Lycée Hoche
Versailles

Automatique – Systèmes combinatoires

Philippe Bourzac
2000

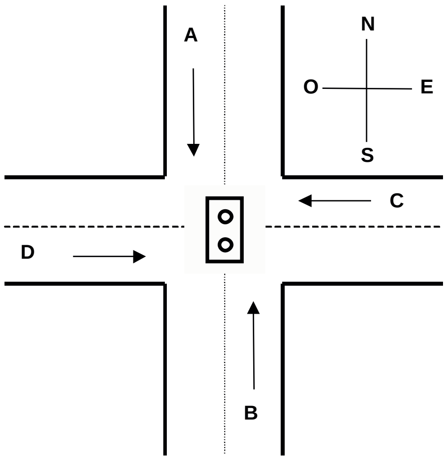
AUTOMATIQUE – SYSTEMES COMBINATOIRES

1. Modèle de fonctionnement logique d'un système combinatoire.

Définition : Un modèle de fonctionnement logique est un ensemble de relations causales qui traduisent le fonctionnement du système.

Exemple : Modèle de fonctionnement logique d'un système de gestion d'un croisement.

La figure suivante représente une intersection entre une route principale et une route secondaire.



Des capteurs de présence des voitures ont été placés le long des voies :

- c et d pour la route principale ;
- a et b pour la route secondaire.

Les sorties de ces capteurs sont à 1 quand il y a des voitures, à 0 quand il n'y en a pas.

Les feux de circulation se trouvant à cette intersection sont commandés par les règles de décision suivantes :

- S'il y a des voitures dans la voie C et dans la voie D Alors le feu E-0 est vert.
- S'il y a des voitures dans la voie C ou dans la voie D et dans la voie A ou dans la voie B mais pas dans les deux Alors le feu E-0 est vert.
- S'il y a des voitures dans la voie A et dans la voie B et dans la voie C ou dans la voie D mais pas dans les deux Alors le feu N-S est vert.
- S'il y a des voitures dans la voie A ou dans la voie B et aucune dans les voies C et D Alors le feu N-S est vert.

Relation causale : Une relation causale s'écrit : Si « cause » Alors « effet ».

Proposition logique : « cause » et « effet » sont des propositions logiques faisant intervenir respectivement les entrées et les sorties du système. Une proposition logique est vraie ou fausse.

Variables binaires : Pour simplifier l'écriture du modèle de fonctionnement logique, on utilise des variables binaires associées à des propositions logiques.

Proposition logique vraie $\Rightarrow$ Variable binaire associée = 1
Proposition logique fausse $\Rightarrow$ Variable binaire associée = 0

Exemple : Association de variables binaires.

- il y a des voitures dans la voie A $\Rightarrow$ va
- il y a des voitures dans la voie B $\Rightarrow$ vb
- il y a des voitures dans la voie C $\Rightarrow$ vc
- il y a des voitures dans la voie D $\Rightarrow$ vd

- le feu E-0 est vert $\Rightarrow$ E-0
- le feu N-S est vert $\Rightarrow$ N-S

Fonctions logiques : Pour simplifier encore l'écriture du modèle de fonctionnement logique, on utilise l'algèbre de Boole qui nous permet d'exprimer les sorties comme des fonctions logiques des entrées.

Exemple : Modèle de fonctionnement logique utilisant des fonctions logiques.

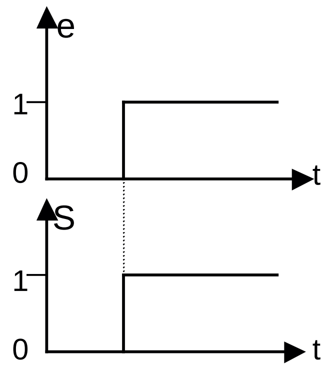
- E-O = 1 si [vc ET vd] = 1
- E-O = 1 si [(vc OU vd) ET (NON (va ET vb))] = 1
- N-S = 1 si [va ET vb ET (NON (vc ET vd))] = 1
- N-S = 1 si [(va OU vb) ET (NON vc) ET (NON vd)] = 1

Donc : **E-O = [vc ET vd] OU [(vc OU vd) ET (NON (va ET vb))]**
N-S = [va ET vb ET (NON (vc ET vd))] OU [(va OU vb) ET (NON vc) ET (NON vd)]

2. Le “temps” et les systèmes combinatoires.

La “causalité à temps nul” des modèles logiques

Les modèles logiques reposent tous sur une hypothèse temporelle fondamentale : l’effet d’une cause se produit à l’instant même où la cause apparaît.
Cette “causalité à temps nul” est celle de l’implication mathématique e $\Rightarrow$ S

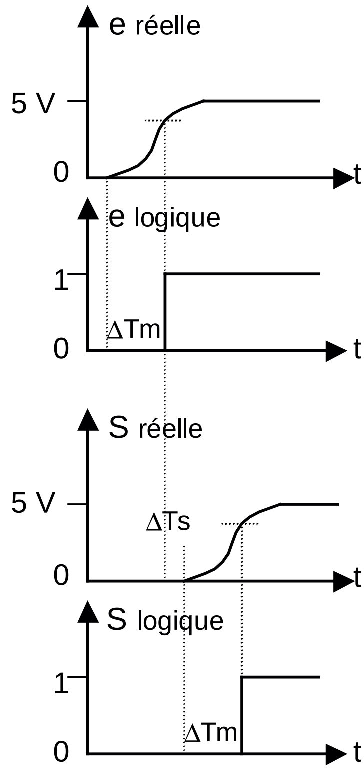


Temps de réponse des constituants logiques

Les composants et les constituants logiques réels, présentent, eux, un retard entre cause et effet.
Ce retard résulte de la conjonction de deux types de phénomènes physiques :

- un seuil de réaction ΔT_S (ex : rattrapage de jeu pour un constituant mécanique; temps interne de traitement pour un constituant programmable etc).
- un temps de changement d’état ΔT_m (temps de passage, pour la sortie réelle, d’une valeur initiale à une valeur finale).

On peut distinguer deux temps de réponse du constituant :
- Tm = $\Delta T_S + \Delta T_m$ (temps de montée)
- Td similaire à Tm (temps de descente)



3. Fonctions logiques combinatoires.

3.1. Définition.

Si à tout instant la sortie **S** d’une fonction logique peut être exprimée par une combinaison de ses entrée $\mathbf{e_1}$, $\mathbf{e_2}$, ... : **S** = $f(\mathbf{e_1}, \mathbf{e_2}, ...)$, on dit que cette fonction logique est combinatoire.

3.2. Table de vérité d'une fonction logique.

Définition : C'est un tableau donnant l'état des sorties d'un modèle logique en fonction des état possibles des entrées.

Exemple :

Entrées		Sorties	
a	b	X	Y
0	0	1	1
0	1	0	1
1	0	0	0
1	1	1	1

3.3. Opérateurs logiques de bases.

3.3.1. L'opérateur OUI $S=f(e)=OUI(e)=e$

Table de vérité

e	S
0	0
1	1

Logigramme

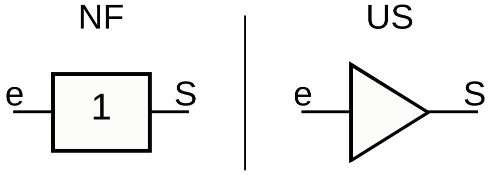


Diagramme temporel

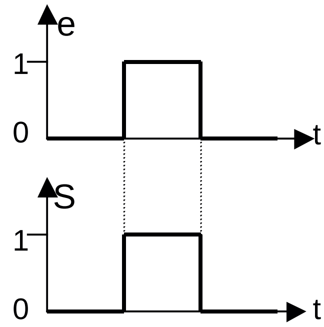
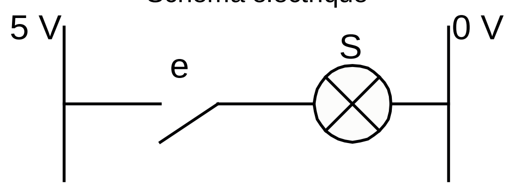


Schéma électrique



3.3.2. L'opérateur NON $S=f(e)=NON(e)=\bar{e}=/e$ (e barre)

Aussi appelé : Fonction complément ou inverse.

Table de vérité

e	S
0	1
1	0

Logigramme

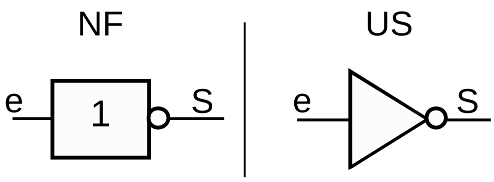


Diagramme temporel

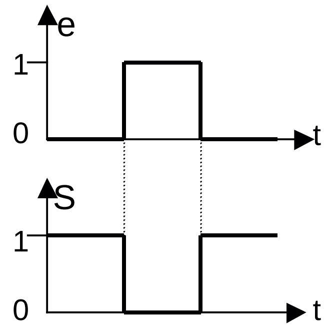
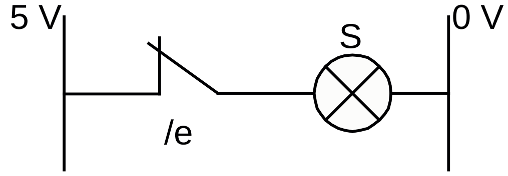


Schéma électrique



3.3.3. L'opérateur ET $S=f(a,b)=aETb=a.b$ Aussi appelé : Produit logique

Table de vérité

a	b	S
0	0	0
1	0	0
0	1	0
1	1	1

Logigramme

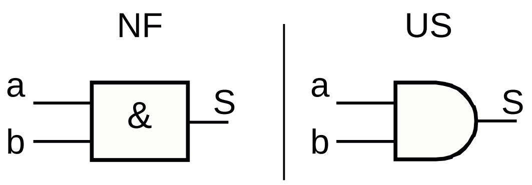


Schéma électrique

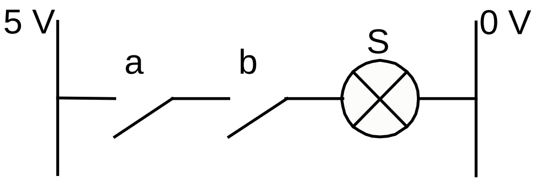
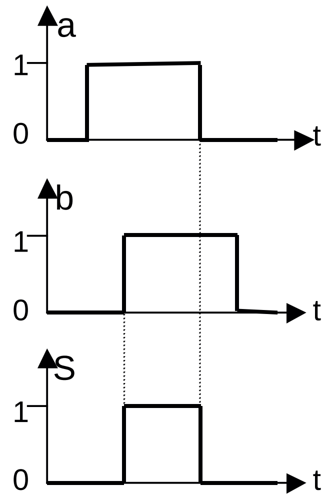


Diagramme temporel



3.3.4. L'opérateur OU $S=f(a,b)=aOUb=a+b$ Aussi appelé : Somme logique

Table de vérité

a	b	S
0	0	0
1	0	1
0	1	1
1	1	1

Logigramme

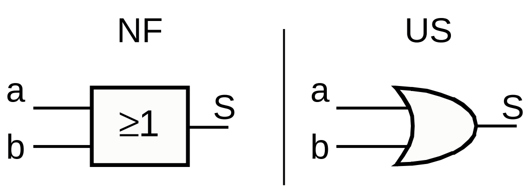


Diagramme temporel

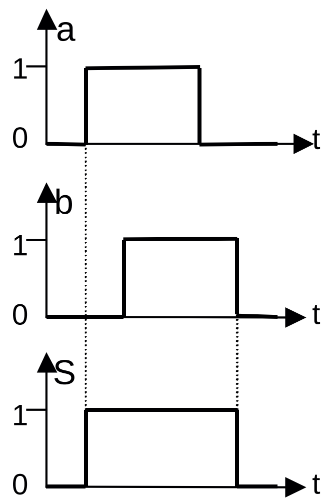
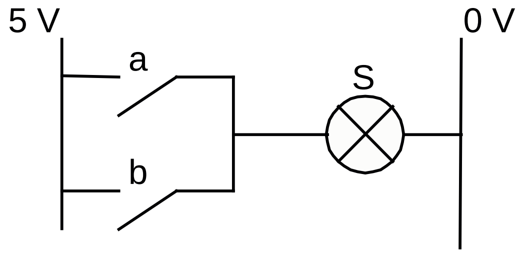


Schéma électrique



3.3.5. L'opérateur OU EXCLUSIF $S = f(a,b) = a\,OU\ EXCLUSIF\,b = a \oplus b$

Table de vérité

a	b	S
0	0	0
1	0	1
0	1	1
1	1	0

Donc $S = \overline{a}.b + a.\overline{b}$

Logigramme

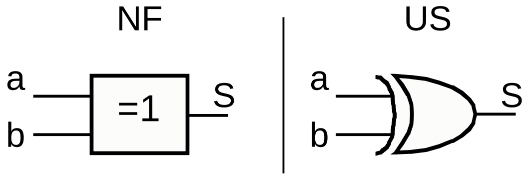


Diagramme temporel

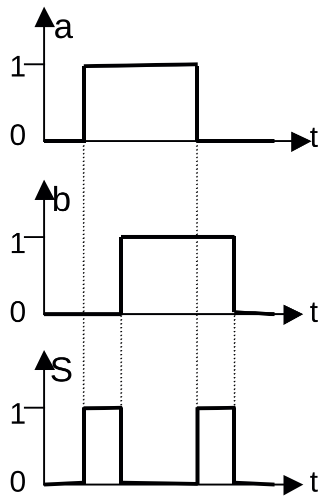
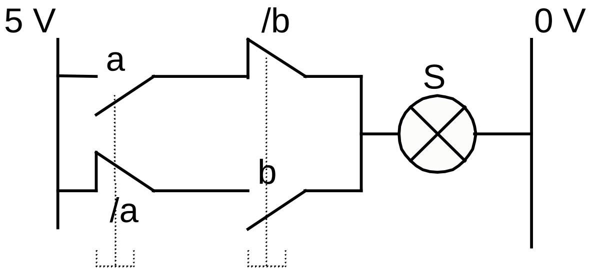


Schéma électrique



3.3.6. L'opérateur IDENTITE $S = f(a,b) = a\,IDENTITE\,b = \overline{a \oplus b}$

Aussi appelé : Opérateur Egalité

Table de vérité

a	b	S
0	0	1
1	0	0
0	1	0
1	1	1

Donc $S = \overline{a}.\overline{b} + a.b$

Logigramme

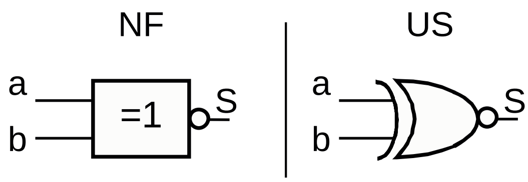
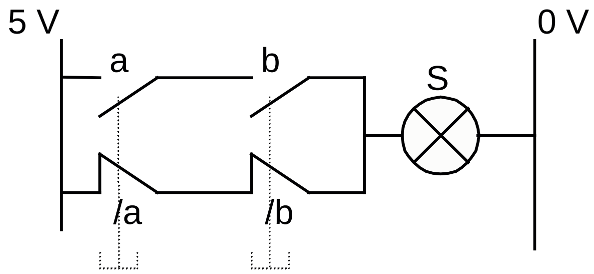


Schéma électrique



3.4. Propriétés des fonctions logiques de base.

- Commutativité : a.b=b.a
  a+b=b+a
- Associativité : (a.b).c=a.(b.c)
  (a+b)+c=a+(b+c)
- Distributivité : a.(b+c)=(a.b)+(a.c)
  a+(b.c)=(a+b).(a+c)
- Eléments neutres : a+0=a
  a.0=0
  a+1=1
  a.1=a
- Absorption : a+a.b=a
- Idempotence : a+a=a
  a.a=a
- Complémentation : a+/a=1
  a./a=0

Remarque : l'opérateur ET est prioritaire par rapport à l'opérateur OU.

3.5. Théorèmes de De MORGAN.

3.5.1. 1[er] théorème : Fonction NON OU (NOR).

Le complément d'une somme logique de plusieurs variables est égal au produit du complément de chacune des variables. $\overline{\sum_{i=1}^{n} ai}=\prod_{i=1}^{n}\overline{ai}$

Exemple : $S=f(a,b)=aNORb=\overline{a+b}=\overline{a}.\overline{b}$

Table de vérité

a	b	S
0	0	1
1	0	0
0	1	0
1	1	0

Logigramme

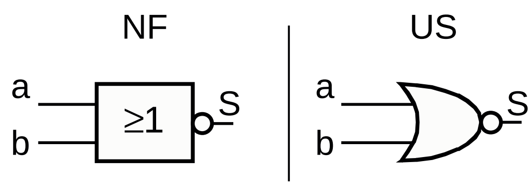
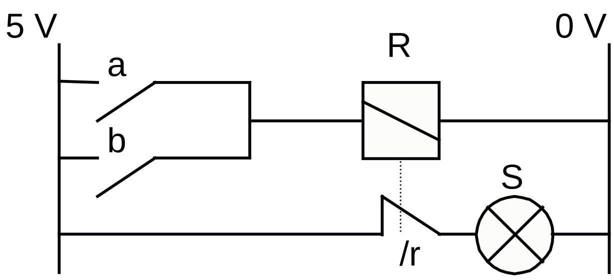


Schéma électrique



3.5.2. 2ème théorème : Fonction NON ET (NAND).

Le complément d'une produit logique de plusieurs variables est égal à la somme du complément de chacune des variables. $\overline{\prod_{i=1}^{n} ai} = \sum_{i=1}^{n} \overline{ai}$

Exemple : $S=f(a,b)=a NAND b=\overline{a.b}=\overline{a}+\overline{b}$

Table de vérité

a	b	S
0	0	1
1	0	1
0	1	1
1	1	0

Logigramme

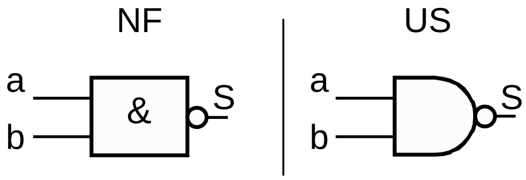
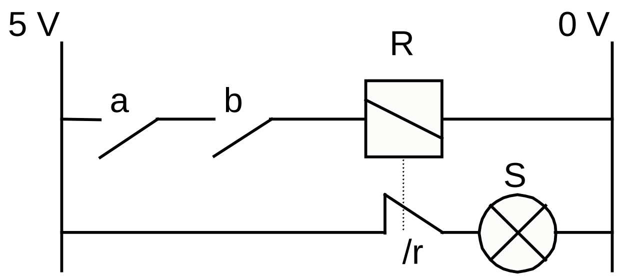


Schéma électrique

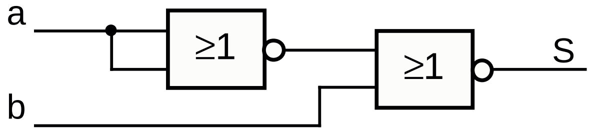


3.5.3. Intérêt des théorèmes de De Morgan.

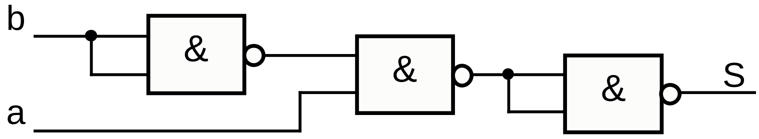
Toute fonction logique peut-être réalisée en utilisant uniquement des cellules NOR ou NAND. Cela permet de réduire les types de composants nécessaires et d'optimiser les circuits électroniques retenus. En effet, sur un circuit électronique, on trouve généralement un minimum de quatre cellules identiques.
Les opérateurs NOR et NAND sont dits complets.

Exemple : $S=f(a,b)=a.\overline{b}$

- Réalisation du circuit n'utilisant que des cellules NOR.
  $S=a.\overline{b}=\overline{\overline{a.\overline{b}}}=\overline{\overline{a}+b}=\overline{\overline{a+a}+b}$
  Logigramme correspondant :



- Réalisation du circuit n'utilisant que des cellules NAND.
  $S=a.\overline{b}=\overline{\overline{a.\overline{b}}}=\overline{\overline{a.\overline{b.b}}}$
  Logigramme correspondant :



3.6. Fonctions logiques d'une variable.

Il s'agit de lister toutes les fonctions logiques d'une variable.

Pour les 2 états différents (0 et 1) de l'entrée, la sortie peut prendre les valeurs 0 ou 1. Il existe donc $2^{2=}4$ fonctions logiques d'une variable : $Si=f(e)$.

e	0	1		
S1	0	0	S1=0	Fonction nulle
S2	0	1	S2=e	Fonction OUI
S3	1	0	S3=/e	Fonction NON
S4	1	1	S4=1	Fonction unité

3.7. Fonctions logiques de deux variables.

Pour les 4 états différents des entrées, la sortie peut prendre les valeurs 0 ou 1. Il existe donc 4^2=16 fonctions logiques de 2 variables différentes : $Si=f(a,b)$.

a	0	1	0	1		
b	0	0	1	1		
S1	0	0	0	0	$S1=0$	Nulle
S2	0	0	0	1	$S2=a\cdot b$	Et
S3	0	0	1	0	$S3=\overline{a}\cdot b$	Inhibition
S4	0	0	1	1	$S4=b$	
S5	0	1	0	0	$S5=a\cdot\overline{b}$	Inhibition
S6	0	1	0	1	$S6=a$	
S7	0	1	1	0	$S7=a\oplus b$	Ou Exclusif
S8	0	1	1	1	$S8=a+b$	Ou
S9	1	0	0	0	$S9=\overline{a+b}$	Nor
S10	1	0	0	1	$S10=\overline{a\oplus b}$	Identité
S11	1	0	1	0	$S11=\overline{a}$	
S12	1	0	1	1	$S12=\overline{a}+b$	Implication
S13	1	1	0	0	$S13=\overline{b}$	
S14	1	1	0	1	$S14=a+\overline{b}$	Implication
S15	1	1	1	0	$S15=\overline{a\cdot b}$	Nand
S16	1	1	1	1	$S16=1$	Unité

3.8. Expressions canoniques d'une fonction logique.

Elles sont utilisées pour trouver l'expression logique d'une fonction à partir de la table de vérité.

3.8.1. Somme de produits canoniques.

- Toute fonction d'une variable s'écrit : $S=f(e)=\overline{e}.f(0)+e.f(1)$
  Exemple : $S3=f(e)=\overline{e}.1+e.0=\overline{e}$

- Toute fonction de deux variables s'écrit : $S=f(a,b)=\overline{a}.\overline{b}.f(0,0)+a.\overline{b}.f(1,0)+\overline{a}.b.f(0,1)+a.b.f(1,1)$
  Exemple : $S11=f(a,b)=\overline{a}.\overline{b}.1+a.\overline{b}.0+\overline{a}.b.1+a.b.0=\overline{a}.\overline{b}+\overline{a}.b=\overline{a}$

3.8.2. Produit de sommes canoniques.

- Toute fonction d'une variable s'écrit : $S=f(e)=[e+f(0)][\overline{e}+f(1)]$
  Exemple : $S3=[e+1][\overline{e}+0]=\overline{e}$

- Toute fonction de deux variables s'écrit :
  $S=f(a,b)=[a+b+f(0,0)][\overline{a}+b+f(1,0)][a+\overline{b}+f(0,1)][\overline{a}+\overline{b}+f(1,1)]$
  Exemple : $S11=[a+b+1][\overline{a}+b+0][a+\overline{b}+1][\overline{a}+\overline{b}+0]=[\overline{a}+b][\overline{a}+\overline{b}]=\overline{a}+\overline{a}.\overline{b}+b.\overline{a}+b.\overline{b}=\overline{a}$

4. Simplification des fonctions logiques.

4.1. Le code GRAY.

Le code Gray, encore appelé code binaire symétrique ou réfléchi, permet de coder les nombres en utilisant les chiffres binaires 0 et 1. Le codage est tel que :

- Il permet de passer d'un nombre au suivant ou au précédent en ne changeant d'état qu'un seul digit de l'expression.
- Des « symétries » dans le code conduisent à la même remarque pour deux nombres « symétriques ».

Le tableau suivant donne la correspondance avec les nombres binaires naturels, décimaux et hexadécimaux :

Hexadécimal	Décimal		Binaire				Gray			
16^0	10^1	10^0	2^3	2^2	2^1	2^0				
0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	1
2	0	2	0	0	1	0	0	0	1	1
3	0	3	0	0	1	1	0	0	1	0
4	0	4	0	1	0	0	0	1	1	0
5	0	5	0	1	0	1	0	1	1	1
6	0	6	0	1	1	0	0	1	0	1
7	0	7	0	1	1	1	0	1	0	0
8	0	8	1	0	0	0	1	1	0	0
9	0	9	1	0	0	1	1	1	0	1
A	1	0	1	0	1	0	1	1	1	1
B	1	1	1	0	1	1	1	1	1	0
C	1	2	1	1	0	0	1	0	1	0
D	1	3	1	1	0	1	1	0	1	1
E	1	4	1	1	1	0	1	0	0	1
F	1	5	1	1	1	1	1	0	0	0

4.2. Tableau de KARNAUGH d'une fonction logique.

C'est une table de vérité matricielle de la fonction logique. Le codage des lignes et des colonnes se fait selon le code gray. Les cases de la matrice contiennent l'état de la fonction logique.

Exemple : $S=f(a,b,c,d)=\overline{b}.c.d+b.\overline{c}.d+a.\overline{b}.c+\overline{a}.b.\overline{c}+a.b+\overline{a}.c$

	a	0	0	1	1
	b	0	1	1	0
c	d				
0	0	0	1	1	0
0	1	0	1	1	0
1	1	1	1	1	1
1	0	1	1	1	1

4.3. Simplification par la méthode de KARNAUGH.

On exploite les particularités du code Gray.

Soit un système à q entrées. On considère les regroupements de 2^n cases « symétriques » à l'état 1 dans le tableau de Karnaugh. Chaque regroupement représente le produit logique des (q-n) variables d'entrée qui ne changent pas d'état. Les autres variables sont indifférentes pour le regroupement considéré.

Remarques :

- Les regroupements peuvent se superposer.
- Il faut prendre en compte, par des regroupements, l'ensemble des cases à l'état 1.
- Il faut chercher les regroupements les plus gros. Ils donnent des produits logiques les plus simples.
- En raisonnant sur les cases à l'état 0, on obtient l'expression du complément de la fonction considérée.

Exemple :

	a	0	0	1	1
c	d / b	0	1	1	0
0	0	0	1	1	0
0	1	0	1	1	0
1	1	1	1	1	1
1	0	1	1	1	1

$$S=f(a,b,c,d)=b+c$$

5. Réalisation pratique des opérateurs logiques de base.

5.1. Technologie électrique.

5.1.1. Le relais électromagnétique.

Le relais électromagnétique (ou plus simplement relais) est utilisé pour réaliser des *réseaux à contacts.* Le schéma de principe d'un relais est donné figure 1. Il est constitué d'un électro-aimant et d'une barrette mobile qui supporte des paires de contacts (une seule paire est représentée sur le schéma). Ces paires de contacts sont donc couplées mécaniquement, mais elles sont électriquement indépendantes. Chaque paire est constituée de deux contacts qui ont un point commun. On distingue le *contact travail* et le *contact repos* (cf. figure 1.). Il y a deux types de bornes électriques :

les 2 bornes d'alimentation de la bobine, notées A_1 et A_2,

les 3 bornes de chaque paire de contacts, notées a_1, a_2, et a_3.

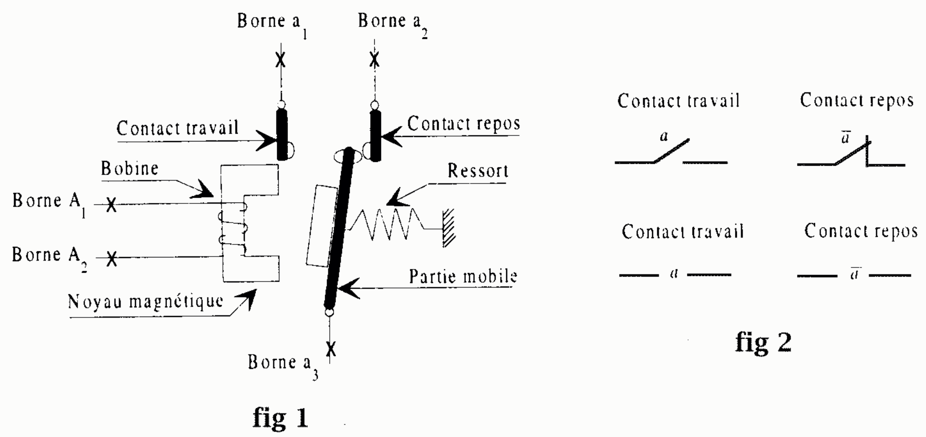


fig 1

fig 2

Lorsqu'un courant électrique traverse la bobine (bornes A_1 et A_2 sous tension), la partie mobile est attirée et le *contact travail* est fermé (un courant électrique peut circuler entre les bornes a_1 et a_3), tandis que le *contact repos* est ouvert (aucun courant ne peut circuler entre les bornes a_2 et a_3). Lorsqu'aucun courant ne traverse la bobine, la partie mobile est rappelée par le ressort et le *contact travail* est ouvert, tandis que le *contact repos* est fermé.

Dans un schéma électrique de réseaux de contacts, on adoptera l'une des deux conventions représentées sur la figure 2 dans laquelle *a* représente un contact travail et */a* un contact repos.

Exemples :
Sur le schéma de la figure 3a, la lampe sera allumée si *a* = 1, c'est-à-dire si le contact travail est fermé et, pour cela, si la bobine A (non représentée ici) est alimentée (A = 1). On a donc réalisé la fonction logique L = *a.*

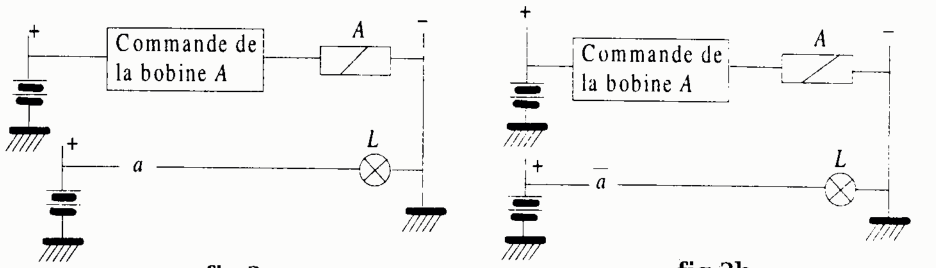


fig 3a **fig 3b**

Par contre, sur le schéma de la figure 3b, la lampe sera allumée si */a* = 1, ou *a* = 0, c'est-à-dire si le contact repos est fermé et, pour cela, si la bobine A n'est pas alimentée (A = 0). On a réalisé la fonction *L = /a.*

Remarque : Nous avons volontairement distingué l'alimentation de la bobine et de sa commande de celle de la lampe. Cela correspond à la réalité de nombreuses applications. En effet, le relais est aussi un amplificateur de puissance, et, par exemple, l'alimentation de la bobine pourra se faire en basse tension, alors que celle de la lampe se fera sous une tension beaucoup plus élevée. De plus, ces tensions peuvent être continues ou alternatives.

5.1.2. Opérateurs de base (ET, OU).

Avec des contacts, on réalise simplement les opérateurs de base.
Dans le réseau de la figue 4a, la lampe s'allume si et seulement si les deux contacts a et b sont fermés. Ce réseau réalise donc la fonction L = a.b.
Dans le réseau de la figue 4b, la lampe s'allume si et seulement si l'un des deux contacts a ou b au moins est fermé. Ce réseau réalise donc la fonction L = a+b.

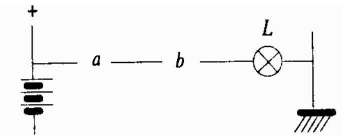


fig 4a

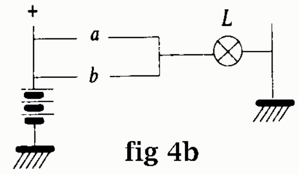


fig 4b

5.2. Technologie électronique.

Les systèmes digitaux modernes, tels que ceux que l'on trouve dans les ordinateurs, sont constitués d'un très grand nombre de composants qui contiennent chacun un très petit nombre d'éléments différents. Ces éléments de base sont les portes logiques. Elles ont un nombre fixé d'entrées et réalisent une opération logique sur ces entrées.
Afin de pouvoir réaliser n'importe quelle expression logique, il faut disposer d'un ensemble de portes capables de réaliser toutes les opérations de l'algèbre binaire. Le premier ensemble qui vient naturellement à l'esprit est constitué des trois opérateurs de base de l'algèbre binaire : NON, ET, OU. Nous verrons également qu'il existe d'autres ensembles, plus réduits et surtout beaucoup plus utilisés, qui permettent également de réaliser toutes les fonctions logiques ; ce sont les opérateurs NAND et NOR.

5.2.1. Le transistor.

Le module de base de la technologie électronique est le transistor (figure 5).
Les trois électrodes sont (figure 6) :

- L'émetteur E ;
- La base B
- Le collecteur C.

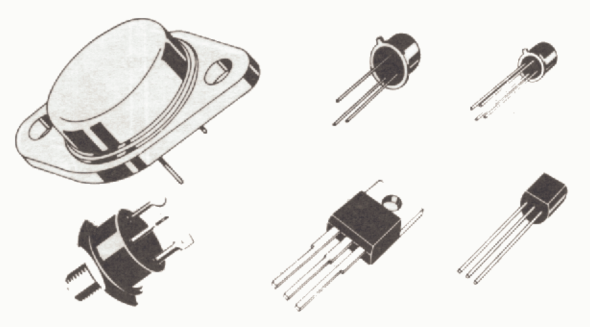


fig 5 (doc. Thomson - CSF).

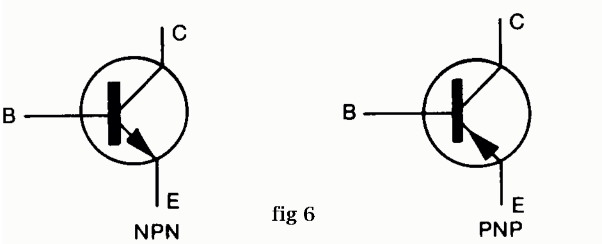


fig 6

La flèche indique toujours l'émetteur.
Les sens de la flèche permet de reconnaître le type : NPN (ne pénètre pas) ou PNP (pénètre)

Fonctionnement :
Le courant collecteur IC dépend du courant base IB (figure 7 et 8).
Faisons varier le courant base IB et notons les variations correspondantes de IC.

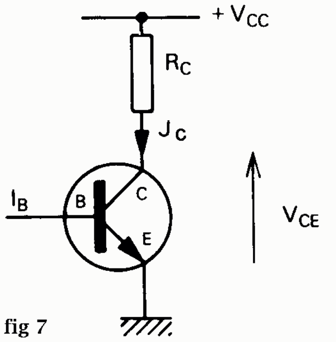


fig 7

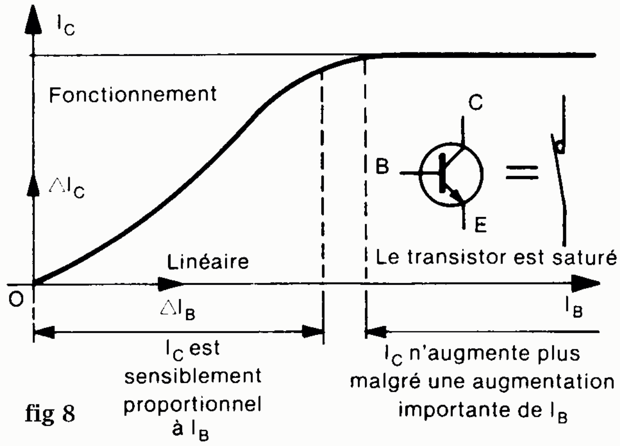


fig 8

Transistor saturé : à partir d'une certaine valeur de IB, le courant IC se stabilise et reste constant, même si IB continue à croître. Le transistor conduit totalement, il se comporte comme un interrupteur fermé. On dit qu'il est saturé (VCE ≈ 0).
Transistor bloqué : pour un courant IB nul, le courant collecteur est également nul. Le transistor se comporte comme un interrupteur ouvert. On dit qu'il est bloqué (VCE = VCC).

Utilisation en composant logique (figure 9)
a est la variable d'entrée et S la fonction de sortie.

Si a = 0, U_{CE} = 5 v (le transistor est bloqué) et S = 1 ;
Si a = 1, U_{CE} = 0,1 à 0,8 v (le transistor est saturé) et S = 0.

Dans ces conditions de branchement, un transistor se comporte donc comme une fonction NON.

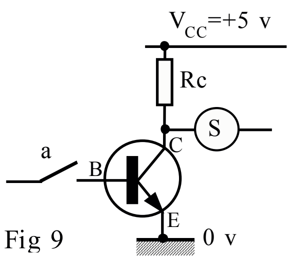


Fig 9

5.2.2.Opérateurs NAND et NOR.

Le transistor, module de base, permet de constituer, sous forme intégrée, différentes fonctions logiques appelées portes.
Les plus utilisées sont les portes NAND et NOR :

Dans le réseau de la figure 10, la sortie S est active si a ou b est ouvert. Ce réseau réalise donc la fonction : $S = f(a,b) = a\,NAND\,b = \overline{a.b} = \overline{a} + \overline{b}$

Dans le réseau de la figure 11, la sortie S est active si a et b sont ouverts. Ce réseau réalise donc la fonction : $S=f(a,b)=aNORb=\overline{a+b}=\overline{a}.\overline{b}$

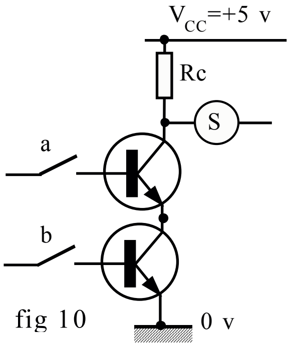


fig 10

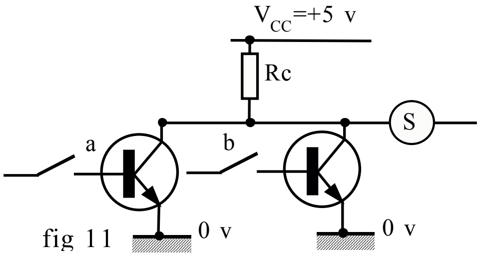


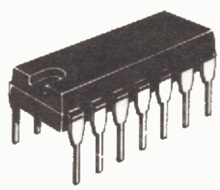
fig 11

5.2.3.Les composants technologiques.

Les composants élémentaires (transistors, résistances…) peuvent être montés sur une carte et constituer ainsi un circuit imprimé. La soudure de ces composants est cependant longue et délicate, aussi préfère-t-on utiliser des circuits intégrés miniaturisés. Il sont embrochables et comportent généralement plusieurs portes.

Exemples :

circuit intégré : circuit Logique TTL



TO-116
(CB-2)

(doc. Thomson - CSF).

circuit intégré quadruple porte NOR à deux entrées :

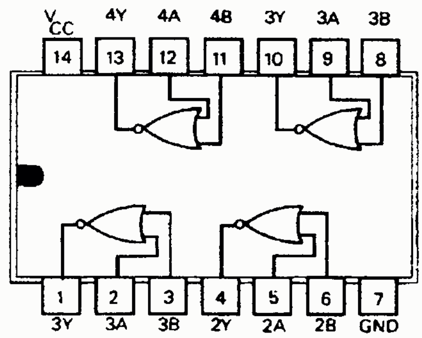
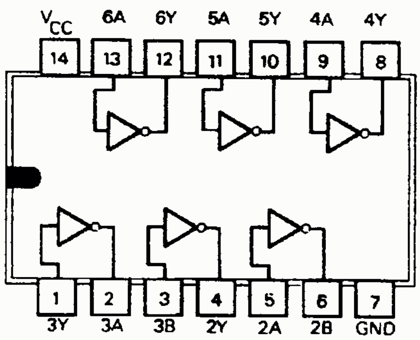
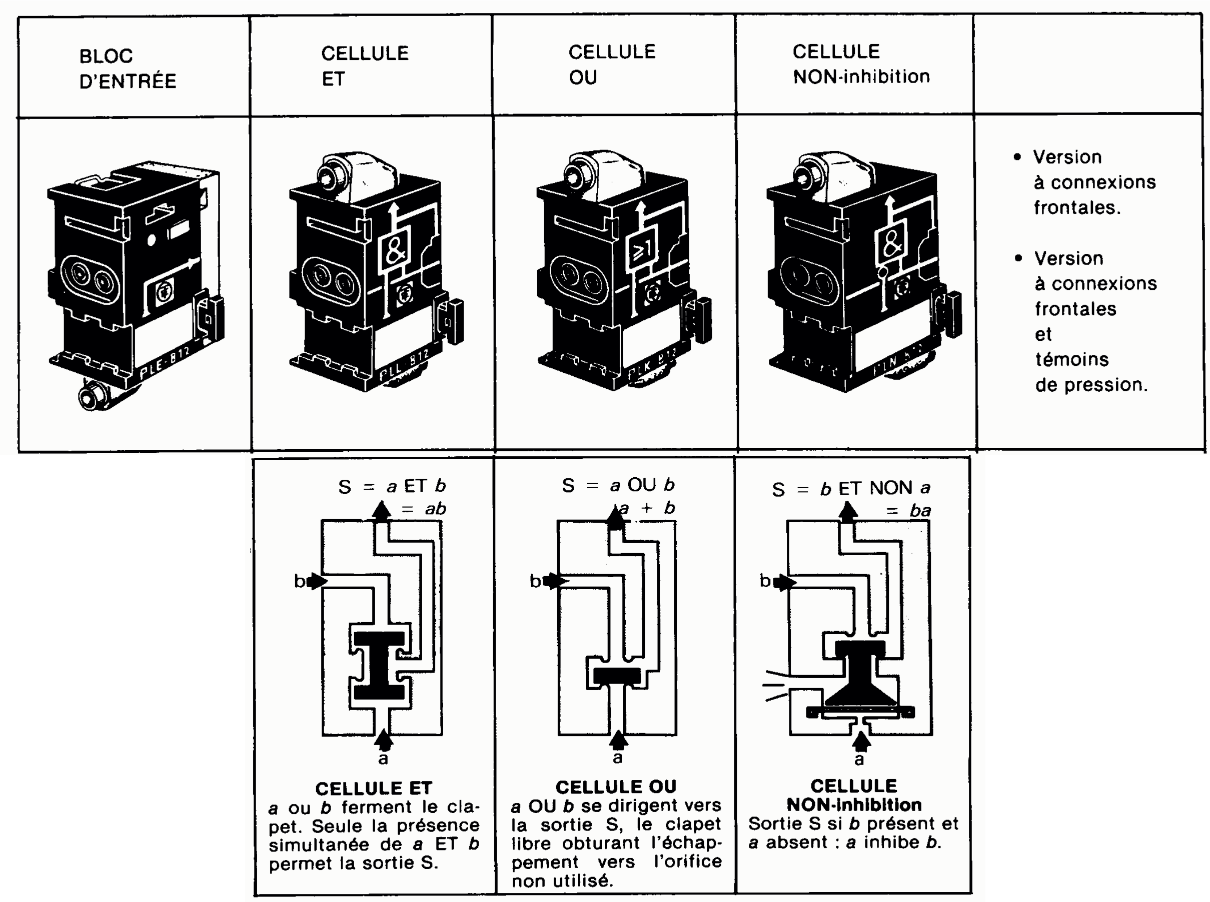


fig 12

circuit intégré six portes NON (inverseurs) :



5.3. Technologie pneumatique.



(D'après Télémécanique)