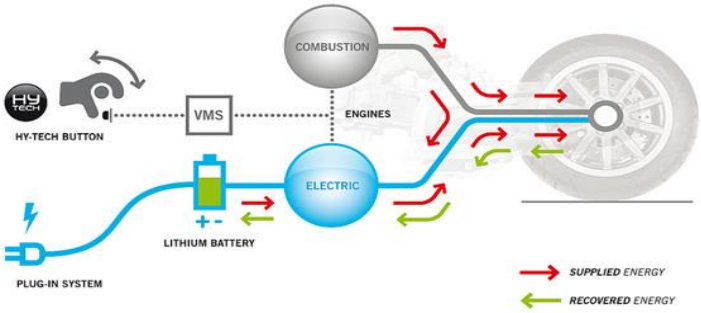
Lycée ROBERT GARNIER La Ferté-Bernard	***Enseignement Transversal Commun***	*Fiche de cours*
	TRAITEMENT COMBINATOIRE DE L'INFORMATION	

I – DEFINITION :

L'information codée disponible dans un système n'est pas directement adaptée à la prise de décision par une unité de commande automatique.

Par exemple sur le scooter hybride, on dispose des informations suivantes :

a : la clef de contact est en position active
b : la batterie est correctement chargée
c : le moteur thermique est en fonctionnement
d : le véhicule roule en palier à plus de 30 km/h



💣 La commande automatique doit-elle mettre en marche le moteur électrique. (fonctionnement en mode hybride) ?

📖 La décision de commande automatique résultant de ces informations est obtenue par un traitement du type : **« SI a ET b ET c ET d sont vraies, ALORS activer le mode hybride »**

ou : **« SI a OU b OU c OU d est fausse, ALORS désactiver le mode hybride »**

Un traitement de ce type est appelé traitement combinatoire car il consiste à définir les différentes combinaisons des variables binaires a,b,c,d qui activent la fonction résultante.

II – L'ALGEBRE DE BOOLE :

Les fonctions logiques sont des fonctions permettant de mettre en équation tous les problèmes d'automatisation, en utilisant l'algèbre de BOOLE.

Dans cet algèbre de BOOLE, la variable ou la fonction ne peut prendre que deux valeurs : **(0)** ou **(1)**.

Ces deux valeurs sont complémentaires, c'est à dire que si la variable ou la fonction ne possède pas la valeur **(0)** elle possède la valeur **(1)** et inversement.

Si une valeur Booléenne est représentée par la lettre X son complément s'écrit $\overline{X}$

On dit que $\overline{X}$ est le complément ou l'inverse de **X**.

Donc si **X = 0** $\overline{X}$ **= 1** et si **X = 1** $\overline{X}$ **= 0**

Par convention, **une variable d'entrée est représentée par une lettre minuscule.**

Ex : boutons poussoirs, capteurs, etc… s1, s2, s3, s4, s5, …

une variable de sortie est représentée par une lettre majuscule.

Ex : voyants, contacteurs, électrovannes, H1, KA, KM1, YV1…

III – PROPRIETES DE L'ALGEBRE DE BOOLE :

La mise en œuvre d'opérateurs logiques permet de décrire le comportement du système ainsi réalisé sous forme d'équations logiques ou équations booléennes.

La recherche de la plus simple équation de fonctionnement est possible, par l'application des règles concernant les propriétés de l'algèbre de Boole.

III .1 – La commutativité :

S = a . b peut s'écrire S = b . a

S = a + b peut s'écrire S = b + a

III .2 – L'associativité :

S = a . (b . c) peut s'écrire S = (a . b) . c

S = a + (b + c) peut s'écrire S = (a + b) + c

III .3 – La distributivité :

S = a . (b + c) peut s'écrire S = (a . b) + (a . c)

S = a + b . c peut s'écrire S = (a + b) . (a + c)

IV – REGLE DE COMPLEMENTATION :

Egalement appelé « Théorème de DE MORGAN », cette règle est propre à l'algèbre de Boole.

Le théorème de DE MORGAN permet de transformer l'inverse d'une expression logique .

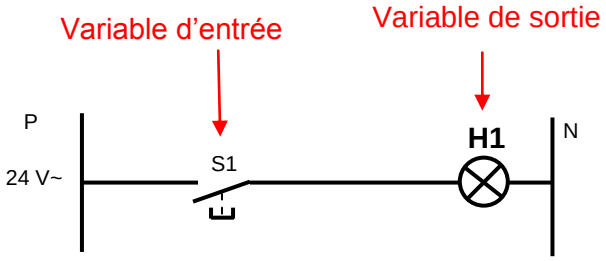
Le complément d'une expression logique s'obtient en complémentant chacun des termes et en complémentant chaque signe. (les signes + en signe . et inversement)

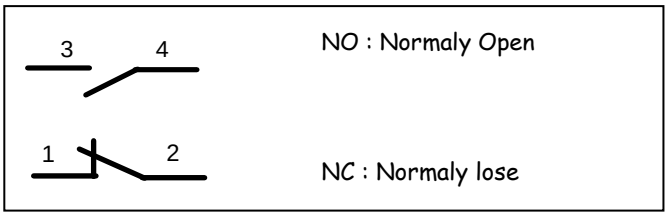
APPLICATION DU THEOREME $\overline{a + b} = \bar{a} . \bar{b}$ $\overline{a . b} = \bar{a} + \bar{b}$

V – LES DIFFERENTES REPRESENTATIONS :

Si j'appui sur l'interrupteur à l'entrée de ma chambre. Je réalise une fonction ***OUI***

V.1 – Le schéma électrique :





V.2 – La table de vérité :

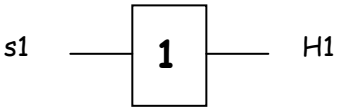
Entrée	SORTIE
s1	H1
0	0
1	1

On attribue la valeur **0** à tout **contact non actionné** et la valeur **1** à tout **contact actionné**.
On attribue la valeur **0** à tout **récepteur non alimenté** et la valeur **1** à tout **récepteur alimenté**.

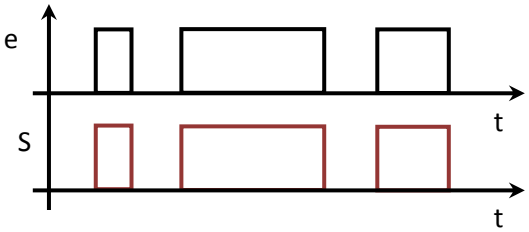
V.3 – Forme algébrique (équation) :

H1 = s1

V.4 – Symbole opérateur :



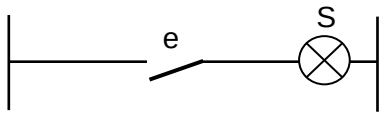
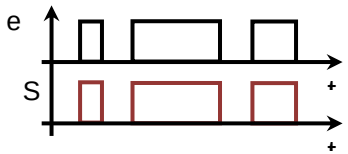
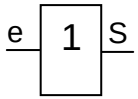
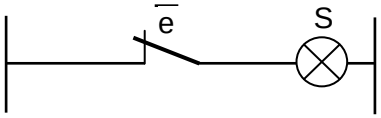
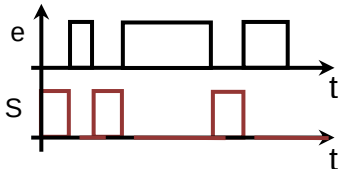
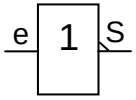
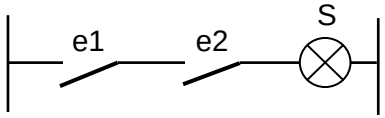
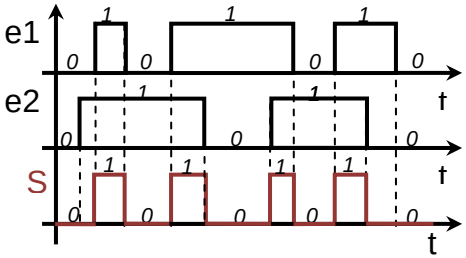
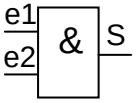
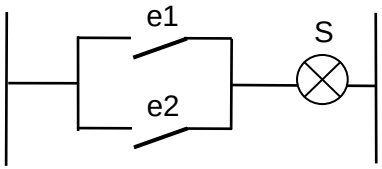
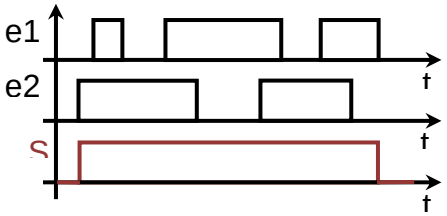
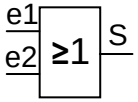
V.5 – Le chronogramme :

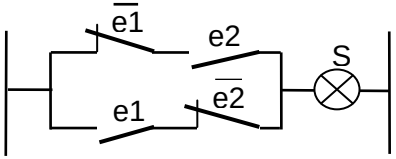
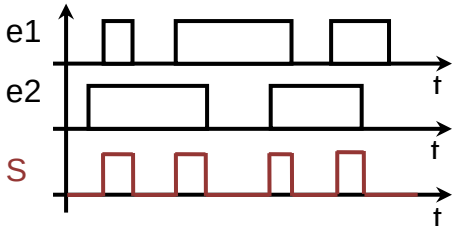
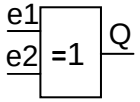


Un opérateur logique combinatoire (l'état des sorties ne dépend que de l'état des entrées) permet de réaliser avec des variables binaires, des opérations logiques, c'est-à-dire des opérations dont le résultat ne peut s'exprimer que par deux états, vrai ou faux, ou encore états logiques 1 ou 0.

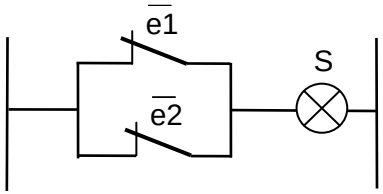
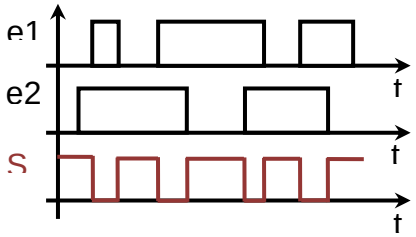
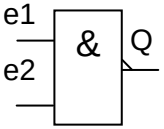
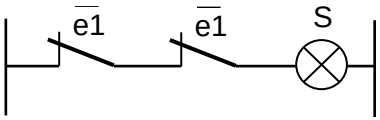
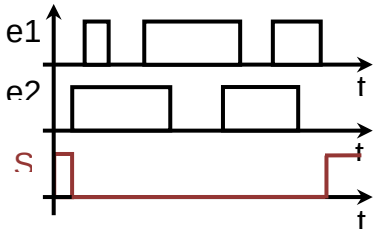
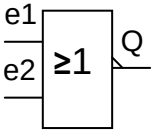
Un opérateur logique est désigné par l'opération logique réalisée, ou **FONCTION LOGIQUE**.

VI - LES OPERATEURS DE BASES :

Fonction	*Equation*	*Schéma*	*Table de vérité*	*Chronogramme*	*Symbole*	*Proposition*
OUI	**S= e**					La sortie **S** est identique a l'état de l'entrée "**e**".
NON	S= $\overline{e}$					La sortie **S** est complémentaire de l'état de l'entrée "**e**".
ET (AND)	**S= e1.e2**					La sortie **S** est à 1, si et seulement si l'entrée "**e1**" ET l'entrée "**e2**" sont à l'état 1.
OU (OR)	**S= e1+e2**					La sortie **S** est à 1, si et seulement si l'une des deux entrées, "**e1**" OU l'entrée "**e2**" est à l'état 1 ou si les deux entrées sont à l'état **1**.

Fonction	*Equation*	*Schéma*	*Table de vérité*	*Chronogramme*	*Symbole*	*Proposition*
OU exclusif	$S = \overline{e1}.e2 + e1.\overline{e2} = e1 \oplus e2$		e2 e1 S 0 0 0 0 1 0 1 0 0 1 1 1			La sortie **S** est à 1, si et seulement si les deux entrées "**e1**" et "**e2**" sont à l'état 1

VII - LES OPERATEURS COMPLEMENTES :

Fonction	*Equation*	*Schéma*	*Table de vérité*	*Chronogramme*	*Symbole*	*Proposition*
NON ET (NAND)	$S = \overline{e1.\ e2}$ $= \overline{e1} + \overline{e2}$		e2 e1 S 0 0 1 0 1 1 1 0 1 1 1 0			La sortie **S** est à 0, si et seulement si l'entrée "**e1**" et l'entrée "**e2**" sont à l'état 1
NON OU (OR)	$S = \overline{e1 + e2}$ $= \overline{e1}\ .\ \overline{e2}$		e2 e1 S 0 0 1 0 1 0 1 0 0 1 1 0			La sortie **S** est à 1, si et seulement si les deux entrées "**e1**" ET "**e2**" sont à l'état **0**.

VIII – ETABLISSEMENT D'UN EQUATION LOGIQUE A PARTIR D'UNE TABLE DE VERITE :

Une table de vérité définit les relations entrée(s)/sortie(s) en faisant la liste de toutes les possibilités.
Une table de vérité contient 2^n lignes n étant le nombre d'entrées.
On peut avoir plus d'une sortie.
Ex : si 3 entrées et 2 sorties la table de vérité comportera 5 colonnes.

Exemple : Un aquarium est composé :

- D'un filtre fonctionnant grâce à une pompe qui débarrasse l'eau de ses impuretés.
- D'un thermomètre qui mesure la température de l'eau. Celle-ci ne doit pas excéder 20°C sous peine de tuer les poissons
- D'un buller qui apporte l'oxygène aux poissons.

Une alarme prévient l'utilisateur en cas de problème, c'est à dire lorsqu'au moins une des conditions suivantes est vérifiée :

- La pompe (p) est arrêtée (p=0 : pompe arrêtée ; p=1 : pompe en marche)
- La température (t) est supérieure à 20°C (t=0 si temp =< 20°C; t=1 si temp > 20°C)
- Le bulleur (b) est arrêté (b=0 : bulleur arrêtée ; b=1 : bulleur en marche)

Etablir par une table de vérité.

p	t	b		A
0	0	0		1
0	0	1		1
0	1	0		1
0	1	1		1
1	0	0		1
1	0	1		0
1	1	0		1
1	1	1		1

Déterminer les équations de fonctionnement.

Pour obtenir l'équation logique à partir d'une table de vérité, il suffit de rechercher les différentes combinaisons des variables d'entrées qui permettent d'obtenir la sortie égale à 1.

Pour A on peut donc écrire : A = **1**
si p = 0 **ET** t = 0 **ET** b = 0
OU si p = 0 **ET** t = 0 **ET** b = 1
OU si p = 0 **ET** t = 1 **ET** b = 0
OU si p = 0 **ET** t = 1 **ET** b = 1
OU si p = 1 **ET** t = 0 **ET** b = 0
OU si p = 1 **ET** t = 1 **ET** b = 0
OU si p = 1 **ET** t = 1 **ET** b = 1

Donc $A = \overline{p}.\overline{t}.\overline{b} + \overline{p}.\overline{t}.b + \overline{p}.t.\overline{b} + \overline{p}.t.b + p.\overline{t}.\overline{b} + p.\overline{t}.b + p.t.b$

IX – SIMPLIFICATION D'UNE EQUATION LOGIQUE:

La simplification de l'équation permet de réduire la taille du logigramme donc le nombre de portes nécessaires.
Deux méthodes peuvent être utilisées, mais la plus rapide et la plus sûr est la simplification par les tableaux de Karnaugh.

IX.1 – Simplification algébrique :

Elle permet grâce à des regroupements et des propriétés des fonctions logiques de simplifier une équation.

Propriétés

- Élément neutre : A+0 = A et A.1 = A
- Élément absorbant : A+1 = 1 et A.0 = 0
- Idempotence : A+A=A et A.A=A
- Complément : $\overline{\overline{A}} = A$

✍ Simplifier algébriquement l'équation de l'aquarium : **$A = \overline{p}.t.\overline{b}$**

IX.2 Simplification par les tableaux de Karnaugh

✍ Construction :

A 3 variables d'entrée

Etats possibles pour les entées b et c

Etats possibles pour l'entée a

a \ bc	00	01	11	10
0	0	1	1	1
1	0	1	0	0

Etats possibles de la sortie en fonction des entrées a,b et c

On utilise obligatoirement le code de Gray

A 4 variables d'entrée

ab \ cd	00	01	11	10
00	0	1	1	1
01	0	1	0	0
11	1	0	1	1
10	1	0	1	0

✍ Principe de simplification :

- Réaliser des groupements de '1' adjacents, dans l'ordre, par 16, 8, 4 ,2 ou 1.

- Il faut toujours s'arranger à regrouper le maximum de '1' pour diminuer la taille des termes.
- Lorsqu'il ne reste plus de '1' isolé, les regroupements sont terminés.
- L'équation simplifiée est déduite de ces groupements

Exemples :

b change — c change

a \ bc	00	01	11	10
0	0	1	1	1
1	1	0	0	0

$S_1 = a.\overline{b}.\overline{c} + \overline{a}.c + \overline{a}.b$

On ne prend pas en compte les variables qui changent d'état dans le regroupement.

ab \ cd	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$S_2 = \overline{d}$

ab \ cd	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$S_3 = \overline{b}.\overline{d}$

ab \ cd	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	1	1
10	0	1	1	1

$S_4 = c + \overline{b}.d$

Erreurs à ne pas commettre	A faire
Réaliser des groupements de 3, 5, 7 etc …	Réaliser des groupements de 16, 8, 4, 2 ou 1
Utiliser le code binaire pour les variables d'entrée.	Utiliser le code de Gray pour les variables d'entrée.

✍ Simplifier par tableau de Karnaugh l'équation de l'aquarium

tb p	00	01	11	10
0	1	1	1	1
1	1	0	1	1

X – ASSOCIATION D'OPERATEURS LOGIQUES : LOGIGRAMME

Le traitement logique des informations peut nécessiter la mise en œuvre d'un nombre important **d'opérateurs binaires qui sont interconnectés.**

La représentation graphique de l'association de plusieurs opérateurs binaires est un logigramme ou diagramme logique.

Exemple :Système de commande de l'ouverture de porte de garage d'un hôtel

Pour ENTRER dans le garage :

SI autorisation d'entrée délivrée par le réceptionniste **ET** demande d'entrée du client

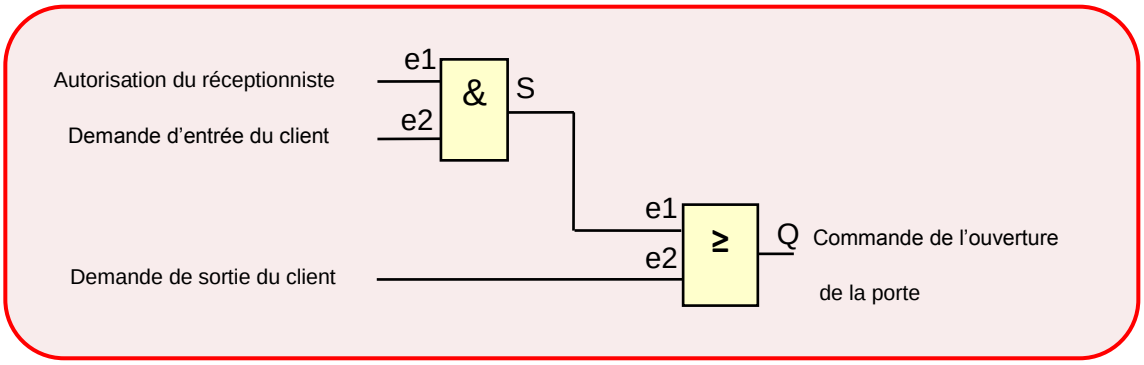
ALORS on permet **Ouverture porte**

Pour SORTIR du garage :

SI seule la demande de sortie du client

ALORS on permet **Ouverture porte**

Ce comportement peut être représenté à l'aide d'un logigramme qui associe un opérateur **ET** à 2 entrées avec un opérateur **OU** à 2 entrées.



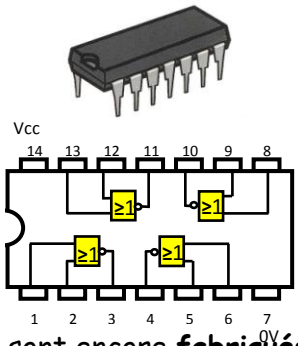
XI – LES CIRCUITS INTEGRES STANDARD

Il y a quelques années, les montages électroniques nécessitant des circuits logiques utilisaient des fonctions logiques élémentaires contenues dans les circuits intégrés des familles 74xxx ou CD4000.

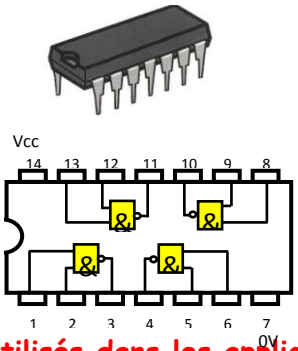
La mise en œuvre de fonctions regroupant plusieurs de ces circuits nécessitait le **câblage de nombreuses connexions** et éventuellement la réalisation de **circuits imprimés parfois complexe.**

Ceci avait pour conséquences un **prix de revient élevé**, une **mise en œuvre complexe et un circuit imprimé de taille importante.**

BROCHAGE DU CIRCUIT INTEGRE 4001

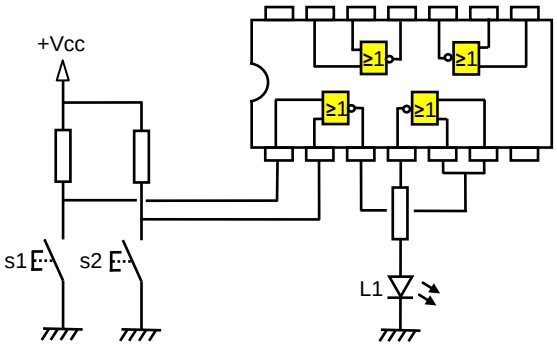


BROCHAGE DU CIRCUIT INTEGRE 4011



Ces circuits sont encore **fabriqués et vendus** mais **ne sont plus utilisés dans les applications industrielles.**

Exemple de câblage



L = s1 + s2

La led L1 s'allumera si un des BP s1 ou s2 sont actionnés.

XIII – LES CIRCUITS LOGIQUES PROGRAMMABLES

XIII.1 – Introduction :

A l'origine, les portes logiques étaient réalisées avec des composants discrets (diodes) portes ET, OU, mais la logique était incomplète (pas de portes NON). Il fallait créer un système de logique complet, il était alors nécessaire d'utiliser les transistors.
Puis, pour miniaturiser, les transistors ont été intégrés directement sur le silicium dans des circuits intégrés standards (TTL ou CMOS) pour obtenir les familles de composants logiques qu'il fallait associer sur des circuits imprimés.

Les progrès technologiques ont permis de multiplier le nombre de portes sur une puce et les fournisseurs ont proposés des solutions programmables en réalisant directement sur le silicium l'interconnexion des portes.
Dans le courant des années 1980, la société Californienne XILINX invente alors, une nouvelle famille de puces programmable appelée FPGA.

Dès 1985 des contrôleurs et des circuits périphériques complexes commencent à être implantés autour des microprocesseurs. Ces cartes sont utilisées de plus en plus pour effectuer du traitement du signal et dans le domaine des télécoms. De leur côté, les FPGA se développent.

En 1987, une première standardisation des langages de descriptions du hardware est créée. Cette standardisation permet d'étendre le design de circuit aux particuliers.

Le développement des mémoires utilisées en informatique fut à l'origine des premiers circuits logiques programmables (PLD : programmable logic device).

Ce type de produit peut intégrer dans un seul circuit plusieurs fonctions logiques programmables par l'utilisateur.

Sa mise en œuvre se fait très facilement à l'aide d'un programmateur, d'un micro-ordinateur et d'un logiciel adapté (voir liens à la page suivante).

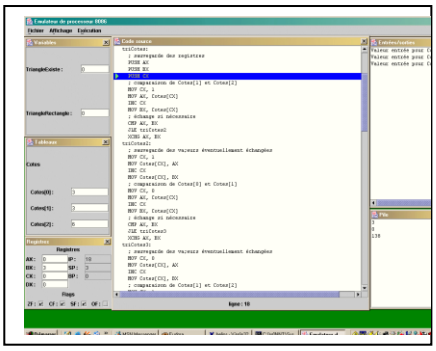
L'utilisateur peut créer, dans ces circuits, toutes les fonctions logiques qu'il souhaite avec comme limitations, la place disponible dans le circuit choisi et / ou la vitesse de fonctionnement de celui-ci.

Les outils de développement mis à la disposition des utilisateurs par les fabricants de ces circuits doivent donc permettre de passer de la description du comportement d'une fonction logique à son câblage dans le circuit et cela de la manière la plus simple possible

Ce type de composant électronique est communément désigné par les sigles anglais :

- **FPGA** (*field-programmable gate array*, réseau de portes programmables *in situ*),
- **PLD** (*programmable logic device*, circuit logique programmable),
- **EPLD** (*erasable programmable logic device*, circuit logique programmable et effaçable),
- **CPLD** (*complex programmable logic device*, circuit logique programmable complexe),
- **PAL** (*programmable array logic*, réseau logique programmable),
- **PLA** (*programmable logic array*, réseau logique programmable), etc…

XIII.2 – Présentation :



Les fonctions logiques programmables sont des circuits disposants d'entrées et de sorties dont l'utilisateur peut programmer le schéma logique d'après les besoins liés à la fonction souhaitée : **Logique combinatoire et/ou séquentielle.**

Avantages :

✍ Un seul circuit peut remplacer à lui seul l'équivalent de plusieurs circuits en technologie standard TTL ou CMOS.

✍ Composant de faible coût car produits en grand nombre.

✍ **La fonction logique réalisée peut être modifiée par programmation, donc sans qu'il soit nécessaire de redessiner un nouveau circuit imprimé, cela dépend de la technologie utilisée..**

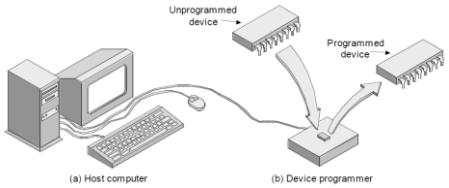
Inconvénients :

✍ Phase de programmation qui nécessite du temps de main d'œuvre et l'immobilisation de moyens de production (programmateur/Logiciel)

✍ Logiciels et matériel compatibles avec le composant à programmer

✍ Compétences supplémentaires.

XII.3 – Caractéristiques :



Un circuit logique programmable est caractérisé par:

- le nombre d'entrées
- le nombre de sorties
- la technologie
- le temps de propagation (vitesse)
- la consommation qui augmente avec la commutation
- le nombre de termes produits par sortie

XII – LES MODULES LOGIQUES PROGRAMMABLES :

XII.1 – Introduction :

Ce type de module est utilisé pour réaliser des traitements de logiques combinatoire.
Le comportement peut être décrit avec des logigrammes (comportement combinatoire) mais le traitement en interne est séquentiel lié aux cycles de lecture/écriture des E/S.

Très utilisés en domotique pour la gestion d'applications liées notamment à la gestion de l'énergie, ces modules permettent de développer des applications simplement même si elles sont parfois complexes.

Ils sont de plus en plus utilisés avec la gestion d'installations qui prennent en compte le développement durable.

Les modules logiques sont destinés à la commande automatique d'installations domestiques ou industrielles.

Ils peuvent être utilisés partout où des fonctions logiques, des fonctions temporisées, des fonctions de comptage, etc… sont nécessaires.

Ils conviennent particulièrement pour la commande de chauffage (PAC), éclairage, ventilation, régulation de température, gestion de récupérateur d'eau de pluie, gestion de chauffage par zone…

Ces modules peuvent assurer également la commande et le contrôle d'installations a distance.

Certains modèles permettent le traitement de grandeurs analogiques 0–10 V avec possibilité de détection de seuils et de fonctions de régulation.
Des capteurs peuvent y être associés en entrée si leur conditionnement le permet et les rendent compatibles avec le module d'entrée. Acquisition de tous types d'informations avec traitement analogique des valeurs si cela est nécessaire.

Les sorties peuvent être de type tout ou rien a relais ou a transistor.

XII.2 – Intérêt des modules logiques :

Les modules logiques sont des contrôleurs programmables qui permettent que les machines effectuent des processus sans avoir besoin d'une intervention humaine.

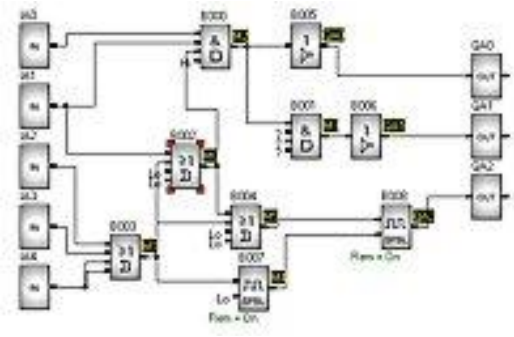
Ce sont des contrôleurs intelligents.

Grâce aux modules logiques modernes et économiques il est possible de nos jours d'utiliser ces contrôleurs dans des processus simples ou même dans le domaine privé. Des applications qui auparavant avaient besoin d'une série d'activateurs et de temporisateurs et beaucoup de travail de connexions, pourront maintenant s'effectuer avec les modules logiques.

Il est souvent possible d'y connecter en option un écran LCD et / ou des modules supplémentaires. La programmation s'effectue à travers d'un logiciel dédié, qui réalise la simulation des paramètres programmés et le téléchargement du programme

XII.3 – Programmation :

Les modules logiques sont de petits contrôleurs programmables. Contrairement aux commutateurs traditionnels, dans lesquels les fonctions de commutation se déterminent par les connexions de fils, ces fonctions sont programmables et enregistrées dans la mémoire interne du module logique. Etant donné que ce type de mémoire peut s'effacer et permettre une nouvelle écriture, un module logique est à usage très flexible. La programmation des modules logiques s'effectue directement dans le module, ou pour les programmes plus importants, en utilisant un PC et le logiciel, et en téléchargeant le programme vers le module.



Le programme se compose de différents composants individuels interconnectés. en plus des modules logiques pour la connexion logique des signaux (AND, OR, NOT, etc.), les modules logiques ont une série d'autres modules de fonctions, comme par exemple, les compteurs (fonction séquentielle).

XII.4 - Différents types de modules :

Module ZELIO de SCHNEIDER

8 entrées

4 sorties

Module LOGO de SIEMENS

8 entrées

4 sorties