

Circuits intégrés

Notions introduites

- architecture d'un microcontrôleur
- modèle de Harvard
- système sur puce

En 1975, Gordon E. Moore (cofondateur de la société Intel) énonça la conjecture suivante sur l'évolution des capacités des circuits intégrés, appelés familièrement puces électroniques :

Dans les microprocesseurs, le nombre de transistors sur une puce va doubler tous les deux ans.

Bien que fondée sur un constat empirique de l'industrie des fabricants de circuits entre les années 1965 et 1975, cette prédiction, qu'on appelle aussi loi de Moore, s'est révélée incroyablement juste. On est ainsi passé de 2250 transistors en 1971 sur un microprocesseur Intel 4004 (un des premiers microprocesseurs) à plusieurs dizaines de milliards aujourd'hui sur les derniers microprocesseurs où la taille des transistors n'est que de 7 nanomètres, soit à peine plus que l'épaisseur de quelques dizaines d'atomes de silicium. Cette loi de Moore, qui est (plus ou moins) généralisable à tous les composants électroniques (mémoire, etc.), a permis non seulement une augmentation de la puissance de calcul des ordinateurs (en augmentant leur fréquence de fonctionnement grâce à une diminution de la distance entre les composants), mais également une baisse des coûts (en rassemblant plusieurs composants en un seul). La diminution de la taille des transistors a également permis de baisser la tension électrique pour les faire fonctionner, ce qui a engendré une diminution de leur consommation énergétique à puissance de calcul équivalente.

La miniaturisation des circuits électroniques est telle qu'il est possible aujourd'hui de rassembler sur une même puce tous les composants essentiels d'un ordinateur (microprocesseur, mémoire, interfaces d'entrées-sorti etc.). Ces systèmes complets, que l'on retrouve dans tous les systèmes embarqués, portent les noms de microcontrôleur, système sur puce (en anglais System on Chip) ou circuit logique programmable.

1) Microcontrôleurs

Les microcontrôleurs sont des circuits intégrés qui regroupent sur même puce (de quelques centimètres carrés) un microprocesseur, de la mémoire, des ports d'entrée-sortie, des périphériques et des bus de communication. La puissance de calcul et la capacité en mémoire de ces puces sont en dessous des composants que l'on trouve sur les cartes mères des ordinateurs. Ainsi, la fréquence d'horloge (qui détermine le nombre d'instructions exécutées par seconde) est généralement de quelques dizaines de mégahertz (MHz) et la taille de la mémoire se mesure seulement en kilo octets (Ko). En revanche, les microcontrôleurs ont une consommation électrique très faible ce qui leur confère une très grande autonomie lorsqu'ils sont alimentés par des batteries. Enfin, leur coût de fabrication est aussi très réduit et les microcontrôleurs s'achètent pour seulement quelques euros. Ces « mini »-ordinateurs sont principalement utilisés dans des systèmes informatiques embarqués, par exemple dans des avions, des voitures, des robots, etc. Les rôles des microcontrôleurs dans ces systèmes sont très spécifiques mais ils partagent néanmoins certaines caractéristiques : acquisition de données (grandeurs physiques), contrôle d'un processus (actions mécaniques ou électroniques), contraintes de temps.

Prenons l'exemple d'un régulateur de vitesse d'une voiture qui maintient le véhicule à une vitesse définie par le conducteur :

- il récupère la vitesse de la voiture via un capteur externe branché à l'un de ses ports d'entrée ;
- il est en charge de contrôler l'accélération ou le freinage automatique du véhicule pour conserver la vitesse définie par le conducteur et pour cela il envoie des ordres à des actionneurs, externes également, reliés à ses ports de sortie ;
- il est soumis à des contraintes temporelles (temps de réponse) très fortes pour garantir la sécurité des passagers.

Des périphériques auxiliaires, intégrés dans la puce du microcontrôleur peuvent être nécessaires pour réaliser cette tâche. Par exemple, un filtre de conversion analogique/numérique est utilisé pour interpréter les données analogiques en entrée du capteur. Un timer peut aussi être utilisé pour mesurer le temps afin de s'assurer que les réponses seront envoyées dans un certain délai.

Étant données les capacités de calcul et de mémoire des microcontrôleurs, il n'est pas envisageable d'y faire tourner un système d'exploitation (OS) qui prendrait beaucoup trop de place et de temps d'exécution. De plus, un OS complique fortement la mise au point d'applications temps réel car il faut maîtriser par exemple quand celui-ci va interrompre les processus en activité, combien de temps prend le basculement entre les tâches, etc.

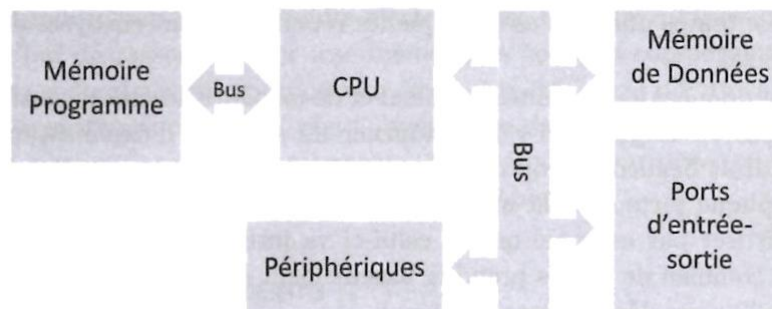
Architecture d'un microcontrôleur.

Les microcontrôleurs sont principalement des composants autonomes qui, dès qu'ils sont sous tension, exécutent le programme contenu dans leur mémoire. Ainsi, la principale différence entre l'architecture d'un microcontrôleur et celle d'un ordinateur basé sur le modèle de von Neumann est que la mémoire qui contient les programmes n'est pas la même que celle qui contient les données. La mémoire Programme est une mémoire morte, c'est-à-dire une mémoire qui ne perd pas l'information qu'elle contient quand le microcontrôleur n'est plus alimenté en électricité. Il y a plusieurs types de mémoire morte :

- ROM (Read-Only Memory), désigne une mémoire dont le contenu est enregistré à la construction de la mémoire et ne peut être lu ;
- EEPROM (Electrically-Erasable Programmable Read-Only Memory), désigne une mémoire dont le contenu peut être écrit et effacé plusieurs fois. L'accès en lecture/écriture est assez lent, mais on peut effacer et écrire son contenu octet par octet ;
- FLASH, désigne un type de mémoire similaire à l'EEPROM mais avec un accès très rapide en lecture et (un peu moins) rapide en écriture. Contrairement à l'EEPROM, la mémoire FLASH est découpée en secteurs physiques (dont la taille varie de quelques centaines d'octets à plusieurs kilo octets) et ne peut être effacée ou écrite octet par octet comme l'EEPROM, mais uniquement secteur par secteur. De plus, pour réécrire un octet, on est obligé d'effacer tout le secteur qui le contient avant d'écrire (un secteur entier).

La mémoire *Données* peut être composée d'une partie de mémoire morte et d'une autre de mémoire vive (aussi appelée mémoire RAM, pour Random Access Memory), c'est-à-dire de la mémoire dans laquelle on peut lire et écrire à convenance (avec un temps d'accès identique quelle que soit la partie de la mémoire qui est accédée), mais qui perd toutes ses données dès qu'elle n'est plus alimentée en

électricité. Certains microcontrôleurs profitent de la séparation des mémoires de programmes et de données pour implémenter une architecture dite de Harvard.



Architecture de Harvard.

Dans cette architecture (figure ci-dessus), on accède aux mémoires Programme et Données (ou ports d'entrée-sortie) à travers deux bus distincts, ce qui procure plusieurs avantages. Tout d'abord, cela permet de gagner du temps en transférant simultanément au microprocesseur les instructions et les données sur lesquelles elles agissent. Ensuite, cela permet d'avoir des mots mémoires différents pour les instructions et les données. Par exemple, on peut utiliser une mémoire morte avec des mots de 16 bits pour stocker les instructions et seulement des mots de 8 bits pour la mémoire de données.

Jeu d'instructions.

Lorsqu'ils sont basés sur une architecture Harvard, les microcontrôleurs ont un jeu d'instructions réduit (en anglais RISC, Reduced Instruction Set Computer), composé d'instructions simples, avec une longueur et un format d'instruction fixe. Cela simplifie à la fois le circuit de décodage mais également l'architecture globale. Par exemple, les instructions pour déplacer directement des données entre deux cellules mémoires sont impossibles (il faut passer par des registres du microprocesseur).

Périphériques.

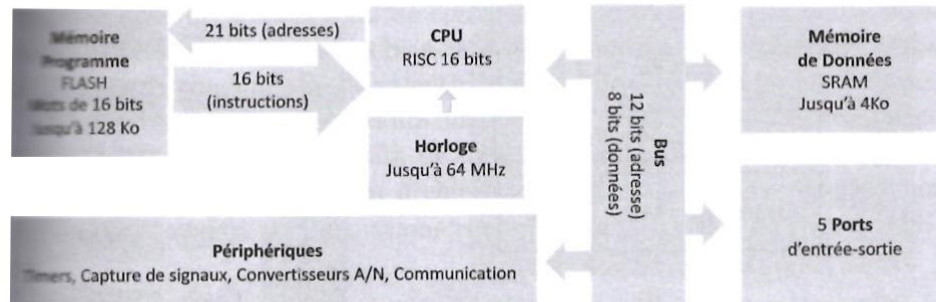
Les microcontrôleurs sont équipés de nombreux périphériques pour effectuer des tâches bien spécifiques. Voici ci-dessous quatre familles de périphériques couramment utilisés.

- Les Timers sont utilisés pour la gestion du temps. Ils servent à déclencher une action au bout d'un laps de temps t . Ceci est réalisé à l'aide d'un compteur, initialisé à t , qui est décrémenté à chaque cycle d'horloge. L'action est déclenchée quand la valeur du compteur atteint 0 ;
- Les modules de capture de signaux permettent de récupérer la valeur d'un *timer* au moment où un événement extérieur se déclenche, par exemple quand un signal d'entrée change d'état ;
- Les convertisseurs Analogique/Numérique permettent de transformer une valeur de tension analogique en un nombre binaire ;
- Les modules de communication permettent de faire communiquer les microcontrôleurs entre eux.

Exemple.

Les microcontrôleurs PIC sont fabriqués par la société Microchip. Il s'agit d'une gamme de microcontrôleurs qui s'appuient sur une architecture de Harvard avec un jeu d'instructions RISC.

Il existe plusieurs familles de puces PIC. Chaque famille se distingue par le type et la quantité de mémoires utilisées, le nombre de bits par instruction, la fréquence d'horloge du CPU, le nombre de ports d'entrée-sortie ou encore les périphériques présents dans la puce.



Architecture des microcontrôleurs de la famille PIC 18F.

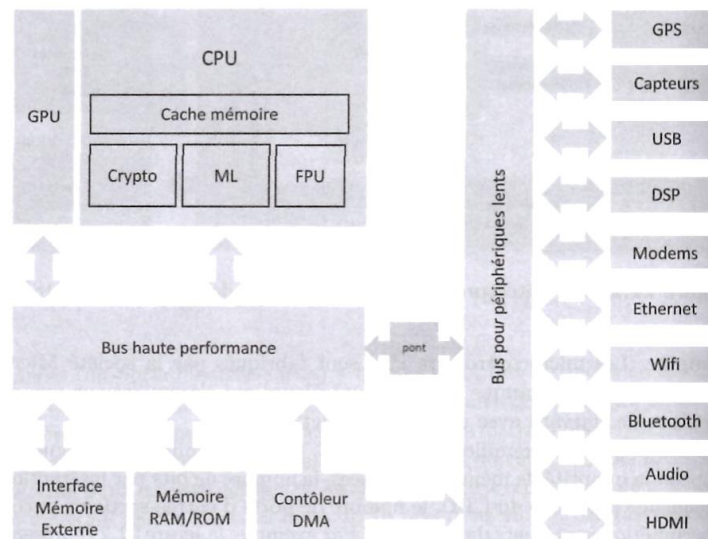
Par exemple, la figure ci-dessus représente l'architecture de la famille des microcontrôleurs PIC 18F. On y distingue les caractéristiques suivantes :

- Un CPU avec jeu d'instructions RISC avec des registres de 8 bits et des instructions codées sur 16 bits.
- Une mémoire programme de type FLASH dont la capacité peut aller jusqu'à 128 Ko (même si, en théorie, une capacité de 2 Mo est possible).
- Un bus de communication vers la mémoire programme sur 21 bits d'adressage (permet donc d'accéder jusqu'à 2 Mo de mémoire) et 16 bits de large pour récupérer en un cycle une instruction sur 16 bits.
- Une horloge pouvant être cadencée jusqu'à 64 MHz.
- Une mémoire de données de type SRAM (Static RAM) de 4 ko. Il s'agit d'un type de mémoire RAM qui ne nécessite pas de rafraîchissement pour maintenir ses informations.
- Un bus de communication sur 12 bits d'adressage et 8 bits de données.
- Jusqu'à 5 ports d'entrée-sortie bidirectionnels.
- Des périphériques comme des *timers*, des modules de capture de signaux, des convertisseurs A/N et des modules de communication.

2) Système sur puce

Un système sur puce, appelé System on Chip (SoC) en anglais, rassemble sur un même circuit intégré tous les composants habituellement présents sur la carte mère d'un ordinateur. Les caractéristiques d'un SoC sont de fait très proches de celles d'un ordinateur. Ils ont une puissance de calcul comparable qui repose sur des microprocesseurs de dernière génération avec de nombreux cœurs cadencés à plusieurs gigahertz, ainsi que des processeurs dédiés (graphique, sécurité). Leur capacité mémoire se mesure en gigaoctets et ils incluent des mémoires de type RAM et FLASH. Enfin, ils contiennent de nombreux périphériques. Tout cela sur une puce d'environ 100 mm².

Architecture d'un système sur puce.



Architecture d'un *System on Chip*.

Les SoC reposent habituellement une architecture comparable à celle donnée dans le schéma de la figure ci-dessus. Contrairement à l'architecture de Harvard sur laquelle repose les microcontrôleurs, on retrouve un modèle de von Neumann dans lequel il n'y a de différence entre la mémoire des programmes et celle de données. Les unités de calcul sont ainsi connectées à la mémoire (RAM ou ROM) ou aux périphériques par un unique bus de communication.

Les microprocesseurs des SoC sont presque toujours basés sur un jeu d'instructions RISC, avec des registres de 32 ou 64 bits. Contrairement aux microcontrôleurs, ces CPU sont cadencés à des fréquences tellement élevées qu'ils sont fortement ralentis dès qu'ils lisent ou écrivent des données dans la mémoire ou qu'ils dialoguent avec des périphériques. Cette différence de vitesse entre unités de calculs, mémoires et périphériques est telle qu'il est crucial de concevoir des moyens de communication entre ces composants qui soient bien calibrés en fonction des vitesses et débits de données de chacun, afin de ralentir le moins possible le microprocesseur.

Ainsi, parce que les composants mémoire ont généralement une fréquence de fonctionnement et des débits de données bien plus importants que les autres périphériques, certaines architectures de SoC mettent en place un système de transfert d'informations reposant sur deux bus de communication :

- Un bus *haute performance* est dédié à l'échange d'information entre le microprocesseur (CPU), les processeurs dédiés à une tâche particulière comme les processeurs graphique (GPU) et les différents composants mémoire ;
- Un deuxième bus est utilisé pour la communication avec les périphériques plus lents. Ce bus est relié au bus rapide via un *pont* de communication qui met en œuvre un protocole d'échange.

Cette architecture permet d'éviter que les bus de communication ne deviennent des goulots d'étranglement pour le CPU qui, lorsqu'il souhaite communiquer avec des composants très rapides comme les mémoires, se retrouve bloqué en attente d'une réponse d'un périphérique beaucoup plus lent.

Afin de limiter encore plus l'impact de la différence de vitesse entre le CPU et les autres composants, les SoC disposent d'un mécanisme d'accès direct à la mémoire (en anglais Direct Memory Access — DMA) similaire aux cartes mères des ordinateurs. Ce mécanisme permet de transférer directement des

données de la mémoire vers ou depuis un périphérique, sans aucune intervention du CPU. Pour cela, le CPU ou les périphériques envoient des informations concernant les adresses et la quantité de données à échanger au contrôleur DMA, qui se charge en retour de réaliser ces transferts et d'avertir quand ils sont effectués.

Même en utilisant un bus à haute performance, le microprocesseur risque d'être fortement ralenti lorsqu'il accède à la mémoire centrale (RAM ou ROM). Pour limiter l'effet de ce goulot d'étranglement, un mécanisme de *mémoire cache*, identique à celui des microprocesseurs des ordinateurs, est intégré dans les systèmes sur puce. Cela consiste à insérer une mémoire très rapide entre le CPU et la mémoire centrale afin de servir de tampon en lecture et en écriture. Ainsi, lorsqu'une instruction doit accéder à la mémoire centrale, elle vérifie d'abord si la mémoire cache contient une copie des cellules qu'elle recherche. Si c'est le cas, alors elle écrit ou lit directement dans le cache. Sinon, il se produit ce qu'on appelle un *défaut de cache* et les tâches à réaliser dans ce cas sont différentes selon qu'il s'agisse d'un accès en lecture ou en écriture :

- **Accès en lecture.** Un défaut de cache va provoquer un chargement des données de la mémoire centrale vers le cache, puis les données seront ensuite envoyées au CPU. Ce chargement nécessite de libérer de l'espace dans le cache pour accueillir les nouvelles données. Le choix de cet espace n'est pas anodin car le CPU pourrait avoir besoin de son contenu quelques instructions après. La *politique de remplacement* mise en œuvre dans une mémoire cache est donc un point important de son efficacité.
- **Accès en écriture.** Si le cache ne contient pas la partie de la mémoire dans laquelle le CPU souhaite écrire, il commence par la charger puis elle est modifiée dans le cache. Pour qu'il y ait une cohérence entre mémoire dans le cache et la mémoire centrale, une *politique d'écriture* est mise en place. Là encore, le choix de cette politique a une grande influence sur l'efficacité du SoC.

Bien que l'architecture globale d'un SoC soit celle d'un microprocesseur RISC, le CPU peut également contenir des circuits dédiés à certaines opérations afin d'accélérer certains calculs. Par exemple, on peut trouver aujourd'hui :

- Une unité de calcul pour nombres flottants (FPU) simple ou double précision ;
- Un circuit dédié aux opérations sur les matrices, par exemple pour accélérer les algorithmes d'apprentissage automatique (en anglais *Machine Learning* — ML) très utilisés par les applications d'intelligence artificielle. Ce circuit peut par exemple exploiter les possibilités de parallélisation des opérations sur ces structures ;
- Un composant pour la sécurité qui implémente des opérations élémentaires pour accélérer les algorithmes cryptographiques ;

Enfin, la liste des périphériques présents dans un système sur puce n'est limitée que par l'espace physique qu'ils occupent sur la puce. Voici ci-dessous une liste non exhaustive des composants fréquemment présents sur de telles puces :

- des modems (2G/3G/ 4G),
- des circuits radio (WiFi, Bluetooth),
- une puce GPS,
- des ports d'entrée-sortie (USB, Ethernet, HDMI, Audio),
- des capteurs (CCD).

Avantages et inconvénients. Les SoC sont aujourd'hui les composants incontournables de l'informatique nomade (smartphones, tablettes). Ils sont également de plus en plus utilisés dans le monde plus large des systèmes embarqués (voitures, robots, etc.). Ce succès est dû aux avantages suivants :

- **Énergie.** Il est admis qu'une grande partie de la consommation électrique d'un circuit est liée au câblage (et donc au transport) entre les composants. Comme tous les composants d'un SoC sont connectés entre eux sur des distances très petites, et de manière interne sans câblage énergivore, les gains sont très importants. Cette faible consommation énergétique implique aussi une très faible perte de chaleur qui évite de recourir à un ventilateur pour refroidir la puce. Les SoC sont donc silencieux ;
- **Coût.** Le prix d'un système sur puce est très petit si on le compare à celui d'une carte mère rassemblant les mêmes composants. Ceci s'explique d'une part par son coût de production, très bas, résultant de la forte automatisation du processus de fabrication des puces de l'industrie du hardware et, d'autre part, par des volumes de production importants ;
- **Sécurité.** Le circuit d'un SoC est conçu de manière globale, tant pour la partie hardware que pour celle du micrologiciel (en anglais firmware), c'est-à-dire des programmes intégrés dans les différents composants qui leur permettent d'assurer leurs fonctions. Cette vue d'ensemble du système, sans aucune possibilité d'ajouter ou changer les composants, est un avantage important pour garantir la sécurité du système.

Bien qu'ayant de nombreux avantages, les systèmes sur puce ont aussi quelques inconvénients. Contrairement à un ordinateur équipé d'une carte mère, où chaque composant est connecté et peut être remplacé en cas de panne ou pour une nouvelle version, les SoC ne permettent pas de mise à jour, aucune extension n'est possible et, si un seul transistor est endommagé, il ne sera pas possible de réparer l'unité défaillante.

Exemple. Le système sur puce *A.13 bionic*, disponible sur les téléphones et tablettes de la société Apple, est un SoC qui repose sur une architecture similaire à celle présentée dans cette section. Cette puce électronique contient 8,5 milliards de transistors gravés avec une précision de 7 nanomètres. La liste ci-dessous résume ses principaux composants :

- Un CPU disposant d'un jeu d'instructions RISC et de registres de 64 bits. Il contient six cœurs dont deux rapides (2.65GHz) et quatre plus lents (1.8GHz), ces derniers étant plus économes en énergie. Les cœurs rapides intègrent un module (AMX) pour accélérer les opérations de *Machine Learning* comme les multiplications de matrices. Ensemble, ces modules sont capables d'effectuer 1000 milliards d'opérations 8 bits par seconde ;
- Un processeur graphique (GPU) avec quatre cœurs pour toutes les opérations 3D ;
- Un processeur avec huit cœurs, appelé *Neural Engine*, qui s'occupe de tous les traitements d'informations et calculs liés à l'intelligence artificielle, pour la reconnaissance faciale et la réalité augmentée notamment ;
- Un module spécialisé pour les opérations cryptographiques (AES) ;
- Une capacité mémoire de 4Go de RAM, directement sur la puce.

À retenir.

La miniaturisation des circuits intégrés aujourd'hui est telle qu'il est possible d'avoir des ordinateurs sur une unique puce d'une centaine de millimètres carrés. Ces mini-ordinateurs, appelés microcontrôleurs ou System on Chip (SoC), intègrent tous les composants habituellement présents sur la carte mère d'un ordinateur (CPU, mémoire et périphériques). Les microcontrôleurs, peu coûteux et avec des capacités de calculs et de mémoire limitées, reposent souvent sur une architecture de Harvard où la mémoire des programmes est dissociée de celle des données. Les SoC sont de véritables ordinateurs avec des CPU très puissants, tellement rapides qu'une architecture avec deux bus est nécessaire pour éviter les goulots d'étranglement.

Exercices

Exercice 1

Quels sont les spécificités et les avantages d'une architecture de Harvard par rapport à celle de von Neumann ?

Exercice 2

Que signifie l'acronyme RISC ?

Exercice 3

Quels sont les éléments essentiels d'un microcontrôleur ?

Exercice 4

Donner quelques critères pour choisir d'utiliser un microcontrôleur.

Exercice 5

Quelles sont les spécificités et les avantages d'un système sur puce ?

Exercice 6

Ecrire un programme Python qui affiche toutes les n secondes la température du CPU d'un ordinateur. Pour cela, on peut utiliser la bibliothèque `psutil` qui donne accès à de nombreuses informations concernant les composants de la machine. En particulier, la fonction `sensors_temperatures()` renvoie des données renvoyées par les sondes de température sous la forme d'un dictionnaire :

```
{...
  'coretemp' : [ shwtemp(...,current=26,...),...],
  ...
}
```

Une entrée `'coretemp'` de ce dictionnaire contient les informations de température pour le CPU rassemblées sous la forme d'un tableau qui décrit chaque cœur du CPU. L'information d'un cœur est donnée sous forme d'un n-uplet nommé qui contient, entre autres, un champ `current` qui donne sa température actuelle. Ainsi, si on appelle `tempinfos` ce dictionnaire, la température courante du premier cœur sera accessible par `tempinfos['coretemp'][0].current`.