

Les circuits

« Les circuits, c'est le grand moteur de l'électronique, de l'informatique. Sans les circuits il n'y aurait pas ! C'est la progression des circuits qui permet le monde numérique. »

G. Berry Collège de France Avril 2008

« Toute ligne de code, écrite avec votre langage de programmation préféré, conçue avec votre paradigme de programmation préféré est exécutée par des opérateurs logiques constitués d'atomes de silicium assemblés dans un micro circuit au cœur d'un système numérique. Ces opérateurs nanoscopiques sont gravés sur le silicium par un processus de fabrication dont les données techniques de réalisations sont-elles mêmes issues d'un flot de conception d'une chaîne logicielle complexe. »

Pour les classes de Terminale et première NSI

Auteur : Patrick Guérangé

Relecteur : André Crépel

Académie de Grenoble

Document version 2.0 relecture OK

P.G le 14 janvier 2021

Sommaire

Introduction	4
Les circuits de leurs genèses à l'IA embarquée.....	5
Les fondements.....	5
Les premières cartes numériques	5
Le calculateur AGC des missions Apollo	6
L'intégration augmente un exemple les unités arithmétiques et logiques.....	8
Le premier microprocesseur : le 4004 d'Intel	9
Une brève description	9
Quelques éléments de ses performances.....	9
Les grandes parties constituantes des microprocesseurs	10
La course à la miniaturisation	11
C'est quoi un transistor ?	11
Du transistor aux portes logiques	11
Des portes logiques au microprocesseur 4004	12
La course à l'intégration	13
La loi de Moore	13
Les limites à l'intégration maximum	14
Les microprocesseurs d'aujourd'hui.....	14
Vers l'intégration du système minimum	14
Un microprocesseur n'est jamais seul	14
Le microcontrôleur : l'intégration du système minimum.....	15
Vers l'intégration maximale d'un système on chip ou SOC.....	16
Le SoC : Système on Chip	16
Toujours plus loin dans l'intégration.....	19
Intégrer des algorithmes d'Intelligence Artificielle.....	19
Des puces multi technologies les systèmes embarqués au CEA	19
La logique programmable.....	20
L'apport de la vitesse et d'une interprétation hardware	20
Une rapidité de conception, une rapidité de mise en œuvre	20
Principe du fonctionnement	20
Fonctionnement simplifié de la logique programmable	21
Une première équation	21
Synthèse de l'additionneur de deux bits	22
Synthèse de l'additionneur de trois bits.....	22
Synthèse d'une UAL en VHDL	23
Le symbole du composant désiré.....	23
Les entrées sorties du circuit	23
L'architecture	23
Les circuits mémoires	24
Présentation.....	24
Les mémoires vives RAM	24
Les mémoires mortes	24
Le compromis Hardware / Software Surface / Temps	25

Exercices avec correction	26
Exercices de logiques.....	26
Quelques règles élémentaires de logique à connaître	26
Mise en équation à partir d'une table de vérité	26
Réalisation d'une cellule d'addition complète, addition avec retenue	27
Représentation des opérateurs	28
Table de vérité d'une fonction NOR ou NON-OU.....	29
A vous de compléter les schémas :	29
Applications des règles de simplification démonstration de propriétés.....	29
Réalisation d'une fonction multiplexeur	30
Exercices structures UAL.....	30
Analyse d'une UAL '381	30
Exercices synthèse de portes logiques	31
L'inverseur	31
Une porte NAND	32
Une porte NOR.....	33
Une porte AND.....	33
La porte NOR de la NASA.....	33
Passage à l'échelle.....	34
Passage à une échelle quelconque un opérateur ET à n entrées noté ET_n	34
Comparaison de deux structures	35



Retour au sommaire

Introduction

Dans cette ressource nous nous intéresserons aux 'circuits intégrés'. Sous ce vocable se cache la réalité matérielle qui permet la mise en œuvre des algorithmes sur des machines, ou systèmes de traitements de l'information.

Le sujet est vaste, technique, et surtout passionnant et toujours en pleine évolution. Nous décrirons quelques pistes pour en découvrir les principaux aspects dans ce document. Notre objectif n'est pas d'y être exhaustif, ce n'est pas possible, mais par différents parcours de proposer au lecteur enseignant puis par ricochet à ses élèves une meilleure compréhension des 'circuits intégrés', de ce qu'ils sont, de ce qu'ils font, de comment ils sont conçus puis de comment ils sont fabriqués.

Les différents chapitres exposeront au travers de cette ressource et de ressources complémentaires jointes en annexes les différents aspects de cet univers technologique. Les activités proposées organisées en petits modules pourront alors facilement être utilisées telles quelles selon les besoins de l'enseignant et de son enseignement dispensé dans les classes de 1^{ère} et T^{le} NSI.

En particulier nous nous proposons d'illustrer les points suivants au cours de nos études :

- Montrer les grands axes de l'intégration des systèmes sur 'puces'.
- Découvrir qu'il existe beaucoup de façons de résoudre un problème particulier qui vont de l'implantation matérielle sur silicium au programme exécuté par un microprocesseur en passant par une implantation en logique programmable.
- Que chacune de ces technologies ou implémentations possède ses avantages et ses inconvénients.
- Qu'il existe une dualité surface / vitesse ou espace / temps autrement dit si on veut calculer vite on augmente alors la surface du circuit utilisé pour résoudre la fonction.
- Qu'il existe une dualité hardware / software, en effet pour résoudre un problème tout n'est pas confié au programme principal mais au contraire une répartition entre une réalisation matérielle et une réalisation logicielle est inévitable et alors où se trouve la frontière ?
- Dans la découverte de la conception des circuits intégrés, numériques, nous retrouverons des principes tel que diviser pour régner qui s'impose y compris au plus bas niveau de la réalisation matérielle des circuits ou puces.

Nous utiliserons le support des conférences et séminaires autour de l'informatique et du monde numérique faites au Collège de France.

Par ailleurs nous restreindrons notre étude aux circuits dits numériques, les circuits traitant de signaux non numériques, c'est-à-dire non binaire, signaux analogiques ne faisant pas partie du programme de NSI. Nous en évoquerons quelques aspects dans la perspective de l'augmentation de l'intégration des SoC avec en exemple les PSoC de la société Cypress. Tendance inéluctable d'essayer de mettre tout le design sur une seule puce dans tous ses aspects : c'est le domaine du Système On Chip.

L'ensemble du thème 'circuit' ne saurait être abordé profondément dans ce document. L'objectif est d'en comprendre l'histoire, les technologies en présence, les liens avec le domaine de l'informatique. Pour chaque thème abordé il sera proposé des exercices qui permettront d'approfondir les notions dans le cadre de devoir à la maison ou d'exercices fait en classe à la discrétion de l'enseignant.



Les circuits de leurs genèses à l'IA embarquée

Nous illustrons dans cette partie quelques grandes étapes technologiques de l'histoire des circuits numériques. Des exercices seront proposés pour approfondir l'apprentissage et la découverte de différents aspects complémentaires.

Les fondements

Les premières cartes numériques

La numérisation de l'information, l'essor de la logique fondée sur une représentation avec deux états différents : vrai / faux, ouvert / fermé, oui / non, 1 / 0 a tout d'abord été réalisée avec des systèmes électromécaniques puis avec des tubes électroniques ou 'lampes'. Ces technologies ont été supplantées complètement par l'invention du transistor à semi-conducteur en 1947 puis par l'implantation de quelques transistors dans un seul circuit appelé 'intégré' en 1958.

À la suite des travaux sur les semi-conducteurs, le **transistor** a été inventé le 23 décembre 1947 par les Américains John Bardeen, William Shockley et Walter Brattain, chercheurs des Laboratoires Bell. Ces chercheurs ont reçu pour cette **invention** le prix Nobel de physique en 1956.



[fr.wikipedia.org > wiki > Transistor](https://fr.wikipedia.org/wiki/Transistor)

[Transistor — Wikipédia](https://fr.wikipedia.org/wiki/Transistor)

Le physicien Jack St. Clair Kilby (1923-2005) inventa le **circuit intégré** en 1958. Le physicien Robert Norton Noyce (1927-1990) l'inventa de son côté, six mois plus tard. Noyce utilisa le silicium comme matériau semi-conducteur, et Kilby le germanium. 18 nov. 2013



[www.futura-sciences.com > Sciences > Dossiers](http://www.futura-sciences.com)

[1958 : le circuit intégré | Dossier - Futura-Sciences](http://www.futura-sciences.com)

[Exercices de logique](#)

Ces premiers circuits, qui ne contiennent au départ que quelques portes logiques, sont assemblés ensembles sur des cartes électroniques pour réaliser des fonctions plus complexes et répondre aux besoins technologiques du moment. Au fur et à mesure des avancées technologiques l'intégration est de plus en plus grande. Il apparaît alors des grandes lignes de circuits différents et complémentaires à savoir :

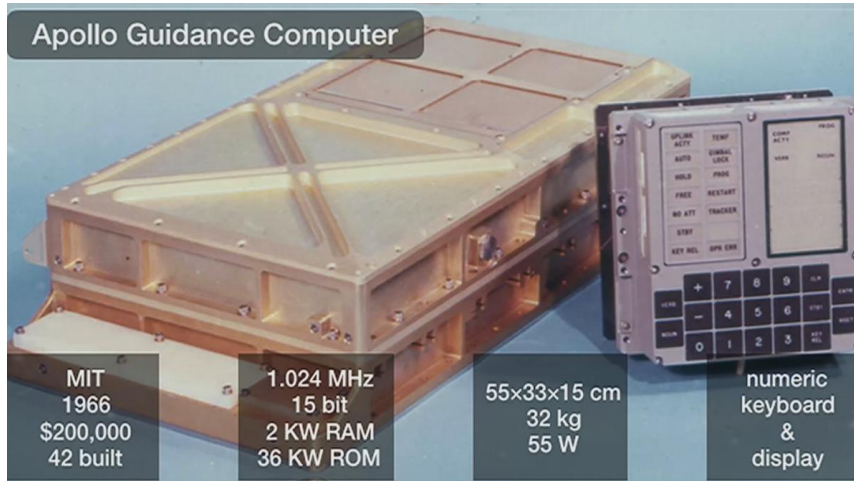
- Les circuits logiques combinatoires ou séquentiels
- Les unités arithmétiques et logiques ou ALU
- Les mémoires vives (volatiles) ou mortes (permanentes)
- Les circuits analogiques



Le calculateur AGC des missions Apollo¹

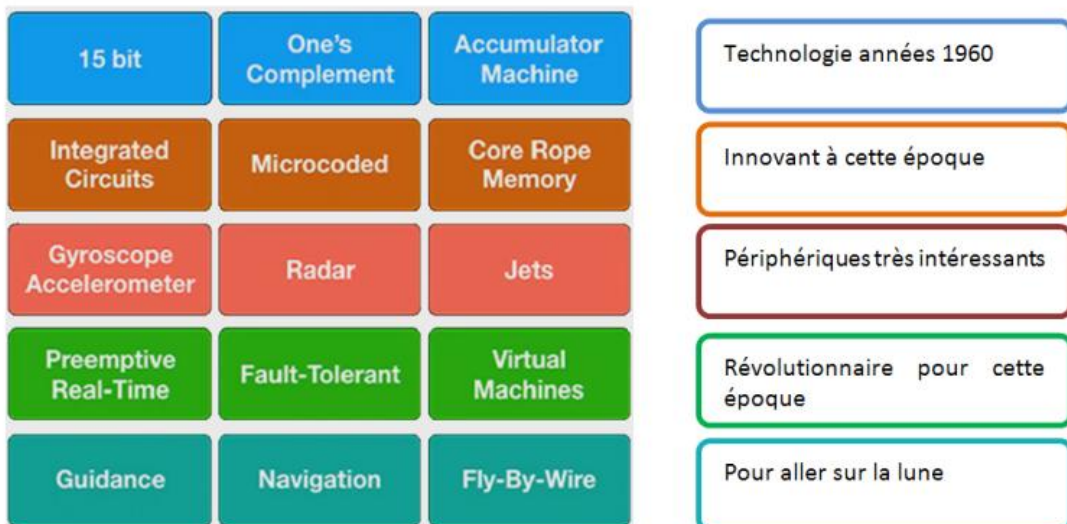
Le calculateur de navigation du programme Apollo ou **A**pollo **G**uidance **C**omputer

Un calculateur complet réalisé uniquement avec des portes NOR à 3 entrées. Le programme *Apollo* a contribué à l'essor de l'informatique : le développement des programmes de navigation et de pilotage des vaisseaux Apollo voit apparaître la scission entre matériel et logiciel. Les méthodes de programmation et de test sont également en partie nées des exigences de fiabilité et de la complexité des logiciels développés pour le programme. Enfin, le projet lance l'utilisation des circuits intégrés qui ont fait leur apparition en 1961. La NASA achète au début du programme 60 % de la production mondiale pour les besoins des ordinateurs des vaisseaux Apollo².



Ce calculateur réalisé uniquement en circuits câblés démarre en 1ms ! La mémoire est réalisée en tores de ferrite entièrement connectées à la main par des ouvrières du textile.

Le système d'exploitation pouvait exécuter six tâches simultanément et gérer l'éjection de tâches moins prioritaires s'il était surchargé. L'AGC a été élaboré sous la supervision de l'équipe de Margaret Hamilton³.



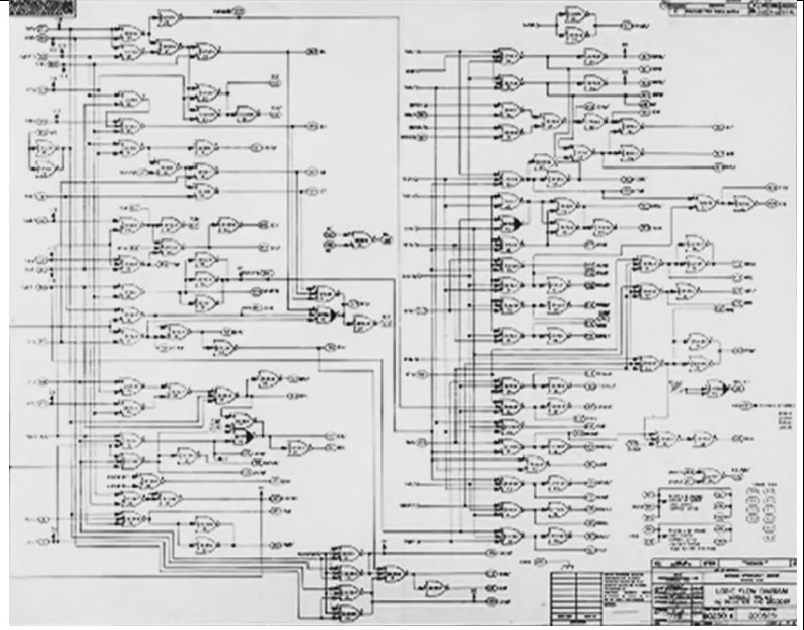
¹ <https://www.youtube.com/watch?v=xx7Lfh5SKUQ> 34C3 - The Ultimate Apollo Guidance Computer Talk, Michael Steil Christian Hessmann, consulté le 30 novembre 2020, réutilisation autorisée.

² https://fr.wikipedia.org/wiki/Programme_Apollo

³ [https://fr.wikipedia.org/wiki/Margaret_Hamilton_\(scientifique\)](https://fr.wikipedia.org/wiki/Margaret_Hamilton_(scientifique))

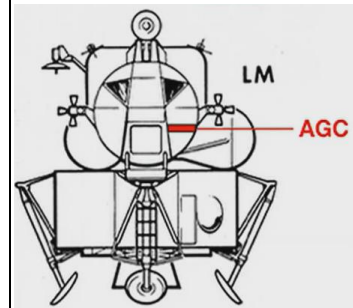
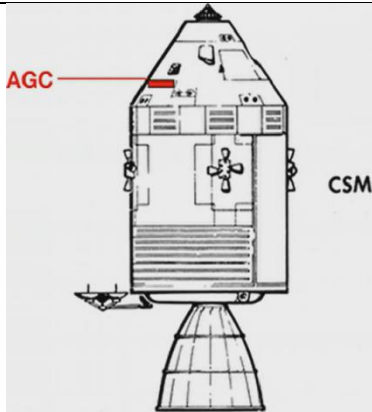
Margaret Hamilton se tenant auprès du code du logiciel de navigation qu'elle et son équipe du MIT Draper Lab ont produit pour le programme Apollo (en 1969).

Une page du schéma (1 / 16) de l'AGC, les circuits NOR y sont bien omniprésents et ont assisté les astronautes dans leurs missions vers la lune.



Module de commande

Le module lunaire



Le premier microprocesseur : le 4004 d'Intel⁴

Cinq années après l'exploit de la NASA le premier microprocesseur apparaît créé par une société dont le nom vous est sûrement connu : Intel.

Une brève description

Le **4004** d'Intel est le premier microprocesseur commercialisé, c'est-à-dire la première intégration réussie de toutes les fonctions d'un processeur sur un seul et unique circuit intégré.

Il est d'abord produit en exclusivité pour l'industriel qui a commandité son développement, Busicom, en mars 1971. Après avoir fait lever la clause d'exclusivité, Intel annonce sa commercialisation le 15 novembre 1971.

La documentation technique de ce composant est disponible sans difficulté sur internet et peut être qualifiée d'historique. Son analyse va nous fournir beaucoup d'informations intéressantes.



Malgré une intégration plus grande le 4004 ne peut fonctionner seul il est associé au minimum à un circuit contenant le programme en mémoire morte, le 4001 une ROM, et à un circuit assurant la mémoire vive pour stocker les données fugitives non permanentes ainsi qu'une interface vers d'autres éléments, le 4002.

Notez ici que le sigle ROM signifie **R**ead **O**nly **M**emory. Cette mémoire ne peut pas être programmée par l'utilisateur. Il est donc impossible d'en fixer le contenu interne, donc d'y mettre son code programme. Celui-ci ne peut qu'être implanté au moment de la fabrication du circuit. On n'est pas dans une grande souplesse. Cela viendra avec l'évolution des technologies.

Quelques éléments de ses performances

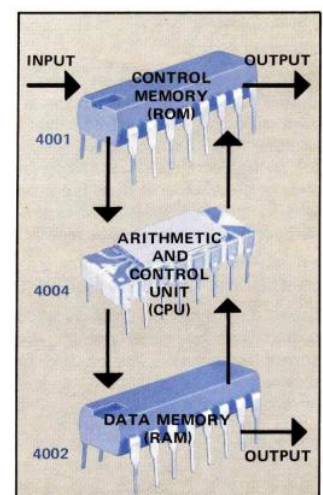
Le résumé de ses performances est indiqué dans la documentation technique. Citons :

- Microprocesseur 4 bits.
- Un jeu de 45 instructions différentes.
- Calcul en binaire et en décimal.
- Additionne deux nombres de 8 chiffres en 850 μ s
- Un temps de cycle de 10,8 μ s

Si nous comparons avec les éléments fournis dans la vidéo de présentation de l'AGC de la NASA nous y trouvons que celui-ci réalise une addition en 22,7 μ s. En simplifiant nous observons que la logique en opérateurs câblés est plus rapide qu'une logique microprogrammée.

C'est pourtant celle-là qui va s'imposer pour plusieurs raisons : l'intégration toujours plus grande élargira à l'infini les possibilités du couple matériel / logiciel, les vitesses de calcul sont toujours de plus en plus grandes, avec cependant des problèmes d'échauffement toujours plus compliqués à gérer.

Mais à technologies similaires, une logique spécialisée réalisée directement gravée sur une puce, ou bien dans des puces configurables, présentées plus loin seront toujours beaucoup plus rapide qu'une logique à base du couple logiciel / matériel.

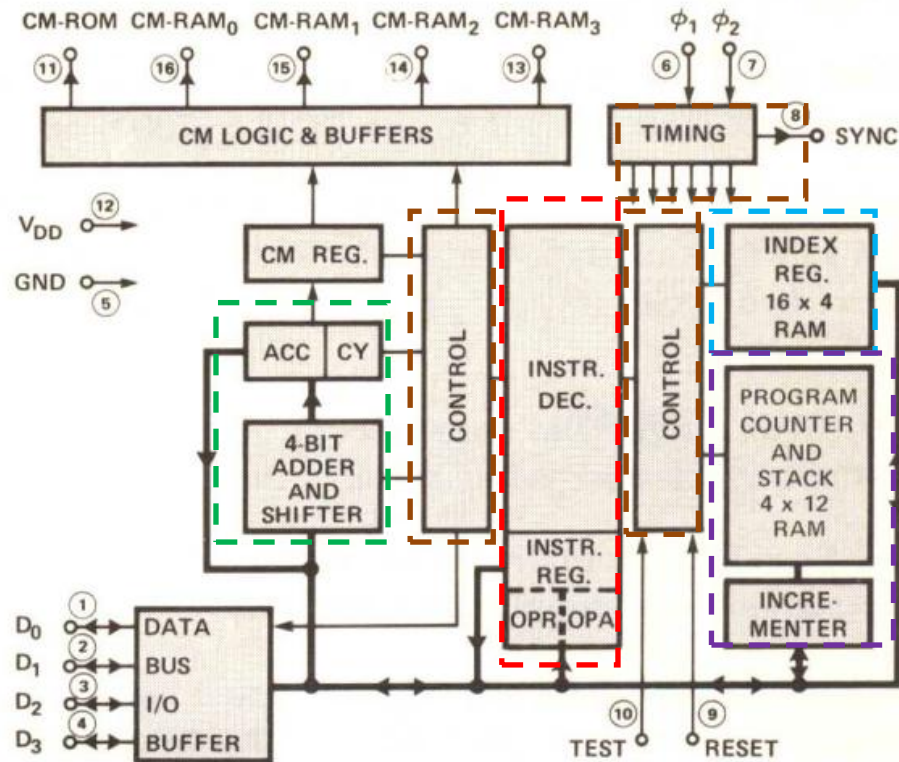


⁴ https://fr.wikipedia.org/wiki/Intel_4004

Les grandes parties constituant des microprocesseurs

Le 4004 est encore 'à échelle humaine'. Nous pouvons donc y retrouver les principaux constituants des microprocesseurs :

- Un décodeur d'instruction.
- Une UAL.
- Des registres de mémoires vives.
- Un compteur de programme Program Counter ou PC.
- La gestion des horloges internes le contrôle du séquençage des opérations.



Le décodeur d'instruction : celui-ci analyse le code de l'opération lu dans la mémoire ROM du programme. Ce code binaire après analyse permettra de piloter le fonctionnement de la puce pour exécuter l'instruction.

L'UAL : déjà étudiée réalise les opérations logiques et arithmétiques prévues par le jeu d'instruction du 4004.

Les registres de mémoires vives : ils stockent les résultats des calculs intermédiaires sans les stocker en mémoire externe donc plus rapidement.

Le compteur de programme : il indique quelle instruction du programme est à exécuter. Il indique donc une adresse dans la mémoire programme.

Le contrôle : il gère le bon ordonnancement des opérations à partir des cycles des signaux d'horloge.

Point d'étape numéro 2

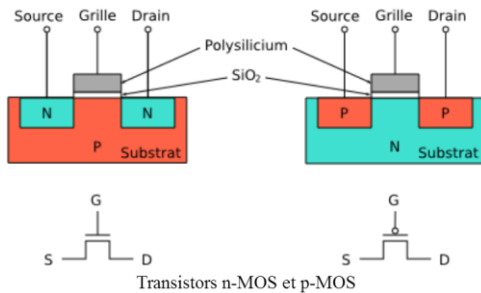
1971 Les microprocesseurs sont nés. Commence alors une course à l'intégration : concevoir et fabriquer des circuits de plus en plus petits contenant de plus en plus de transistors. Cette course est toujours d'actualité aujourd'hui et n'est rendu possible que par la conception assisté par ordinateur. L'heure n'est plus où l'ingénieur rectifiait 'à la main' sur son logiciel de CAO la géométrie des transistors de ces circuits.

« L'informatique est une science qui s'invente elle-même » G.Berry Collège de France.



La course à la miniaturisation

C'est quoi un transistor ?⁵



Les indications N et P désignent des couches ou des zones de silicium qui sont artificiellement modifiées ou 'dopées'.

La structure atomique cristalline de l'atome de silicium est 'dérangée' par l'ajout d'impureté, c'est le terme consacré.

L'équilibre atomique est ainsi modifié et on a au final un matériau possédant un dopage avec excès d'électrons le dopage N, ou bien un défaut d'électrons le dopage P.

Sur ces matériaux est fondée toute l'industrie des composants électroniques.

Il existe plusieurs technologies de transistors, ceux utilisés dans les circuits intégrés sont de type MOS. Le fonctionnement peut être comparé à un interrupteur commandé. Le signal appliqué sur la borne G (grille) commande la fermeture de la liaison D (drain) et Source (Source).

Il y a deux modèles de transistors qui sont en quelque sorte 'miroir' l'un par rapport à l'autre : les transistors de type p-mos et n-mos. Vous pouvez constater sur le dessin de principe d'une coupe du composant que la nature du substrat est particulière pour chaque type : substrat p pour le n-mos et substrat n pour le p-mos.

Le tableau ci-dessous présente le fonctionnement du transistor, fonctionnement simplifié mais suffisant pour comprendre les principes de la conception des portes logiques. A partir d'un niveau logique haut noté 1 (présence de la tension d'alimentation de 5V) ou d'un niveau logique bas noté 0 (présence du 0V de l'alimentation) on réalise la commande des deux transistors MOS. Ils sont complémentaires le n-mos est commandé avec un 1 sur sa borne G et le p-mos est commandé par un 0.

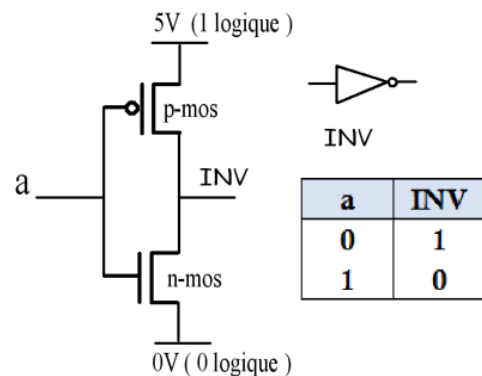
Illustration du fonctionnement des transistors MOS en interrupteurs	
n-mos	p-mos
Illustrations : <i>Transistor MOS - efreidoc.fr</i>	

Du transistor aux portes logiques

L'assemblage de ces transistors permet de réaliser les premières portes intégrées. La réalisation de l'inverseur est très simple puisqu'elle ne met en œuvre que deux transistors.

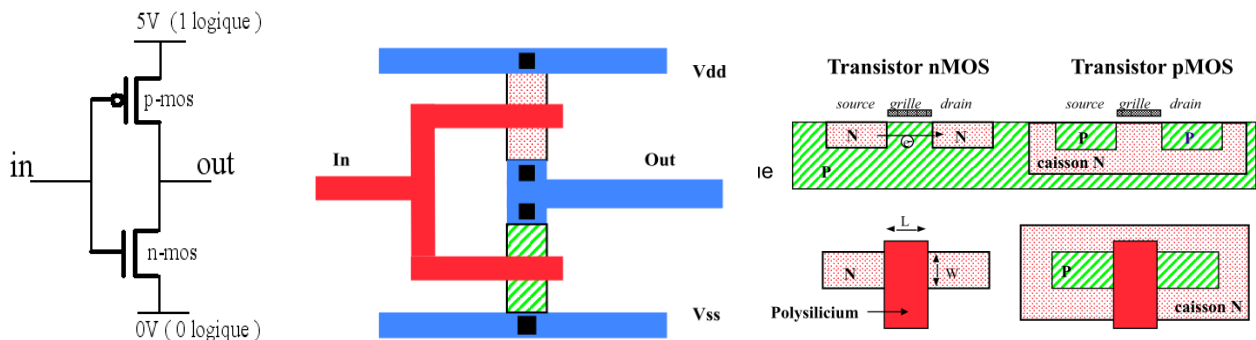
Son fonctionnement sera explicité dans la rubrique exercices.

[Exercices conception de portes logiques](#)



⁵ <https://www.irif.fr/~carton/Enseignement/Architecture/Cours/Gates/> consulté le 3 décembre 2020.

La réalisation matérielle de ces portes est faite avec un empilement de couches successives de silicium dopé N ou P :



Source de l'illustration : https://perso.univ-rennes1.fr/olivier.sentieys/teach/2Cellules_MOS.pdf

Le Collège de France propose depuis 2008 des cycles de conférences sur le thème de la science informatique. Laurent Thenie nous propose dans ce cadre une intervention intitulée 'La course à l'infiniment petit et ses challenges technologiques'.

Dans cette vidéo les enjeux et processus de fabrication des circuits y sont clairement présentés.

Adresse : <https://www.college-de-france.fr/site/gerard-berry/seminar-2008-02-01-11h30.htm>

Auteur : Laurent Thenie

Des portes logiques au microprocesseur 4004

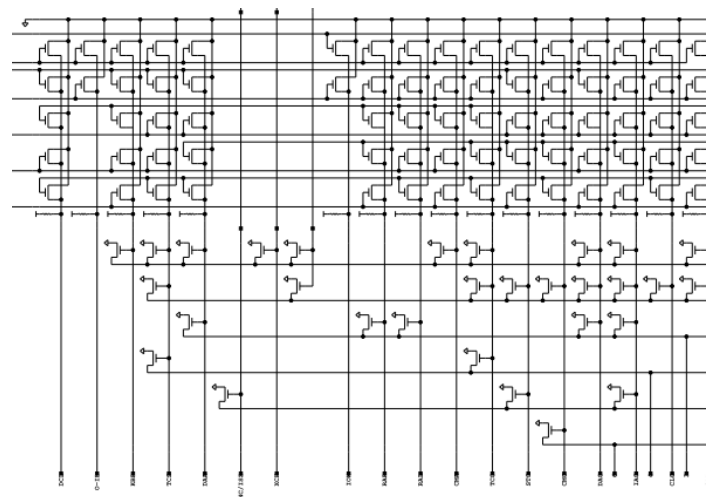
Une vision très partielle du schéma⁶ interne du microprocesseur 4004 est visible sur la figure de droite. Ce microprocesseur contient en fait 2300 transistors. Son schéma tient sur trois pages.

Conçu avec un logiciel de CAO approprié le concepteur peut encore travailler la conception en direct en agissant directement sur les tracés de la métallurgie des transistors.

Ce n'est absolument plus possible actuellement, sur les données de 2008 le processeur PENRYN contient 820 millions de transistors. Sur la base des éléments du 4004 son schéma interne est de l'ordre de 1 million de pages si l'on représente le tracé élémentaire de chaque transistor.

Ce n'est plus possible aujourd'hui, de la même manière que pour le passage d'un langage compilé vers le code exécuté sur une puce, le passage du design du processeur vers les fichiers de production pilotant les machines de fabrication du fondeur* se fait avec des outils informatiques.

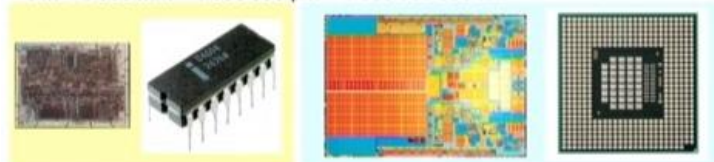
Les implantations 'on chip' ou les compilations de codes sont optimisées par des algorithmes d'autant plus performant que les problèmes à résoudre sont très complexes. Nous en avons ici un bref aperçu.



Comparaison de 2 microprocesseurs INTEL

	Date	Horloge (kHz)	Alim(v)	Taille transistor (nm)	Nombre transistors	Boitier (pattes)
4004	1971	108	15	10 000	2 300	16
PENRYN	2007	3 000 000	0.8	45	820 000 000	603

1 m = 1 000 mm = 1 000 000 μm = 1 000 000 000 nm



⁶ <https://4004.com/assets/redrawn-4004-schematics-2006-11-12.pdf>

La course à l'intégration

L'histoire des microprocesseurs est intimement liée à celle de la technologie des semi-conducteurs et à l'intégration toujours de plus en plus poussée de ceux-ci dans les circuits dont voici les principales étapes⁷ :

Nom	Signification	Année de sortie	Nombre de transistor	Nombre de portes logiques par boîtier
SSI	<i>small-scale integration</i>	1964	1 à 10	1 à 12
MSI	<i>medium-scale integration</i>	1968	10 à 500	13 à 99
LSI	<i>large-scale integration</i>	1971	500 à 20 000	100 à 9 999
VLSI	<i>very large-scale integration</i>	1980	20 000 à 1 000 000	10 000 à 99 999
ULSI	<i>ultra large-scale integration</i>	1984	1 000 000 et plus	100 000 et plus

Pour comprendre le processus de fabrication voir la vidéo ci-dessous :

https://www.youtube.com/watch?v=NFr-WyytNfo&feature=emb_logo

Le procédé de fabrication des circuits, organisé en traitement de couches successives, est spécifié en fonction de la qualité de la gravure : c'est-à-dire de la finesse du plus petit motif qu'il est possible de graver sur le silicium.

Voilà quelques ordres de grandeurs industriels :

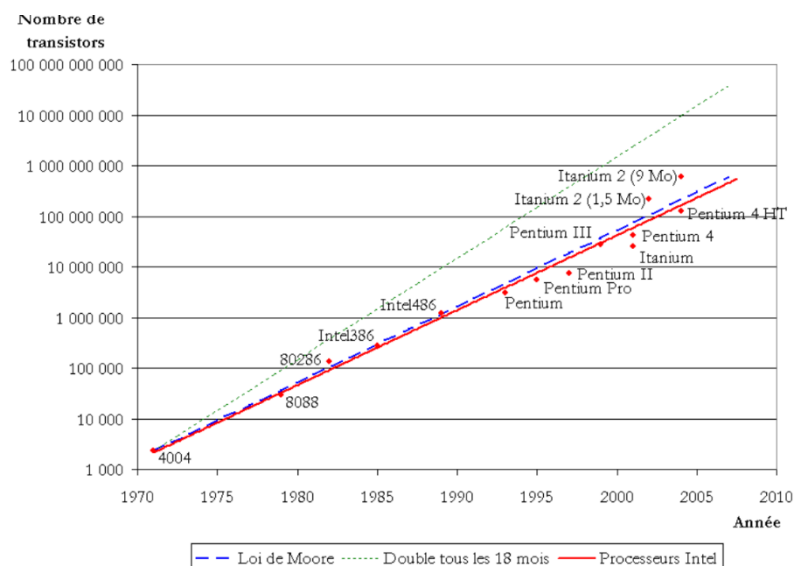
Année	Finesse de la gravure	
2004	130 nm	90 nm
2006	60 nm	30 nm
2015	14 nm	
2018	10 nm	

La loi de Moore⁸

En 1975, Gordon E. Moore, l'un des trois fondateurs d'Intel, posa que le nombre de transistors des microprocesseurs (et non plus de simples circuits intégrés moins complexes) sur une puce de silicium double tous les deux ans.

Bien qu'il ne s'agisse pas d'une loi physique mais seulement d'une extrapolation empirique, cette prédiction s'est révélée étonnamment exacte. Entre 1971 et 2001, la densité des transistors a doublé chaque 1,96 année.

En conséquence, les machines électroniques sont devenues de plus en plus petites et de moins en moins coûteuses tout en devenant de plus en plus rapides et puissantes.



⁷ https://fr.wikipedia.org/wiki/Circuit_int%C3%A9gr%C3%A9

⁸ https://fr.wikipedia.org/wiki/Loi_de_Moore

Les limites à l'intégration maximum

Nous pouvons citer quelques limites à une intégration toujours plus poussée. Tout d'abord aux plus fortes intégrations on approche des limites des dimensions des atomes. Ensuite plus les transistors sont nombreux plus la dissipation thermique est importante, voire rédhibitoire.

Un autre grand facteur de limitation vers le chemin d'une intégration toujours plus poussée est le coût de l'outil de production. Celui-ci toujours de plus en plus élevé n'est accessible qu'à quelques fondeurs sur la planète.

Les microprocesseurs d'aujourd'hui

Pour comprendre l'architecture des microprocesseurs d'aujourd'hui vous pouvez lire l'excellent article de M. PHILIPPE PRADOS [Consultant – OCTO Technology] intitulé : Au cœur des microprocesseurs. Article paru dans la revue GNU Linux au numéro 218.

Apple II 1977⁹



Point d'étape numéro 3

Les microprocesseurs envahissent notre vie quotidienne. Nous pensons tout d'abord à ceux présents dans les ordinateurs qui deviennent progressivement des ordinateurs personnels à la fin des années 70.

Mais une autre évolution bien plus importante mais plus invisible est celle des systèmes embarqués ou enfouis. Tous les objets du quotidien sont progressivement pilotés par des puces. La course à l'intégration prend alors une autre direction : celle de mettre le microprocesseur et tous les circuits périphériques qui lui permettent d'accéder au monde extérieur dans une seule puce.

C'est l'heure des microcontrôleurs.

Vers l'intégration du système minimum

Un microprocesseur n'est jamais seul

Aucun microprocesseur ne peut fonctionner seul. Ces circuits ne contiennent que quelques registres de mémoires en interne pour y stocker les résultats immédiats des calculs. De plus ils ne possèdent aucun accès vers l'extérieur mis à part l'accès à leur espace d'adresses mémoires.

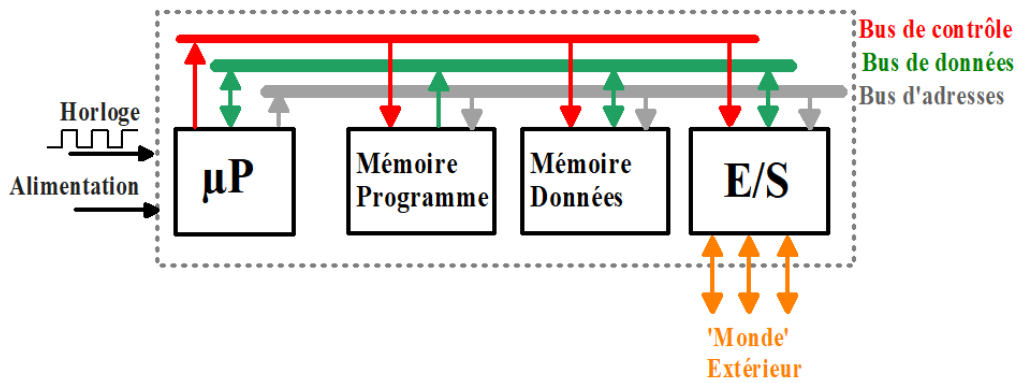
Pour pouvoir fonctionner cet espace d'adressage est rempli avec d'autres circuits : des mémoires vives pour les données fugitives et mortes (au sens de permanentes même après la mise hors tension du circuit) pour le programme, des gestionnaires de périphériques d'entrées sorties, des horloges, pour réaliser ce qu'on appelle un système minimum.

La mémoire de données et la mémoire de programme sont dans le même espace mémoire du microprocesseur. C'est l'architecture de Von Neumann. Le microprocesseur contrôle l'ensemble du fonctionnement et l'échange des informations en utilisant des liaisons appelées BUS.

- Le bus de contrôle : il contient tous les signaux nécessaires au fonctionnement des échanges entre les circuits.
- Le bus d'adresses : il permet au microprocesseur d'envoyer une adresse pour rechercher une information dans l'un des circuits connectés dans son espace mémoire. Cette information peut-être une micro instruction de la mémoire programme, une donnée dans la mémoire de données ou bien une donnée du système de gestion des entrées sorties E/S.
- Le bus de données : il permet le transit de l'information entre le circuit adressé via le bus d'adresses et le microprocesseur.



⁹ <https://www.system-cfg.com/detail.php?ident=96> Reproduit avec autorisation.



Ce système minimum comme nous l'avons vu avec le processeur 4004 peut être réalisé en interconnectant plusieurs circuits sur une même carte. Les progrès de l'intégration permettent d'intégrer l'ensemble de ce système minimum sur une seule puce qui s'appelle un microcontrôleur.

Le microcontrôleur : l'intégration du système minimum

Le microcontrôleur est l'intégration du système minimum sur une seule puce, dans un seul circuit intégré. Les conséquences immédiates sont :

- La réduction de la taille des circuits imprimés sur lesquels sont interconnectés les différents composants d'un système.
- La réduction des coûts de fabrication des cartes.

Un exemple le block diagramme du microcontrôleur AVR ATMEGA328

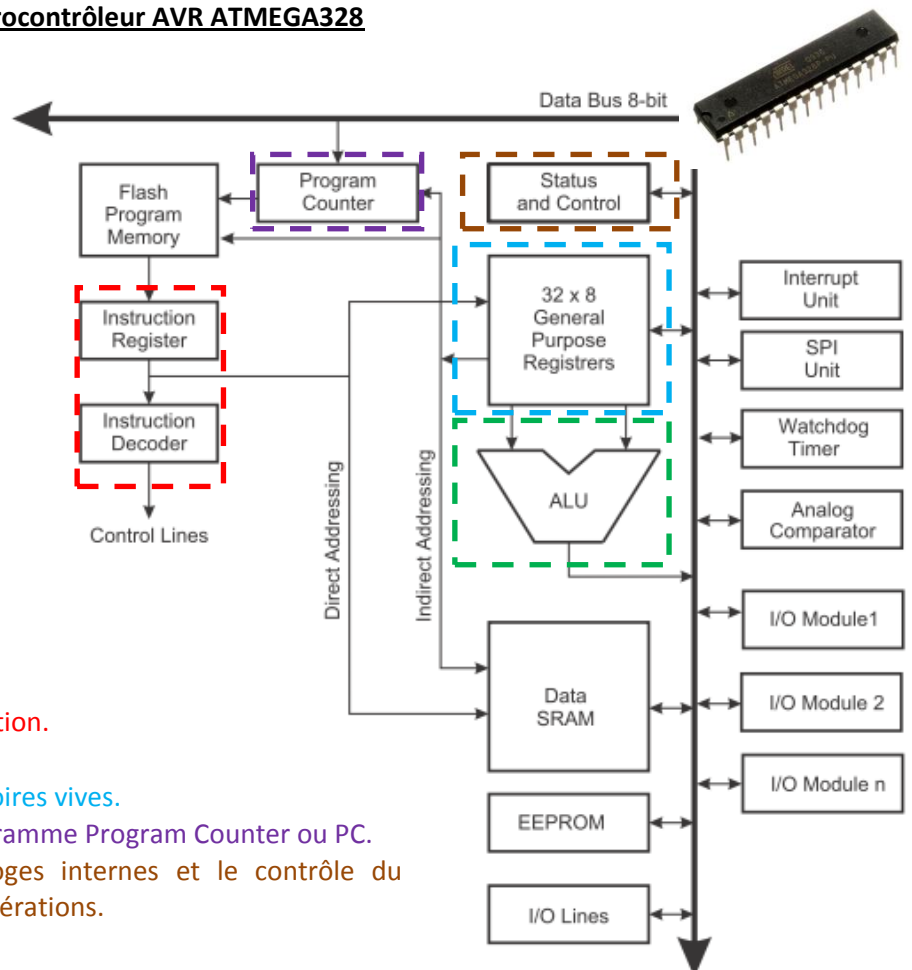
Ce microcontrôleur 8 bits possède 32 registres 8 bits d'usage général ce qui facilite la compilation des programmes écrits en langage C.

La plupart de ses instructions sont exécutées en un seul cycle d'horloge. Il possède une mémoire FLASH¹⁰ (EEPROM) pour le programme entre 4ko et 32ko¹¹.

Une mémoire FLASH permet la sauvegarde de données utilisateur.

En reprenant le découpage fonctionnel du processeur 4004 (p.11) on voit bien toutes les parties intégrées supplémentaires dans ce microcontrôleur.

- Le décodeur d'instruction.
- L'UAL.
- Les registres de mémoires vives.
- Un compteur de programme Program Counter ou PC.
- La gestion des horloges internes et le contrôle du séquençement des opérations.



¹⁰ Les sigles concernant les mémoires sont expliqués dans un paragraphe particulier.

¹¹ Pour donner une idée des possibilités un programme écrit en langage Pascal Multitâches pour microcontrôleur AVR comprenant au final 16 pages de listing occupe 16 ko de code une fois compilé.

- **High Endurance Non-volatile Memory Segments**
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48P/88P/168P/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48P/88P/168P/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48P/88P/168P/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
- **High Performance, Low Power AVR® 8-Bit Microcontroller**
- **Advanced RISC Architecture**
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier



Point d'étape numéro 4

Les familles des microcontrôleurs se développent intensivement, fondée sur leurs racines les microprocesseurs. Les catalogues des fabricants s'enrichissent de multiples déclinaisons avec dans chacune des familles plus ou moins de mémoire, plus ou moins de périphériques intégrés...

Néanmoins, mis à part la gestion de l'alimentation et les convertisseurs analogiques-numériques et/ou numériques-analogiques parfois présents, ces puces restent de nature numériques.

La prochaine évolution dans la ligne de l'intégration maximale est l'incorporation de multiples technologies toujours à l'intérieur d'une seule puce.

Un circuit de gestion d'un téléphone mobile intégrant les aspects analogiques avec le traitement de la voix, le codage et le cryptage réalisés en numérique, la gestion de l'écran, mais également la liaison radio et l'accès au réseau GSM tout cela sur une seule puce, ou bien un circuit de télé HD unique dans une télévision ce n'est plus impossible.

Nous y sommes ce sont les circuits SoC ou PSoC.

Vers l'intégration maximale d'un système on chip ou SOC

Le SoC : Système on Chip

Pour simplifier les coûts les concepteurs de circuits électroniques recherchent en permanence l'intégration maximale. L'objectif est d'implanter toutes les fonctions d'un système sur une seule puce. La difficulté provient alors de la diversité des technologies qui doivent cohabiter, sans interférer les unes avec les autres, sur la même puce.

Ces technologies peuvent être (liste non exhaustive) :

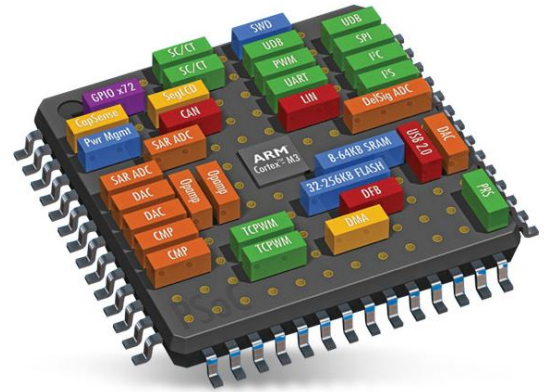
- Des fonctions d'électronique analogique
- Des fonctions d'électronique de puissance
- Des fonctions de communication, d'accès aux réseaux, au Bluetooth,
- Accès à l'IoT Internet des Objets.

Ces systèmes se rencontrent dans l'informatique embarquée, embedded, c'est-à-dire dans les systèmes de traitement de l'information qui agissent en toute autonomie, enfouies à l'intérieur de nos objets du quotidien. La possibilité d'avoir une puce versatile dans tous les domaines diminue la taille des systèmes, augmente l'intégration des cartes ainsi que la fiabilité. Toutes les fonctions usuellement rencontrées dans ces circuits sont implantées autour d'un cœur 32-bit ARM Cortex-M3 et sont ainsi entièrement pilotable par le programme implanté sur la puce.

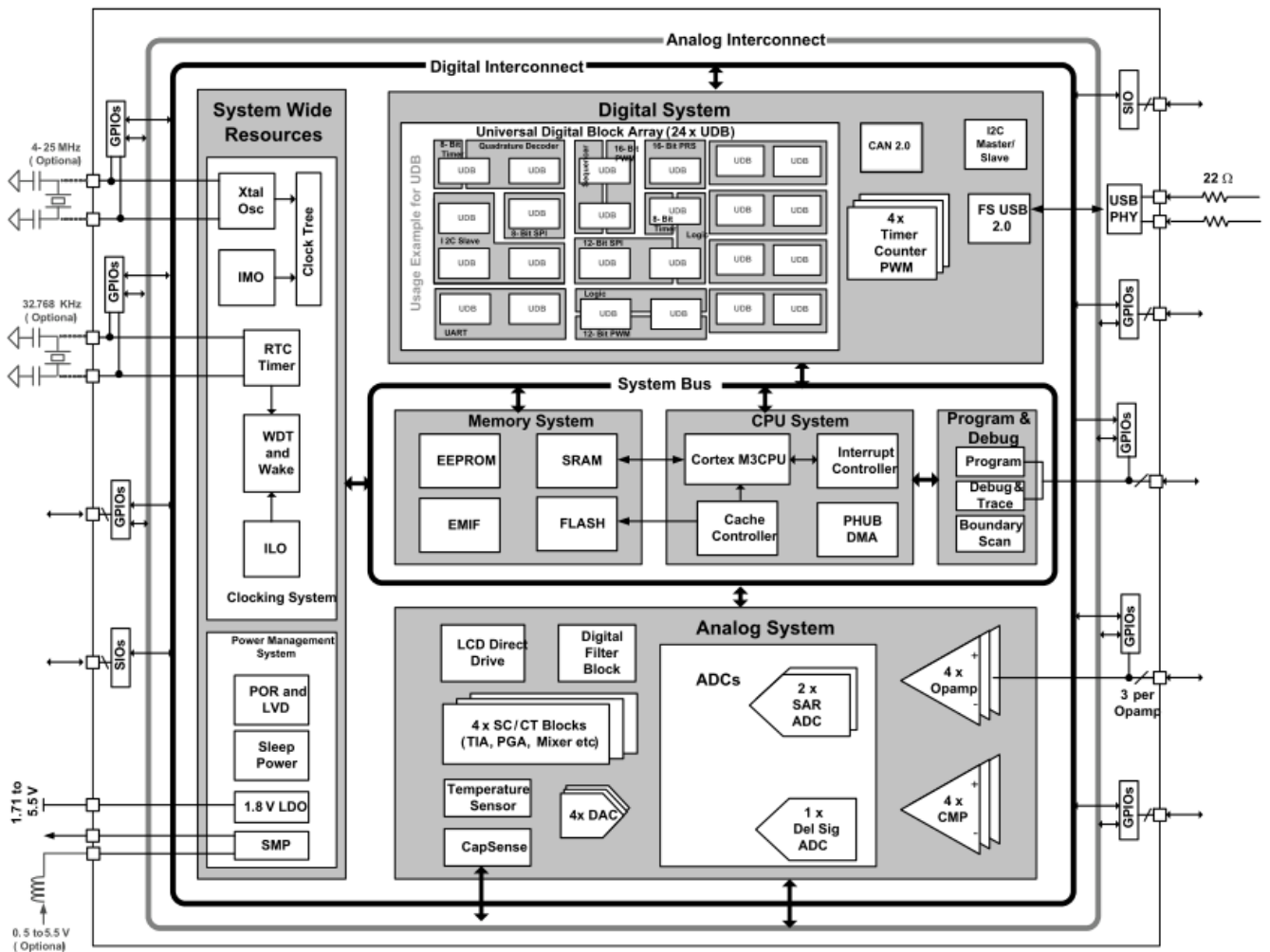


Voici quelques fonctions parmi les centaines implantables :

- Une commande directe d'afficheur LCD 'customisés' sans circuits additionnels
- Une détection d'appui sur un clavier statique fonctionnant par effet capacitif
- Un bloc de filtrage numérique
- Des ressources analogiques, convertisseurs CAN, CNA, comparateur de tensions
- Des blocs de logiques programmables¹² permettant de configurer en 'hardware' des fonctions numériques additionnelles complexes
- Une gestion avancée de l'alimentation fonctionnant entre 0,5V et 5V
- Un routage complet en interne des broches d'entrées sorties
-



PSoC 5LP Cypress

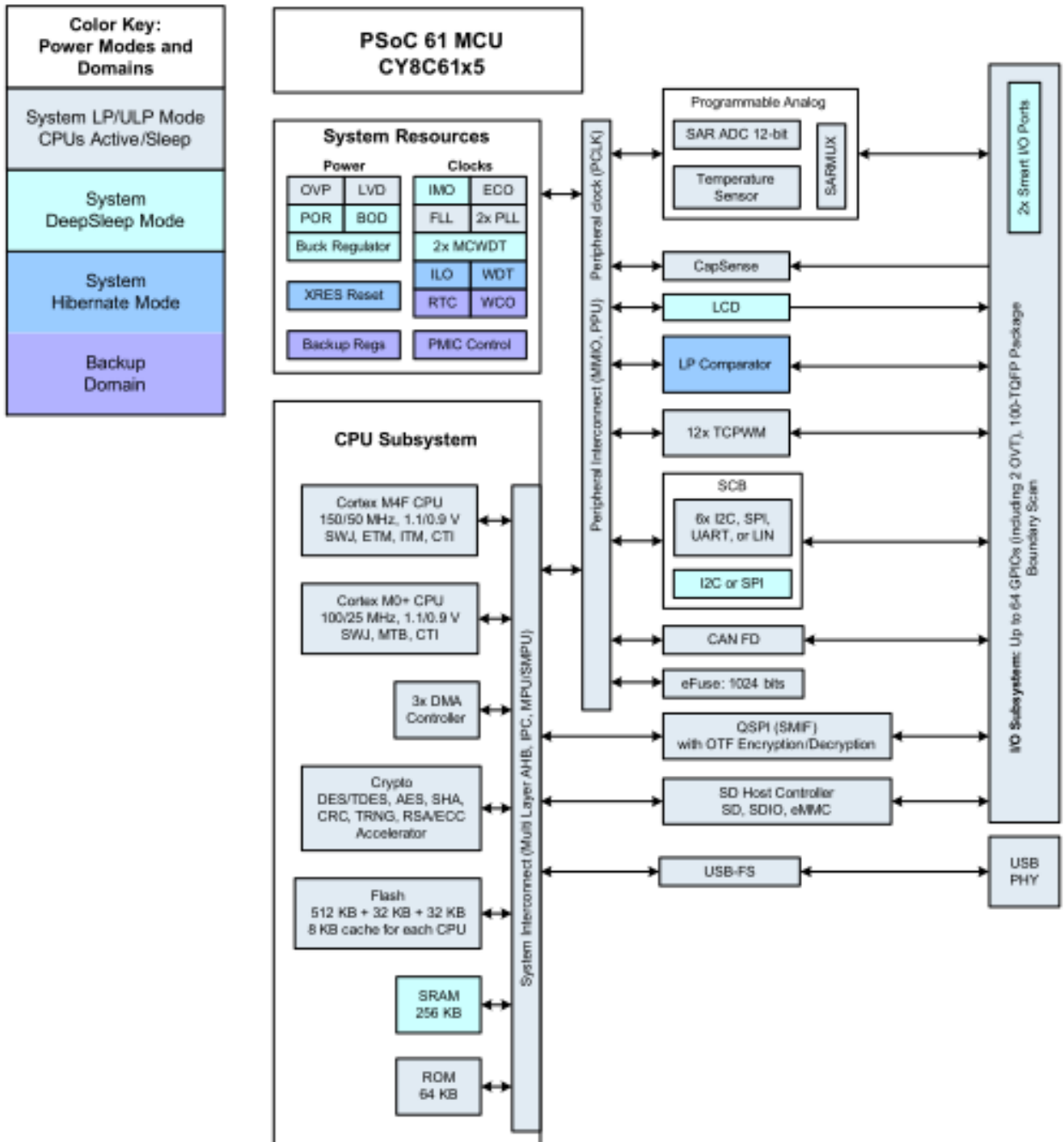


¹² Les principes de la logique programmable seront expliqués plus bas.

Les enjeux d'aujourd'hui tournés vers le développement de l'internet des objets, ou IoT Internet of Things, demandent une intégration poussée doublée d'une sécurité renforcée dans sa mise en œuvre. Un exemple de réponse dans le domaine des circuits est montré ci-dessous avec la famille PSoC6 de Cypress.

Cette puce contient deux cœurs ARM différents, une unité de cryptage symétrique AES 128-256 bits ou asymétrique RSA, un générateur de nombre aléatoire qui est fondamental pour les algorithmes cryptographique. Ce générateur, basé sur un phénomène physique aléatoire génère un vrai bruit numérique (True random number generation TRNG).

LE PSoC 6 de Cypress



Toujours plus loin dans l'intégration

Intégrer des algorithmes d'Intelligence Artificielle

La tendance aujourd'hui consiste à réaliser les calculs au plus proche des capteurs. Des algorithmes comme les réseaux de neurones sont intégrés directement dans des puces programmables de type FPGA. Ces réseaux sont préalablement entraînés.



Key features

- ECP5 FPGA consuming under 1 W of power consumption
- Flexible video connectivity with support for MIPI CSI-2, eDP, HDMI, GigE Vision, USB 3.0, and more

Des puces multi technologies les systèmes embarqués au CEA¹³

Les systèmes embarqués confèrent des fonctions « intelligentes » à de nombreux produits de notre quotidien : dispositifs mobiles, appareils ménagers, automobiles, stimulateurs cardiaques et caméras intelligentes, ...

Dans un avenir proche, ils devraient jouer un rôle de premier plan dans l'Internet des objets, qui permettra à des milliards d'objets dans le monde de communiquer entre eux, sans fil, via Internet. Le défi consiste aujourd'hui à intégrer les systèmes embarqués dédiés sur une seule puce assurant la connectivité, le traitement des données, la gestion de la consommation d'énergie et la génération de signaux.

Pour relever ce défi, le CEA a créé la plateforme Conception et systèmes embarqués.

Avec ses 3000 m2 de locaux, ses 280 collaborateurs et plusieurs dizaines de millions d'euros d'équipements en CAO, la plateforme Conception et systèmes embarqués n'a pas d'équivalent dans la recherche française. Elle propose à ses partenaires industriels (STMicroelectronics, EADS, Schneider Electric, Delphi, PME...) de réaliser des circuits intégrés spécifiques à forte valeur ajoutée : architecture en rupture, miniaturisation poussée, faible consommation, performances de haut niveau...

Les circuits intégrés sont étudiés sur place, des spécifications de départ aux tests finaux sur prototypes. Une trentaine de brevets est déposée chaque année pour protéger leur conception et leur fabrication.

Les vidéos conception et systèmes embarqués :

1. Introduction
2. Calcul numérique
3. Méthodes de conception
4. Imageurs et capteurs
5. Sans fil



¹³ <https://www.cea.fr/Pages/domaines-recherche/nouvelles-technologies/conception-systemes-embarques.aspx>

La logique programmable

L'apport de la vitesse et d'une interprétation hardware

Les circuits de logique programmable permettent l'implantation d'algorithmes de toutes natures qui sont alors exécutés dans une 'puce' dédiée. Ces algorithmes sont d'abord transformés en schémas logiques par des analyses logicielles spécifiques à partir d'une description dans un langage de haut niveau. Le fonctionnement (traitement des informations) est alors semblable à celui d'une carte électronique (réalisée avec des portes et circuits logiques), et ne s'apparente plus à un fonctionnement programmé.

Cette approche a pour avantage la rapidité du calcul logique dans des circuits logiques interconnectés.

Une rapidité de conception, une rapidité de mise en œuvre

La logique programmable déclinée avec des circuits de différentes tailles permet de répondre rapidement à la conception d'un nouveau design. En effet la réalisation d'un nouveau processeur par les fondeurs¹⁴ est un processus très lourd et très long. Ce processus est incompatible avec certains besoins du marché. De plus elles peuvent être facilement reconfigurées pour s'adapter aux évolutions des besoins et contraintes du marché, le time to market.

Principe du fonctionnement

Ces circuits de logique programmable sont de différentes nature et complexité. Pour en comprendre le fonctionnement nous nous intéresserons à des exemples simples illustrant les principes mis en œuvre, le passage à l'échelle est une affaire de spécialiste.

Rappelons simplement que le point d'entrée pour ce flux de conception est un programme décrivant l'ensemble du projet numérique à implanter, programme écrit à l'aide de langages dédiés de description hardware, VHDL ou VERILOG.

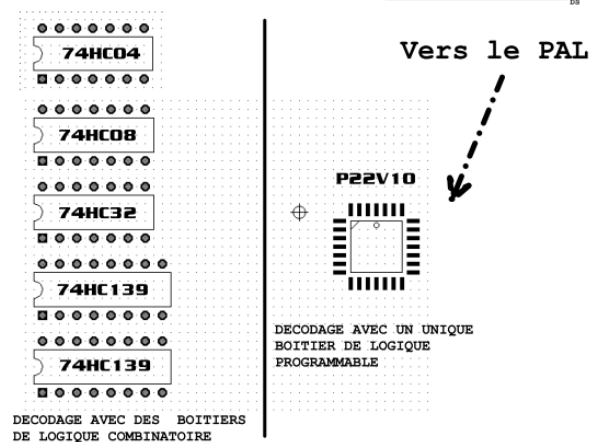
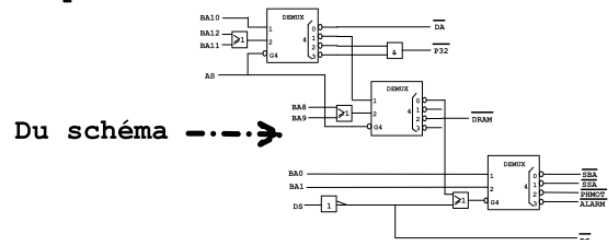
Ces langages proches dans leurs apparences du langage C par exemple sont d'un abord assez simple. Voire même avec certains outils c'est un schéma logique interconnectant des 'boîtes' de haut niveau qui est directement saisi par l'utilisateur puis automatiquement interprété et implanté dans la puce cible.

Dans certains circuits, les PSoC de Cypress, les blocs de logique programmable y sont très puissants et surtout très transparent à l'utilisateur qui peut ainsi augmenter les ressources numériques de son chip selon ses besoins. Ressources réalisées matériellement donc très rapide contrairement à des fonctions émulées par logiciel.

Les FPGA¹⁵ qui sont les plus gros circuits de logique programmable permettent de synthétiser des cœurs de microprocesseurs. Nous avons vu plus haut dans ce document l'intégration de réseaux de neurones dans des circuits de ce type.



Synthèse d'un schéma logique complexe :



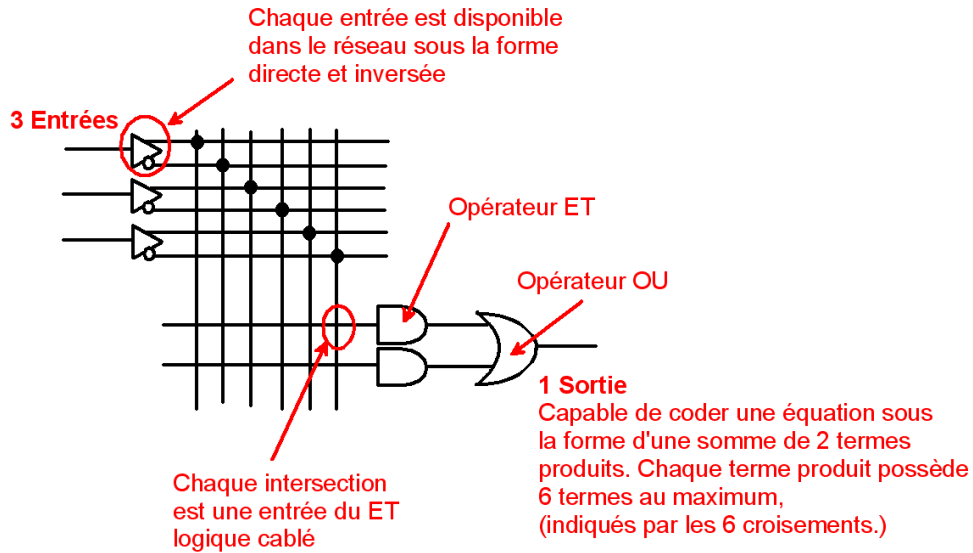
¹⁴ On appelle les fabricants de circuits électroniques des fondeurs

¹⁵ https://fr.wikipedia.org/wiki/Circuit_logique_programmable

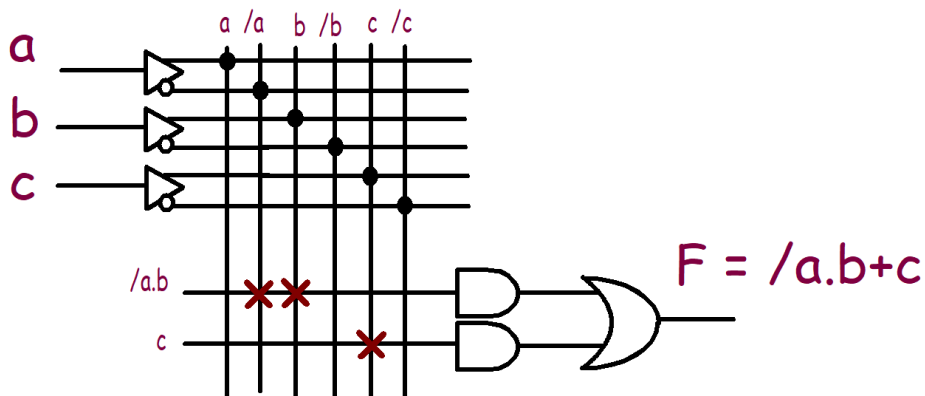
Fonctionnement simplifié de la logique programmable

Pour certaines familles de composants programmables le principe de fonctionnement est basé sur le fait qu'à l'intérieur du composant existe tout un ensemble de liaison configurable par le logiciel de programmation de la puce. Les liaisons sont détruites de manière à réaliser dans un canevas préfixé les équations logiques que l'on souhaite implanter.

La liste des liaisons à détruire est obtenue par compilation du code décrivant la ou les fonctions que l'on souhaite réaliser.



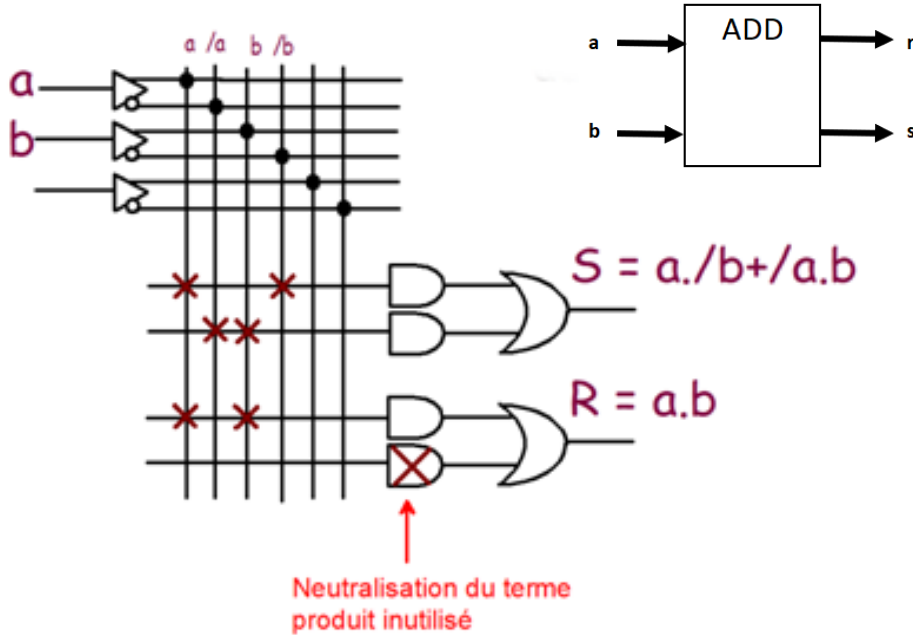
Une première équation



- ✗ La croix indique que la liaison ligne / colonne est conservée donc active
- + Les autres liaisons sont détruites par la programmation du circuit



Synthèse de l'additionneur de deux bits



Synthèse de l'additionneur de trois bits

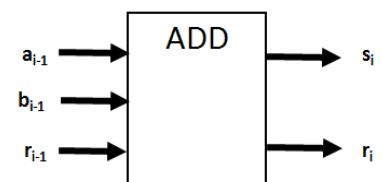
La description est donnée en VHDL. Sans être spécialiste on voit que la partie encadré en pointillé vert l'Entité spécifie les entrées sorties du composant à synthétiser. Le pointillé orange encadre l'Architecture décrivant les équations logiques du fonctionnement attendu.

Les outils permettent de simuler les design et de vérifier ainsi le bon comportement et timing avant de passer à la mise en œuvre des puces.

Aperçu du langage utilisé : le VHDL

```
ENTITY add1 IS
    PORT ( ai,bi,rin1 : IN STD_LOGIC;
          si : OUT STD_LOGIC;
          ri : OUT STD_LOGIC);
END add1;
```

Les entrées sorties du composant



```
ARCHITECTURE logique OF add1 IS
BEGIN
    si <= ai xor bi xor rin1;
    ri <= (ai and bi) or
          (bi and rin1) or
          (rin1 and ai) or
          (ai and bi and rin1);
END logique;
```

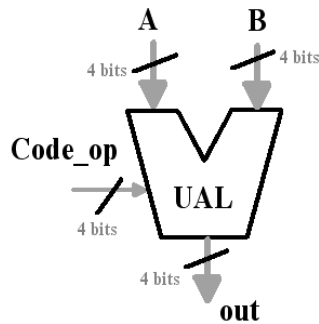
Les équations de fonctionnement



Synthèse d'une UAL en VHDL

Le symbole du composant désiré

- Deux mots d'entrée sur 4 bits.
- Un code opératoire sur 4 bits.
- Les sorties sur 4 bits.



Les entrées sorties du circuit

```
entity UAL_2x4 is
```

```
port (  
    Mot_A : in std_logic_vector ( 3 downto 0 );  
    Mot_B : in std_logic_vector ( 3 downto 0 );  
    CodeOp : in std_logic_vector ( 3 downto 0 );  
    UAL_output : out std_logic_vector ( 3 downto 0 )  
);
```

```
end UAL_2x4;
```

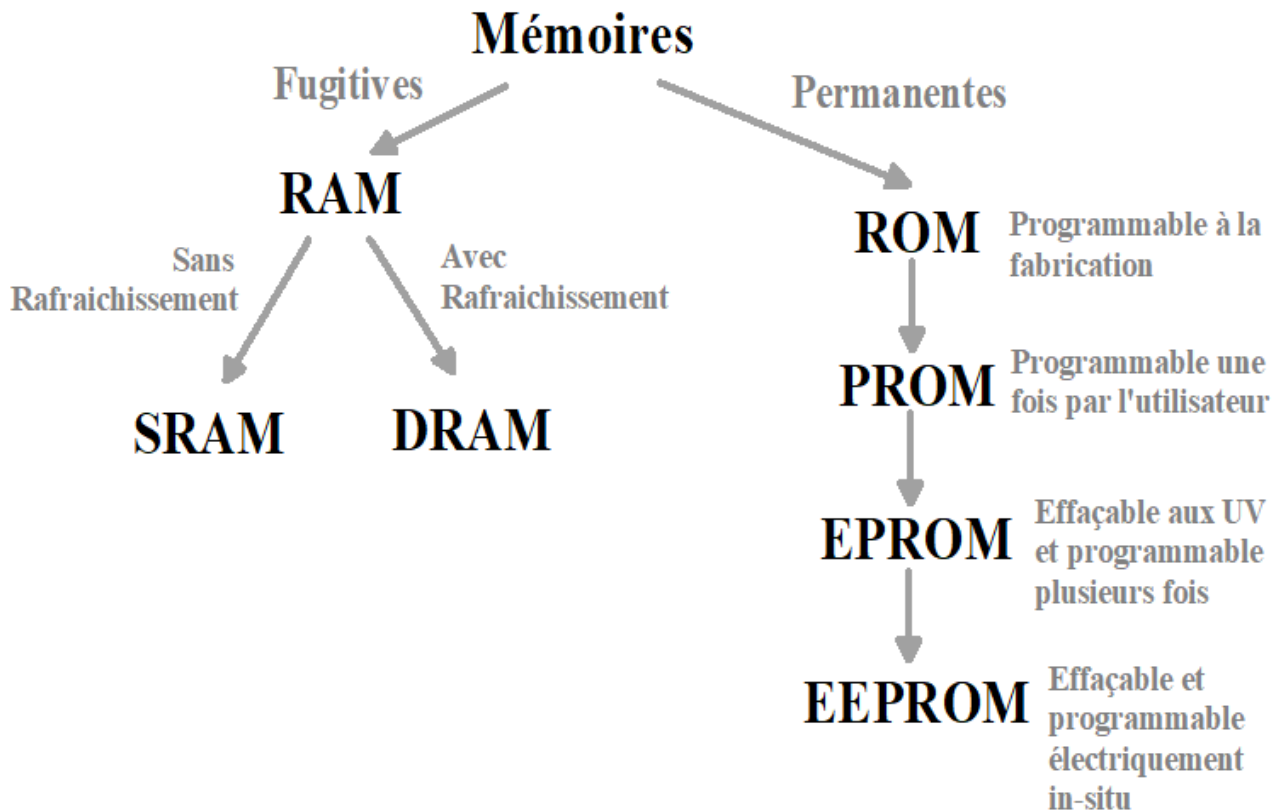
L'architecture

```
architecture equation of UAL_2x4 is  
begin  
    UAL : process (Mot_A, Mot_B, CodeOp)  
    begin  
        if ( CodeOp = "0000" )  
            then UAL_output <= "0000";  
        elsif ( CodeOp = "0001" )  
            then UAL_output <= Mot_A;  
        elsif ( CodeOp = "0010" )  
            then UAL_output <= Mot_B;  
        elsif ( CodeOp = "0011" )  
            then UAL_output <= not(Mot_A);  
        elsif ( CodeOp = "0100" )  
            then UAL_output <= not(Mot_B);  
        elsif ( CodeOp = "0101" )  
            then UAL_output <= Mot_A and Mot_B;  
        elsif ( CodeOp = "0110" )  
            then UAL_output <= Mot_A or Mot_B;  
        elsif ( CodeOp = "0111" )  
            then UAL_output <= Mot_A xor Mot_B;  
        else  
            UAL_output <= "0000";  
        end if;  
    end process;  
end equation;
```

Quel chemin depuis
l'AGC de la NASA.

Le process est une entité qui s'active dès que l'un des signaux présents dans sa liste de paramètre change de valeurs. Le VHDL est un langage de description de schéma ne l'oublions pas. Une fois activé le code est réalisé de manière itérative jusqu'à la fin du process.





Le compromis Hardware / Software Surface / Temps

Une bonne présentation des grands principes qui guident la conception des circuits. Permet de bien comprendre le compromis essentiel : **gain de temps de calcul mais au prix d'une perte de surface (du silicium) et d'énergie.**

Une conférence du Collège de France : Des circuits aux systèmes sur puces.

Adresse : <https://www.college-de-france.fr/site/gerard-berry/course-2008-02-01-10h30.htm>

Auteur : Gérard Berry

Questionnaire accompagnant la vidéo :

Des_circuits_aux_systèmes_sur_puces.docx



Exercices avec correction

Exercices de logiques

Quelques règles élémentaires de logique à connaître

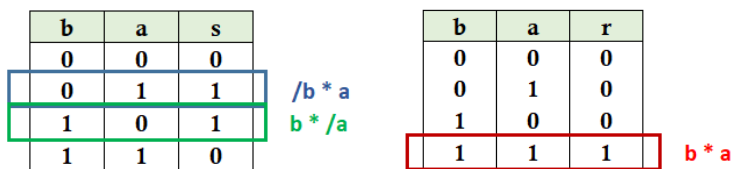
- $1 + x = 1$ $x + /x \cdot y = x + y$ (absorption)
- $1 \cdot x = x$ $/x \cdot x = 0$
- $0 + x = x$ $/x + x = 1$
- $0 \cdot x = 0$ $/x \cdot y + x \cdot /y = x \oplus y$ (ou exclusif)
- $x \cdot (y + z) = x \cdot y + x \cdot z$ (distributivité)
- $/(x + y) = /x \cdot /y$ $/(x \cdot y) = /x + /y$ (Règles de De Morgan)
- $/(x \cdot y)$ opérateur NON ET ou NAND
- $/(x+y)$ opérateur NON OU ou NOR

Mise en équation à partir d'une table de vérité

L'addition de deux bits peut se décrire sous la forme du dessin ci-dessous où nous trouvons les deux bits d'entrées a et b et les deux bits de résultats : r la retenue et s la somme. Nous pouvons décrire le comportement attendu de notre circuit d'addition par sa table de vérité.



A partir de cette table nous pouvons écrire les équations logiques des sorties s (somme) et r (retenue) :



Il nous suffit de regrouper toutes les lignes de la table de vérité pour lesquelles la fonction vaut 1 ce qui nous donne :

$$S = /b * a + b * /a = a \oplus b \quad \text{et} \quad r = b * a$$

Avec ici les représentations classiques : l'opérateur ET logique (ou 'produit logique') noté *, la représentation du OU (ou 'somme') logique avec + (ne pas confondre avec l'addition) et la représentation de l'opérateur d'inversion / et enfin le ou exclusif ou XOR noté \oplus .

Pour rappel voilà les tables de vérité de ces trois opérateurs :

b	a	ET
0	0	0
0	1	0
1	0	0
1	1	1

b	a	OU
0	0	0
0	1	1
1	0	1
1	1	1

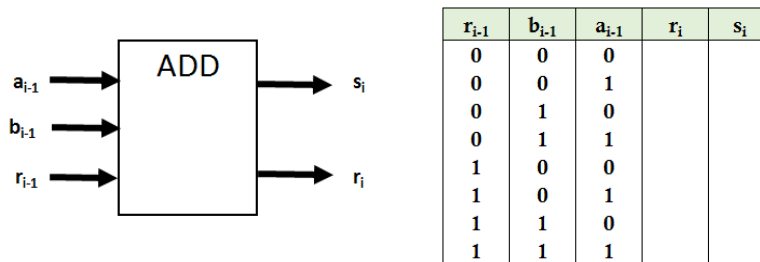
b	a	XOR
0	0	0
0	1	1
1	0	1
1	1	0

a	INV
0	1
1	0



Réalisation d'une cellule d'addition complète, addition avec retenue

L'addition vue en exemple ci-dessus n'est pas très utile car elle n'additionne que deux bits. Pour passer à l'échelle de manière simple il nous faut une cellule d'addition avec la possibilité d'entrée une retenue issue d'un calcul de rang $N-1$ quand on est sur la cellule de rang N .



Q1 : Compléter la table de vérité de l'additionneur.

Q2 : Écrire sans les simplifier les équations des sorties s_i et r_i

Q3 : Dessiner un schéma présentant l'interconnexion de quatre cellules d'addition pour réaliser l'addition de deux mots de quatre bits notés $A : a_3a_2a_1a_0$ et $B : b_3b_2b_1b_0$

Cette réalisation d'un additionneur n'est pas optimale. Dans la vidéo 'Des circuits aux systèmes sur puces' vous avez appris la nécessité de calculer différemment pour gagner en vitesse. En effet ici nous avons une complexité en $O(n)$ où n étant le nombre de bits de chacun des opérandes.

L'additionneur de Von Neumann permet une complexité en $O(\log_2 n)$ au prix d'une plus grande surface de calcul.

Q4 : Préciser en particulier la valeur à positionner sur l'entrée r_{i-1} de l'additionneur de rang 0.










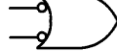
[↩](#) (Retour les premières cartes numériques)

Représentation des opérateurs

Nous utilisons ici la représentation usitée dans les schémas des circuits que nous étudions à savoir la représentation dites Américaine.

La deuxième forme est obtenue avec l'application des lois de De Morgan à savoir :

$$\overline{a + b} = \overline{a} * \overline{b} \quad \text{et} \quad \overline{a * b} = \overline{a} + \overline{b}$$

	INV	OU	ET	NOR	NAND	XOR
Symbole de l'opérateur						
Symbole équivalent obtenu avec les lois de De Morgan						

Voilà les tables de vérité des opérateurs supplémentaires :

b	a	NAND
0	0	1
0	1	1
1	0	1
1	1	0

b	a	NOR
0	0	1
0	1	0
1	0	0
1	1	0

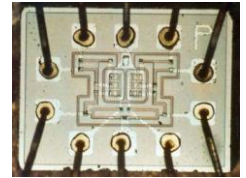
b	a	XNOR
0	0	1
0	1	0
1	0	0
1	1	1

[↩](#) (Retour les premières cartes numériques)



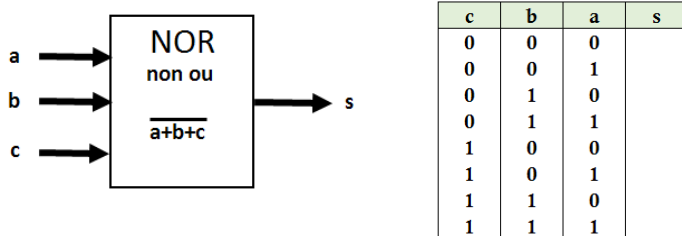
Table de vérité d'une fonction NOR ou NON-OU

Nous allons voir dans ce document que la porte NOR avec trois entrées est à la base du calculateur de vol de la mission Apollo. Avec quelques milliers de ces portes c'est un calculateur multitâche qui est réalisé assistant les astronautes pour la conduite de la capsule Apollo et du LEM.



Le circuit intégré utilisé par la NASA.

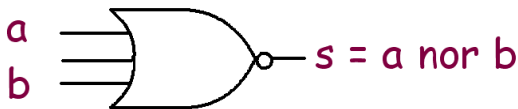
Q1 : Établir la table de vérité du circuit intégré double porte NOR à 3 entrées.



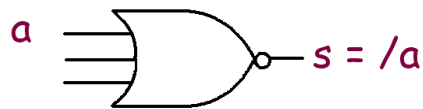
La théorie nous enseigne que toutes fonctions logiques peut être réalisée avec un assemblage d'opérateurs ET, OU et INV. Une fonction NOR est un assemblage d'un opérateur OU suivi d'un inverseur INV. Mais alors comment réaliser un inverseur INV, un opérateur ET ou bien un OU avec des portes NOR à trois entrées les seules disponibles dans le module AGC ?

A vous de compléter les schémas :

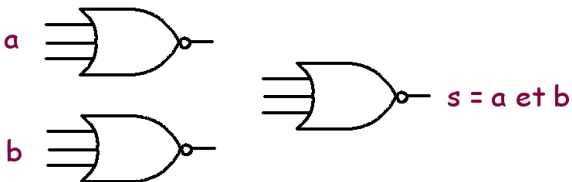
Q1 : Comment réaliser un NOR à deux entrées ?



Q2 : Comment réaliser un inverseur ?



Q3 : Comment réaliser un ET à deux entrées ?



[↩](#) (Retour les premières cartes numériques)

Applications des règles de simplification démonstration de propriétés

Un moyen de vérifier des règles entre opérateurs logiques est de dresser la table de vérité complète des deux alternatives à comparer puis de les comparer pour conclure. Exemple avec De Morgan :

b	a	a*b	$\overline{a*b}$	\overline{b}	\overline{a}	$\overline{a+b}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

$$\overline{a * b} = \overline{a} + \overline{b}$$

[↩](#)

Q1 : Vérifier la règle d'absorption : $x + /x \bullet y = x + y$

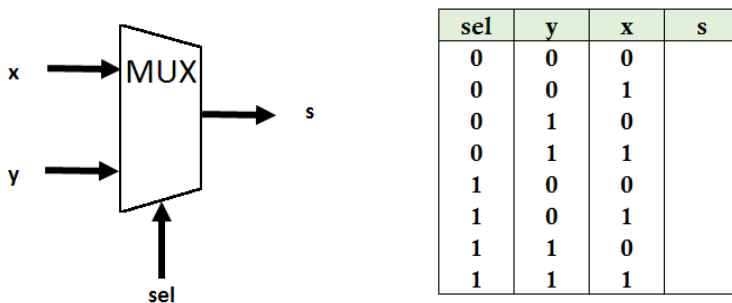
y	x	/x	/x . y	x + /x . y	x + y
0	0				
0	1				
1	0				
1	1				

[↩](#) (Retour les premières cartes numériques)

Réalisation d'une fonction multiplexeur

Un multiplexeur est une fonction qui réalise la sélection de données. Pour cela une entrée, ou un groupe d'entrées, particulière que l'on peut appeler entrée d'adressage réalise la sélection pour la sortie s soit de l'entrée x soit de l'entrée y. Tous les signaux sont des signaux logiques valant 0 ou 1.

Q1 : Avec une entrée de sélection constituée d'un seul fil combien de choix pouvons-nous gérer ?



Q2 : On souhaite obtenir le fonctionnement suivant : si la sélection vaut 0 alors on sélectionne l'entrée x, donc $s = x$, sinon, si la sélection vaut 1 on sélectionne l'entrée y donc $s = y$. Remplir la table de vérité ci-dessus.

Q3 : Écrire sans chercher à la simplifier l'équation de la sortie s.

Q4 : Écrire cette même équation uniquement avec sel, x et y et /. Cette façon de procéder est fondamentale car il faut réfléchir 'logique'. On raisonne sur les variables logiques des problèmes à résoudre et on trouve directement la solution. Cela libère du côté besogneux des tables de vérités.

[↩](#) (Retour les premières cartes numériques)

Exercices structures UAL

Analyse d'une UAL '381

Nous rappelons les opérateurs logiques élémentaires dans le tableau ci-dessous.

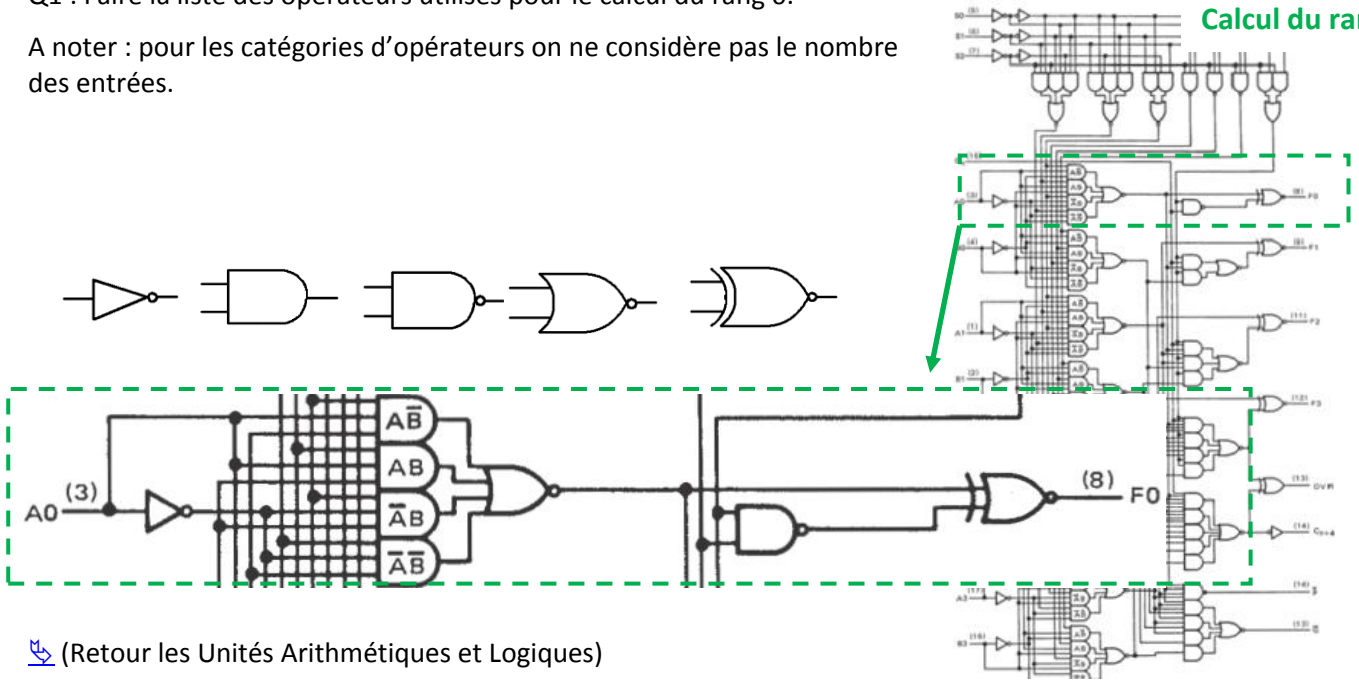
INV	OU	ET	NOR	NAND	XOR	XNOR (Équivalence)



Q1 : Faire la liste des opérateurs utilisés pour le calcul du rang 0.

A noter : pour les catégories d'opérateurs on ne considère pas le nombre des entrées.

Calcul du rang 0

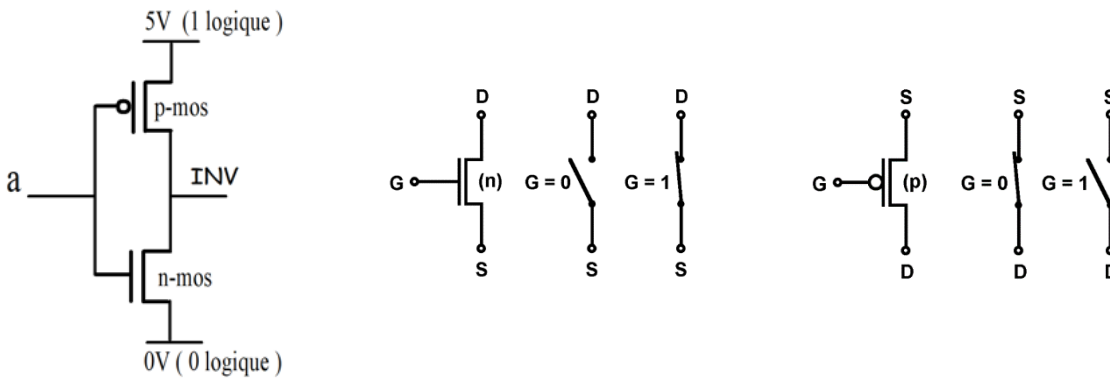


[↩](#) (Retour les Unités Arithmétiques et Logiques)

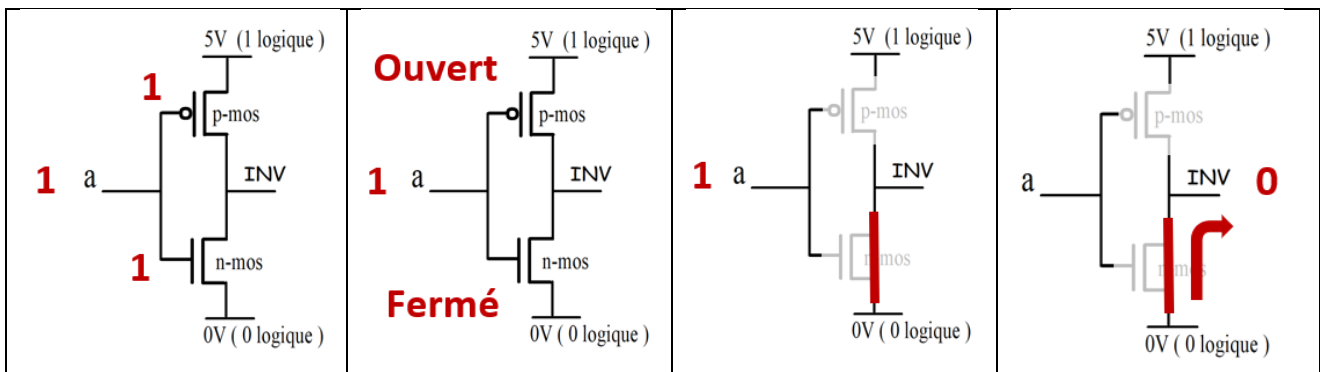
Exercices synthèse de portes logiques¹⁸

L'inverseur

Analysons le fonctionnement de l'inverseur :

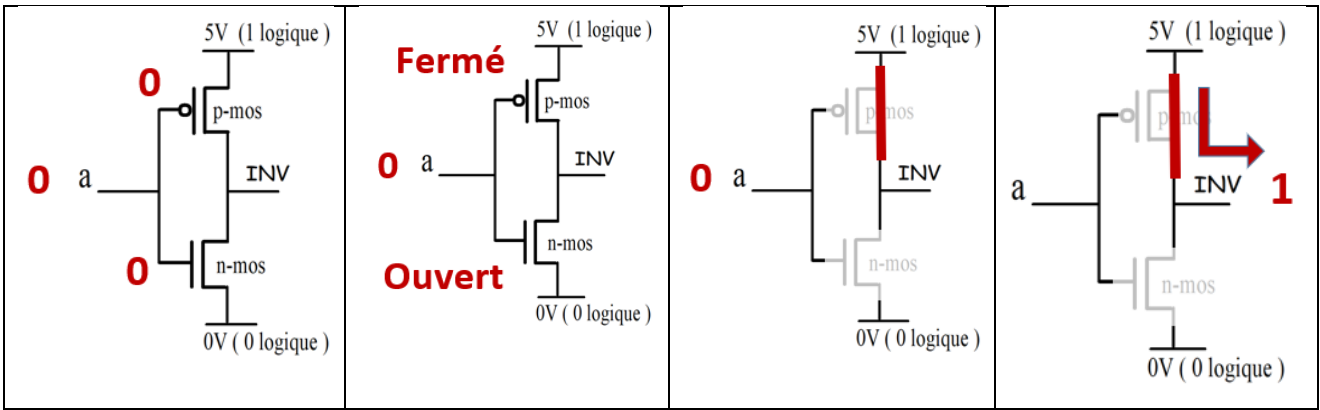


Le raisonnement est simple, chaque transistor ne peut prendre que deux états ouvert ou fermé. Lorsqu'il est ouvert on peut le supprimer du schéma et lorsqu'il est fermé on peut le remplacer par un fil. Voyons tout cela :



[↩](#)

¹⁸ Cette partie est réalisée à partir de la page <https://www.irif.fr/~carton/Enseignement/Architecture/Cours/Gates/>



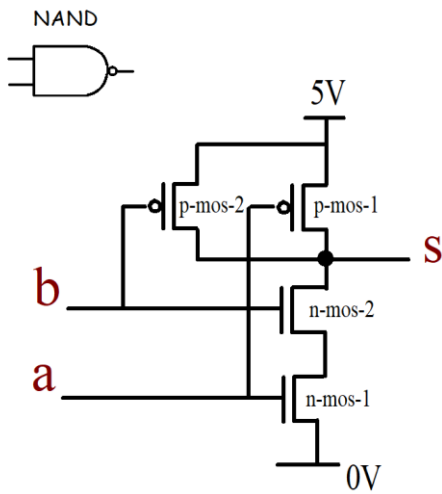
Bilan :

a	p-mos	n-mos	INV	INV
0	Fermé	Ouvert	5V	1
1	Ouvert	Fermé	0V	0

Nous retrouvons bien le fonctionnement attendu d'un inverseur.

[↩](#) (Retour conception des portes logiques)

Une porte NAND

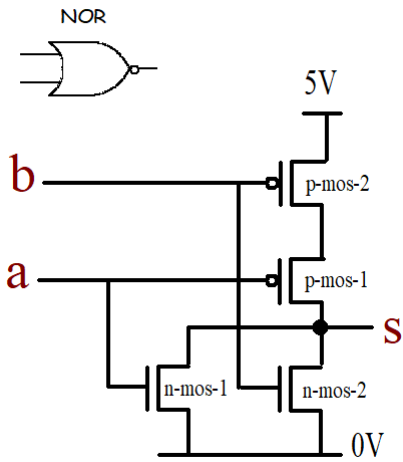


b	n-mos-2	p-mos-2	a	n-mos-1	p-mos-1	s
0			0			
0			1			
1			0			
1			1			

[↩](#) (Retour conception des portes logiques)



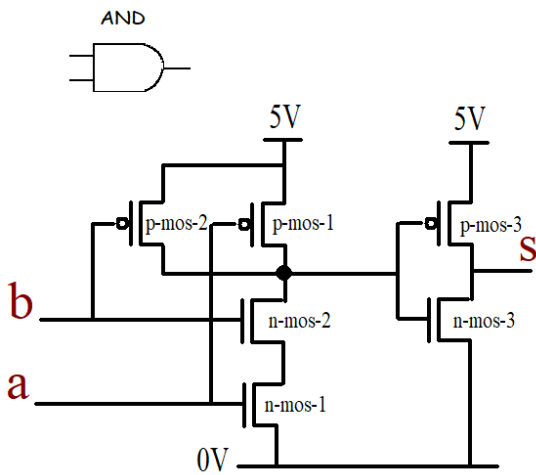
Une porte NOR



b	n-mos-2	p-mos-2	a	n-mos-1	p-mos-1	s
0			0			
0			1			
1			0			
1			1			

[\(Retour conception des portes logiques\)](#)

Une porte AND

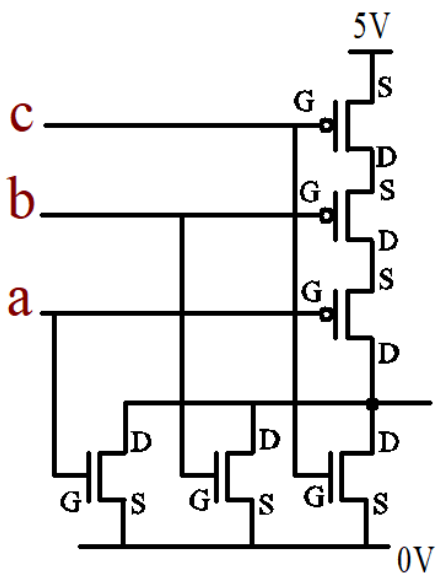


Pour des raisons technologiques la porte ET est réalisée par association de deux opérateurs.

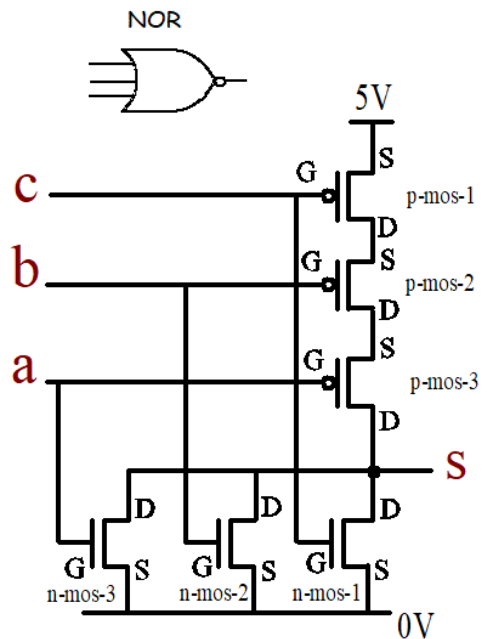
Identifiez ces deux opérateurs et entourez-les avec un trait de couleur.

[\(Retour conception des portes logiques\)](#)

La porte NOR de la NASA



Pour s'y retrouver commençons par identifier tous les transistors.



[↩](#)

Remplir l'état des transistors avec les valeurs Fermé ou Ouvert en fonction des valeurs logiques de a, b et c.

Puis déduire la valeur de s

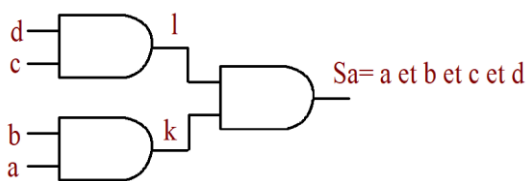
c	n-mos-1	p-mos-1	b	n-mos-2	p-mos-2	a	n-mos-3	p-mos-3	s
0			0			0			
0			0			1			
0			1			0			
0			1			1			
1			0			0			
1			0			1			
1			1			0			
1			1			1			

[↩](#) (Retour conception des portes logiques)

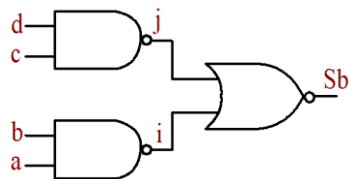
Passage à l'échelle

Voilà deux propositions pour la réalisation d'un opérateur ET à quatre entrées.

Proposition A



Proposition B



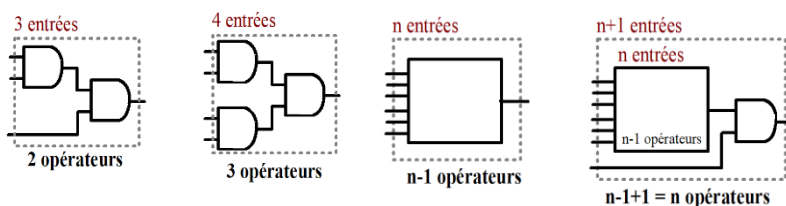
Q1 : Vérifier que les deux propositions sont équivalentes. Pour le faire écrire l'équation de la sortie Sb. Puis simplifier l'expression en utilisant les règles de De Morgan.

$$\overline{\overline{x + y}} = \overline{\overline{x} \cdot \overline{y}} \quad \overline{\overline{x \cdot y}} = \overline{\overline{x} + \overline{y}}$$

Q2 : Quelle est la proposition la plus avantageuse en termes de transistors mos, donc de surface de silicium utilisée ?

Passage à une échelle quelconque un opérateur ET à n entrées noté ET_n.

Q3 : Montrer par récurrence que pour tout n>3 le nombre d'opérateurs ET à deux entrées nécessaires pour réaliser la fonction est égal à n-1.

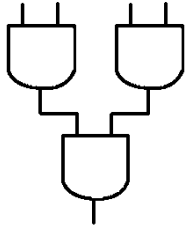


Comparaison de deux structures

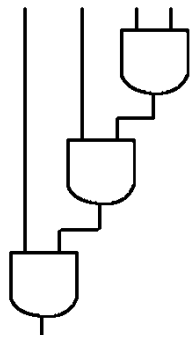
Nous avons deux structures pour réaliser un opérateur à entrées multiples. Les opérateurs logiques réels possèdent un temps de réponse noté t_d .

Q4 : Quelle est la structure qui possède le plus faible temps de réponse ?

Structure en arbre



Structure en cascade



Q5 : Pour réaliser la fonction ET_n on utilise une structure en forme d'arbre. Pour n entrées donner le nombre de couches nécessaires et le temps de réponse correspondant. Quel principe d'algorithmique est utilisé dans ce cas.

Nom du fichier : RESSOURCE CIRCUITS INTEGRES ELEVES V2_0.docx

Auteur : Patrick Guérangé Lycée Vaucanson Grenoble

Version n° 2.0 du 14 janvier 2021

