

États logiques (LST)

Cours Systèmes numérique

Hes·so VALAIS
WALLIS



Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Orientation : Informatique et systèmes de communication (ISC)

Cours : Systèmes numérique (DiD)

Auteur : Christophe Bianchi, François Corthay, Pierre Pompili, Silvan Zahno

Date : 25 août 2022

Version : v2.1

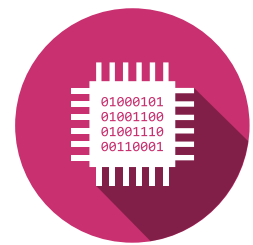


Table des matières

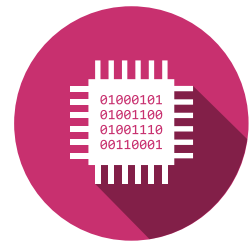
1	Introduction	2
2	État logiques de base	3
2.1	État logique '0'	3
2.2	Etat logique '1'	3
2.3	Réalisation électronique des états logiques	3
3	Portes logiques ne fournissant qu'un état	4
3.1	Interconnexion de sorties	4
3.2	Portes à open-drain (drain ouvert)	4
3.3	Portes à open-source (source ouverte)	5
4	Portes logiques avec sortie à haute impédance	7
4.1	Tampon à haute impédance	7
5	États logiques pour simulateurs	8
5.1	État inconnu	8
5.2	État non initialisé	8
6	État Logique pour synthèse	9
6.1	État non spécifié	9
	Références	10
	Acronymes	10



1 Introduction

Les états logiques de l'algèbre booléenne sont le '0' et le '1'. Toutefois, le concepteur de circuits logiques connaît d'autres états logiques permettant d'interconnecter les sorties de différents circuits ou de comprendre les résultats affichés par le simulateur.

Ce chapitre rappelle les **2 états logiques de base** et présente **d'autres états utilisés** pour la réalisation de circuits ou pour leur simulation.



2 État logiques de base

2.1 État logique '0'

L'état logique '0' correspond soit à la condition "faux" soit au chiffre binaire 0.

2.2 Etat logique '1'

L'état logique '1' correspond soit à la condition "vrai" soit au chiffre binaire 1.

2.3 Réalisation électronique des états logiques

Dans les circuits électroniques numériques, les états '0' et '1' sont réalisés en tirant la ligne électrique de sortie respectivement au potentiel négatif ou positif de l'alimentation.

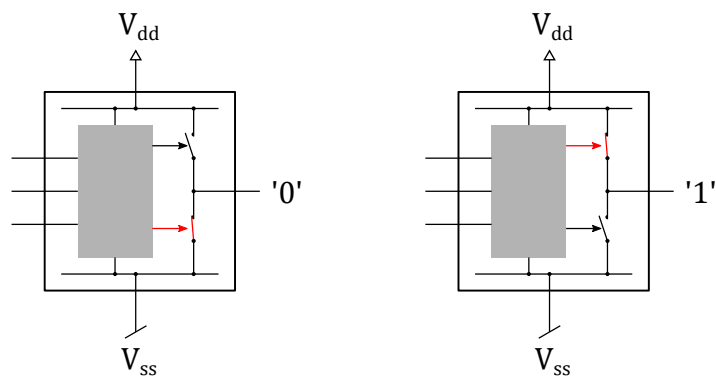


FIGURE 1 – Réalisation électronique des niveaux logiques

Les interrupteurs sont réalisés électroniquement par des transistors bipolaires pour les circuits **Transistor Transistor logic** (TTL) et par des transistors **metal-oxide-semiconductor** (MOS) complémentaires pour les circuits **Complementary metal-oxide-semiconductor** (CMOS).



3 Portes logiques ne fournissant qu'un état

3.1 Interconnexion de sorties

Il est à bannir d'interconnecter deux ou plusieurs sorties de circuits standards. En effet, si un circuit essaie d'imposer un '0' et l'autre un '1', les deux circuits réalisent un court-circuit sur l'alimentation, comme le présente la figure 2.

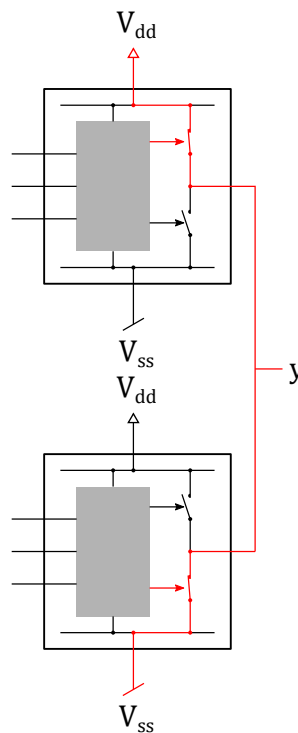


FIGURE 2 – Interconnexion de sorties réalisant un court-circuit

3.2 Portes à **open-drain** (**drain ouvert**)

Les portes à **open-drain** (**drain ouvert**) ou à **open-collector** (**collecteur ouvert**), en technologies **CMOS** et **TTL** respectivement, sont caractérisées par un étage de sortie ne comportant qu'un interrupteur vers le potentiel négatif de l'alimentation.

Ces portes ne sont donc capables de fournir qu'un état '0'. Pour les utiliser, il faut brancher une résistance entre la sortie et la borne positive de l'alimentation, de manière à tirer le potentiel de sortie à l'état '1' lorsque la porte ne le force pas à '0'. Cette caractéristique permet d'interconnecter entre elles plusieurs sorties de portes à **drain ouvert**. Dans ce cas, la ligne correspondante prendra la valeur logique correspondant à la fonction ET des sorties que tentent d'imposer les portes. Ce montage est couramment appelé un **wired-AND** (**ET-câblé**). Il est représenté à la figure 3.



Les portes à **drain ouvert** s'utilisent pour connecter plusieurs circuits à une même ligne électrique sans avoir à l'interrompre par une porte ET.

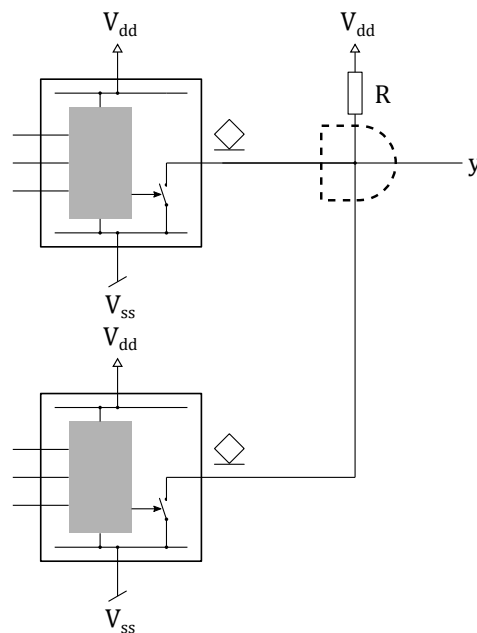
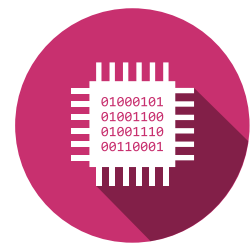
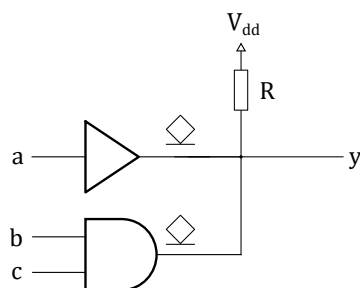


FIGURE 3 – Portes à drain ouvert

Exemple

La figure 1 présente un circuit avec des portes à **drain ouvert** et donne la table de vérité de la fonction réalisée.



<i>a</i>	<i>b</i>	<i>c</i>	<i>y</i>	<i>driver</i>
0	0	0	0	buffer, and
0	0	1	0	buffer, and
0	1	0	0	buffer, and
0	1	1	0	buffer
1	0	0	0	and
1	0	1	0	and
1	1	0	0	and
1	1	1	1	R

TABLE 1 – Circuit avec portes à **drain ouvert**

La sortie n'atteint le niveau logique '1' que lorsqu'aucune des portes ne fournit un '0'.

3.3 Portes à open-source (source ouverte)

Les portes à **open-source** (**source ouverte**) ou à **open-emitter** (**émetteur ouvert**), en technologies **CMOS** et **TTL** respectivement, sont caractérisées par un étage de sortie ne comportant qu'un interrupteur vers le potentiel positif de l'alimentation.

Symétriquement aux portes à **drain ouvert**, ces portes sont capables de ne fournir qu'un état '1'. Il faut brancher une résistance entre la sortie et la borne négative de l'alimentation. L'interconnection des sorties de plusieurs portes à **source ouverte** réalise un **wired-OR** (**OU-câblé**). Ceci est représenté à la figure 4.

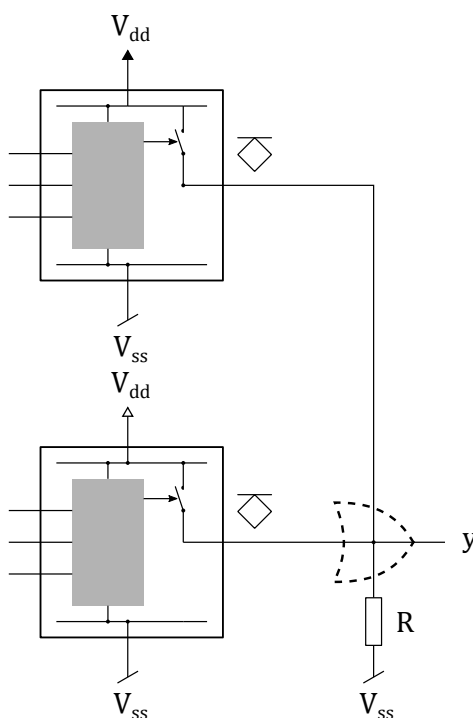
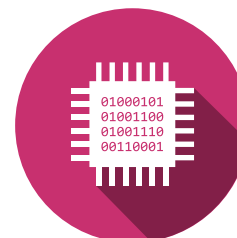
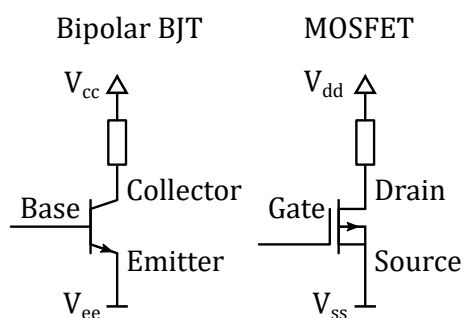
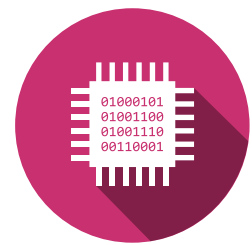


FIGURE 4 – Portes à source ouverte



Pour des raisons historiques, les portes **TTL** étant aptes à tirer une ligne à '0' plutôt qu'à '1', les circuits à **source ouverte** sont bien plus rares que les circuits à **drain ouvert**.


FIGURE 5 – Désignations **BJT** et **MOSFET**



4 Portes logiques avec sortie à haute impédance

Dans les circuits électroniques numériques, les états '0' et '1' sont réalisés en tirant la ligne de sortie respectivement au potentiel négatif et positif de l'alimentation. Certaines portes logiques ont la possibilité supplémentaire de laisser la sortie flottante, c'est-à-dire non reliée à un des potentiels de l'alimentation, comme le montre la figure 6. Cet état est considéré comme étant à **high impedance, hi-Z (haute impédance)**. Il est représenté par la lettre 'Z'. Comme ces portes présentent 3 états possibles de sortie, elles sont aussi référencées comme étant à **tri-state output (sortie 3-états)**.

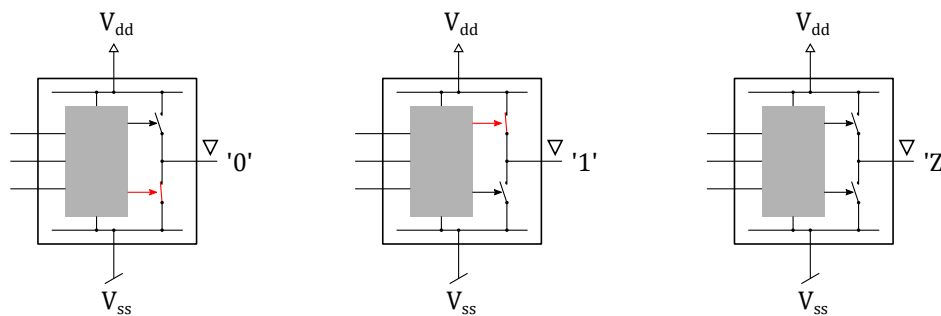


FIGURE 6 – Porte à sortie à haute impédance

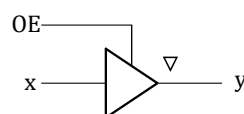


Les sorties à **haute impédance** s'utilisent pour permettre à plusieurs circuits d'écrire des valeurs sur une même ligne ou un même bus. L'utilisation typique est un bus de données où processeurs, mémoires mortes, mémoires vives et périphériques échangent de l'information.

Contrairement à ceux pilotées par des portes à **drain ouvert**, les bus à **haute impédance** nécessitent une coordination entre les circuits pour éviter que deux ou plusieurs d'entre eux ne tentent d'écrire des valeurs différentes sur une même ligne.

4.1 Tampon à haute impédance

La figure 2 présente un tampon à **haute impédance** et sa table de vérité.



OE	x	y
0	0	Z
0	1	Z
1	0	0
1	1	1

TABLE 2 – Tampon à haute impédance

Ce circuit transmet le signal d'entrée lorsque la commande **Output Enable (OE)** est active.



5 États logiques pour simulateurs

5.1 État inconnu

Dans certaines conditions, lesquelles ne devraient normalement pas apparaître dans un circuit réalisé correctement, le simulateur ne peut déterminer l'état logique d'un signal. C'est par exemple le cas lorsqu'une entrée d'une porte logique n'est reliée à rien. Dans ce cas, le simulateur affiche cette condition par l'état **unknown** (**inconnu**). Celui-ci est représenté par la lettre 'X'.

Une ligne qui n'est relié à rien délivre un signal à haute impédance, représenté, rappelons-le, par la lettre 'Z'.



FIGURE 7 – Etat inconnu



Dans un circuit physique, le potentiel électrique associé à la ligne à l'état 'X' dépend des caractéristiques électroniques des circuits logiques concernés. Ce potentiel peut se retrouver en-dehors des régions définissant les états logiques '0' ou '1', ce qui peut poser des problèmes de fonctionnement de l'électronique.

5.2 État non initialisé

A la mise sous tension d'un circuit, les valeurs mémorisées par des éléments de mémoire peuvent avoir un état quelconque. Les simulateurs tiennent généralement compte de cet état de fait en définissant un état **uninitialized** (**non-initialisé**). Cet état est représenté par la lettre 'U'.

Exemple : élément de mémoire

La figure 8 présente l'exemple d'un élément de mémoire. Au démarrage du circuit, et tant que la mise à un n'est pas activée, l'état de la sortie ne peut être déterminé par le simulateur. Celui-ci affichera donc un état 'U' alors que le circuit physique fonctionnerait déjà, même s'il est impossible d'en déterminer l'état de manière univoque.

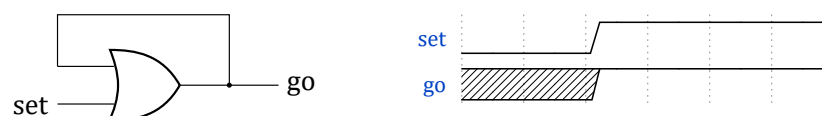


FIGURE 8 – Etat non-initialisé



L'existence de signaux à l'état 'U' qui ne peuvent être mis dans un autre état par une séquence des entrées du système indique que celui-ci n'est pas testable, même s'il peut être fonctionnel.



Certains simulateurs ne connaissent pas d'état 'U' et le remplacent par l'état 'X'.



6 État Logique pour synthèse

6.1 État non spécifié

Lors de la conception de circuits logiques, certaines conditions ne se présentent jamais (condition “never happens”) ou, lorsqu’elles se présentent, permettent d’attribuer n’importe quelle valeur à certains signaux (condition “don’t care”) car ils ne sont alors pas pris en compte.

Un état non spécifié est représenté par la lettre ‘-’.

Exemple : fonction incomplètement définie

La figure 9 présente un affichage à 7 segments utilisé pour représenter un chiffre décimal. La fonction qui convertit un nombre binaire codés sur 4 bits dans les 7 signaux de commande pour l’allumage des 7 segments est définie pour les valeurs correspondant aux chiffres décimaux de 0_d à 9_d . Si les nombres de 10_d à 15_d n’apparaissent jamais, on peut en profiter pour simplifier le circuit qui réalise la fonction de conversion.

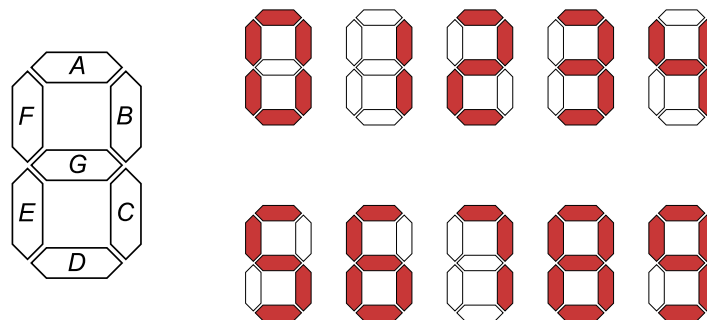


FIGURE 9 – Fonction incomplètement définie

La figure 10 présente la simplification sous forme de somme de produits de la fonction du segment B .

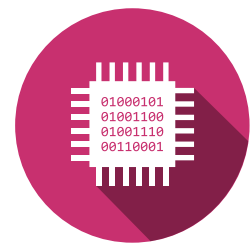
	n_2 n_3		
	1	1	-
n_0	1	0	-
n_1	1	1	-
	1	0	-

$$\bar{n}_2 + \bar{n}_1 \bar{n}_0 + n_1 n_0$$

FIGURE 10 – Simplification de la fonction du segment B



Le résultat de cette simplification permet d’obtenir un circuit logique plus simple que celui où les états non spécifiés sont remplacés par un état logique donné, ‘0’ ou ‘1’.



Références

- [1] Suhail ALMANI. *Electronic Logic Systems*. second edition. New-Jersey : Prentice-Hall, 1989.
- [2] Jean Michel BERNARD et Jean HUGON. *Pratique Des Circuits Logiques*. quatrième édition. Paris : Eyrolles, 1987.
- [3] Michael D. CILETTI et M. Morris MANO. *Digital Design*. second edition. New-Jersey : Prentice-Hall, 2007.
- [4] David J. COMER. *Digital Logic and State Machine Design*. Saunders College Publishing, 1995.
- [5] Marcel GINDRE et Denis ROUX. *Electronique Numérique, Logique Combinatoire et Technologie*. Paris : McGraw-Hill, 1987.
- [6] Ronald J. TOCCI et André LEBEL. *Circuits Numériques : Théorie et Applications*. deuxième édition. Ottawa : Editions Reynald Goulet inc. / Dunod, 1996.

Acronymes

BJT bipolar junction transistor. 6

CMOS Complementary metal-oxide-semiconductor. 3–5

collecteur ouvert open-collector. 4

drain ouvert open-drain. 4–7

ET-câblé wired-AND. 4

haute impédance high impedance, hi-Z. 7

inconnu unknown. 8

MOS metal-oxide-semiconductor. 3

MOSFET metal-oxide-semiconductor field-effect transistor. 6

non-initialisé uninitialized. 8

OE Output Enable. 7

OU-câblé wired-OR. 5

sortie 3-états tri-state output. 7

source ouverte open-source. 5, 6

TTL Transistor Transistor logic. 3–6

émetteur ouvert open-emitter. 5