Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Carthage

Ecole Nationale d'Ingénieurs de Carthage



وزارة التعليب العالي و البدك العلمين جامعــة قرطح المدرمــة الوطنية الممندمين بقرطح

Systèmes embarqués Niveau: 2 ING INFO

Année Universitaire 2020/2021

Plan du module

- Introduction aux systèmes embarqués
 - Cible logicielle: Les microcontrôleurs STM32
 - Cible matérielle: Langage VHDL et carte FPGA

Plan du module

- Introduction aux systèmes embarqués
 - Cible logicielle: Les microcontrôleurs STM32
 - Cible matérielle: Langage VHDL et carte FPGA

Observation

Se réveiller avec un radio réveil





Programmer **une machine** à **café** pour boire ...



Calculateur indiquant la mise ou non de la ceinture par affichage ou voix...



Télécommandes



Smartphones

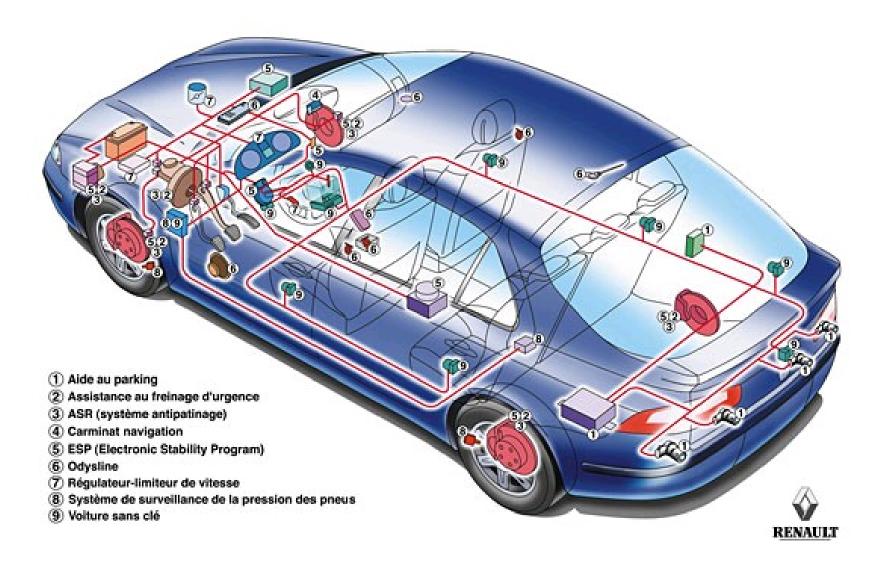




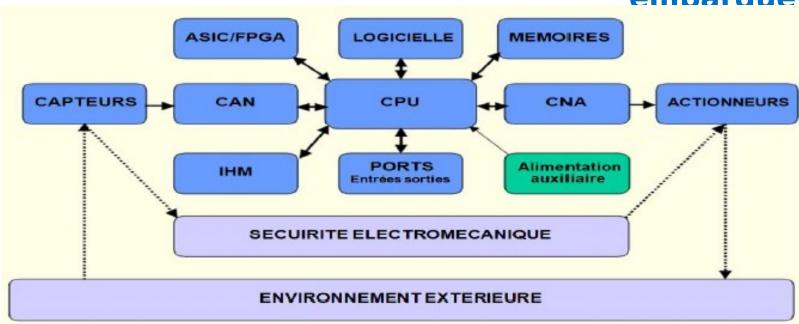
Définition

- Un système embarqué (Embedded System) est défini un système électronique et informatique autonome dédié à une tâche précise, souvent en temps réel, possédant une taille limitée et ayant une consommation énergétique restreinte.
- Les systèmes embarqués sont des systèmes de traitement d'information enfouis dans des produits plus larges [Peter Marwedel]

AUTOMOBILE: 100 PROCESSEURS

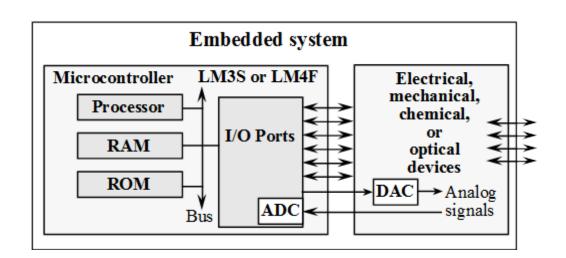


Exemple de prototype d'un Systèmeembarqué





Plus spécifiquement

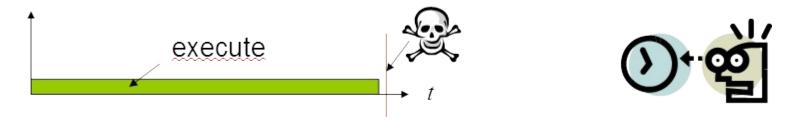


- ۲
 - On y trouve essentiellement:
 - ✓ des capteurs généralement analogiques couplés à des convertisseurs A/N.
 - ✓ des actionneurs généralement analogiques couplés à des convertisseurs N/A.
 - ✓ le calculateur mettant en œuvre un processeur embarqué et ses périphériques d'E/S. Il peut être complété d'un circuit FPGA jouant le rôle de coprocesseur afin de proposer des accélérations matérielles au processeur.
 - Contrairement au PC dans un bureau, le rôle de l'environnement extérieur est un paramètre très important: Variations de la température, Vibrations, chocs, Variations des alimentations, Interférences RF, Corrosion, Eau, feu, radiations.⁹

Contraintes

Temps réel

- Temps réel : ne signifie pas rapide mais plutôt respecter des contraintes temporelles
- un système embarqué doit généralement respecter des contraintes temporelles et l'on y trouve enfoui un système d'exploitation ou un noyau Temps Réel (Real Time Operating System, RTOS).



- Un système temps réel est dit dur (hard real time), si le non respect de la contrainte peut engendrer une catastrophe © provoque une faute
- Dans le cas contraire, le système est dit temps réel **mou** (**soft real time**) © provoque une dégradation des performances.

Basse consommation

- La durée de vie de la batterie est un facteur important pour les systèmes opérant sur batterie
- Ourée de vie de la batterie : dépend de la consommation d'énergie du système; elle est calculée en cycles de charge.
- L'utilisation et la recharge d'une batterie à 100 % de sa capacité correspondent à un cycle de charge complet.
- Énergie ou capacité batterie exprimée en Joules ou Wh
- Énergie massique: par exemple, 720 Joules/gramme pour les batteries Lithium-Ion
- 1 instruction processeur nécessite ~1nJ

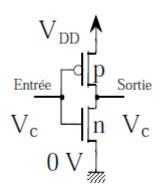
Question: 1 milliard d'instructions exécutées consomment en mg?

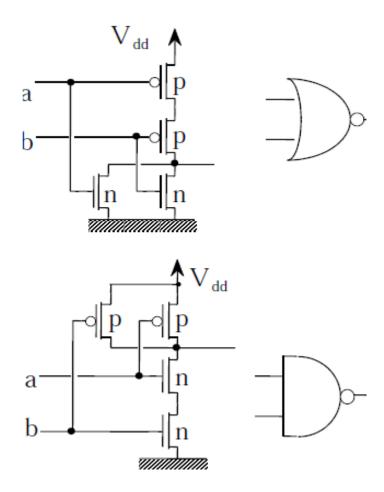
La consommation dépend aussi de la technologie utilisée et du nombre d'éléments logiques intervenants lors de l'ecécution d'un programme



 Exemple: Porte Inverseur avec la technologie MOS (Metal Oxyde Silicium). Elle repose sur des transistors à effet de champs MOSFET.

Porte NOT





۲

□ Trois composantes de la consommation d'une porte logique (inverseur par exemple)

- Consommation dynamique: $P_{dyn} = 1/2 \text{ CV}^2$ (C: capacité de la porte, V: Voltage d'alimentation) ~ 85%
- Consommation statique: $P_{stat} = V I_{fuite} \sim 10\%$
- Consommation de court-circuit: $P_{cc} = K.\tau.(V-2V_{seuil})^3$ avec K: constante technologique, τ : temps de montée descente du signal, V_{seuil} tension seuil

Généralisation simpliste (dynamique)

- Puissance active ~ Activité * 1/2 CV² * Fréquence
- Energie = Temps * Puissance
- Energie ~ Cycles * Activité * ½ CV²

Avec Activité: nombre de porte commutant.

Rq: Cette modélisation n'est pas précise pour un circuit dont le comportement n'est pas stationnaire (ex: processeur)

- Réduction de la puissance active:
- ✓ Réduire l'activité
- √ Réduire la capacité parasite C: plus la technologie est intégrée plus la capacité est diminuée
- √ Réduire le voltage V (Contribution la plus importante)
- **Solutions:** complexité réduite des différents composants (moins de registres, architectures RISC), gestion dynamique de la fréquence d'horloge, gestion dynamique de l'alimentation...

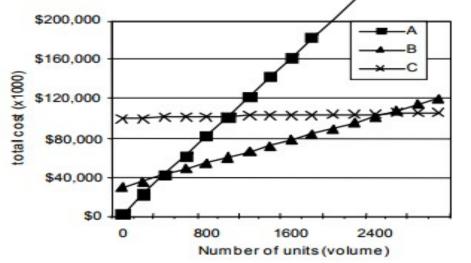
Cout réduit

- Coûts:
- ✓ Coût unitaire : le coût de fabrication chaque unité du système, à l'exclusion des coûts non récurrents
- ✓ (NRE) NRE (Non-Recurring Engineering cost): Le coût de conception du système
- ✓ Coût total = NRE + coût unitaire* Nbr d'unités
- ✓ Coût total par unité = coût total / Nbr d'unités = (NRE / Nbr d'unités) + coût unitaire

Exemple - NRE=\$2000, coût unitaire=\$100

- Pour 10 unités: Coût total = \$2000 + 10*\$100 = \$3000
- Coût total par unité = \$2000/10 + \$100 = \$300L'amortissement du NRE sur toutes les unités ajoute un coût \$200 par unité **15**

- Comparaison de technologies la meilleure dépend de la quantité
- -Technologie A: NRE=\$2,000, coût unitaire=\$100
- Technologie B: NRE=\$30,000, coût unitaire =\$30
- Technologie C: NRE=\$100,000, coût unitaire =\$2



Mais, il faut aussi considérer le temps de mise en marché!

- Le temps c'est de l'argent : Time-to-market
- Le temps de conception a un effet double :
- ✓ Coût non récurrent (homme/année)

Intégrati

- •onLa capacité d'intégration = nombre de transistors par unité de surface.
- Selon la loi de Moore, cette capacité double chaque 18 mois. Toutefois, depuis 2015, cette loi ne se vérifie plus, car les évolutions n'arrivent plus à suivre la même cadence, et sont désormais plus lentes © On s'approche des tailles atomiques.
- VLSI (Very large scale intégration)
- Conséquence: les composants discrets qui étaient connectés sur une carte PCB (Printed circuit board) sont intégrés sur une même puce On parle de SoC (system on

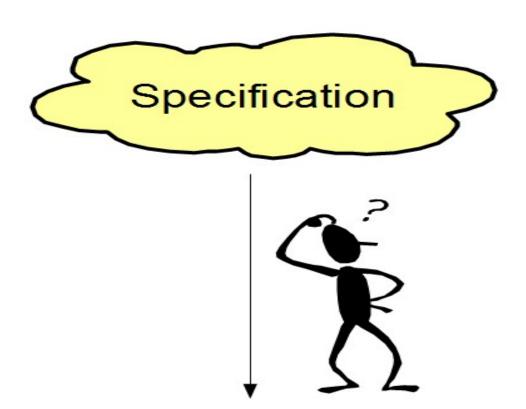


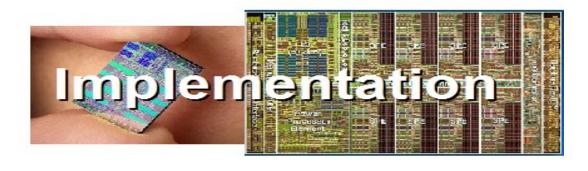
From PCB to SoC

5mm

Flot de conception des SoC







Cible Logicielle

- Ce sont des cibles programmables où on peut modifier l'application juste en modifiant le code sans toucher à l'architecture
- ☐ Les processeurs généralistes (GPP : General Purpose Processor)
- ✓ ARM, MIPS, PowerPC, SH4, Intel
- ✓ Jeu d'instruction :RISC / CISC
- ✓ Architecture : Harvard / Von Neumann
- ☐ Les DSP (Digital Signal Processing)
- ✓ Processeurs dédiés et optimisés pour le traitement numérique du signal (filtrage, extraction de signaux, etc.).
- ✓ Exemple : Texas Instrument C6x
- ☐ Les ASIP : (Application specific processors)
- ✓ Processeurs optimisés pour une certaine application
- √ Générés automatiquement par les outils de compilation
- ✓ Exemple : Tensilica

Les microcontrôleurs

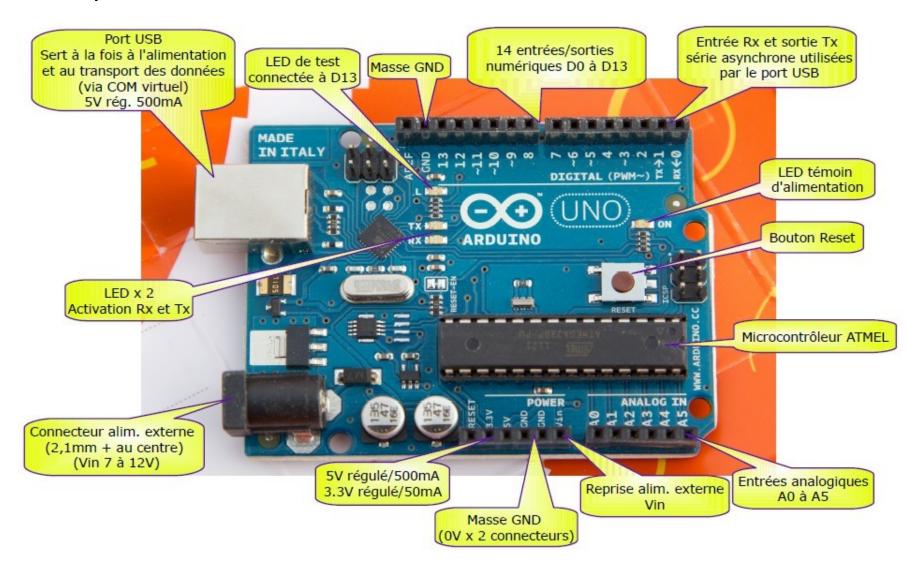
- Les microcontrôleurs: Cas particulier (simple) de SoC
- ✓ Construit autour d'un Processeur GPP et quelques périphériques
- ✓ Convient pour des petites/moyennes applications
- ✓ Performances limitées
- ✓ Basse consommation
- Exemple : STM32 de STmicroelectronic



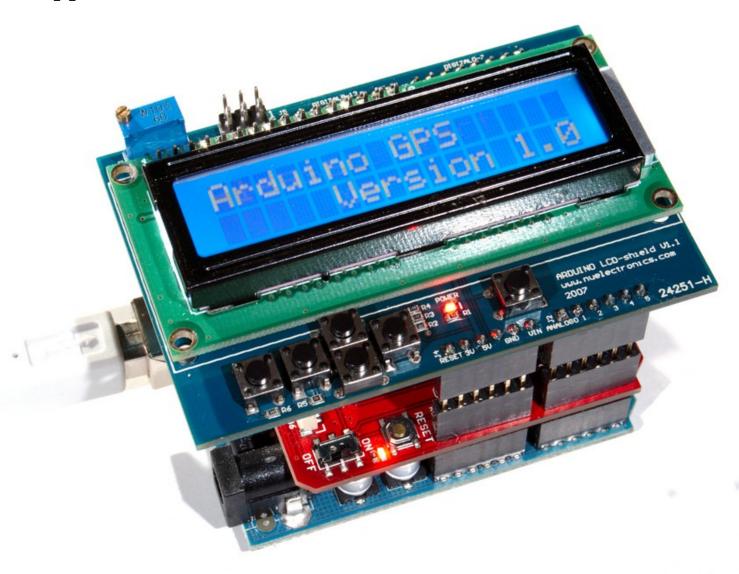
• Exemple : Microcontrôleurs PIC



Exemple 3: cartes Arduino

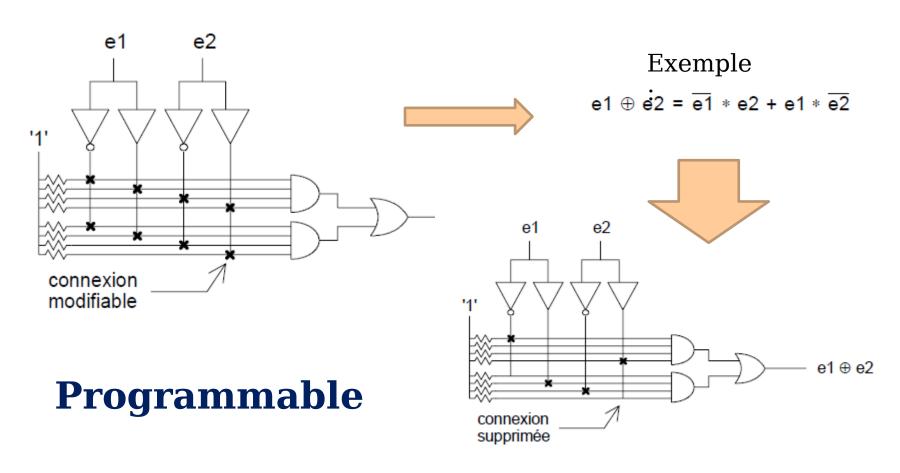


Application sous Arduino



Cible matérielle

Exemple introductif: sous la base que toute fonction logique combinatoire peut se mettre sous forme d'une somme (Ou logique) de produits (ET logique), prenons cas des fonctions à deux variables



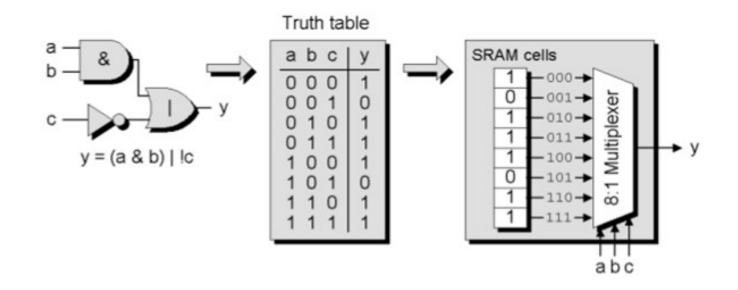


• Un circuit programmable est un assemblage d'opérateurs logiques combinatoires et de bascules dans lequel la fonction réalisée n'est pas fixée lors de la fabrication. Ce qui est programmable dans un circuit concerne donc les interconnexions et les opérateurs.

Idées:

- ✓ Toute fonction logique est une interconnexion de transistors (niveau très bas)
- ✓ Toute fonction logique combinatoire peut se mettre sous forme d'une somme de produits ou forme disjonctive
- ✓ Toute fonction logique combinatoire peut se mettre sous forme d'un produit de sommes ou forme conjonctive
- ✓ Toute fonction logique combinatoire peut être réalisée par un multiplexeur.
- √ Toute fonction logique combinatoire peut être réalisée par un décodeur

Exemple



Cible matérielle Par utilisation des circuits programmables: Ce sont des cibles fixes conçues (avec des langages de description du matériel tels que VHDL, VERILOG) pour réaliser des tâches bien déterminées



Deux grandes familles :

- ☐ Les ASIC : « Application specific integrated circuits »
- ✓ ASIC « Full custom »: Conception au niveau transistor: Effort de conception très important mais performances maximales
- ✓ ASIC « Standart cell » ou semi-custom: utilise des librairies de cellules primitives (AND, OR, registres, SRAM...): Effort de conception réduit, performances souvent proches du fullcustom
- Les circuits reconfigurables: PLA, CPLD (Complex Programmable Logic Device), FPGA

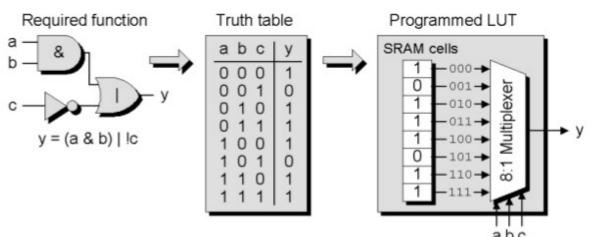
FPGA (Field Programmable Gate Arrays)
Arrangement Matriciel de blocs logiques avec configuration des :

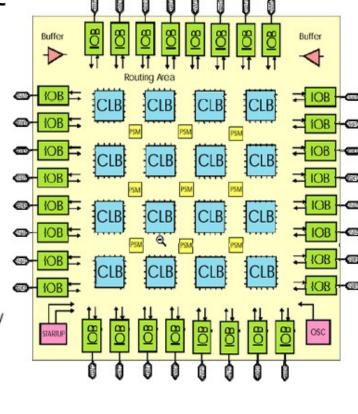
✓ Interconnexions entre les blocs logiques

✓ La fonction de chaque bloc

- CLB: Configurable Logic Bloc

- Élément clé : LUT (lookup table)





Caractéristiques	FPGA	ASIC	
		Semi custom	Full custom
Densité	Faible	Moyenne	Grande
Flexibilité	Grande	Moyenne	Faible
Analogique	Non	Oui	Oui
Rapidité	Faible	Bonne	Très bonne
Temps de conception	Très petit	Moyen	Grand
Coût de conception	Très petit	Moyen	Très grand
Utilisation des outils	Simple	Complexe	Très complexe
Volume de production	Petit	Grand	Grand