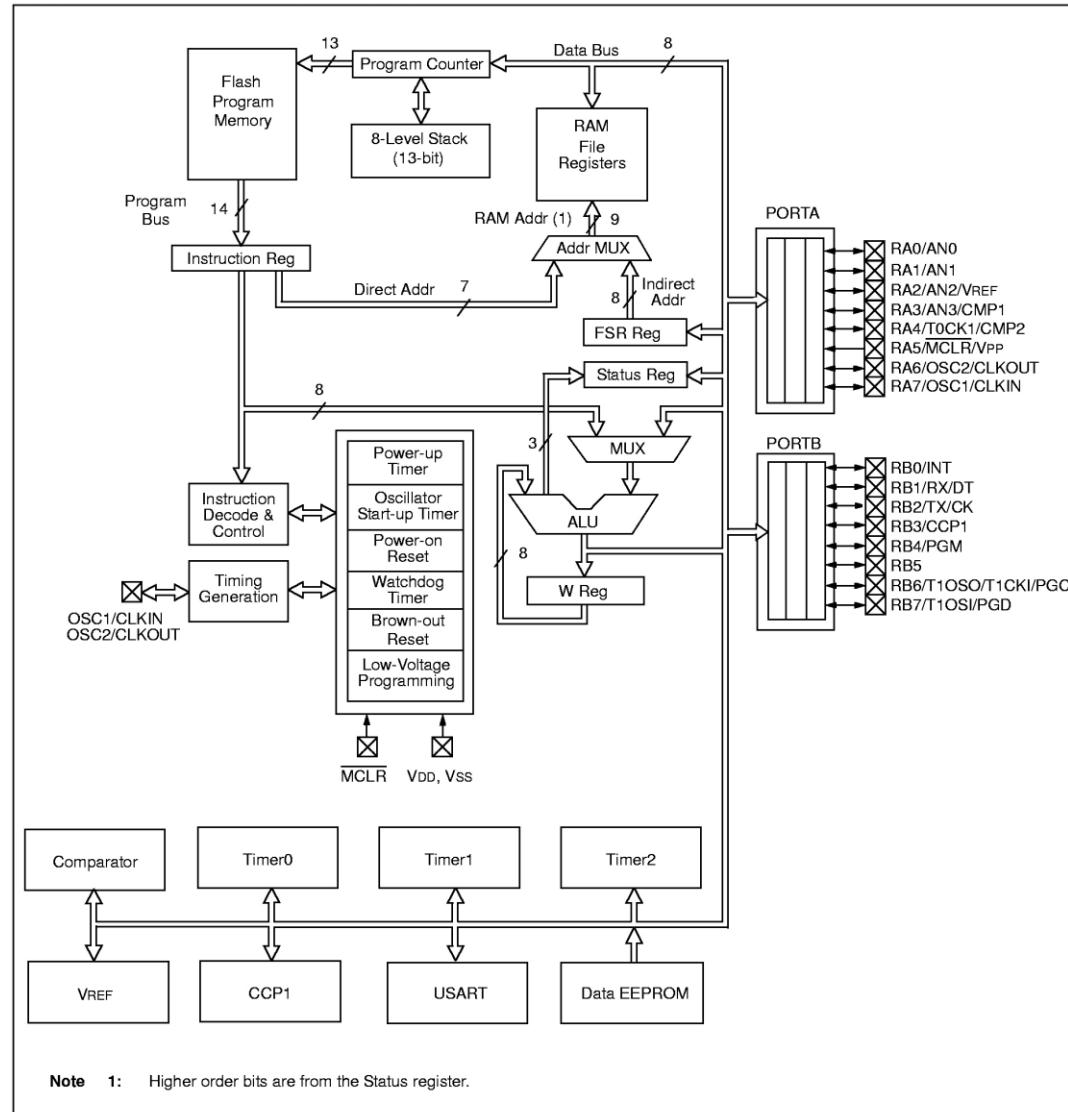
Microprocesseurs

Il existe essentiellement deux types de processeurs:

- **CISC** : Complex Instruction Set Computer
 - Grand nombre d'instructions
 - Type de processeur le plus répandu
- **RISC** : Reduced Instruction Set Computer
 - Nombre d'instructions réduit (sélection des instructions pour une exécution plus rapide)
 - Décodage des instructions plus rapide
- Aujourd'hui il n'existe plus d'écart important en performance entre les deux types.

Structure des système micro-programmés

FIGURE 3-1: BLOCK DIAGRAM



Un système micro-programmé comme un microcontrôleur présente un grand nombre de composants

Exemple:

Le microcontrôleur **PIC16F628** de Microchip

Issu de la documentation technique du PIC16F628 (Microchip)

Les bus d'un système micro-programmé

« *Un bus est un jeu de lignes partagées pour l'échange de mots numériques.* » (Traité de l'électronique, Paul Horowitz & Winfield Hill)

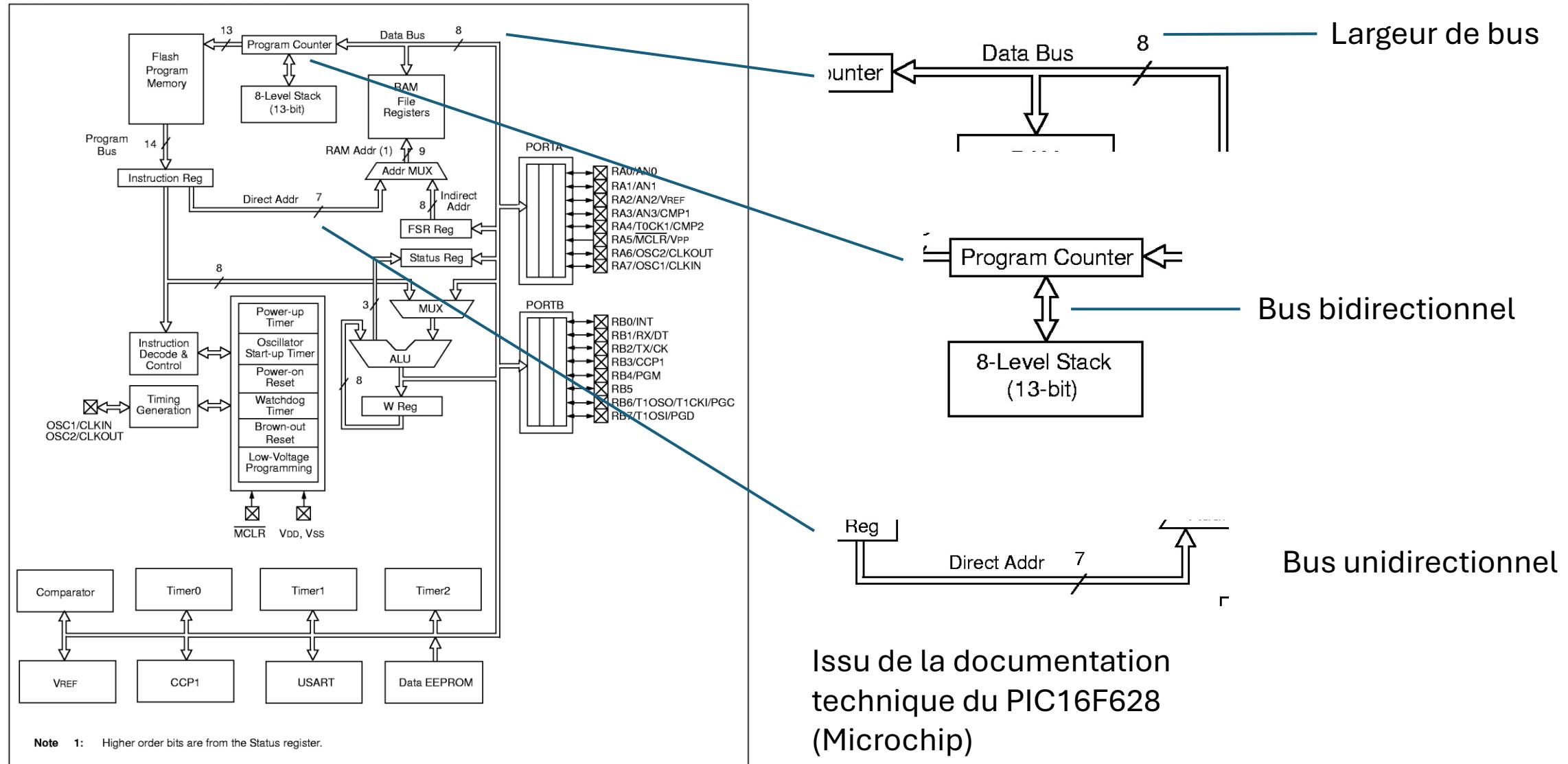
Définition: Un bus permet de faire transiter (liaison série/parallèle) des informations codées en binaire entre deux points. Typiquement les informations sont regroupées en mots : octet (8 bits), word (16 bits) ou double word (32 bits).

Caractéristiques d'un bus:

- nombres de lignes
- fréquence de transfert

Les bus sur un diagramme de bloc

FIGURE 3-1: BLOCK DIAGRAM



Différents types de bus

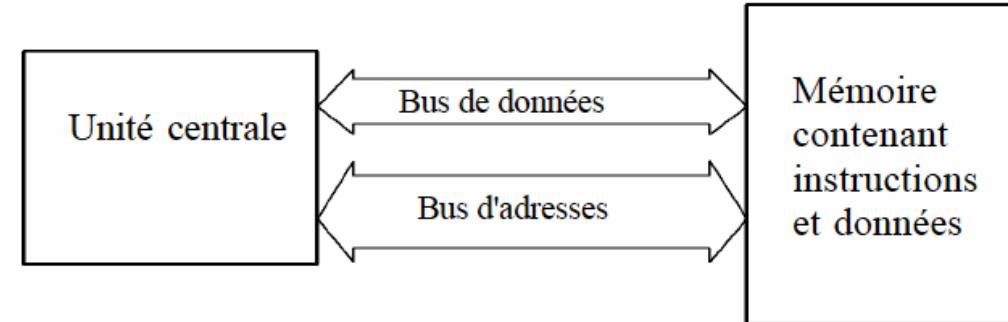
Il existe 3 types de bus

- **Bus de données** : permet de transférer entre composants des données, p.ex. résultat d'une opération, valeur d'une variable, etc.
- **Bus d'adresses** : permet de transférer entre composants des adresses, p.ex. adresse d'une case mémoire, etc.
- **Bus de contrôle** : permet l'échange entre les composants d'informations de contrôle [bus rarement représenté sur les schémas].
p.ex. périphérique prêt/occupé, erreur/exécution réussie, etc.

Une **adresse** est un nombre binaire qui indique un emplacement dans une zone mémoire

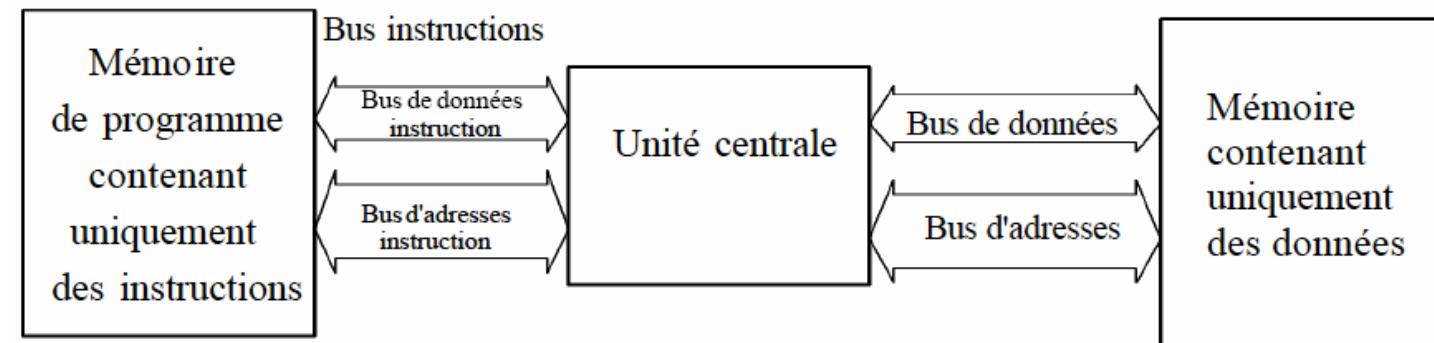
Structures des systèmes micro-programmés

- Structure von Neumann



Extraits du cours intitulé « Les systèmes micro-programmés »

- Structure de Harvard

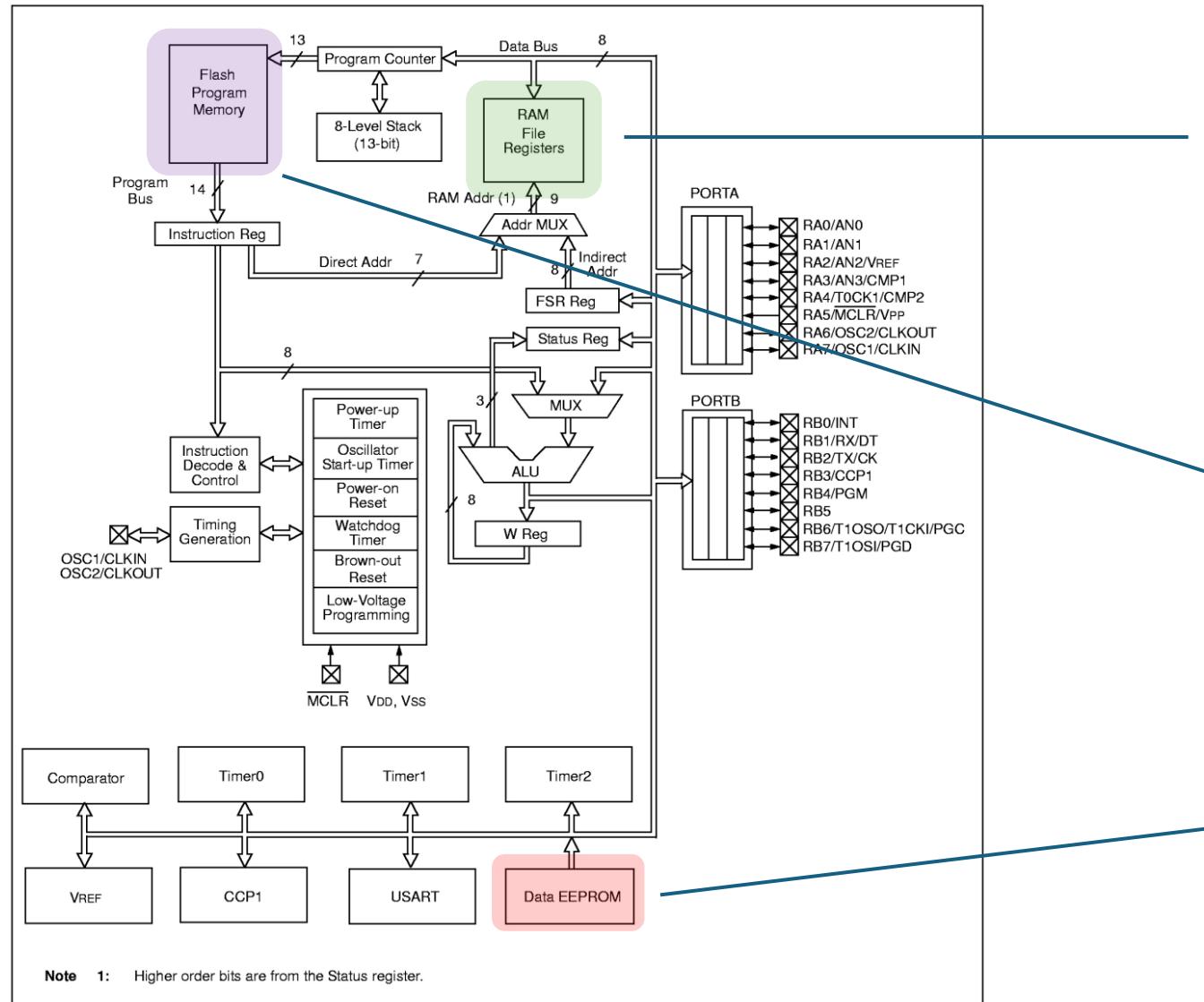


La différence se situe au niveau de la **séparation ou non des mémoires programmes** et données. La structure de Harvard permet de transférer données et instructions simultanément, ce qui permet un gain de performances.

Le PIC16F628 présente une structure de Harvard.

Eléments dans un diagramme de bloc

FIGURE 3-1: BLOCK DIAGRAM



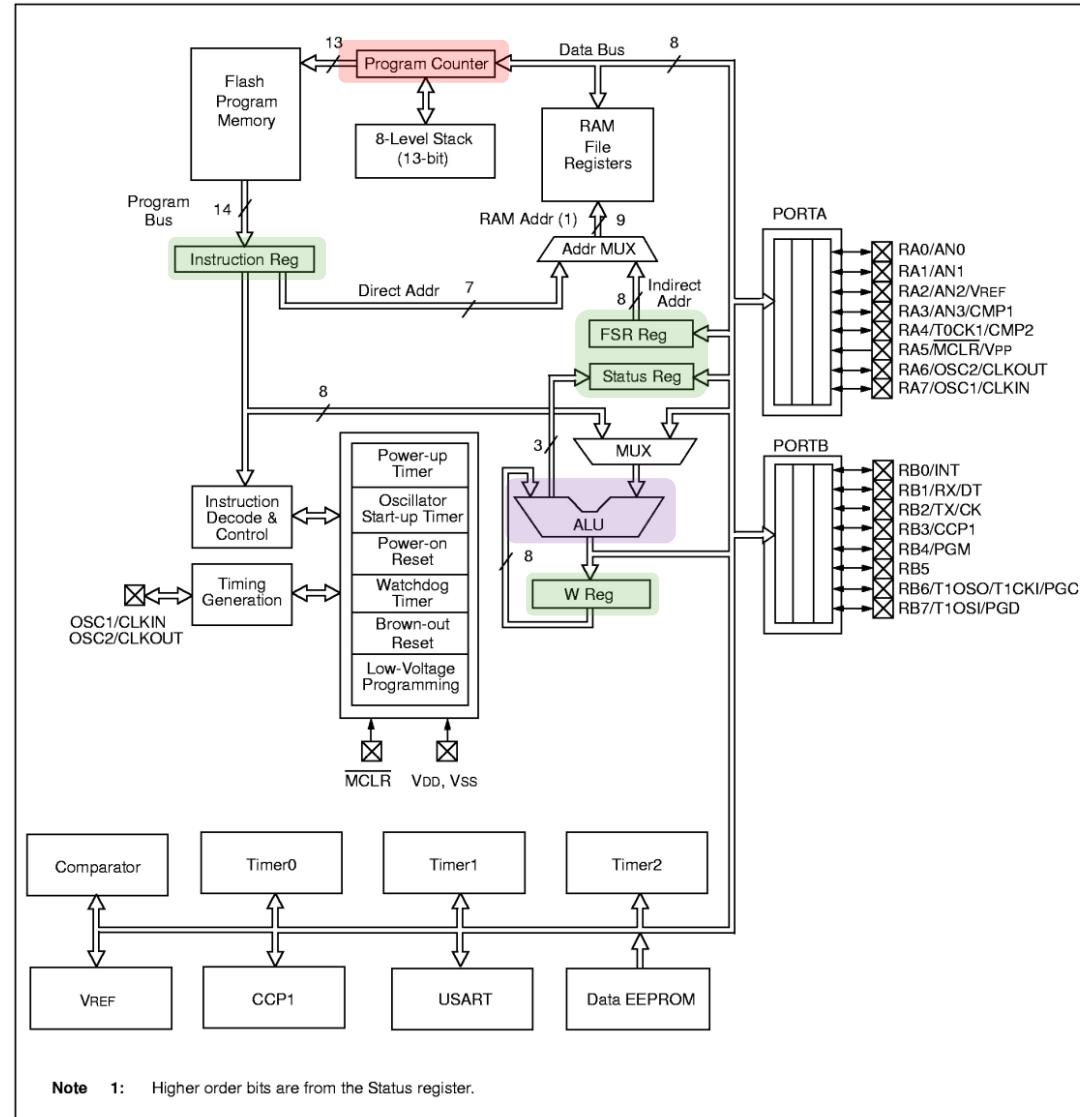
RAM (random access memory)
Mémoire rapide qui permet de stocker temporairement des données

ROM (read only memory)
Mémoire à lecture seule; contient le programme

EEPROM (electrically erasable programmable read-only memory)
mémoire lente qui permet de stocker des données sans alimentation

Eléments dans un diagramme de bloc

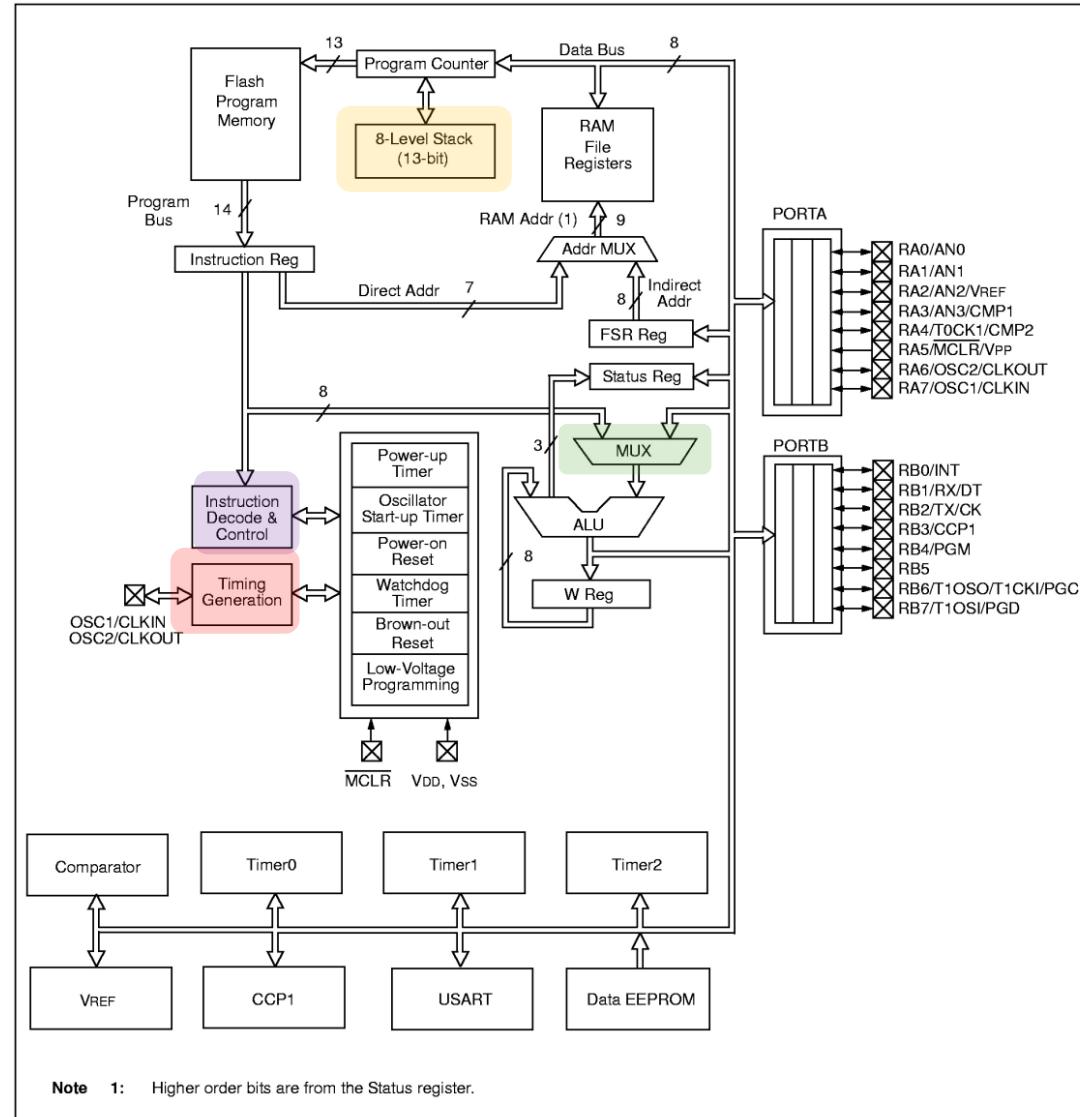
FIGURE 3-1: BLOCK DIAGRAM



- PC (Program counter)
- Registres (cases mémoire)
- ALU (arithmetic & logical unit)
- Multiplexeur
- Décodeur d'instructions
- Horloge
- Stack (pile): LIFO = last in, first out

Eléments dans un diagramme de bloc

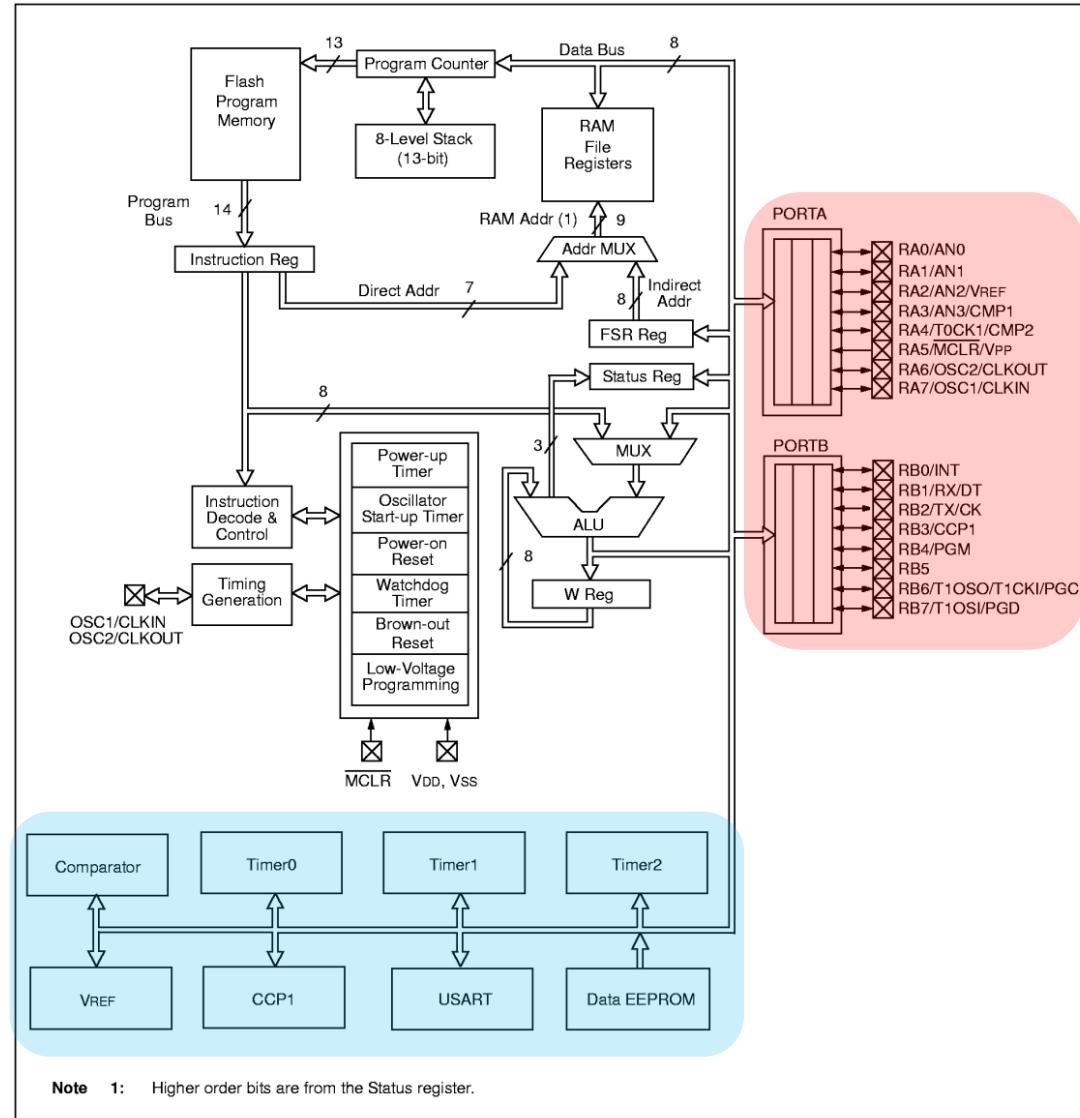
FIGURE 3-1: BLOCK DIAGRAM



- PC (Program counter)
- Registres (cases mémoire)
- ALU (arithmetic & logical unit)
- Multiplexeur
- Décodeur d'instructions
- Horloge
- Stack (pile): LIFO = last in, first out

Eléments dans un diagramme de bloc

FIGURE 3-1: BLOCK DIAGRAM



- Ports d'entrée/sorties
→ *interaction avec périphériques*

- USART (Universal Synchronous Asynchronous ReceiverTransmitter)
interface de communication série
- CPP (Capture/Compare/PWM)
Modulation en largeur d'impulsions
- Timer
- Comparateur
- DAC / ADC
- Référence de tension

Les éléments de choix

Architecture

- ALU (8, 16, 32, 64 bits)
- Structure du processeur (Harvard, Von Neumann)
- Type de processeur (RISC, CISC)
- Taille des mémoires programme et données
- Nombre de ports d'entrée/sortie

Fonctionnalités

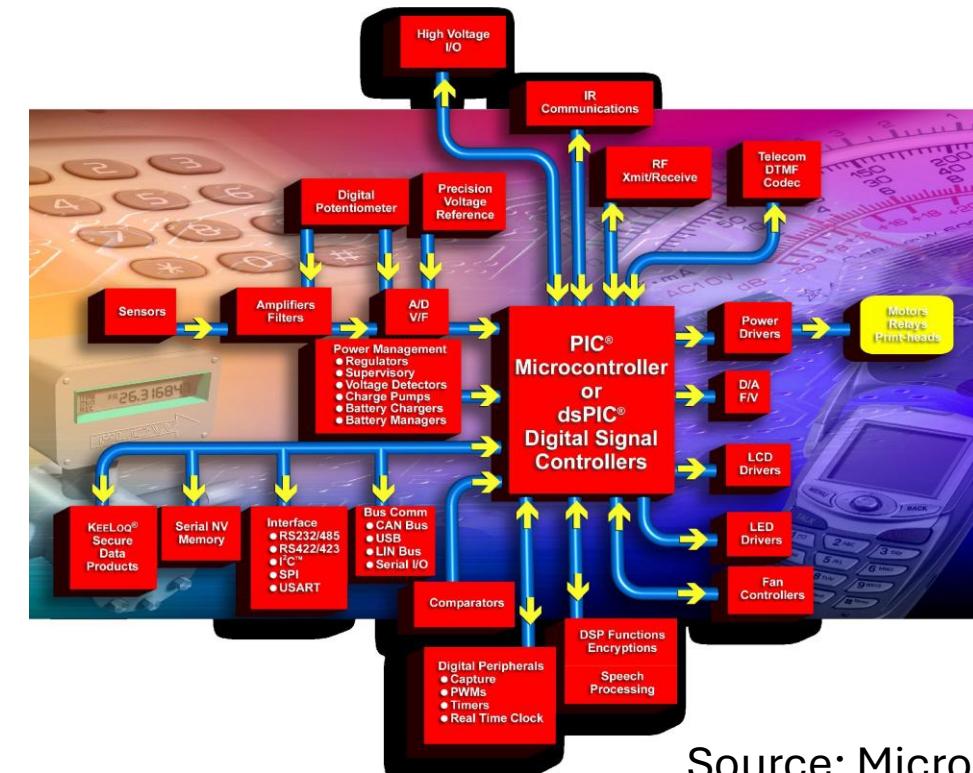
- Fonctions analogiques : ADC, DAC, Comparateur, ...
- Fonctions de timing : Timer, Watchdog, ...
- Fonctions de communication : USART (Communication série), USB, I2C, ...
- Facilité de programmation

Mise en œuvre, maintenance

- Coût de développement : outils de développement, formation, ...
- Suivi du microcontrôleur : production suivie, disponibilité, composant obsolète, ...

Caractéristiques électriques

- Fréquence d'horloge
- Tensions d'alimentation
- Consommation d'énergie, modes faible consommation d'énergie, ...



Source: Microchip

Exercices de programmation en Scratch

- Rendez-vous sur Arche:
Informatique Industrielle - UFR MIM 2025-26
- 9 exercices dans le document
Programmation en Scratch exercices
- Montrer chaque exercices à l'enseignant avant de commencer le suivant
- Soumettre les exercices sur Arche
- Sauvegardez vos logiciels sur le serveur **Espace personnel pédagogique**
volume O:\
<\\etu.ad.univ-lorraine.fr\\<login>>

