



Systèmes d'Information Numérique

SI : Systèmes combinatoires

RI.07 Partie I – BUTI GEII

Michelle VIEIRA (responsable du module)

Contact : michelle.vieira@univ-grenoble-alpes.fr

2024-2025

Planning des séances

Sem	CTD SIN I (18h)	TP SIN I (3h)	CC, DS
36	Présentation SIN I Chap. 1 Variables logiques		
36	Chap. 1 Variables logiques		
37	Chap. 2 Fonctions logiques		CCI
37	Chap. 2 Fonctions logiques		
38	Chap. 2 Fonctions logiques		
39	Chap. 3 Technologie des CI		
40	Chap. 4 Fonctions combinatoires		CC2
41	Chap. 4 Fonctions combinatoires		
42		TP Câblage	
43			
44			
45	Chap. 5 Arithmétique binaire		
46			DS 14/11

Acquis d'apprentissage visés au semestre I

A l'issue de cette 1^{ère} partie de la ressource Automatismes I, les étudiants seront capables de :

- Convertir un nombre décimal en un nombre codé en binaire ou en hexadécimal et inversement,
- Décomposer une fonction logique en blocs combinatoires (logigramme)
- Ecrire une table de vérité à partir d'un besoin client simple et d'en déduire les équations logiques
- Mettre en œuvre un logigramme simple à l'aide d'un logiciel approprié et vérifier son action sur une unité de traitement

Evaluation de SIN I

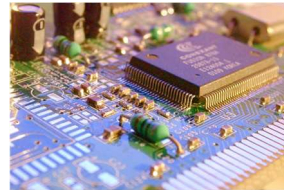
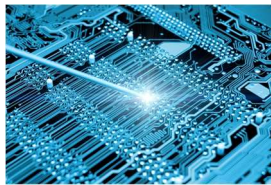
• Devoir surveillé de 1h30 en semaine 46 le 14/11

- Coefficient 1
- Modalité de l'épreuve :
 - Seule une feuille A4 recto/verso de notes personnelles est autorisée
 - Pas de calculatrice
- En **cas d'absence justifiée au DS**, en plus de fournir votre justificatif au secrétariat, vous devez **obligatoirement** prévenir Mme Vieira par mail (michelle.vieira@univ-grenoble-alpes.fr) **au plus tard 24h après le DS** pour pouvoir être autorisé à participer au rattrapage

• Moyenne des 2 contrôles continus et du TP

- coefficient 0,5
- Une absence injustifiée à un CC et au TP vous faudra la note de 0 donc pensez à fournir à votre enseignant de TD votre justificatif

Chapitre 1 : Variables logiques



Plan du chapitre :

- I. Introduction
- II. Codage et numération

2

I. Introduction :

- une **variable logique** ne peut prendre que :
 - la valeur « 0 », associée à la notion de faux,
 - ou la valeur « 1 », associée à la notion de vrai
- quand on relie des variables logiques par des **opérateurs logiques**, on obtient une **fonction logique**, l'ensemble des opérations forme **l'algèbre de Boole**

George Boole : mathématicien britannique 1815-1864

→
Logique binaire



Mode de raisonnement **très bien adapté** à l'électricité et à l'électronique
→ **on l'utilise** comme formalisme !

3

I. Introduction : la base 2

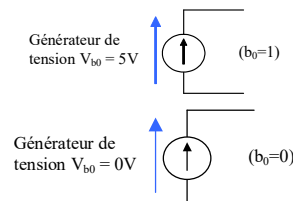
Le codage des nombres est fait en **base 2** : $N \text{ (base 10)} = 36$
→ à partir de deux chiffres : **le zéro et le un** $N \text{ (base 2)} = 100100$

Un **chiffre** binaire → a **binary digit** → a « **bit** » → un « bit »

- La **représentation électrique** élémentaire d'un bit

Le support de l'information → C'est une **tension**

Il y a donc deux états distincts du bit → deux **niveaux de tension** nécessaires



4

I. Introduction : représentation des nombres

Il faut faire la distinction entre:

- État logique **0** ou **1**
- Niveau logique (qui correspond à un niveau de tension)

En électronique numérique :

- H comme High correspond à la tension la plus élevée (5V ou 3,3V)
- L comme Low correspond à la tension la plus faible (0V)

Niveau électronique de tension	Etat logique correspondant	
	Logique dite positive	Logique dite négative
H (5 Volts par exemple)	« 1 »	« 0 »
L (0 Volt)	« 0 »	« 1 »

En Automatismes : on utilisera la logique POSITIVE (sauf contre indication)

5

I. Introduction : représentation des nombres

En électronique numérique :

Un **chiffre** binaire est représenté par **une** tension

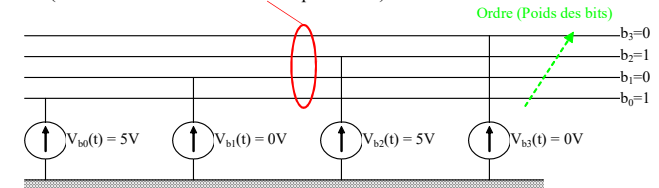
Un **nombre** binaire sera donc représenté → par **DES** tensions

→ On considère **plusieurs fils** (et donc **plusieurs générateurs** associés);

Exemple sur 4 bits:

On appelle cela un **bus**

(un nombre devient en électronique un bus)



Attention → les fils sont **ordonnés** ! → on parle du **poids** des bits, des fils

→ Le bit de **poids** le plus **faible**

→ le **LSB** (least significant bit)

→ Le bit de **poids** le plus **fort**

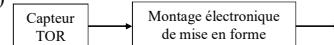
→ le **MSB** (most significant bit)

6

I. Introduction : d'où viennent les variables ou grandeurs logiques ?

D'une manière générale les variables logiques viennent :

- De **sources de tension fixes** : câblage à 0V ou bien câblage à 5V
- De **capteurs TOR** (Tout Ou Rien) raccordés à une électronique de mise en forme (qui **délivre un niveau logique 0 ou 5V**)



- D' **interrupteurs** ou **boutons-poussoirs** (qui **délivre une tension 0V ou 5V**) voir diapo 11

Dans ce cas, les variables logiques correspondent à des **ENTREES** :

- En **électronique numérique** : on les nomme souvent par une minuscule ou bien par le nom de la patte du composant correspondant à leur adresse (voir Info 2)
- En **automatisme** : suivant le constructeur ils sont repérées par %I (comme Input) pour Schneider Electric ou tout simplement I pour Eaton suivies d'un ou plusieurs indices correspondant à leur adresse physique et seront affectées d'un commentaire (voir doc EASYSOFT)

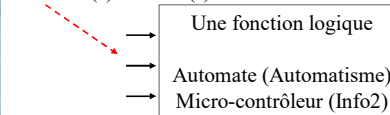
7

I. Introduction : schéma bloc

Schéma Fonctionnel (ou schéma bloc) d'un système automatisé:

À partir de la (ou des) variable(s) d'entrée(s)

La fonction fournit la grandeur logique de sortie



Il y a deux types de blocs fonctions:

1^{er} semestre

$$S = f(E_i)$$

Les fonctions qui ne dépendent que des entrées
→ toujours le même comportement pour des entrées données

→ **Fonctions combinatoires**

$$S = f(E_i, S_{t-1})$$

2^{ème} semestre

Les fonctions dont la sortie dépend des entrées (bien sur) et :

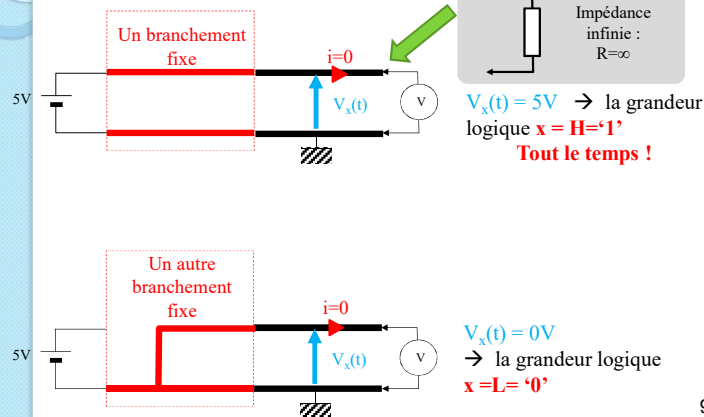
→ De ce qui c'est passé avant dans le temps : S_{t-1}

→ **Fonctions séquentielles** (dépend de l'état du système)

8

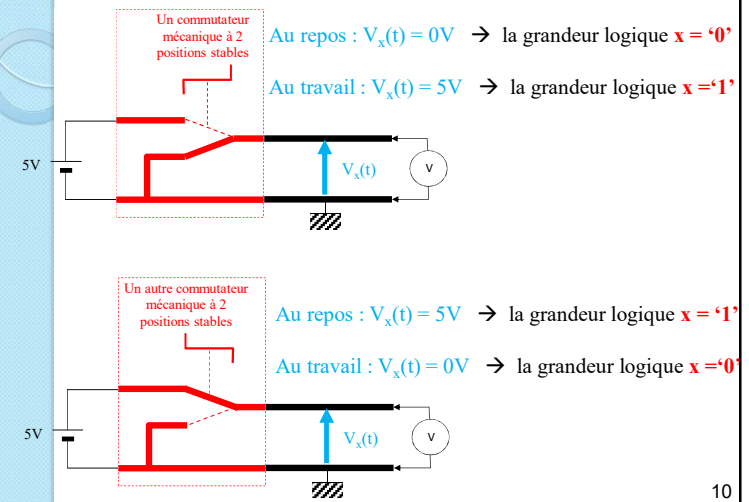
I. Introduction : exemples de montages de base pour des entrées

Lecture de la tension (au voltmètre) du bit, de l'information:



9

I. Introduction : cas d'un commutateur



10

I. Introduction : 4 autres montages de base

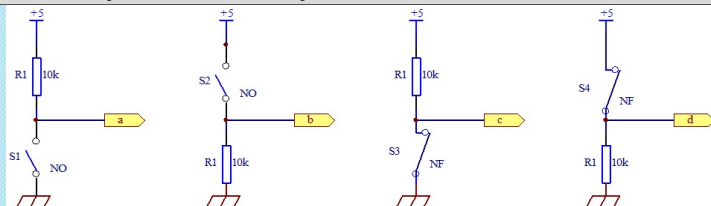
Si l'appareillage n'est pas manœuvré son état est dit « normal » ou « au repos »
Lorsque l'appareillage (ici des BP) est manœuvré son état est dit « au travail »

Attention : certains appareillage sont ouverts à l'état « repos » (les NO), d'autres sont « fermés » à l'état repos (les NF ou NC) ...

NO = Normalement Ouvert = Normaly Open

NF = NC = Normalement Fermé = Normaly Closed

Remarque : par convention de représentation des schémas électriques, un appareillage est **TOUJOURS représenté dans son état « repos »**



Indiquez pour chaque montage quel est le niveau des variables a, b, c, d lorsque l'appareillage est « activé » $\rightarrow$ On associe alors à l'état « travail » de l'appareillage l'état « actif » de la variable.

État actif de 'a'

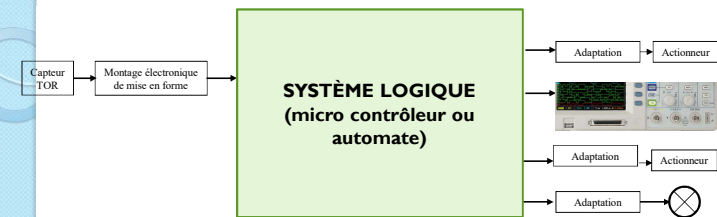
État actif de 'b'

État actif de 'c'

État actif de 'd'

11

I. Introduction : où vont les grandeurs logiques ?



Les grandeurs logiques (simple fil ou bus) peuvent aller vers :

$\rightarrow$ des **blocs ou fonctions INTERNES**

Les grandeurs logiques peuvent être tout simplement des SORTIES reliées physiquement à :

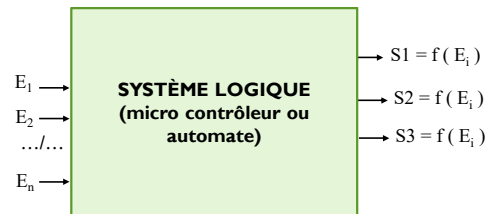
$\rightarrow$ des **actionneurs** permettant de **piloter un procédé physique** (vannes, relais ...) au travers d'une adaptation (montage ou carte électronique d'adaptation ou **d'amplification** de puissance)

$\rightarrow$ des équipements d'IHM (Interface Homme Machine) : voyants, LEDs, afficheurs ... au travers d'une adaptation

$\rightarrow$ du matériel d'instrumentation et de **mesure** : oscilloscope, voltmètres

12

I. Introduction : où vont les grandeurs logiques ?



Dans le cas où les fonctions logiques correspondent à des **SORTIES**

- En électronique numérique : on les nomme souvent par une MAJUSCULE ou bien par le nom de la patte du composant (voir Info 2)
- En automatisme : suivant le constructeur ils sont repérés par %Q pour Schneider Electric ou tout simplement Q pour Eaton (voir EasySoft) suivies d'un ou plusieurs indices et affectées d'un commentaire.

13

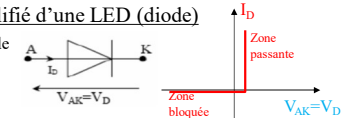
I. Introduction : montages de base pour les sorties

Dans ce cas les fonctions logiques correspondent à des SORTIES

- Exemple classique en électronique numérique : sortie = fonction matérialisée par une LED

Rappel sur le fonctionnement simplifié d'une LED (diode)

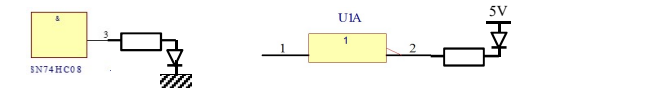
La LED s'allume si la diode est passante (le + sur l'anode)
La LED est éteinte si la diode est bloquée (le + sur la cathode)



Proposez un montage électronique pour chacun des montages de telle sorte que la LED soit allumée lorsque la sortie est à son état actif

La première sortie est **active à l'état HAUT** (noté 'H' comme High)

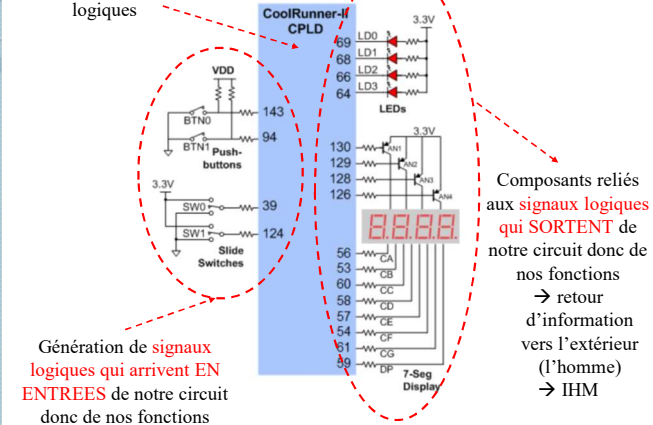
La seconde sortie est **active à l'état BAS** (noté 'L' comme Low) car présence en sortie du symbole ▽



14

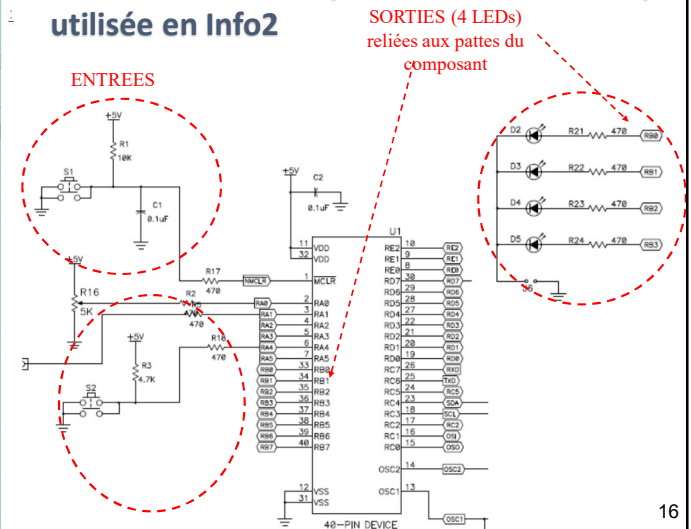
I. Introduction : exemple de carte électronique

Circuit électronique qui réalise des fonctions logiques



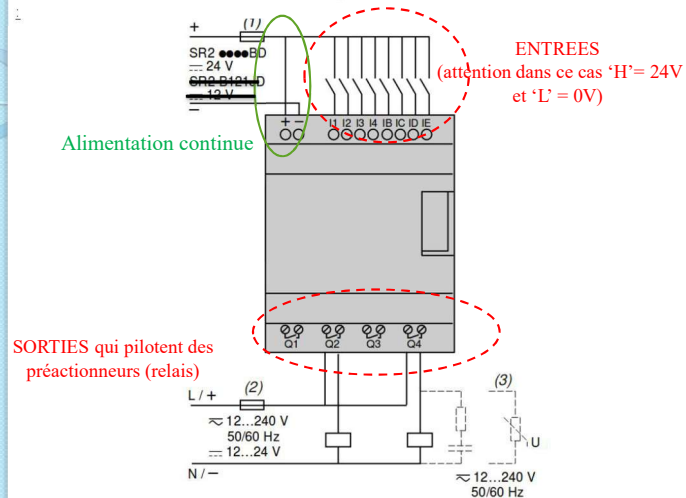
15

I. Introduction : exemple de carte électronique utilisée en Info2



16

I. Introduction : exemple en automatisme



17

Plan du chapitre :

- I. Introduction
- II. Codage et numération

18

II. Codage et Numération

- **Quelques définitions**
- **Systèmes de numération :**
 - décimal, binaire, hexadécimal,...
- **Différents codages binaires:**
 - Code BCD, GRAY,...

19

Quelques définitions

- **Mot:** séquence de symboles ordonnés et représentant une information.
Les symboles appartiennent à un alphabet.
Ex : « deux mille quatorze », 2014, MMXIV
- **Mot binaire:** mot constitué avec un alphabet binaire dont les symboles {0, 1} sont appelés **bits** (binary digit) et peuvent être véhiculés sur des lignes de plusieurs fils appelés **bus**.
Ex : 11111011110
- **Quartet (Nibble):** mot binaire de 4 bits
- **Octet (Byte):** mot binaire de 8 bits

20

Quelques définitions

En info2

- **char**: mot binaire de 8 bits
- **unsigned short int**: mot de 16 bits (non signé donc >0)
- **unsigned short long int**: mot de 24 bits (non signé)
- **unsigned long int**: mot de 32 bits (non signé donc >0)

Note: ces variables sont implémentées (situées) dans un plan mémoire (zone mémoire dédiée)

21

Quelques définitions

En Automatismes avec EASYSOFT

- **MB (Byte)** : mot binaire de 8 bits
- **MW (Word)** : mot de 16 bits
- **DW (Double Word)**: mot de 32 bits

Exemples: MB13 correspond à l'octet situé à l'emplacement mémoire 13

MW26 correspond au mot de 16 bits situé à l'emplacement mémoire 26

Voir documentation EASYSOFT § 6.1.15

En Automatismes avec SCHNEIDER ELECTRIC

- **%MW (Word)** : mot de 16 bits

Exemple: %MW0 le mot de 16 bits se situe à l'adresse 0 du plan mémoire

Note: ces variables sont implémentées (situées) dans un plan mémoire (zone mémoire dédiée) et M pour memory.

22

Quelques définitions

- **Codage** : action de faire correspondre une séquence de symboles à une information **numérique ou alphanumérique**.
- **Code** : groupe de symboles.

Dans ce chapitre, nous n'aborderons que le codage des **nombre entiers naturels** c'est à dire positifs.

23

II. Codage et Numération

- Quelques définitions
- **Systèmes de numération** :
 - décimal, binaire, hexadécimal,...
- Différents codages binaires:
 - Code BCD, GRAY,...

24

Système de numération : Système décimal

- Système à base 10
- 10 chiffres différents $\{0,1,2,3,4,5,6,7,8,9\}$
- Décomposition suivant les puissances de 10:

$$(1789)_{10} = 1.10^3 + 7.10^2 + 8.10^1 + 9.10^0$$

$$(390)_{10} = 3.10^2 + 9.10^1 + 0.10^0$$

- Système difficile à adapter aux équipements électroniques car ceux-ci devraient fonctionner avec 10 niveaux de tension différents.

25

Système de numération : Principe

Numération : juxtaposition ordonnée de symboles appelés chiffres. Le nombre de chiffres possibles définit la base du système de numération.

Base b : base comportant b chiffres différents

$$\{S_0, S_1, \dots, S_{b-1}\}.$$

Ecriture d'un nombre N dans la base b :

Le nombre N se décompose dans la base b suivant les puissances de b:

$$N = a_{n-1}.b^{n-1} + \dots + a_2.b^2 + a_1.b^1 + a_0.b^0$$

$$= (a_{n-1} \dots a_1 a_0)_b \text{ est un mot de } n \text{ symboles où}$$

chaque $a_j \in \{S_0, S_1, \dots, S_{b-1}\}$ pour $j=0, \dots, n-1$

26

Système de numération : Principe

Soit $N = (a_{n-1} \dots a_1 a_0)_b$

- Chaque chiffre a_j est repéré par son **rang** (position dans le mot) et son **poids** (valeur): le chiffre de rang j a pour poids b^j .
- Le chiffre le plus à gauche a_{n-1} est le plus significatif. On dit qu'il est de **poids fort**.
- Le chiffre le plus à droite a_0 est le moins significatif. On dit qu'il est de **poids faible**.

27

Système de numération : Système binaire ou binaire naturel

- Système binaire ou système à base 2 plus simple et adapté aux systèmes logiques basés sur le « tout ou rien ».
- *Rappel*: 2 chiffres $\{0,1\}$ appelés **bits** (binary digit).
- **Ecriture d'un nombre dans la base 2** : (0b)

$$(0010 \ 1001)_2 = 0b0010 \ 1001$$

Bit de poids fort (MSB) →

← Bit de poids faible (LSB)

Rappel:

- LSB (less significant bit) : Bit le plus à droite du mot binaire
- MSB (most significant bit) : Bit le plus à gauche du mot binaire

28

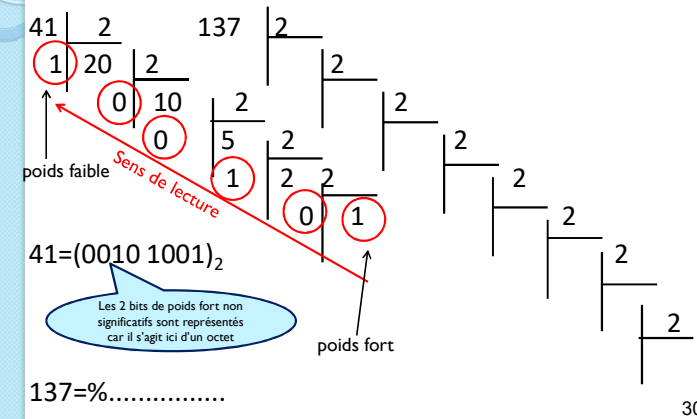
Système de numération : Système binaire

Conversion base 10 vers base 2 par divisions successives

- obtenue par divisions successives par 2 jusqu'à obtenir un quotient nul.
- Le nombre dans la base 2 correspond aux restes des divisions faites dans le sens inverse où ils ont été obtenus (le dernier reste est le MSB).

29

Système de numération : Système binaire



30

Système de numération : Système binaire

Conversion base 10 vers base 2 par soustractions successives

- obtenue par soustractions successives des puissances de 2 maximales jusqu'à obtenir un résultat nul.
- Le nombre dans la base 2 est obtenu en représentant par 1 les poids présents et par 0 les poids absents.

31

Système de numération : Système binaire

Exemple : représentation de (41)₁₀ en base 2

$$\begin{array}{r}
 41 \\
 - 32 \\
 \hline
 9 \\
 - 8 \\
 \hline
 1 \\
 - 1 \\
 \hline
 0
 \end{array}$$

... 64 32 16 8 4 2 1

 x x x

$(41)_{10} = (00101001)_2$

- Convertir (137)₁₀ en binaire

32

Système de numération : Système binaire

- **Conversion base 2 vers base 10 :**

On utilise la valeur décimale des différentes puissances de 2 pour faire la somme des termes élémentaires $a_i \cdot 2^i$

ex :

2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1
1	0	1	0	0	1

$$\begin{aligned}
 N &= 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\
 &= 32 + 8 + 1 \\
 &= 41
 \end{aligned}$$

33

Système de numération : Système hexadécimal

- Système à base $16=2^4$

- 16 chiffres différents

{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

- Très utilisé car permet d'abréger les mots binaires

Un chiffre hexadécimal (ou digits hexadécimal) correspond donc à un mot binaire de 4 bits

Ex : 6 = 0b0110 F = 0b1111 D=0b.....

- **Ecriture d'un nombre dans la base 16:**

$$(6FD)_{16} = \$6FD = 0x6FD = 6FDh$$

34

Système de numération : Système hexadécimal

- **Conversion base 16 vers base 2 puis base 10:**

car les puissances de 2 sont plus connues que les puissances de 16...

ex :

$$\begin{aligned}
 0x6FD &= (0110 \ 1111 \ 1101)_2 \\
 &= 1024 + 512 + 128 + 64 + 32 + 16 + 8 + 4 + 1 \\
 &= (1789)_{10}
 \end{aligned}$$

35

Système de numération : Système hexadécimal

- **Conversion base 16 vers base 10 :**

On utilise la valeur décimale des différentes puissances de 16 pour faire la somme des termes élémentaires $a_i \cdot 16^i$

16^2	16^1	16^0
256	16	1
6	F	D

$$N = 6 \cdot 16^2 + 15 \cdot 16^1 + 13 \cdot 16^0 = 1789$$

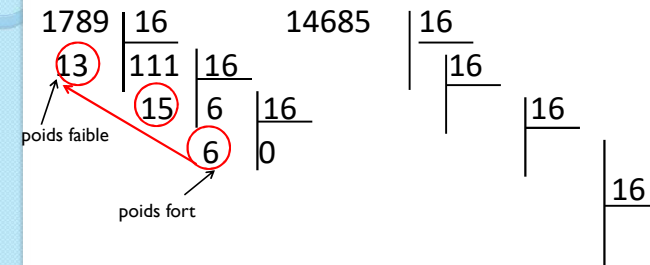
36

Système de numération : Système hexadécimal

- **Conversion base 10 vers base 16 par divisions successives**
 - obtenue par divisions successives par 16 jusqu'à obtenir un quotient nul.
 - le nombre dans la base 16 correspond aux restes des divisions faites dans le sens inverse où ils ont été obtenus (le dernier reste est le MSB).

37

Système de numération : Système hexadécimal



1789 = (6FD)₁₆
14685 = 0x.....

38

Système de numération : Tableau de conversion

décimal	hexa	binaire	décimal	hexa	binaire
0	0	0000	10	A	1010
1	1	0001	11	B	1011
2	2	0010	12	C	1100
3	3	0011	13	D	1101
4	4	0100	14	E	1110
5	5	0101	15	F	1111
6	6	0110	16	10	1 0000
7	7	0111	17	11	1 0001
8	8	1000	31	1F	1 1111
9	9	1001	32	20	10 0000

39

II. Codage et Numération

- Quelques définitions
- Systèmes de numération :
 - décimal, hexadécimal,...
- **Différents codages binaires:**
 - **Code binaire naturel, BCD, GRAY,...**

40

Différents codes binaires : Code BCD

- Code Décimal Codé en Binaire
- Chaque chiffre d'une valeur décimale est codé séparément en binaire sur 4 bits

Exemple: Milliers Centaines Dizaines Unité
(M C D U)

- Ex : $(5250)_{10} = (0101\ 0010\ 0101\ 0000)_{bcd}$
 $(1000\ 0110\ 0111\ 1001)_{bcd} = (8679)_{10}$
- Non utilisation de 6 combinaisons binaires possibles, ce code
 - facilite la conversion
 - complique grandement les circuits arithmétiques basés sur ce codage.

41

Différents codes binaires : Code binaire réfléchi

- aussi appelé code GRAY
- défini de manière à n'avoir qu'un seul bit qui change d'état entre deux valeurs successives
On dit que ce code est à distance minimale
- utilisé pour la simplification de fonctions logiques à partir du tableau de Karnaugh
- particulièrement intéressant pour les applications très rapides (car un seul bit change entre 2 transitions)

42

Différents codes binaires : Code binaire réfléchi

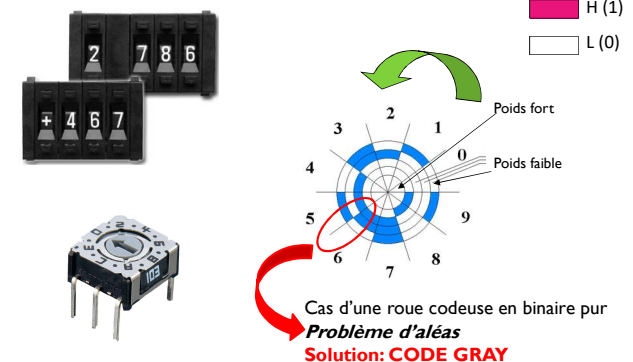
décimal	Gray	décimal	Gray
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

43

Pour aller plus loin...

Différents codes binaires : Code binaire réfléchi

Exemple d'une roue codeuse



44

Différents codes binaires : Code ASCII

- ASCII (code standard américain pour l'échange d'information) qui code une information alphanumérique

	MSB	0	1	2	3	4	5	6	7
LSB		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	^	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	}
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M	]	m	{
E	1110	SO	RS	.	>	N	^	n	~
F	1111	SI	US	/	?	O	_	o	DEL

- ASCII normal sur 7 bits soit 128 caractères
- ASCII étendu sur 8 bits soit 256 caractères

TD Chap 1 Variables logiques

Codage et Numération

Exercice 1 :

Dans un mot de 8 bits (de D7 à D0), trouver la valeur du bit de rang 5 (D5) dans les nombres 0xEB et 0x9F.

On dispose d'un bus de données constitué de 8 lignes D₇D₆... D₀.

- Donner l'équivalent décimal des données sachant que les lignes 1, 2 et 7 sont au niveau H et toutes les autres au niveau L (le N° de la ligne correspond à l'indice du bit)
- Quel est le niveau électrique de la ligne 4 lorsque l'équivalent décimal est 145 ?

Exercice 2 :

Convertir en décimal les nombres suivants :

- 0b1100 1001
- 0b0001 0001 1101
- 0xA3
- 0x34F

Exercice 3 :

Convertir en hexadécimal et en binaire les nombres décimaux suivants :

- 220
- 573

Exercice 4 :

Convertir en code BCD les nombres décimaux suivants :

- 1999
- 2745

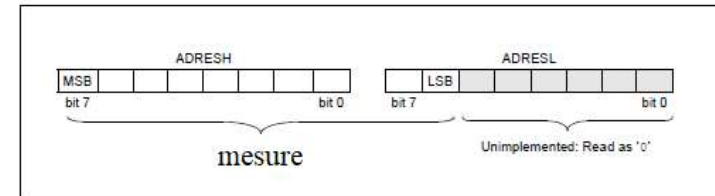
Convertir, si c'est possible, en décimal les codes suivants :

- (0011 1100 0110)_{bcd}
- (0010 0110 1000)_{bcd}

Exercice 5 :

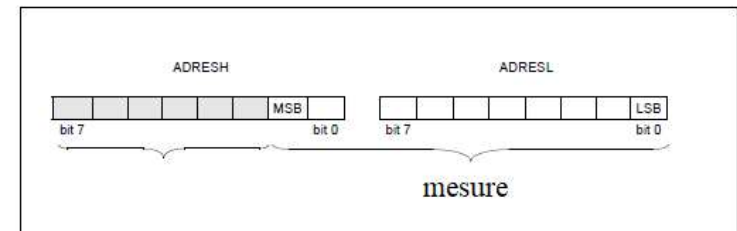
Soit le mot « mesure » de 10 bits en décimal : **340**

1. Convertir ce nombre en binaire
2. Ce nombre peut être rangé de 2 façons différentes
 - 2.1.1^{ère} solution



- 2.1.1. Donner dans ce cas le contenu de la variable (registre) ADRESH en binaire puis en hexadécimal
- 2.1.2. Quelles sont les bits qui ont été supprimés ? Conclure.

2.2.2^{ème} solution



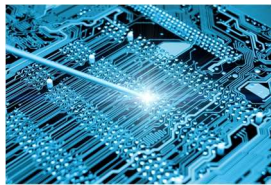
- 2.2.1. Donner dans ce cas le contenu de la variable ADRESH et de la variable ADRESL en binaire puis en hexadécimal
- 2.2.2. Quelles sont les bits qu'il ne faut surtout pas supprimer ?

Exercice 6 :

Soit un mot de 16 bits TMR1

1. TMR1 vaut 2022, donner son équivalent en hexadécimal.
2. Cette variable est rangée dans 2 octets TMR1H et TMR1L. Sachant que TMR1H correspond aux 8 bits de poids fort de TMR1 et TMR1L aux 8 bits de poids faibles, donner le contenu (valeur) de TMR1H et TMR1L.

Chapitre 2 :

Fonctions logiques**Plan du chapitre :**

- I. **Opérateurs élémentaires**
- II. Autres opérateurs
- III. Propriétés des opérateurs
- IV. Représentation d'une fonction logique
- V. Simplification d'une fonction logique

2

I. Opérateurs élémentaires :

- opérateur logique NON (complément)
- opérateur logique OU (somme)
- opérateur logique ET (produit)

3

Opérateur élémentaire : NON (complément)

Il inverse la valeur de la variable logique

$$F(a) = \neg a = \bar{a}$$

$$\bar{0} = 1$$

$$\bar{1} = 0$$

4

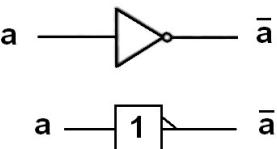
Opérateur élémentaire : NON

Table de vérité :

a	S
L	H
H	L

Il est noté graphiquement par un rond ou par une barre oblique

Symboles :



5

Opérateurs élémentaires : OU (somme)

Soient n termes reliés par des OU. Le résultat est à 1 si au moins un des termes est à 1

$$F(a, b) = a + b$$

$$0 + 0 = 0$$

$$0 + 1 = 1 + 0 = 1$$

$$1 + 1 = 1$$

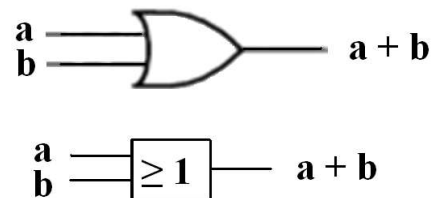
6

Opérateurs élémentaires : OU

Table de vérité :

b	a	S
L	L	L
L	H	H
H	L	H
H	H	H

Symboles :



7

Opérateurs élémentaires : ET (produit)

Soient n termes reliés par des ET. Le résultat est à 1 si tous les termes sont à 1

$$F(a, b) = a \cdot b = a * b$$

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

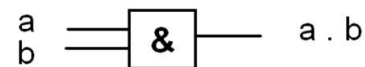
8

Opérateurs élémentaires : ET

Table de vérité :

b	a	S
L	L	L
L	H	L
H	L	L
H	H	H

Symboles :



9

II. Autres opérateurs :

- opérateur logique NON OU
- opérateur logique NON ET
- opérateur logique OU exclusif

10

Autres opérateurs : NON OU (NOR)

Soient n termes reliés par des NOR. Le résultat est à 1 si tous les termes sont à 0

$$F(a,b) = \overline{a + b}$$

$$\overline{0 + 0} = 1$$

$$\overline{0 + 1} = \overline{1 + 0} = 0$$

$$\overline{1 + 1} = 0$$

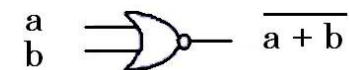
11

Autres opérateurs : NON OU (NOR)

Table de vérité :

b	a	S
L	L	H
L	H	L
H	L	L
H	H	L

Symboles :



12

Autres opérateurs : NON ET (NAND)

Soient n termes reliés par des NAND. Le résultat est à 1 si au moins un des termes est à 0

$$F(a, b) = \overline{a \cdot b}$$

$$\overline{0 \cdot 0} = 1$$

$$\overline{0 \cdot 1} = \overline{1 \cdot 0} = 1$$

$$\overline{1 \cdot 1} = 0$$

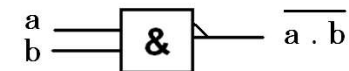
13

Autres opérateurs : NON ET (NAND)

Table de vérité :

b	a	S
L	L	H
L	H	H
H	L	H
H	H	L

Symboles :



14

Autres opérateurs : OU exclusif (XOR)

Soient n termes reliés par des XOR. Le résultat est à 1 si un nb impair de termes est à 1

$$F(a, b) = a \oplus b (= a \text{ :+ } b)$$

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

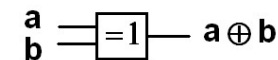
15

Autres opérateurs : OU exclusif (XOR)

Table de vérité :

b	a	S
L	L	L
L	H	H
H	L	H
H	H	L

Symboles :



16

Opérateurs logiques : synthèse

NOM	Table de vérité	Expressions des fonctions réalisées	Symbole normalisé															
INVERSEUR	<table><tr><td>a</td><td>S</td></tr><tr><td>L</td><td>H</td></tr><tr><td>H</td><td>L</td></tr></table>	a	S	L	H	H	L	$a \rightarrow \neg a \rightarrow S$ $/S=a$										
		a	S															
L	H																	
H	L																	
$a \rightarrow \neg a \rightarrow S$ $S=/a$																		
BUFFER	<table><tr><td>a</td><td>S</td></tr><tr><td>L</td><td>L</td></tr><tr><td>H</td><td>H</td></tr></table>	a	S	L	L	H	H	$a \rightarrow a \rightarrow S$ $S=a$										
		a	S															
L	L																	
H	H																	
$a \rightarrow \neg a \rightarrow S$ $/S=/a$																		
OU exclusif	<table><tr><td>b</td><td>a</td><td>S</td></tr><tr><td>L</td><td>L</td><td>L</td></tr><tr><td>L</td><td>H</td><td>H</td></tr><tr><td>H</td><td>L</td><td>H</td></tr><tr><td>H</td><td>H</td><td>L</td></tr></table>	b	a	S	L	L	L	L	H	H	H	L	H	H	H	L	$a \rightarrow b \rightarrow S$ $S=a \oplus b$	
b	a	S																
L	L	L																
L	H	H																
H	L	H																
H	H	L																

17

Opérateurs logiques : synthèse

NOM	Table de vérité	Expressions des fonctions réalisées	Symbole normalisé															
AND (ET)	<table><tr><td>b</td><td>a</td><td>S</td></tr><tr><td>L</td><td>L</td><td>L</td></tr><tr><td>L</td><td>H</td><td>L</td></tr><tr><td>H</td><td>L</td><td>L</td></tr><tr><td>H</td><td>H</td><td>H</td></tr></table>	b	a	S	L	L	L	L	H	L	H	L	L	H	H	H	 $S=a*b$	
		b	a	S														
L	L	L																
L	H	L																
H	L	L																
H	H	H																
 $/S=/a+/b$																		
OR (OU)	<table><tr><td>b</td><td>a</td><td>S</td></tr><tr><td>L</td><td>L</td><td>L</td></tr><tr><td>L</td><td>H</td><td>H</td></tr><tr><td>H</td><td>L</td><td>H</td></tr><tr><td>H</td><td>H</td><td>H</td></tr></table>	b	a	S	L	L	L	L	H	H	H	L	H	H	H	H	 $/S=/a*/b$	
		b	a	S														
L	L	L																
L	H	H																
H	L	H																
H	H	H																
 $S=a+b$																		

18

Opérateurs logiques : synthèse

NOM	Table de vérité	Expressions des fonctions réalisées	Symbole normalisé															
NAND (NON ET)	<table><tr><td>b</td><td>a</td><td>S</td></tr><tr><td>L</td><td>L</td><td>H</td></tr><tr><td>L</td><td>H</td><td>H</td></tr><tr><td>H</td><td>L</td><td>H</td></tr><tr><td>H</td><td>H</td><td>L</td></tr></table>	b	a	S	L	L	H	L	H	H	H	L	H	H	H	L	 $/S=a*b$	 a b S
	b	a	S															
L	L	H																
L	H	H																
H	L	H																
H	H	L																
 $S=/a+/b$																		
NOR (NON OU)	<table><tr><td>b</td><td>a</td><td>S</td></tr><tr><td>L</td><td>L</td><td>H</td></tr><tr><td>L</td><td>H</td><td>L</td></tr><tr><td>H</td><td>L</td><td>L</td></tr><tr><td>H</td><td>H</td><td>L</td></tr></table>	b	a	S	L	L	H	L	H	L	H	L	L	H	H	L	 $S=/a*/b$	 a b S
	b	a	S															
L	L	H																
L	H	L																
H	L	L																
H	H	L																
 $/S=a+b$																		

19

Plan du chapitre :

- I. Opérateurs élémentaires
- II. Autres opérateurs
- III. Propriétés des opérateurs**
- IV. Représentation d'une fonction logique
- V. Simplification d'une fonction logique

20

III. Propriétés des opérateurs (., +, /) :

- Éléments neutres : $a + 0 = 0 + a = a$
 $a \cdot 1 = 1 \cdot a = a$
- Éléments absorbants : $a + 1 = 1$
 $a \cdot 0 = 0$

21

Propriétés des opérateurs (., +, /) :

- Commutativité : $a + b = b + a$
 $a \cdot b = b \cdot a$
- Associativité :
 $(a + b) + c = a + (b + c) = a + b + c$
 $(a \cdot b) \cdot c = a \cdot (b \cdot c) = a \cdot b \cdot c$
- Distributivité : $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
 $a + (b \cdot c) = (a + b) \cdot (a + c)$

22

Propriétés des opérateurs (., +, /) :

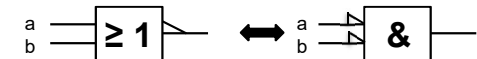
- Complément : $\overline{\overline{a}} = a$
 $a + \overline{a} = 1$
 $a \cdot \overline{a} = 0$
- Idempotence : $a + a + \dots + a = a$
 $a \cdot a \cdot \dots \cdot a = a$

23

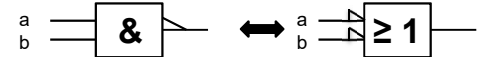
Propriétés des opérateurs (., +, /) :

Théorème de DE MORGAN

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



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

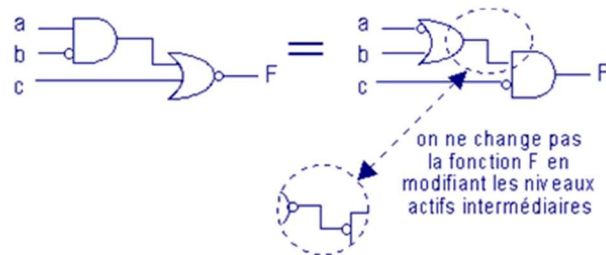


24

Propriétés des opérateurs (., +, /) :

Principe de dualité :

L'expression d'une fonction logique est inchangée lorsque l'on intervertit les opérateurs "somme" et "produit" si l'on inverse également les niveaux actifs hauts et les niveaux actifs bas



25

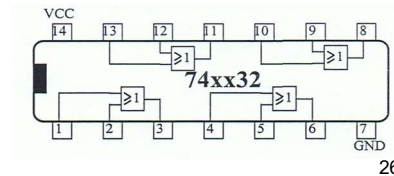
Propriétés des opérateurs (., +, /) :

En pratique :

- plusieurs exemplaires d'une même porte logique simple sont souvent regroupés dans le même **Circuit Intégré** (CI) repéré U1 par exemple

- présence de deux lignes d'alimentation électrique appelée aussi polarisation (**masse** notée GND ou **Vss** et tension positive notée VCC ou V_{DD}) jamais représentées dans les logigrammes

Le circuit 7432 comporte 4 portes OU à 2 entrées (U1A, U1B, U1C, U1D)



26

Propriétés des opérateurs :

Propriétés du XOR :

$$a \oplus 1 = /a$$

$$a \oplus 0 = a$$

$$a \oplus b = c \Leftrightarrow b = a \oplus c$$

27

Plan du chapitre :

- I. Opérateurs élémentaires
- II. Autres opérateurs
- III. Propriétés des opérateurs
- IV. Représentation d'une fonction logique**
- V. Simplification d'une fonction logique

28

IV. Représentation d'une fonction :

Plusieurs possibilités :

- schéma symbolique
- représentation graphique
- chronogramme
- table de vérité
- expression algébrique
- schéma à contact (LADDER en automatisme)

29

Représentation d'une fonction :

schéma symbolique :

=> extraits de datasheet

74AC11008
QUADRUPLE 2-INPUT POSITIVE-AND GATE

description
This device contains four independent 2-input AND gates. It performs the Boolean function $Y = A \cdot B$ or $Y = \overline{A} \cdot \overline{B}$ in positive logic.

FUNCTION TABLE
(each gate)

INPUTS		OUTPUT
A	B	Y
H	H	H
L	X	L
X	L	L

X: état indéterminé H ou L

SN7414, SN74LS14
HEX SCHMITT-TRIGGER INVERTERS

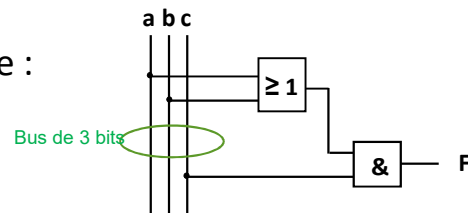
30

Représentation d'une fonction :

représentation graphique :

utilise les portes logiques vues précédemment

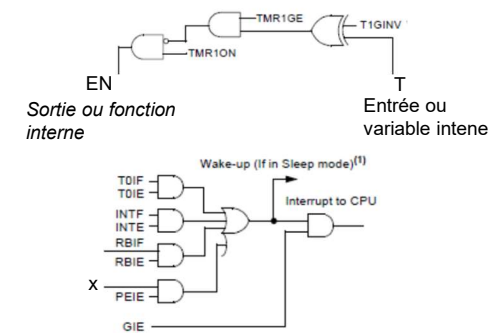
exemple :



31

Représentation d'une fonction :

- Autres exemples issus d'une documentation constructeur :



32

Représentation d'une fonction :

représentation graphique :

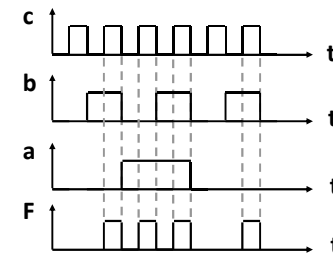
- pour représenter une fonction active à l'état bas => représenter les combinaisons des variables qui permettent d'avoir $F = 0$ et mettre la barre oblique inverseuse (ou le rond) en sortie de la dernière porte logique pour obtenir la représentation de F
- pour représenter une fonction active à l'état haut => représenter les combinaisons des variables qui permettent d'avoir $F = 1$

33

Représentation d'une fonction :

Chronogramme : représente l'évolution temporelle de la fonction logique et des variables logiques qui la composent

Exemple (avec la même fonction F)



34

Représentation d'une fonction :

Table de vérité : indique la valeur de la fonction pour chaque combinaison des valeurs des variables logiques qui définissent cette fonction.

fonction F définie par n variables => table de vérité avec n colonnes à gauche + une colonne à droite pour F et 2^n lignes.

35

Représentation d'une fonction :

Table de vérité :

On associe une colonne à chaque variable, et la dernière colonne donne la valeur de la fonction, elle est séparée des autres colonnes par une double barre.

Rmq : pour avoir toutes les combinaisons des variables il faut :

- alterner les 0 et les 1 dans la colonne des variables la plus à droite,
- alterner deux 0 puis deux 1 dans la colonne précédente,
- alterner quatre 0 puis quatre 1 dans la colonne précédente,...

36

Représentation d'une fonction :

Table de vérité :

exemple (avec la même fonction F)

a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

37

Représentation d'une fonction :

Expression algébrique :

Toute fonction logique F peut s'écrire sous la forme d'une somme de produits de variables (directes ou inverses).

Chaque produit (ou monôme) correspond à une combinaison des variables qui permet d'activer la fonction.

Exemple (avec la même fonction F)

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

38

Représentation d'une fonction :

Expression algébrique :

Si la fonction est active à l'état haut, écrire : $F = \dots$

interprétation des lignes de la table de vérité sur lesquelles F est à 1

Si la fonction est active à l'état bas, écrire : $\overline{F} = \dots$

interprétation des lignes de la table de vérité sur lesquelles F est à 0

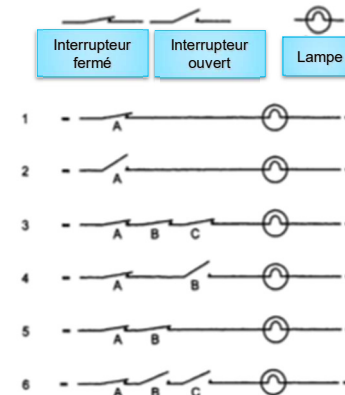
On met en évidence les combinaisons des variables pour lesquelles la fonction est active.

39

Représentation d'une fonction :

schéma à contact (LADDER en automatisme)

Indiquer quel(s) circuit(s) allume(nt) la lampe et quelle est la fonction correspondante



40

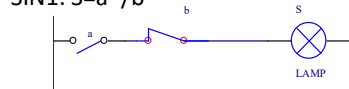
Représentation d'une fonction :

- **schéma à contact (LADDER en automatisme)**

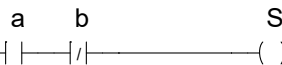
- ❑ La fonction NON est représentée par un contact normalement fermé.
- ❑ La fonction OU est représentée par des contacts en parallèle.
- ❑ La fonction ET est représentée par des contacts en série.

Equivalence entre SIN1 et Automatisme

SIN1: $S = a * b$



Automatisme (Auto2):



41

Plan du chapitre :

- I. Opérateurs élémentaires
- II. Autres opérateurs
- III. Propriétés des opérateurs
- IV. Représentation d'une fonction logique
- v. Simplification d'une fonction logique**

42

V. Simplification d'une fonction :

Soit la table de vérité de la fonction « ou » à deux variables :

a	b	$f(a,b) = a + b$
0	0	0
0	1	1 $\rightarrow /a.b$
1	0	1 $\rightarrow a./b$
1	1	1 $\rightarrow a.b$

$$\Rightarrow f(a,b) = /a.b + a./b + a.b$$

43

Simplification d'une fonction :

La fonction « ou » de la table de vérité s'écrit:

$$\Rightarrow f(a,b) = /a.b + a./b + a.b$$

$\Rightarrow$ trop complexe par rapport à ce qu'elle pourrait être car information redondante

$\Rightarrow$ possible de simplifier cette fonction (simplification mathématique)

44

Simplification d'une fonction :

Simplification mathématique => obtenir la forme minimale (= nombre minimal de termes, et nombre minimal de variables dans chaque terme)

mais :

- ne correspond pas toujours à la réalisation la plus simple et/ou la plus rapide
- la prise en compte de contraintes technologiques peut imposer une complexification d'écriture de l'expression (par ex. faire apparaître toutes les variables dans chaque produit pour une réalisation avec un multiplexeur)
- On préférera donc parfois laisser la fonction sous forme de somme de produit logique $S = \sum \pi$

45

Simplification d'une fonction :

2 possibilités pour simplifier :

- simplification algébrique
- simplification graphique

46

Simplification d'une fonction : simplification algébrique

Règles de simplification :

- regroupement de termes (factorisation)
ex : simplifier $a b c + a b /c + a /b c d$
- ajout de termes déjà existants
ex : simplifier $a b c + /a b c + a /b c + a b /c$
- utilisation des propriétés des opérateurs vues précédemment

47

Simplification d'une fonction : simplification algébrique

identités fréquemment employées pour simplifier les fonctions :

- $a b + /a b =$
- $a + a b =$
- cf. 2 • $a + /a b =$
- $(a + b) (/a + b) =$
- $a (a + b) =$
- $a (/a + b) =$

48

Simplification d'une fonction : simplification algébrique

identités fréquemment employées pour simplifier les fonctions :

- $a \cdot b + \bar{a} \cdot b = (a + \bar{a}) \cdot b = b$
- $a + a \cdot b = a \cdot (1 + b) = a$
- cf. 2 • $a + \bar{a} \cdot b = a + ab + \bar{a}b = a + (a + \bar{a})b = a + b$
- $(a + b) (\bar{a} + \bar{b}) = \bar{a} \cdot \bar{a} + ab + \bar{a}b + \bar{b}b = \bar{a} + ab + \bar{a}b + b = b$
- $a(a + b) = aa + ab = a \cdot (1 + b) = a$
- $a(\bar{a} + b) = a \cdot \bar{a} + ab = a \cdot b$

49

Simplification d'une fonction : simplification graphique => à partir d'un tableau de Karnaugh

On utilise la notion d'adjacence

Deux termes produits sont dits adjacents s'ils ne diffèrent que par une variable : elle doit être complémentée dans un des termes produits et non complémentée dans l'autre

ex : $a \cdot b \cdot c + a \cdot b \cdot \bar{c}$

(produits adjacents car ils ne diffèrent que de c)

50

Simplification d'une fonction : simplification graphique

Karnaugh => mettre en évidence les termes adjacents

C'est le code **Gray** qui est utilisé pour coder les différentes combinaisons des valeurs des variables logiques (code qui assure que deux valeurs successives ne diffèrent que d'un bit (=> adjacence))

51

Simplification d'une fonction : simplification graphique

méthode :

- construire le tableau de Karnaugh
- effectuer les groupements permettant de simplifier la fonction
- en déduire l'expression de la fonction (= termes simplifiés permettant d'activer la fonction)

52

Simplification d'une fonction : simplification graphique

Construction du tableau de Karnaugh :

- c'est une table de vérité à double entrée
- chaque case correspond à une ligne de la table de vérité
- dans chaque case on met la valeur de la fonction correspondant à la combinaison des variables de la ligne et de la colonne concernées
- pour une fonction définie par « n » variables logiques, le tableau est formé de 2^n cases

53

Simplification d'une fonction : simplification graphique

Construction du tableau de Karnaugh pour 3 variables :

		ab			
		00	01	11	10
c	a	b	c	f	
	0	0	0	$f(0,0,0)$	
	0	0	1	$f(0,0,1)$	
	0	1	0	$f(0,1,0)$	
		00	01	11	10
0		$f(0,0,0)$	$f(0,1,0)$	...	
1		$f(0,0,1)$	...		

54

Simplification d'une fonction : simplification graphique

Construction du tableau de Karnaugh pour 4 variables:

		ab			
		00	01	11	10
cd	a	b	c	d	f
	0	0	0	0	$f(0,0,0,0)$
	0	0	0	1	$f(0,0,0,1)$
	0	0	1	0	$f(0,0,1,0)$
	0	0	1	1	$f(0,0,1,1)$
		00	01	11	10
00		$f(0,0,0,0)$	$f(0,1,0,0)$	...	
01		$f(0,0,0,1)$	...		
11		$f(0,0,1,1)$	...		
10		$f(0,0,1,0)$	...		

55

Simplification d'une fonction : simplification graphique

Groupements dans le tableau de Karnaugh :

- on groupe les cases adjacentes contenant des "1"
- 2 cases sont adjacentes si elles ont un côté en commun
- Un "1" peut être utilisé dans plusieurs groupements
- tout "1" doit appartenir (si possible) à au moins un groupement
- on fait des groupes de 2^i cases adjacentes $\forall i \in \mathbb{N}$
- pour obtenir une expression simplifiée minimale il faut rechercher le **nb minimal** de groupements **les plus grands**

56

Simplification d'une fonction : simplification graphique

remarques :

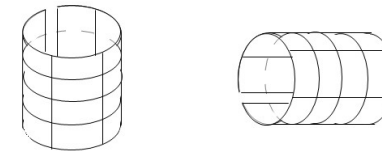
- pour une fonction active à l'état bas, il faut grouper les "0" et non plus les "1"
- si la fonction n'est pas définie pour certaines combinaisons des variables, on met X dans les cases correspondantes. Pour la simplification, X représente "1" si la case est utile à un groupement et "0" dans le cas contraire (pour une fonction active à l'état haut).

57

Simplification d'une fonction : simplification graphique

remarques :

- les cases des extrémités d'une même ligne (respectivement d'une même colonne) sont adjacentes



58

Simplification d'une fonction : simplification graphique

Formation de regroupements

BA \ DC	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

$S=A$

BA \ DC	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

$S=\bar{C}$

BA \ DC	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	1
10	0	0	1	1

$S=DB$

BA \ DC	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	1	0	0	1
10	0	0	0	0

$S=C\bar{A}$

BA \ DC	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$S=C\bar{A}$

BA \ DC	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

$S=CA$

59

Simplification d'une fonction : simplification graphique

expression de la fonction :

- chaque groupement se traduit par un terme correspondant au produit de variables logiques : celles qui ne changent pas de valeur quelle que soit la case du groupement concerné
- les différents termes sont reliés entre eux par des "ou"

60

Simplification d'une fonction : simplification graphique

- Exemple 2: soit la fonction f définie à partir de 4 variables d'entrée a, b, c et d

$$f = \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + ab\bar{c}d + abcd$$

cd \ ab	00	01	11	10
00	0 ₀	0 ₁	1 ₃	1 ₂
01	0 ₄	1 ₅	0 ₇	0 ₆
11	0 ₁₂	1 ₁₃	0 ₁₅	1 ₁₄
10	0 ₈	0 ₉	0 ₁₁	0 ₁₀

Le nombre en bas à droite correspond à l'équivalent décimal de la ligne de la table de vérité

61

Simplification d'une fonction : simplification graphique

$$f = \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + ab\bar{c}d + abcd$$

cd \ ab	00	01	11	10
00	0 ₀	0 ₁	1 ₃	1 ₂
01	0 ₄	1 ₅	0 ₇	0 ₆
11	0 ₁₂	1 ₁₃	0 ₁₅	1 ₁₄
10	0 ₈	0 ₉	0 ₁₁	0 ₁₀

Annotations graphiques (groupes de 2) :

- $\bar{a}\bar{b}cd$ (groupes (3,5) et (7,13))
- $\bar{a}\bar{b}c\bar{d}$ (groupes (3,2) et (7,13))
- $\bar{a}b\bar{c}d$ (groupes (13,14) et (5,6))
- $ab\bar{c}d$ (groupes (13,14) et (15,10))
- $abcd$ (groupes (14,10) et (13,9))

62

Simplification d'une fonction : simplification graphique

$$f = \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + ab\bar{c}d + abcd$$

cd \ ab	00	01	11	10
00	0 ₀	0 ₁	1 ₃	1 ₂
01	0 ₄	1 ₅	0 ₇	0 ₆
11	0 ₁₂	1 ₁₃	0 ₁₅	1 ₁₄
10	0 ₈	0 ₉	0 ₁₁	0 ₁₀

Annotations graphiques (groupes de 2) :

- $/a*/b*c$ (groupes (3,5) et (7,13))
- $b*/c*d$ (groupes (5,6) et (13,14))
- $abcd$ (groupes (14,10) et (13,9))

63

Simplification d'une fonction : exemple simple

Déterminer l'expression de F Conseil: comparer une colonne avec la fonction F

Équivalent décimal

e.d	c	b	a	F
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

On voit directement que $F=c$ ici

d'où
 $F = c*/b + c = c + /b$

64

TD Chap 2 Fonctions logiques

1. Opérateurs Logiques :

Exercice 1

Démontrer les propriétés d'absorption suivantes

$$a+a*b = a$$

$$a+/a*b = a+b$$

$$/a+a*/b = /a+/b$$

Exercice 2

Simplifier algébriquement les expressions suivantes

$$A = a+/a*c+a*c$$

$$B = b+abc$$

$$C = a+b+ab+bc$$

$$D = /b*/c*a+/a*/b*/c$$

$$E = a+ab+a*/b+bc$$

2. Représentation d'une fonction :

Exercice 3

Donner l'expression de F sous forme de somme de produits F active H.

x	y	z	F
L	L	L	L
L	L	H	L
L	H	L	H
L	H	H	L
H	L	L	H
H	L	H	L
H	H	L	H
H	H	H	L

Exercice 4

On désire réaliser une fonction S active à l'état H, à partir de 4 variables x, y, z et t.

L'activation de S doit être obtenue pour :

(x au niveau H ET z au niveau L)

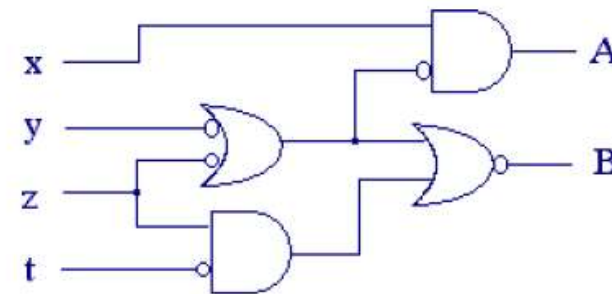
OU (y au niveau H ET z au niveau H ET t au niveau L)

OU (x au niveau H ET y au niveau H ET t au niveau L)

1. Donner l'expression graphique (le logigramme) de la fonction S
2. Donner le schéma à contact (LADDER) de la fonction S

Exercice 5

Soit le logigramme suivant :



1. Exprimer algébriquement ces fonctions
2. Donner la table de vérité des fonctions A et B.

3. Simplification graphique (Karnaugh) :

Exercice 6

- Simplifier par la méthode de Karnaugh les 3 fonctions $F1=f(a, b, c)$, puis $F2=f(a, b, c)$ et enfin $F3=f(a, b, c)$ (X : état indéterminé)

a	b	c	F1	F2	F3
0	0	0	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	1	1
1	0	1	0	1	0
1	1	0	0	1	X
1	1	1	0	1	X

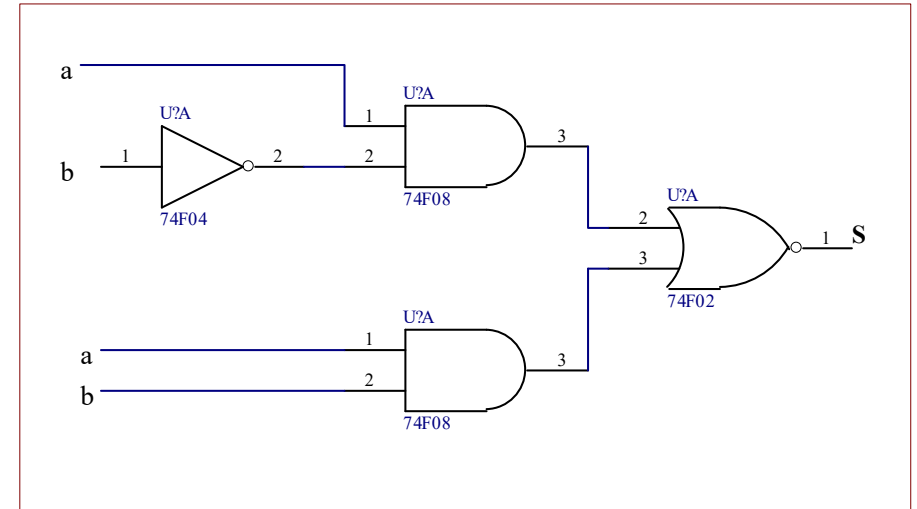
- Même chose pour les fonctions $F4=f(a,b,c,d)$ et $F5=f(a,b,c,d)$

a	b	c	d	F4	F5
L	L	L	L	H	H
L	L	L	H	L	X
L	L	H	L	H	H
L	L	H	H	L	L
L	H	L	L	L	L
L	H	L	H	L	L
L	H	H	L	H	H
L	H	H	H	H	L
H	L	L	L	H	L
H	L	L	H	L	H
H	L	H	L	H	L
H	L	H	H	L	H
H	H	L	L	H	L
H	H	L	H	H	X
H	H	H	L	H	H
H	H	H	H	H	L

4. Synthèse :

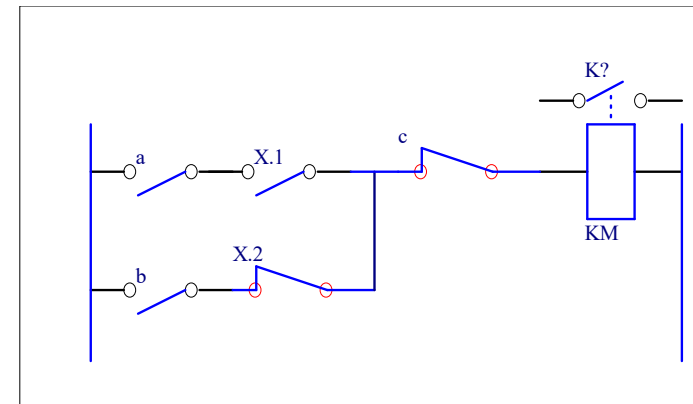
Exercice 7

Quelle est l'expression algébrique simplifiée de S ?



Exercice 8

Donner l'expression algébrique simplifiée de la fonction KM (relais = sortie) ?



Exercice 9 :

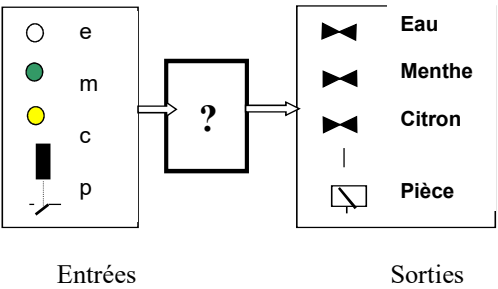
DISTRIBUTEUR DE BOISSONS

Un distributeur de boissons comporte 3 cuves contenant de l’eau (Eau), de la menthe (Menthe), et du citron (Citron).

Trois boutons poussoirs e, m, c permettent d’avoir de l’eau, de la menthe à l’eau, ou du citron à l’eau.

Les sirops de menthe et de citron sont payants et toujours distribués avec de l’eau. L’introduction d’une pièce (bonne) agit sur un contact p.

La restitution de la pièce le cas échéant se fera par un électroaimant Pièce.



- ❑ L’eau est distribuée gratuitement.
- ❑ La menthe et le citron ne sont jamais distribués purs.
- ❑ Les cocktails menthe-citron sont interdits.
- ❑ Si au bout de 10 secondes aucun choix cohérent n’a été fait, la pièce est rendue (la temporisation n’est pas étudiée).
- ❑ Si on demande de la menthe, du citron et de l’eau, on n’aura rien.

1. Compléter la table de vérité suivante :

e	m	c	p	Eau	Menthe	Citron	Pièce
L	L	L	L				
L	L	L	H				
L	L	H	L				
L	L	H	H				
L	H	L	L				
L	H	L	H				
L	H	H	L				
L	H	H	H				
H	L	L	L				
H	L	L	H				
H	L	H	L				
H	L	H	H				
H	H	L	L				
H	H	L	H				
H	H	H	L				
H	H	H	H				

2. Donner les équations simplifiées des fonctions Eau, Menthe, Citron et Pièce en fonction des variables e, m, c et p.
3. Représenter graphiquement (logigramme) les fonctions Eau, Menthe, Citron et Pièce avec des opérateurs élémentaires.
4. Représenter le schéma à contacts (LADDER) des fonctions Eau, Menthe, Citron et Pièce

Chapitre 3 :

Technologie des circuits intégrés

Plan du chapitre :

- I. Généralités
- II. Caractéristiques électriques
- III. Etages de sorties
- IV. Etage d'entrée
- V. Les principales familles logiques

2

Technologie des circuits intégrés

I. Généralités

❑ Caractéristiques générales:

- Caractéristiques mécaniques (type de boîtiers, compatibilité ..)
- Gammes de température d'utilisation (différent de stockage)
 - Gamme à spécification militaire -55°C/125°C (pas réservé à l'armée !! / préfixe 54 en TTL)
 - Gamme industrielle 0°C/70°C (préfixe 74 en TTL)
 - Gamme automobile -25°C/70°C (dépend du constructeur)
- Caractéristiques électriques (statiques et dynamiques)
- Caractéristique fonctionnelle (fonction logique réalisée)

❑ Technologie CI standard

- TTL (obsolète mais ,peut-être « mixée » avec CMOS)
- CMOS: la plus utilisée



Fonctions logiques indépendantes de la technologie

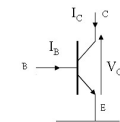
3

Technologie des circuits intégrés

I. Généralités

❑ Technologie TTL:

- TTL (Transistor Transistor Logic): technologie construite autour de transistors bipolaires qui fonctionnent en commutation:



En interne: commande par courant I_B
Schéma simplifié:



- Identification d'un circuit TTL

Exemple: SN 74 AS 169 N ou DM 54 S 283 J

Chaque champ a une signification :

- SN, DM : champ littéral qui indique le constructeur.
- 74 ou 54 : gamme de températures normale (0°C à 70°C) ou militaire (-55°C à +125°C).
- AS, S, ...: technologie ici advanced shottky, shottky.
- 169, 283, ...: fonction logique.
- N, J, P, NT...: type de boîtier (ici DIL plastique ou céramique).

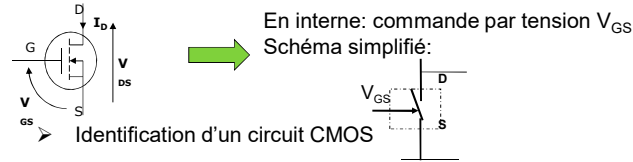
4

Technologie des circuits intégrés

I. Généralités

❑ Technologie CMOS:

- CMOS (Complementary Metal Oxide Semi-conductor): technologie la plus répandue (très faible consommation statique, grande plage d'alimentation)



➤ Identification d'un circuit CMOS

La série 4000 est obsolète, prenons l'exemple du **SN 74HC04D**:

- SN : champ littéral qui indique le constructeur ici Texas Instruments
- 74 ou 54 : gamme de températures normale (0°C à 70°C) ou militaire (-55°C à +125°C)
- AS, S, ...: technologie ici High Speed CMOS (voir d.20)
- 04, ...: fonction logique ici inverseur
- N, J, P, D...: type de boîtier ici SOIC(CMS).

5

Plan du chapitre :

I. Généralités

II. Caractéristiques électriques

III. Etages de sorties

IV. Etage d'entrée

V. Les principales familles logiques

6

Technologie des circuits intégrés

II. Caractéristiques électriques

❑ Caractéristiques statiques:

- Tension d'alimentation: elle dépend de la famille (technologie)
 - **TTL: 5V +/- 5%** 0V: GND (Ground) 5V: V_{CC}
 - **CMOS série 4000 de 3 à 18V**: 0V: V_{SS} ; 15V: V_{DD}
 - **CMOS « moderne » de 3,3V à 1,8V voire moins !** 0V: V_{SS} ; 3,3V V_{DD}
- Courant consommé (par l'alimentation)
- Puissance en statique = tension d'alimentation x courant consommé
- Niveaux logiques (ou de fonctionnement): ils correspondent à des tensions en entrée et en sortie, ils représentent donc les niveaux H et L

7

Technologie des circuits intégrés

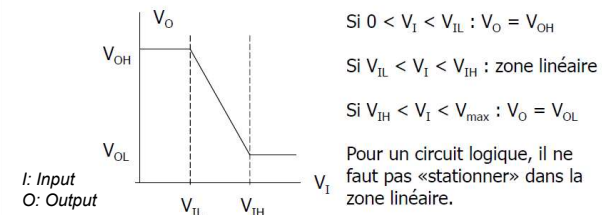
II. Caractéristiques électriques

❑ Caractéristiques statiques:

- Niveaux logiques (ou de fonctionnement):

Niveau	Tension d'entrée	Tension de sortie	En logique dite positive
L	V_{IL}	V_{OL}	0
H	V_{IH}	V_{OH}	1

Exemple de l'inverseur : (courbe idéalisée)



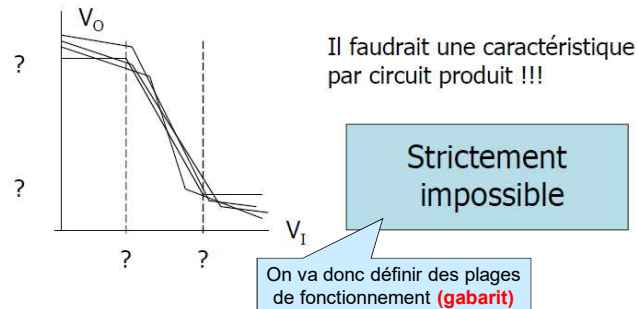
8

Technologie des circuits intégrés

II. Caractéristiques électriques

□ Niveaux logiques (ou de fonctionnement):

Exemple de l'inverseur : (faisceau de courbe)

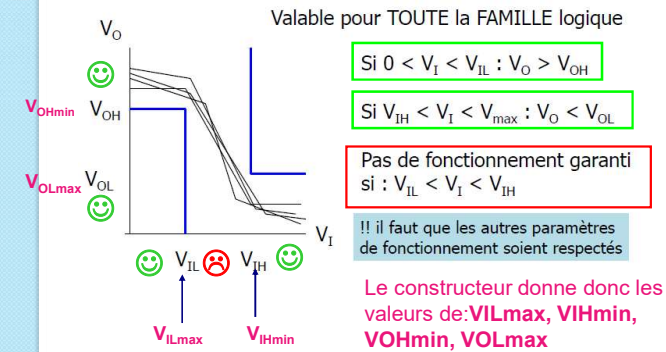


9

Technologie des circuits intégrés

II. Caractéristiques électriques

□ Niveaux logiques (ou de fonctionnement):



10

Technologie des circuits intégrés

II. Caractéristiques électriques

□ Niveaux logiques (ou de fonctionnement): exemple

DC CHARACTERISTICS

Type 74HC04

At recommended operating conditions; voltages are referenced to GND (ground = 0 V).

SYMBOL	PARAMETER	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
		OTHER	V _{CC} (V)				
T _{amb} = 25 °C							
V _{IH}	HIGH-level input voltage		2.0	1.5	1.2	–	V
			4.5	3.15	2.4	–	V
			6.0	4.2	3.2	–	V
V _{IL}	LOW-level input voltage		2.0	–	0.8	0.5	V
			4.5	–	2.1	1.35	V
			6.0	–	2.8	1.8	V
V _{OH}	HIGH-level output voltage	V _I = V _{IH} or V _{IL}					
		I _O = –20 μA	2.0	1.9	2.0	–	V
		I _O = –20 μA	4.5	4.4	4.5	–	V
		I _O = –4.0 mA	4.5	3.98	4.32	–	V
		I _O = –20 μA	6.0	5.9	6.0	–	V
		I _O = –5.2 mA	6.0	5.48	5.81	–	V
V _{OL}	LOW-level output voltage	V _I = V _{IH} or V _{IL}					
		I _O = 20 μA	2.0	–	0	0.1	V
		I _O = 20 μA	4.5	–	0	0.1	V
		I _O = 4.0 mA	4.5	–	0.15	0.26	V
		I _O = 20 μA	6.0	–	0	0.1	V
		I _O = 5.2 mA	6.0	–	0.16	0.26	V

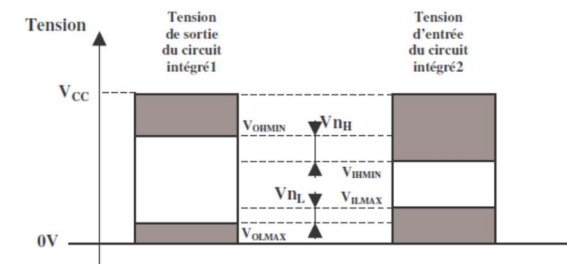
11

Technologie des circuits intégrés

II. Caractéristiques électriques

□ Associations de boîtiers:

- Tous les circuits d'une même famille sont compatibles
- Pour qu'il y ait compatibilité, entre 2 circuits de famille ou technologie différente il faut qu'une certaine marge de bruit existe entre le circuit logique 1 d'attaque et le circuit 2 (voir cours CEM BUT2 ou BUT3)



12

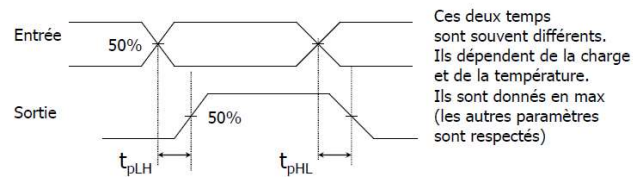
Technologie des circuits intégrés

II. Caractéristiques électriques

□ Caractéristiques dynamiques:

➤ Temps de propagation

Paramètres garantis : Retard dû à la propagation entre une cause (entrée) et un effet (sortie).



Attention : $T_d = (T_{pLH} + T_{pHL}) / 2$ = temps de délai moyen ne correspond à rien de physique

Rem: d'autres paramètres seront abordés en logique séquentiel

13

Plan du chapitre :

I. Généralités

II. Caractéristiques électriques

III. Etages de sorties

IV. Etage d'entrée

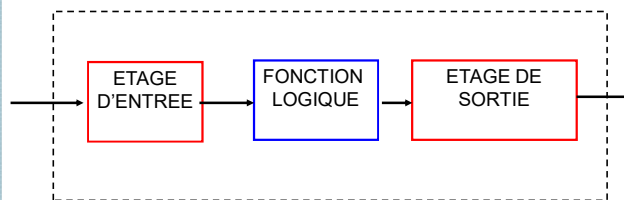
V. Les principales familles logiques

14

Technologie des circuits intégrés

III. Etages de sortie

□ Schéma fonctionnel d'un circuit logique

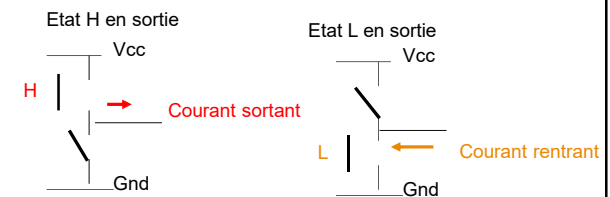
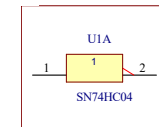


15

Technologie des circuits intégrés

III. Etages de sortie

□ Sorties classiques à transistors : sortie dite « TOTEM_POLE »



Les interrupteurs sont en fait des transistors en commutation

16

Technologie des circuits intégrés

III. Etages de sortie

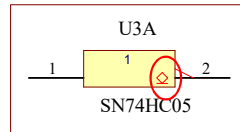
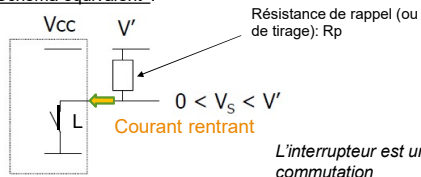
□ Sortie à Collecteur Ouvert CO (TTL) ou Drain ouvert DO (CMOS)

➤ Symbole :



➤ Principe : la sortie qui correspond au collecteur pour du TTL et au drain pour du CMOS est « en l'air », il faut donc impérativement connecter une résistance dit de rappel entre la sortie et une tension externe (V').

➤ Schéma équivalent :



Remarque: Sortie à collecteur ouvert avec résistance de tirage intégrée

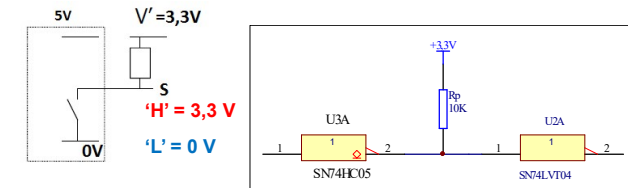
17

Technologie des circuits intégrés

III. Etages de sortie

□ Sortie à Collecteur Ouvert CO (TTL) ou Drain ouvert DO (CMOS)
1^{ère} application : adaptation de niveau (translateur de niveau)

Une technologie 5V (TTL par exemple) pourra fournir des niveaux différents à l'état H si V' , tension d'alimentation externe de la sortie via R_p , est différente (3,3V par exemple)



Les interrupteurs sont en fait des transistors en commutation

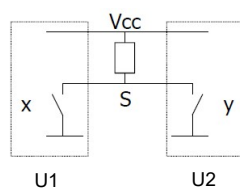
18

Technologie des circuits intégrés

III. Etages de sortie

□ Sortie à Collecteur Ouvert CO (TTL) ou Drain ouvert DO (CMOS)
2^{ème} application : réalisation d'un « ET câblé »

Remarque: il est interdit de connecter plusieurs sorties entre elles (court-circuit!) SAUF dans le cas de **sorties à collecteur ouvert**.



O: ouvert; F: fermé

U1		U2		
X	État de X	Y	État de Y	S
L	F	L	F	L
L	F	H	O	L
H	O	L	F	L
H	O	H	O	H

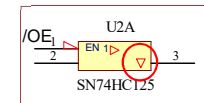
Les interrupteurs sont en fait des transistors en commutation

19

Technologie des circuits intégrés

III. Etages de sortie

□ Sortie haute impédance, Hi-Z ou tri-state ▽

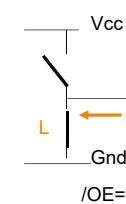


Ce type de porte possède une entrée supplémentaire de mise en HiZ (en fait une /OE active L qui valide le(s) sortie(s))

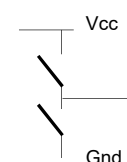
Etat H en sortie



Etat L en sortie



Etat Hi-Z en sortie



/OE=H

La sortie est DECONNECTEE physiquement

Si plusieurs sorties Hi-Z sont branchées ensembles, une seule doit être en basse impédance à la fois; sinon conflit de bus

Autre représentation:



20

Plan du chapitre :

- I. Généralités
- II. Caractéristiques électriques
- III. Etages de sorties
- IV. Etage d'entrée
- V. Les principales familles logiques

25

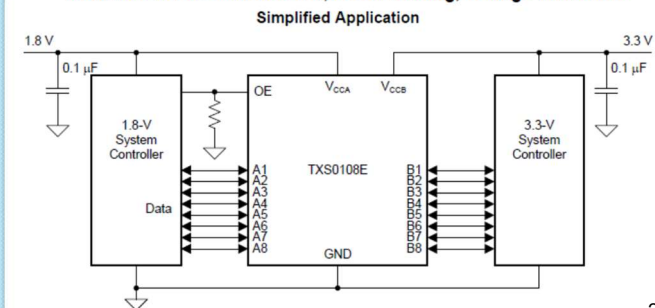
Technologie des circuits intégrés

V. Les principales familles logiques:

Comment interfacer des composants numérique de famille différentes donc des niveaux H et L différents (5V, 3,3V, 1,8V ..) ?

Utilisation de **translateur de niveaux**.

TXS0108E 8-Bit Bi-directional, Level-Shifting, Voltage Translator



26

Technologie des circuits intégrés

Synthèse

On distingue alors 2 types de circuits intégrés numériques:

- ❑ circuits logiques standards (vus dans ce chapitre et en SAE « CRSE »)
- ❑ Circuits logiques **programmables** :
 - Module FPGA du parcours ESE (BUT2 ou BUT3)
 - Prérequis : notions de technologie ...standard

Circuits reconfigurables

27

TD3 Technologie des circuits numériques standards

Exercice 1 : Analyse de documents constructeurs et compatibilité

- Donner les **caractéristiques électriques statiques** des circuits intégrés 74LS00 et 74HC00

1.1. Niveaux de fonctionnement en entrées.

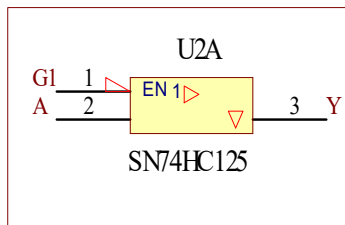
1.2. Niveaux de fonctionnement en sortie.

- Donner **le temps de propagation** des circuits intégrés 74LS00 et 74HC00

Exercice 2 : Analyse de symboles

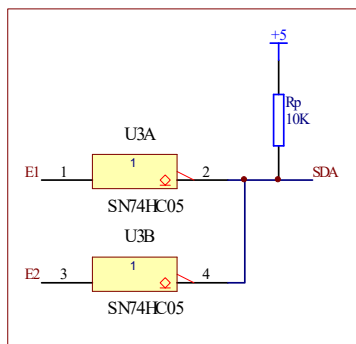
Pour chacun des circuits ou montages proposés, compléter la table de vérité

Circuit 1



G1	A	Y
L	L	
L	H	
H	X	

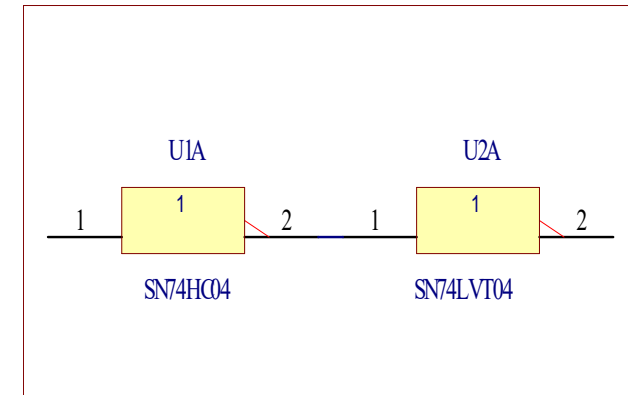
Circuit 2 : bus de donnée I2C (Sn sortie de En)



E1	E2	S1	S2	SDA
L	L			
L	H			
H	L			
H	H			

Exercice 3 :

Soit le montage suivant :



- Les 2 composants U1 et U2 ne sont pas compatibles. Expliquer pourquoi.
- Proposer une solution.

Chapitre 4 :

Fonctions combinatoires standards

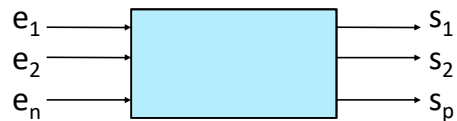
Plan du chapitre :

- I. Système combinatoire
- II. Fonction Commutation : multiplexeur, démultiplexeur
- III. Fonction Codage : décodeur, transcodeur

2

Système combinatoire

- Soit un système logique possédant n entrées $e_1, e_2, \dots, e_n$ et p sorties $s_1, s_2, \dots, s_p$:



- Un tel système est dit combinatoire si chacune de ses sorties est fonction exclusivement des entrées.

$$s_i = f(e_1, e_2, \dots, e_n)$$

3

Système combinatoire

- Pour une combinaison des entrées donnée, chacune des sorties ne peut prendre qu'une seule valeur.
- Ainsi, un système combinatoire peut être entièrement défini par une table de vérité

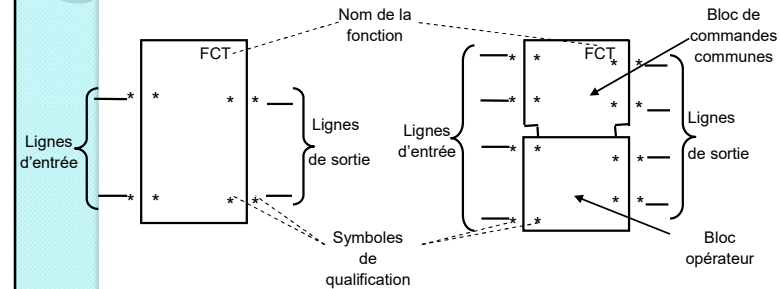
4

Plan du chapitre :

- I. Système combinatoire
- II. Fonction Commutation : multiplexeur, démultiplexeur
- III. Fonction Codage : décodeur, transcodeur

5

Symbolisme



6

Multiplexeur : principe

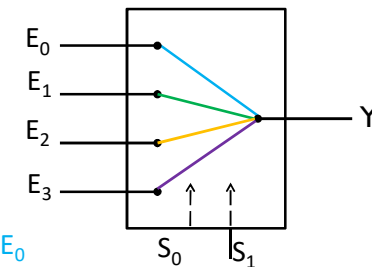
Sélection d'une voie parmi les 2^n entrées mise en sortie avec n bits de sélection ou de commande:

- n entrées de sélection : $S_0, S_1, \dots, S_{n-1}$ (S_i ou A_i)
- $N=2^n$ entrées de données $E_0, E_1, \dots, E_{N-1}$ (E_i ou D_i)
- une sortie Y (ou Q)
- + généralement une entrée de validation EN qui autorise, ou non, le multiplexage.

7

Multiplexeur : principe

Pour $n=2$,
multiplexeur 4 vers 1

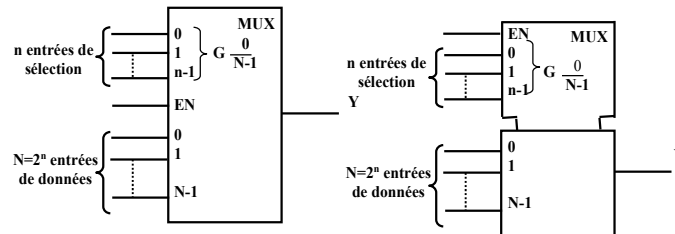


- Si $(S_1 S_0)_2 = 00$ alors $Y = E_0$
- Si $(S_1 S_0)_2 = 01$ alors $Y = E_1$
- Si $(S_1 S_0)_2 = 10$ alors $Y = E_2$
- Si $(S_1 S_0)_2 = 11$ alors $Y = E_3$

$$\text{donc } Y = E_0 \cdot \overline{S_1} \cdot \overline{S_0} + E_1 \cdot \overline{S_1} \cdot S_0 + E_2 \cdot S_1 \cdot \overline{S_0} + E_3 \cdot S_1 \cdot S_0$$

8

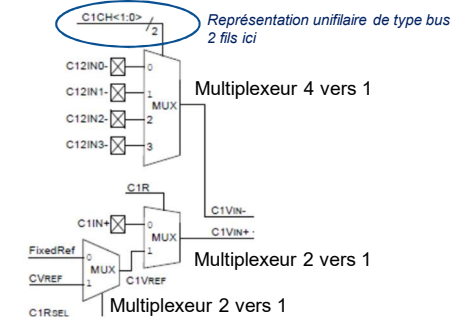
Multiplexeur : symboles



Les valeurs figurant sur chacune des entrées de sélection précisent le rang.
Les indices (en décimal) figurant sur chacune des entrées de données indiquent les valeurs codées pour lesquelles ces entrées sont en sortie.

9

Multiplexeur : autre symbole

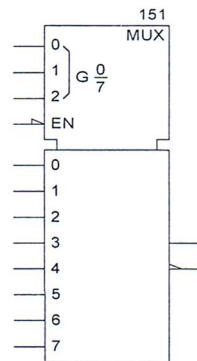


Issu d'une documentation (data sheet) de MICROCHIP
Les entrées de sélection sont appelées ici Cx ou CH (pour CHannel)
Les indices (en décimal) figurant sur chacune des entrées de données indiquent les valeurs codées pour lesquelles ces entrées sont en sortie.

10

Multiplexeur: exemple 74LS151

Multiplexeur « 8 vers 1 » comportant 3 lignes de sélection (A,B,C), 8 lignes d'entrées de données D[0..7], une ligne de validation /E et une sortie directe Q et complémentée /Q



Entrées				Sorties	
de sélection			validat°	Q	$\overline{Q}$
C	B	A	E	Q	$\overline{Q}$
X	X	X	H	L	H
L	L	L	L	D ₀	$\overline{D_0}$
L	L	H	L	D ₁	$\overline{D_1}$
L	H	L	L	D ₂	$\overline{D_2}$
L	H	H	L	D ₃	$\overline{D_3}$
H	L	L	L	D ₄	$\overline{D_4}$
H	L	H	L	D ₅	$\overline{D_5}$
H	H	L	L	D ₆	$\overline{D_6}$
H	H	H	L	D ₇	$\overline{D_7}$

11

Multiplexeur : applications

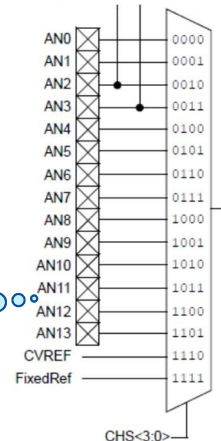
- Conversion parallèle/série: transformation parallèle-série d'une information
Il suffit de présenter les données en parallèle sur les entrées puis de faire « tourner » les bits d'entrées de sélection (à une cadence régulière).
- Sélection de périphérique

12

Cas particulier d'un multiplexeur analogique

- Principe: sélection d'une voie analogique ANx avec n entrées de sélection
- Ici 4 bits de sélection (CHS0 à CHS3) soit $2^4 = 16$ entrées analogiques sélectionnables.

Multiplexeur analogique du module CAN (ADC) d'un microcontrôleur (voir Info 2)



13

Plan du chapitre :

- I. Système combinatoire
- II. Fonction Commutation: multiplexeur, démultiplexeur
- III. Fonction Codage: décodeur, transcodeur

14

Décodeur : principe

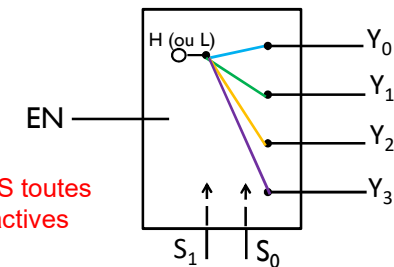
- Un **décodeur n vers N voies** comporte:
 - n entrées de sélection : $S_0, S_1, \dots, S_{n-1}$ (S_i ou A_i)
 - $N=2^n$ sorties $Q_0, Q_1, \dots, Q_{N-1}$
 - + 1 entrée de validation EN (en général)
- Circuit qui active la sortie correspondant au code binaire présent en entrée de sélection. A un instant donné, seule une sortie est active.
- **Exemples:**
 - 74139 : double décodeur de 2 voies vers 4
 - 74138 : décodeur de 3 vers 8 voies

Note: un décodeur est aussi appelé DEMULTIPLEXEUR (fonction « INVERSE » d'un multiplexeur)

15

Décodeur : principe

Pour $n=2$,
Décodeur 2 vers 4

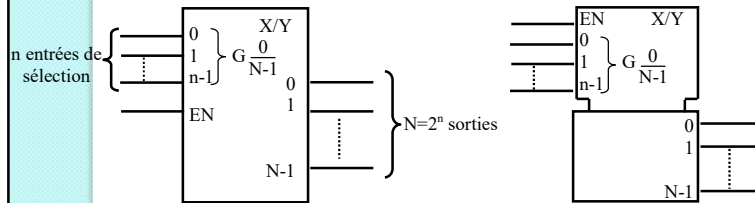


- ❑ Si EN inactive ALORS toutes les sorties Y_i sont inactives
- ❑ Si EN active ALORS

- Y_0 active si $(S_1 S_0)_2 = 0b00$, sinon Y_0 inactive
- Y_1 active si $(S_1 S_0)_2 = 0b01$, sinon Y_1 inactive
- Y_2 active si $(S_1 S_0)_2 = 0b10$, sinon Y_2 inactive
- Y_3 active si $(S_1 S_0)_2 = 0b11$, sinon Y_3 inactive

16

Décodeur : symboles



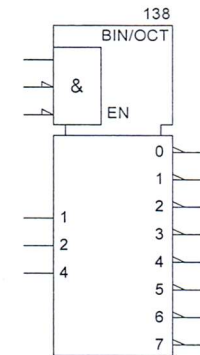
17

Décodeur : exemple du 74138

Décodeur comportant :

- 3 entrées de sélection (A,B,C),
- 8 sorties Y[0..7] actives à l'état bas,
- 3 entrées de validation G1, /G2A et /G2B

$$EN = G1. /G2A. /G2B$$



18

Décodeur : exemple du 74138

- Table de fonctionnement

Entrées de validation		Entrées de sélection			Sorties							
G1	/G2	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	L	H	H	H	H	H
H	L	L	L	L	H	H	H	L	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H
H	L	L	L	L	H	H	H	H	L	H	H	H
H	L	L	L	H	H	H	H	H	H	L	H	H
H	L	L	L	L	H	H	H	H	H	H	L	H

avec /G2=/G2A+/G2B

19

Décodeur : exemple d'application

- Décodage d'adresses

On fait correspondre les variables de poids forts d'un bus d'adresse aux entrées du décodeur.

La sortie correspondante à la combinaison des entrées (active souvent à l'état bas) sera reliée à l'entrée de sélection du boîtier i /CSi (mémoire)

Module complémentaire parcours ESE (BUT2 ou BUT3)

20

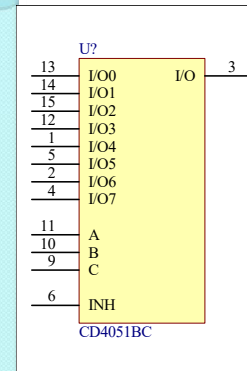
Pour aller plus loin...

Multiplexeur/Démultiplexeur analogique

- Principe: Aiguillage d'une entrée analogique vers N sorties analogiques ou sélection d'une entrée analogique 1 parmi N
- $N=2^n$ avec n entrées de sélection
- Exemple: CD4051 multiplexeur/démultiplexeur analogique 8 voies avec 3 entrées de sélection et une entrée de validation.
 - Fonctionnement en multiplexeur 8 vers 1
 - Fonctionnement en démultiplexeur 1 vers 8

21

Multiplexeur/Démultiplexeur analogique 4051



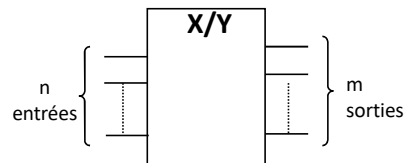
TRUTH TABLES

INPUT STATES				"ON" CHANNEL(S)
INHIBIT	C	B	A	
CD4051B				
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	X	X	X	None

22

Transcodeur : principe

- Circuit qui traduit une valeur d'un code « source » de n bits vers un code « image » de m bits

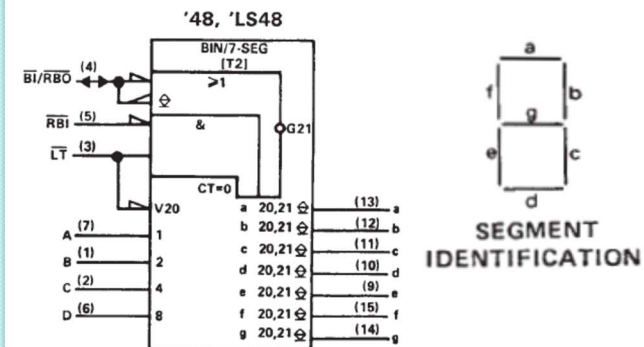


23

Transcodeur : exemple de symbole

- Exemple de circuit:

- 7448 : Transcodeur ou décodeur BCD vers 7 segments



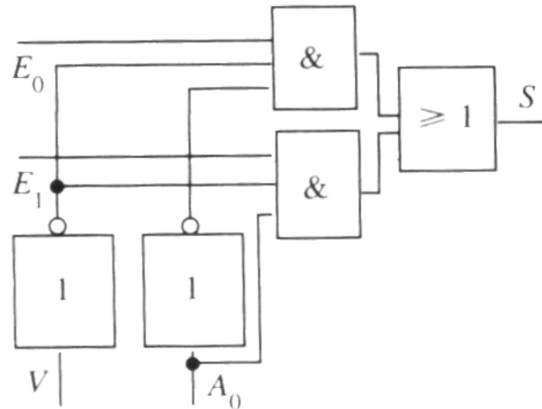
- Fonctionnement: voir document constructeur

24

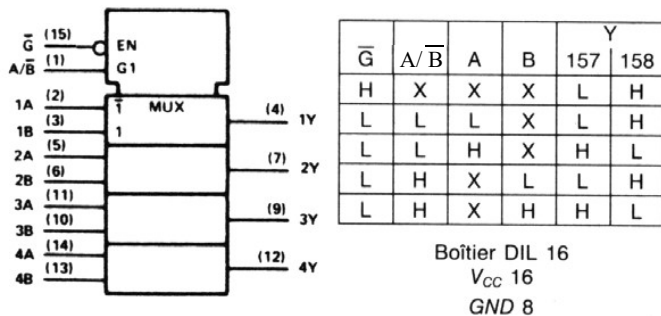
TD Chapitre 4 : Circuits combinatoires standards

Exercice 1 :

Soit le logigramme ci-dessous, E_0 et E_1 étant des entrées de données :



1. En considérant la variable d'entrée V à l'état logique « 0 », donner l'équation logique de la variable de sortie S en fonction de E_0 , E_1 et A_0 .
2. Quand $A_0 = \ll 0 \gg$, que vaut S ? Quand $A_0 = \ll 1 \gg$, que vaut S ?
3. Quel est le rôle de A_0 et de l'entrée V ? Justifier la réponse
4. On souhaite remplacer la structure précédente par le circuit existant 74157. Comment appelle-t-on ce circuit ?
5. Indiquer où présenter les variables d'entrées (E_0 , E_1 , A_0 et V) sur le circuit et où récupérer la sortie.



Exercice 2 : ETUDE PARTIELLE D'UNE STATION METEO OREGON SCIENTIFIC



On vous propose d'étudier une partie de la station météo OREGON du département GEii. Elle permet l'acquisition de plusieurs grandeurs physiques qui seront converties en numériques et traitées par un microcontrôleur PIC 16F887 (calculateur) qui ne sera étudié qu'au semestre 2

Elle possède **13 voies analogiques** (dont 3 pour des mesures de température, une pour la mesure d'hydrométrie une pour la vitesse moyenne du vent, une pour la vitesse instantanée du vent, une pour le sens du vent et une pour l'état de la batterie).

Etude du multiplexage analogique :

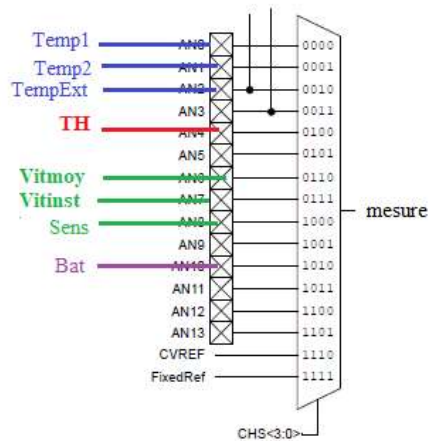
Choix du multiplexeur :

Le calculateur ne peut faire qu'une seule conversion de la donnée analogique en numérique et seule opération à la fois. On utilisera donc **un multiplexeur** qui permet de choisir la voie de mesure parmi les 13 voies

1. Combien faut-il d'entrées numériques de sélection (ou d'adresses) ? Justifier.

Sélection d'une mesure :

En fait le **multiplexeur** est intégré au composant microcontrôleur comme le montre la figure page suivante :



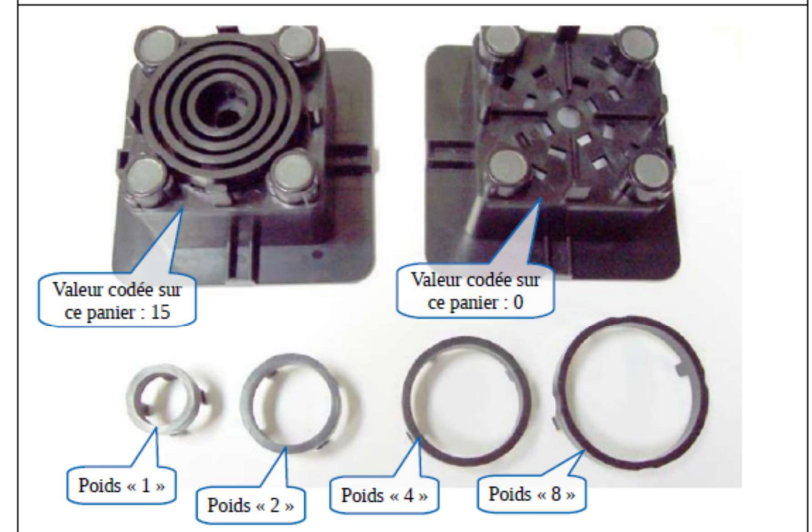
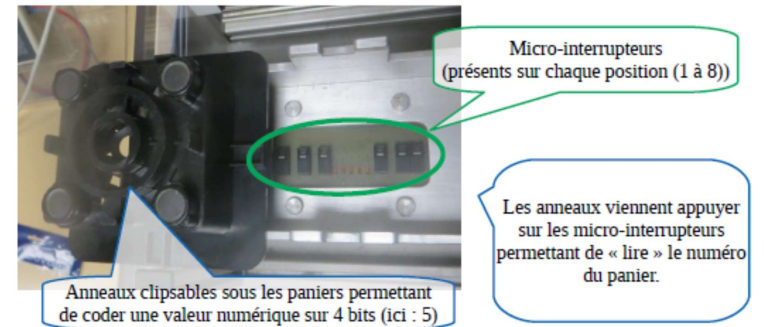
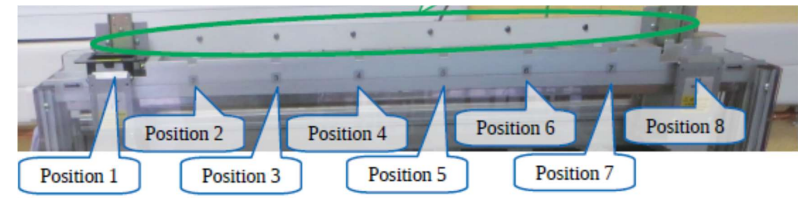
On veut faire une mesure de température extérieure TempExt qui est connectée sur l'entrée AN2

- Indiquer les valeurs des bits de sélection.
- Donner la séquence (suite des bits de sélection) pour effectuer les mesures suivantes : Temp1, Temp2, Vitesse instantanée, sens de vent

Exercice 3 : Etude partielle de la gestion des entrées d'une station de traitement de surface :

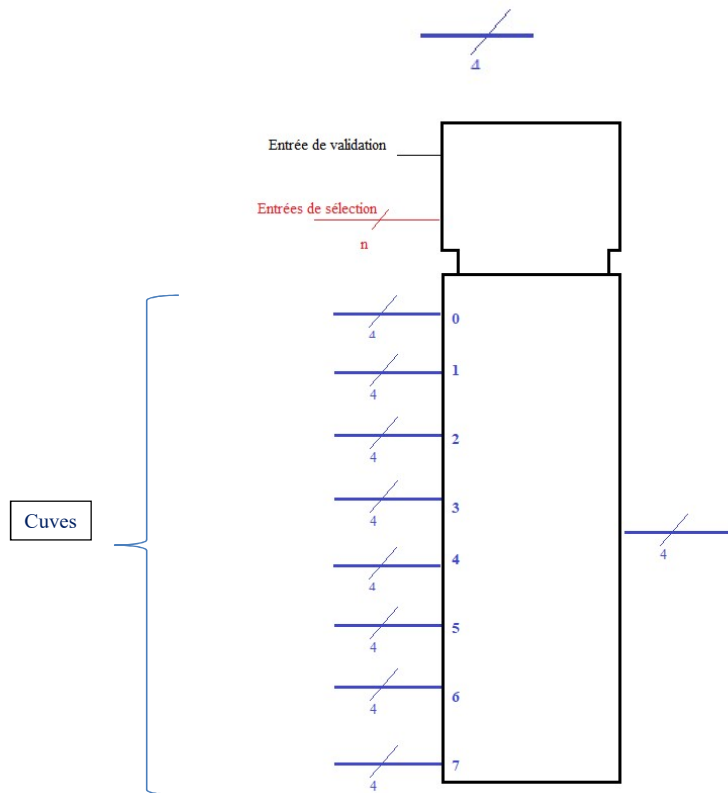
Ce système possède 8 cuves sur laquelle on pourra lire le code du panier positionné. Chaque panier possède 4 anneaux « clipsables »

Note : les photos non contractuelles sont issues de l'appareil d'Eric ESCANDE que l'on remercie pour son reportage.



- Combien faut-il de bit pour sélectionner une cuve parmi les 8 ?
- Combien de paniers peut-on avoir ?
- On peut représenter le multiplexage des cuves par le schéma fonctionnel suivant :

Note Ceci est une représentation unifilaire ici 4 fils



3.1. Quel code de sélection faut-il générer pour savoir quel est le panier sur la cuve 5 ?

3.2. On configure le panier de la façon suivante



Quel est le N° (en décimal) du panier sur la cuve 5 ?

Chapitre 5 : Arithmétique binaire

Introduction :

- code binaire naturel :
 - nombres entiers non signés (= nbs positifs)
 - 2^n combinaisons différentes pour n bits
 - on code de 0 à $2^n - 1$
- nombres binaires :
 - utilisation possible des opérateurs de base :

$$+ \quad - \quad * \quad /$$
 - pour coder des nbs négatifs (résultat soustraction)
 - pour coder des nbs réels (résultat division)

2

Plan du chapitre :

- I. Opérations en binaire
- II. Nombres entiers signés
- III. Particularités
- IV. Exemple d'un circuit arithmétique

3

Plan du chapitre :

- I. **Opérations en binaire**
 - Addition
 - Multiplication
 - division
- II. Nombres entiers signés
- III. Particularités
- IV. Exemple d'un circuit arithmétique

4

Opérations en binaire : addition

L'addition binaire repose sur :

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$



retenue

5

Exemple d'addition :

Addition de $(1001110)_B$ et $(0011011)_B$

$$\begin{array}{r} + 1001110 \\ 0011011 \\ \hline 1101001 \end{array}$$

6

Exercice d'application :

effectuer les additions suivantes :

$$\begin{array}{r} 00011100 \\ + 00101010 \\ \hline \end{array}$$

$$\begin{array}{r} 01101101 \\ + 01100111 \\ \hline \end{array}$$

7

Opérations en binaire : multiplication

La multiplication binaire repose sur :

$$0 * 0 = 0$$

$$0 * 1 = 0$$

$$1 * 0 = 0$$

$$1 * 1 = 1$$

multiplier par $(2)_D$ un nombre binaire $\Leftrightarrow$
décaler les bits d'un rang vers la gauche, et le
LSB prend pour valeur « 0 »

8

Opérations en binaire : division

La division binaire repose sur :

$$0 / 0 = > \text{impossible}$$

$$0 / 1 = 0$$

$$1 / 0 = > \text{impossible}$$

$$1 / 1 = 1$$

diviser par $(2)_D$ un nombre binaire $\Leftrightarrow$ décaler les bits d'un rang vers la droite, et le LSB est perdu

9

Plan du chapitre :

- I. Opérations en binaire
- II. **Nombres entiers signés**
 - complément à 1
 - complément à 2
- III. Particularités
- IV. Exemple d'un circuit arithmétique

10

Nombres entiers signés :

Plusieurs conventions existent, nous verrons les représentations :

- par complément restreint (cà1)
- **par complément vrai (cà2)**

11

Nombres entiers signés : complément restreint

- appelé **complément à 1** (cà1) en binaire
- démarche : opposé arithmétique d'un nombre obtenu en complémentant chacun des bits $\Rightarrow$ **cà1(N) = /N**
- convention :
 - nb positif : bit indicateur de signe vaut 0
 - nb négatif : bit indicateur de signe vaut 1

12

Nombres entiers signés : complément restreint

Exemple sur 3 bits :

	Val. Binaire	Valeur décimale
nombres positif	0 0 0	0
	0 0 1	1
	0 1 0	2
	0 1 1	3
nombres négatifs	1 0 0	-3
	1 0 1	-2
	1 1 0	-1
	1 1 1	-0

13

Nombres entiers signés : complément restreint

Inconvénients :

- pas vraiment adapté aux additions (il faut ajouter la retenue si au moins un des opérandes est négatif)
- deux représentations de la valeur « 0 »

14

Nombres entiers signés : complément vrai

- appelé **complément à 2** (cà2) en binaire
- démarche : l'opposé arithmétique d'un nombre est obtenu en complémentant chacun des bits et en ajoutant 1 au résultat

$$\Rightarrow \text{cà2}(N) = /N + 1$$

- convention :
 - nb positif : bit indicateur de signe vaut 0
 - nb négatif : bit indicateur de signe vaut 1



Le bit indicateur de signe fait partie du nombre

15

Nombres entiers signés : complément vrai

Exemple sur 3 bits :

$(N)_{10}$	N	/N	Cà2(N)	$(-N)_{10}$
0	0 0 0	1 1 1	0 0 0	0
1	0 0 1	1 1 0	1 1 1	-1
2	0 1 0	1 0 1	1 1 0	-2
3	0 1 1	1 0 0	1 0 1	-3

16

Nombres entiers signés : complément vrai

- adapté aux additions
- une seule représentation du « 0 »
- codage sur n bits : 2^n valeurs différentes
 - 2^{n-1} valeurs négatives (de -1 à -2^{n-1})
 - 2^{n-1} valeurs positives (de 0 à $2^{n-1} - 1$)
- par ex, pour n = 8, codage de -128 à +127
=> dans l'ex précédent 100 représente -4

17

Nombres entiers signés : complément vrai

- $-X \Rightarrow X$ s'obtient comme $X \Rightarrow -X$
- ex : $-3 = (101)_b \Rightarrow (010 + 1)_b = (011)_b = 3$
- obtenir la valeur décimale d'un nombre, codé en cà2 :
 - si le nombre binaire commence par 0, on procède comme avec le binaire naturel,
 - si le nombre binaire commence par 1, on le transforme en nombre positif (cà1, +1) puis on procède comme avec le binaire naturel (sans oublier le signe "-" devant la valeur décimale)

18

Nombres entiers signés : complément vrai

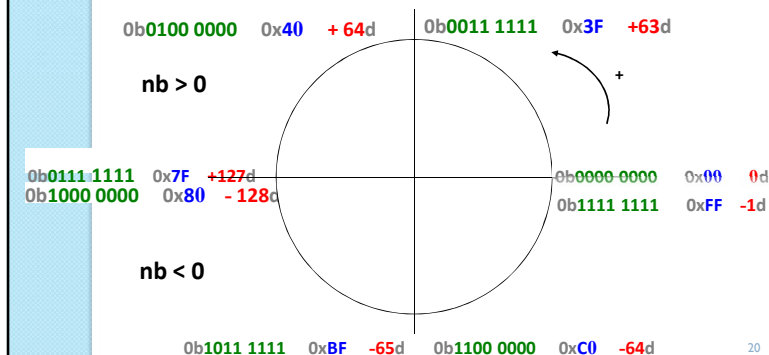
Astuce pour trouver la valeur décimale d'un nombre négatif:

- Si le nombre binaire commence par 1 (MSB) donc négatif, tous les bits en partant de la droite jusqu'au premier 1 rencontré inclus sont conservés puis tous les autres bits sont inversés
- Exemples:
 - 1) Soit **0b1011 1111**
on conserve le **1** puis on inverse les autres bits à gauche
0b0100 0001 soit -65 en décimal
 - 2) Soit -26 on a $|-26| = 26$ cad **0b0001 1010**
on conserve le **10** et on inverse les autres bits à gauche soit **0b1110 0110** soit -26 en décimal

19

Nombres entiers signés : complément vrai

représentation d'1 octet en cà2 sur un cercle



20

Plan du chapitre :

- I. Opérations en binaire
- II. Nombres entiers signés
- III. Particularités
 - Format
 - Flag
 - Débordement
- IV. Exemple d'un circuit arithmétique

21

Particularités :

Calculs effectués par une machine : les nombres sont codés sur un nombre de bits fixé (= format, ex : 16 bits)

- des "0" sont ajoutés devant les bits d'un nombre positif si nécessaire
- des "1" sont ajoutés devant les bits d'un nombre négatif si nécessaire
- le nb de valeurs que l'on peut représenter est limité

22

Particularités :

- Pour savoir si les calculs se sont bien déroulés : interroger les bits du registre d'état (= flags)
- Exemples de **flags** :
 - **CARRY** => vaut "1" quand il y a une retenue lors d'une addition
 - **BORROW** => vaut "1" quand il y a une retenue lors d'une soustraction
 - **ZERO** => vaut "1" quand le résultat de l'opération est nul
 - **OVERFLOW** => vaut "1" quand il y a un dépassement de capacité (= débordement)

23

Particularités :

- Ces indicateurs apparaissent sous la forme suivante (internes aux registres de microcontrôleurs) :

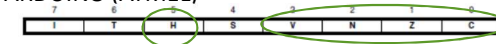
• PIC16F

Z	DC(1)	C(1)
---	-------	------

• STM8:

V	H	H	0	N	Z	C
---	---	---	---	---	---	---

- ARDUINO (ATMEL)



Z: Zero

C: Carry

N: Négatif

V: Overflow

H ou DC: Half Carry, utilisé pour des somme sur 4 bits, généralement pour des opérations en BCD

24

Particularités :

Débordement quand :

- addition de nombres positifs donne un résultat négatif
- addition de nombres négatifs donne un résultat positif

25

Plan du chapitre :

- I. Opérations en binaire
- II. Nombres entiers signés
- III. Particularités
- IV. Exemple d'un circuit arithmétique**
 - comparateurs

26

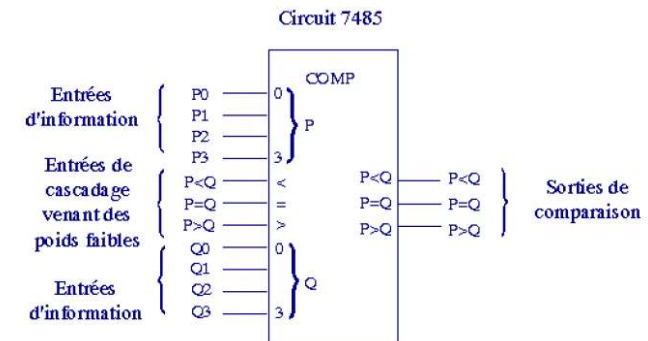
Comparateurs :

- Objectif : détecter si 2 nombres sont égaux, voire indiquer le plus grand ou le plus petit
- Principe : comparaison bit à bit en commençant par le MSB
- Egalité obtenue si l'examen de tous les bits ne permet pas de déterminer la supériorité ou l'infériorité

27

Pour aller plus loin...

Exemple de comparateur :



28

TD Chapitre 5 : Arithmétique binaire

Exercice 1 :

Dans cet exercice, les nombres binaires sont codés en cà2 sur 8 bits. Effectuer les conversions demandées.

$$(-20)_d = (\quad)_b$$

$$(48)_d = (\quad)_b$$

$$(B5)_h = (\quad)_d$$

$$(52)_h = (\quad)_d$$

$$(E3)_h = (\quad)_b$$

$$(01101011)_b = (\quad)_d$$

$$(10001101)_b = (\quad)_d$$

$$(-6)_d = (\quad)_h$$

Exercice 2 :

On considère un codage des nombres entiers signés en complément à 2 sur 12 bits. Indiquer, pour ce codage, les valeurs binaires et décimales :

- du plus petit nombre entier positif représentable
- du plus grand nombre entier positif représentable
- du plus petit nombre entier négatif représentable
- du plus grand nombre entier négatif représentable

Exercice 3 :

Dans cet exercice, les nombres binaires sont codés en cà2 sur 8 bits. Effectuer les opérations demandées et préciser la valeur des flags carry, zero, overflow.

$$\begin{array}{r} 01101011 \\ + 10111101 \\ \hline \end{array}$$

$$\begin{array}{r} 01101100 \\ + 00111110 \\ \hline \end{array}$$

$$\begin{array}{r} 11101100 \\ + 00010100 \\ \hline \end{array}$$

$$\begin{array}{r} 6F \\ + 19 \\ \hline \end{array}$$

$$\begin{array}{r} FC \\ + 04 \\ \hline \end{array}$$

TP SIN 1

Ce TP a pour but de prendre en main le banc de câblage, appelé aussi « comatest », et de câbler des circuits logiques réalisant les fonctions ET, OU, NON ET et NON OU.

La présence du symbole ► indique que vous devez faire valider ce point par un enseignant.

Il est vivement conseillé de regarder l'annexe d'Eric ESCANDE Conseils_TP_Elec_EE_V2

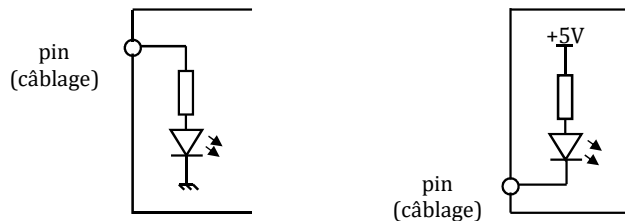
1. Prise en main du comatest :

a) Repérer la masse électrique et l'alimentation 5V que vous pourrez utiliser pour vos câblages.

b) Vérifier le fonctionnement des voyants (L.E.D.) rouges et verts en appliquant 0V, puis 5V sur leur entrée. En déduire l'état actif des LED rouges, puis des LED vertes, telles qu'elles sont câblées dans le comatest.

► Indiquez à quel schéma électrique correspond le câblage des LED rouges.

► Indiquez à quel schéma électrique correspond le câblage des LED vertes.

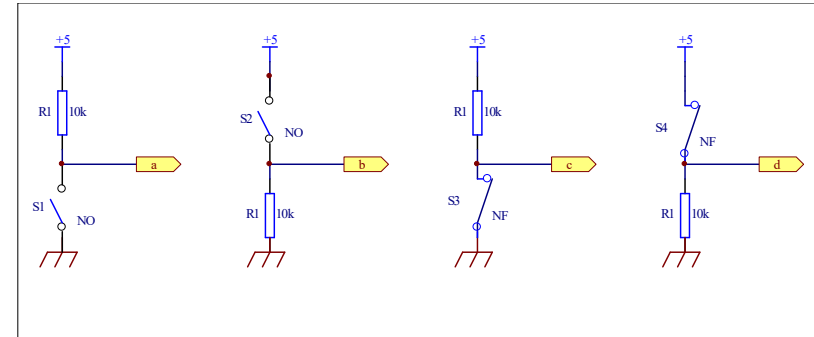


c) Actionner un interrupteur en visualisant sa sortie sur une LED. Mesurer avec un voltmètre sa tension par rapport à la masse électrique. En déduire son état H ou L en fonction de sa position physique.

Remarque : ces interrupteurs possèdent 2 positions stables. En conséquence, il faudra choisir une convention c'est-à-dire convenir de la position pour laquelle on le considère comme étant actionné. Par exemple, on peut choisir qu'ils sont actionnés quand ils sont en "haut".

d) Un bouton-poussoir est soit au REPOS, soit ACTIONNE. Chaque bouton-poussoir du banc de câblage dispose de 2 lignes de sortie. Quand un bouton-poussoir est actionné les deux lignes de sorties sont actives.

► Indiquez à quel(s) schéma(s) électrique(s) pourrai(en)t correspondre le câblage des lignes 12, 14, 16 et 18, puis celui (ceux) des lignes 13, 15, 17 et 19.



2. Circuits logiques élémentaires :

Les datasheet (= documentations des circuits logiques) peuvent être trouvés sur le classeur ou sur internet

Pour ce TP, vous avez à disposition les circuits suivants :

7400

74LS02

74HCT08

74LS32

On associe les variables logiques A et B à deux interrupteurs.

a) En vous aidant des documentations mises à votre disposition, indiquez à quelle fonction logique correspond chaque boîtier.

b) Fonction F1 : on souhaite que la LED rouge (celle de votre choix) s'allume si A ou B est activé.

► Donnez l'expression algébrique puis réalisez cette fonction.

c) Fonction F2 : on souhaite que la LED verte (celle de votre choix) s'allume si A ou B est activé.

► Donnez l'expression algébrique puis réalisez cette fonction.

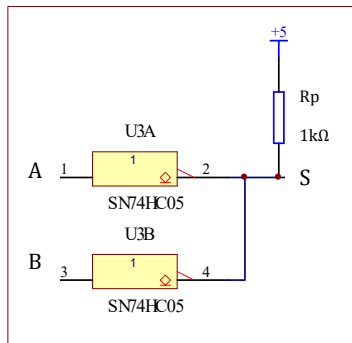
3. Analyse du circuit 74HCT05 :

a) En vous aidant de la datasheet du circuit, indiquez à quelle fonction logique correspond le boîtier et quelle est sa technologie.

b) Câblez la fonction logique $S = \overline{A}$ avec ce circuit en associant A à un interrupteur et S à une LED rouge. Que constatez-vous ? Vous pouvez utiliser un multimètre pour mesurer le niveau de tension de la sortie et de l'entrée.

► Modifiez le montage afin de corriger le problème et ainsi d'obtenir la fonction logique prévue.

- c) On associe les variables logiques A et B à deux interrupteurs et la fonction logique S à une led rouge.



► Réalisez le montage ci-dessus. Donnez l'expression algébrique de la fonction logique S.

4. Synthèse d'une fonction logique :

On associe les variables logiques A, B, C à trois interrupteurs.

a) Fonction F3 : on souhaite que la LED rouge (celle de votre choix) s'allume si A est activé et B est activé ou si B n'est pas activé et C est activé.

► Donnez l'expression graphique puis réalisez cette fonction.

b) Reprendre la fonction F3.

► Donnez l'expression graphique de cette fonction en utilisant que des NAND, puis réalisez cette nouvelle expression de F3.

5. Synthèse de fonctions logiques (BONUS)

Dans cette partie, vous utiliserez deux boutons-poussoirs (P1 et P2), 3 LED actives à l'état haut (V1, V2, et V3), 2 LED actives à l'état bas (V4 et V5) et des portes logiques élémentaires.

Donnez les expressions algébriques puis réalisez les fonctions suivantes :

- V1 est éclairé en actionnant P1 (en utilisant la ligne active par niveau bas).
- V2 est éclairé en actionnant P1 (en utilisant la ligne active par niveau haut) et en actionnant P2 (en utilisant la ligne active par niveau bas).
- V3 est éclairé en n'actionnant pas P1 (en utilisant la ligne active par niveau haut) et en actionnant P2 (en utilisant la ligne active par niveau haut).
- V4 est éteint en actionnant P1 (en utilisant la ligne active par niveau bas) ou en actionnant P2 (en utilisant la ligne active par niveau bas).
- V5 est éteint en n'actionnant pas P1 (en utilisant la ligne active par niveau haut) ou en n'actionnant pas P2 (en utilisant la ligne active par niveau bas).