

BOOLÉENS ET PORTES LOGIQUES

Vendredi 22 Mai

Option Informatique
Ecole Alsacienne

AVANT DE COMMENCER

- TD à rendre
 - Méthodes de chiffrement
 - Le jeu de la vie
 - Tris et complexité
 - Le jeu de la vie – version graphique
 - Date après prolongation : 17 mai
- Date de fin de saisie des notes pour le 3^e trimestre : 1^{er} juin
- Prochaines séances
 - 29 mai : devoir sur table
 - 5 juin : correction du devoir et bilan de l'année

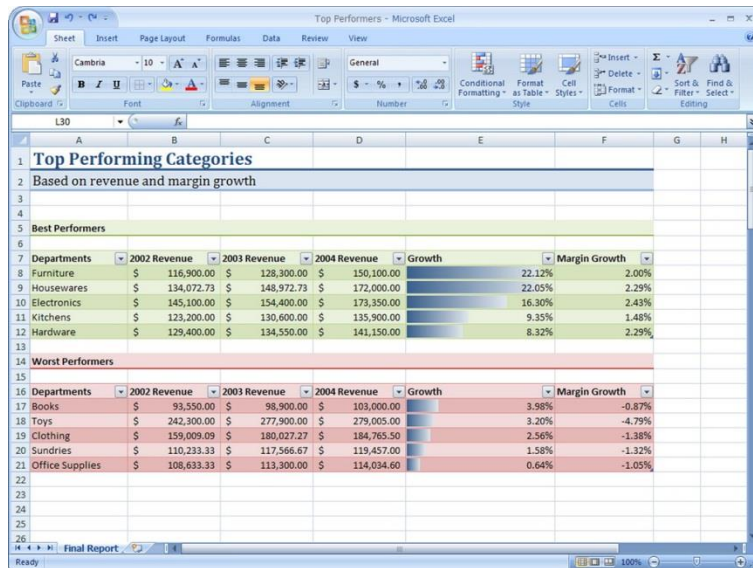
PLAN

1. Introduction
2. Booléens
3. Formules booléennes classiques
4. Propriétés
5. Circuits booléens
6. Exercices

INTRODUCTION

INTRODUCTION

Un ordinateur, ça fait des tas de trucs...



The screenshot shows a Microsoft Excel spreadsheet titled "Top Performers - Microsoft Excel". The spreadsheet is divided into sections for "Top Performing Categories" and "Worst Performers". The data is organized into columns for Departments, 2002 Revenue, 2003 Revenue, 2004 Revenue, Growth, and Margin Growth. The "Top Performers" section includes Furniture, Housewares, Electronics, Kitchens, and Hardware. The "Worst Performers" section includes Books, Toys, Clothing, and Office Supplies.

Departments	2002 Revenue	2003 Revenue	2004 Revenue	Growth	Margin Growth
Furniture	\$ 116,900.00	\$ 128,300.00	\$ 150,100.00	22.12%	2.00%
Housewares	\$ 134,072.73	\$ 148,972.73	\$ 172,000.00	22.05%	2.29%
Electronics	\$ 145,100.00	\$ 154,400.00	\$ 173,350.00	16.30%	2.43%
Kitchens	\$ 123,200.00	\$ 130,600.00	\$ 135,900.00	9.35%	1.48%
Hardware	\$ 129,400.00	\$ 134,550.00	\$ 141,150.00	8.32%	2.29%
Books	\$ 93,550.00	\$ 98,900.00	\$ 103,000.00	3.98%	-0.87%
Toys	\$ 242,300.00	\$ 277,900.00	\$ 279,005.00	3.20%	-4.79%
Clothing	\$ 159,009.09	\$ 180,027.27	\$ 184,765.50	2.56%	-1.38%
Sundries	\$ 110,233.33	\$ 117,566.67	\$ 119,457.00	1.58%	-1.32%
Office Supplies	\$ 108,633.33	\$ 113,300.00	\$ 114,034.60	0.64%	-1.05%



La question : comment ?

INTRODUCTION

Plus simplement, comment un ordinateur sait-il que :

- $40 + 2 = 42$?
- $1 + 1 = 2$?
- $0 \neq 1$?

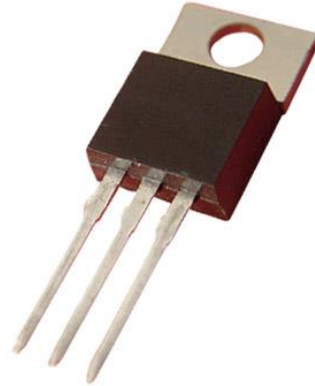
LA RÉPONSE EN DEUX MOTS

0

1

QUELQUES PETITES QUESTIONS

- Qu'est-ce que c'est ?



Un transistor !

- Un exemple concret de porte logique
- Mais au fait, c'est quoi une porte logique ?

BOOLÉENS

0 ou 1

0

Faux

Non

Ouvert

Arrêt

Nul

Tension 1

1

Vrai

Oui

Fermé

Marche

Positif

Tension 2

DÉFINITION

- Un **booléen** est une donnée qui ne peut avoir que deux états possibles.
- Ces deux états, aussi appelés **valeurs de vérité**, sont généralement noté `Vrai` et `Faux`, ou encore 1 et 0
- Un booléen peut être
 - Une **constante booléenne** (dont la valeur de vérité est toujours la même)
 - Une **variable booléenne** (dont la valeur de vérité peut changer)

EXEMPLES DE BOOLÉENS

Constantes

- `Vrai`
- `False`
- `4 est plus grand que 6`

Variables

- `Le nombre P est plus grand que le nombre Q`
- `Il pleut aujourd'hui`

FUNCTION LOGIQUE

- On appelle **fonction logique** une fonction prenant en argument un (ou plusieurs) booléen(s) et renvoyant un (ou plusieurs) booléen(s).

$$f : \mathbb{B}^n \rightarrow \mathbb{B}^m$$

- On parle aussi de **porte logique**, mais plutôt pour désigner la représentation graphique associée ou le composant électronique correspondant.

TABLE DE VÉRITÉ

- Une **table de vérité** est une liste exhaustive des valeurs d'une fonction logique pour toutes les valeurs possibles de ses arguments
- Une telle table est souvent représentée de la façon suivante :

x	y	$f(x, y)$
0	0	$f(0,0)$
0	1	$f(0,1)$
1	0	$f(1,0)$
1	1	$f(1,1)$

EXEMPLE : LA PORTE ET

- La fonction ET réalise la conjonction entre deux booléens :

$$f : \mathbb{B}^2 \rightarrow \mathbb{B}$$

$$f(x, y) = \text{Vrai} \text{ si et seulement si } \begin{cases} x = \text{Vrai} \\ y = \text{Vrai} \end{cases}$$

x	y	$f(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1

FONCTIONS BOOLÉENNES CLASSIQUES


CONSTANTE **VRAI**

- Description : Constante toujours vraie
- Notation : `Vrai` ou `True` ou `T` ("*top*")
- Valeur de vérité : 1
- Représentation graphique :

1
- Nom anglais : `True`

CONSTANTE FAUX

- Description : Constante toujours fausse
- Notation : `Faux` ou `False` ou \perp ("*bottom*")
- Valeur de vérité : 0
- Représentation graphique :


- Nom anglais : `False`

PORTE NON

- Description : Contraire d'un booléen x
- Notation : $\neg x$ ou \bar{x} (" x barre")

- Table de vérité :

x	\bar{x}
0	1
1	0

- Représentation graphique : $x \rightarrow \triangleleft \bar{x}$

- Nom anglais : NOT

PORTE ET

- Description : Conjonction de deux booléens x et y

- Notation : $x \wedge y$

- Table de vérité :

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

- Représentation graphique : 

- Nom anglais : AND

PORTE OU

- Description : Disjonction de deux booléens x et y

- Notation : $x \vee y$

- Table de vérité :

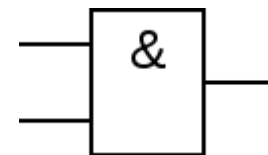
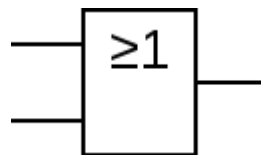
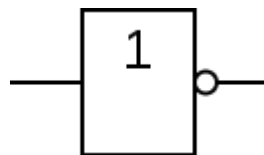
x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

- Représentation graphique : 

- Nom anglais : OR

REMARQUE GÉOGRAPHIQUE

- Les symboles présentés dans les slides précédents sont appelés **symboles américains**
- C'est une norme de représentation (nommée ANSI/IEEE 91-1984) adaptée aux schémas simples et aux tracés à la main.
- Il existe d'autres normes, notamment des **symboles** dits **européens** (norme CEI 60617-12), moins utilisés, mais permettant de représenter davantage de circuits :



PROPRIÉTÉS

COMMUTATIVITÉ

- Les fonctions ET et OU sont commutatives :

$$x \vee y = y \vee x$$

$$x \wedge y = y \wedge x$$

DISTRIBUTIVITÉ

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

x	y	z	$(y \wedge z)$	$x \vee (y \wedge z)$	$(x \vee y)$	$(x \vee z)$	$(x \vee y) \wedge (x \vee z)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

AVEC LES CONSTANTES

- $\text{Vrai} \wedge \text{Vrai} = \text{Vrai}$
- $\text{Vrai} \wedge \text{Faux} = \text{Faux}$
- $\text{Faux} \wedge \text{Vrai} = \text{Faux}$
- $\text{Faux} \wedge \text{Faux} = \text{Faux}$

- $\text{Vrai} \vee \text{Vrai} = \text{Vrai}$
- $\text{Vrai} \vee \text{Faux} = \text{Vrai}$
- $\text{Faux} \vee \text{Vrai} = \text{Vrai}$
- $\text{Faux} \vee \text{Faux} = \text{Faux}$

AVEC LES CONSTANTES

Pour toute variable booléenne x ,

- $\text{Vrai} \wedge x = x$
- $\text{Faux} \wedge x = \text{Faux}$

- $\text{Vrai} \vee x = \text{Vrai}$
- $\text{Faux} \vee x = x$

RÈGLES DE PRIORITÉ

- **Question** : Que vaut cette formule logique ?

$$\text{Vrai} \vee \text{Faux} \vee \text{Vrai} \wedge \text{Faux}$$

- **Réponse** : Vrai

- **Explication** : Le ET est prioritaire sur le OU.

C'est comme si on avait les parenthèses suivantes :

$$\text{Vrai} \vee \text{Faux} \vee (\text{Vrai} \wedge \text{Faux})$$

- Pour modifier ces priorités, on utilise des parenthèses :

$$(\text{Vrai} \vee \text{Faux} \vee \text{Vrai}) \wedge \text{Faux}$$

THÉORÈME DE DE MORGAN

- Théorème de De Morgan :

$$\overline{x \vee y} = \bar{x} \wedge \bar{y}$$

$$\overline{x \wedge y} = \bar{x} \vee \bar{y}$$

x	y	$x \vee y$	$\overline{x \vee y}$	\bar{x}	\bar{y}	$\bar{x} \wedge \bar{y}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

CIRCUITS BOOLÉENS

DÉFINITIONS

- Un **circuit booléen** est une combinaison de plusieurs portes logiques connectées entre elles.
- Ces circuits reçoivent en entrée un certain nombre de booléens, appelés **variables d'entrée**, et renvoient un autre jeu de booléens appelés **variables de sorties**.



LOGIQUE COMBINATOIRE

- On travaille dans le cadre de la **logique combinatoire** : les valeurs de vérité des variables de sortie dépendent uniquement des valeurs de vérité des variables d'entrée.
- Il existe d'autres modèles, dont notamment la logique séquentielle, où la sortie dépend aussi des états précédents des entrées

PORTE NON-ET

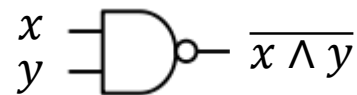
- Description : Contraire d'une porte ET

- Notation : $\overline{x \wedge y}$

x	y	$\overline{x \wedge y}$
0	0	1
0	1	1
1	0	1
1	1	0

- Table de vérité :

- Représentation graphique :



- Nom anglais : NAND

PORTE NON-ET

Circuit booléen correspondant :

x	y	$\overline{x \wedge y}$
0	0	1
0	1	1
1	0	1
1	1	0

PORTE NON-OU

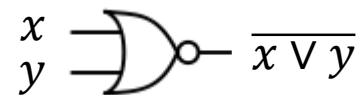
- Description : Contraire d'une porte OU

- Notation : $\overline{x \vee y}$

x	y	$\overline{x \vee y}$
0	0	1
0	1	0
1	0	0
1	1	0

- Table de vérité :

- Représentation graphique :



- Nom anglais : NOR

PORTE NON-OU

Circuit booléen correspondant :

x	y	$\overline{x \vee y}$
0	0	1
0	1	0
1	0	0
1	1	0

PORTE OU-EXCLUSIF

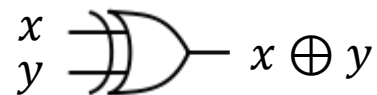
- Description : "Soit l'un, soit l'autre, mais pas les deux"

- Notation : $x \oplus y$

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

- Table de vérité :

- Représentation graphique :



- Nom anglais : XOR

PORTE OU-EXCLUSIF

Circuit booléen correspondant :

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

PORTE NON-OU-EXCLUSIF

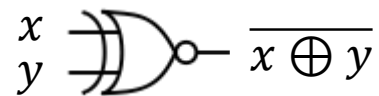
- Description : "Soit aucun, soit les deux"

- Notation : $\overline{x \oplus y}$

x	y	$\overline{x \oplus y}$
0	0	1
0	1	0
1	0	0
1	1	1

- Table de vérité :

- Représentation graphique :



- Nom anglais : XNOR

PORTE NON-OU-EXCLUSIF

Circuit booléen correspondant :

x	y	$\overline{x \oplus y}$
0	0	1
0	1	0
1	0	0
1	1	1

EXERCICES

PREMIERS EXERCICE

- **Question** : Peut-on réaliser un circuit équivalent à une porte AND sans utiliser une porte AND ?

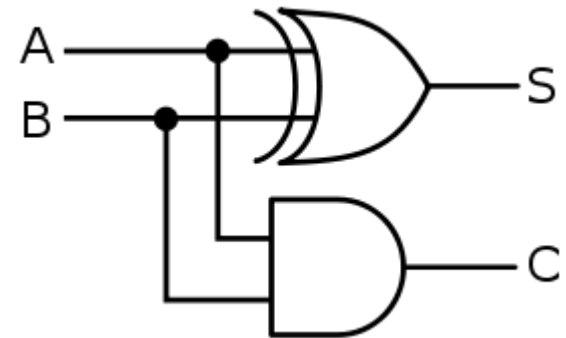
- **Question** : Peut-on réaliser un circuit équivalent à une porte XOR en n'utilisant que des portes NAND ?

ADDITIONNEUR

- Questions :
 - Comment additionner deux nombres ?
 - Comment représenter un nombre ?
 - Comment additionner deux bits ?
 - Sous quelle forme renvoyer la réponse ?

- Demi-additionneur :

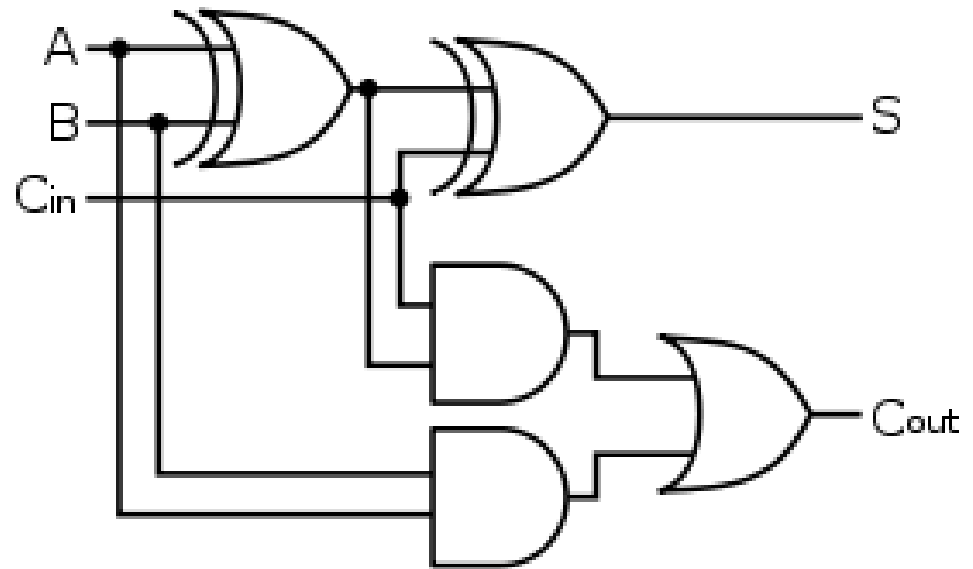
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



ADDITIONNEUR

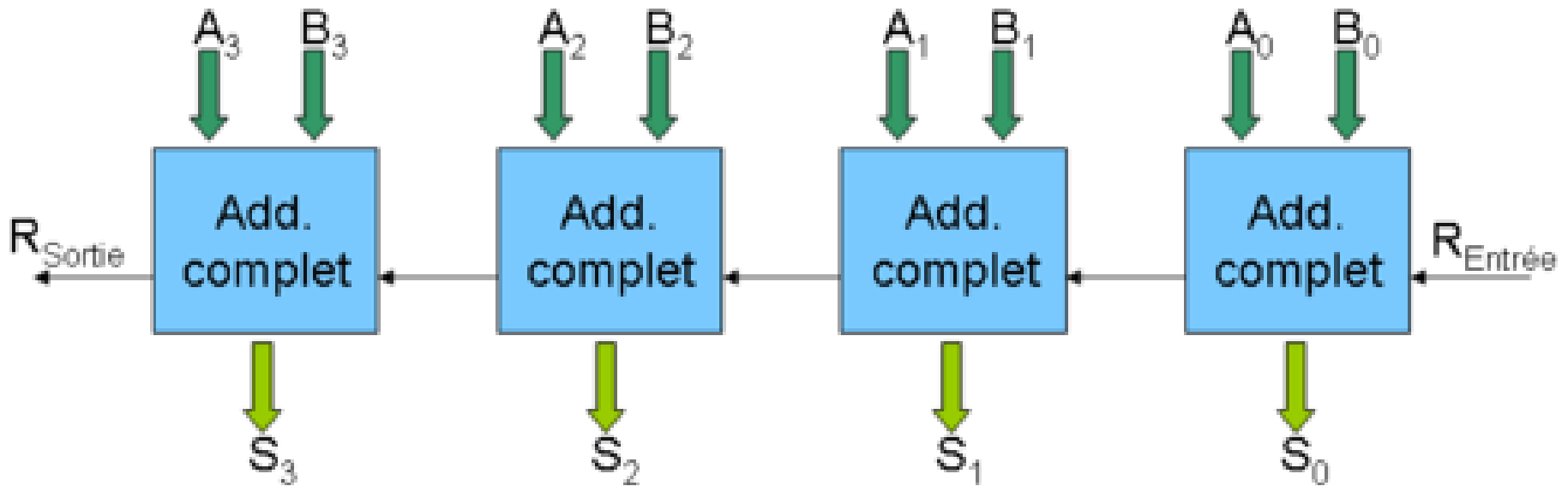
- Additionneur complet :

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



ADDITIONNEUR

- Additionneur parallèle à propagation de retenue



MULTIPLEXEUR

- Entrées et sorties :

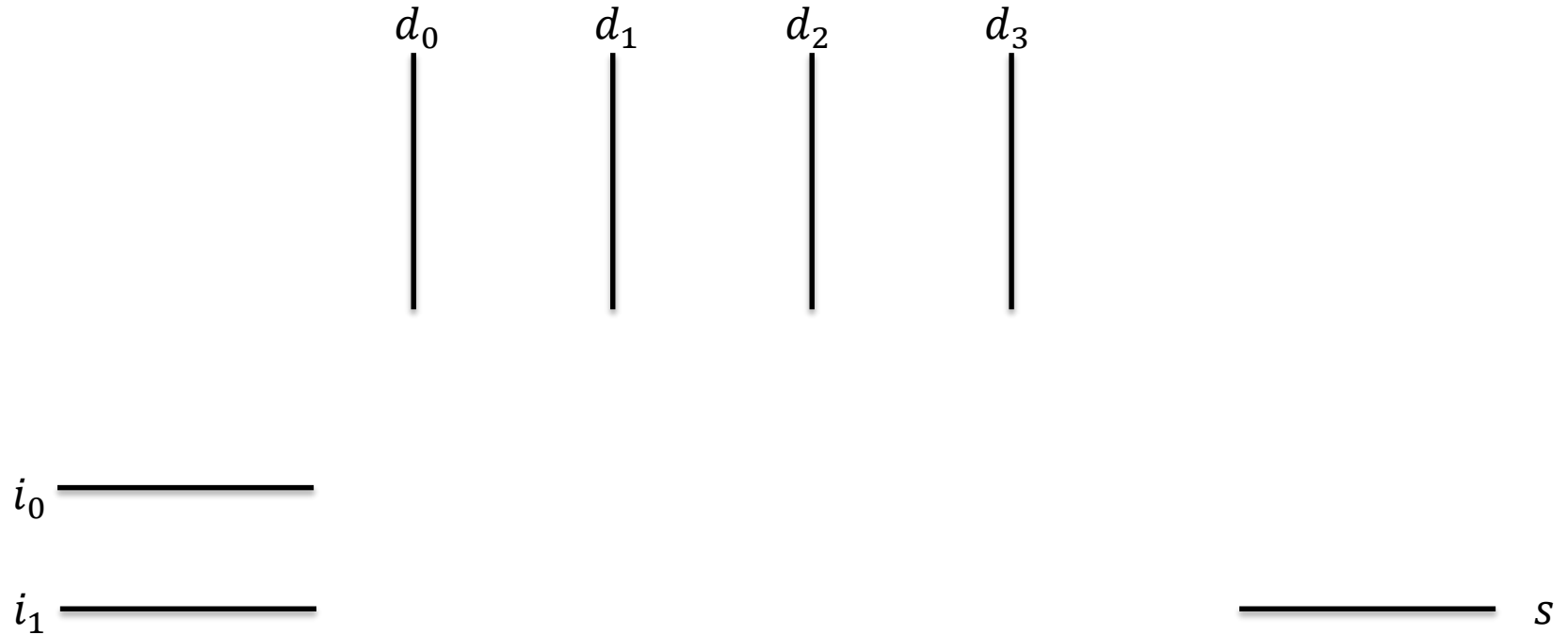
- 4 entrées de données (valeurs stockées sur la machine)
- 2 entrées de sélection
- Une sortie

- Principe :

Les valeurs de vérité des entrées de sélection définissent quelle est l'entrée de données dont la valeur de vérité est envoyée sur la sortie

MULTIPLEXEUR

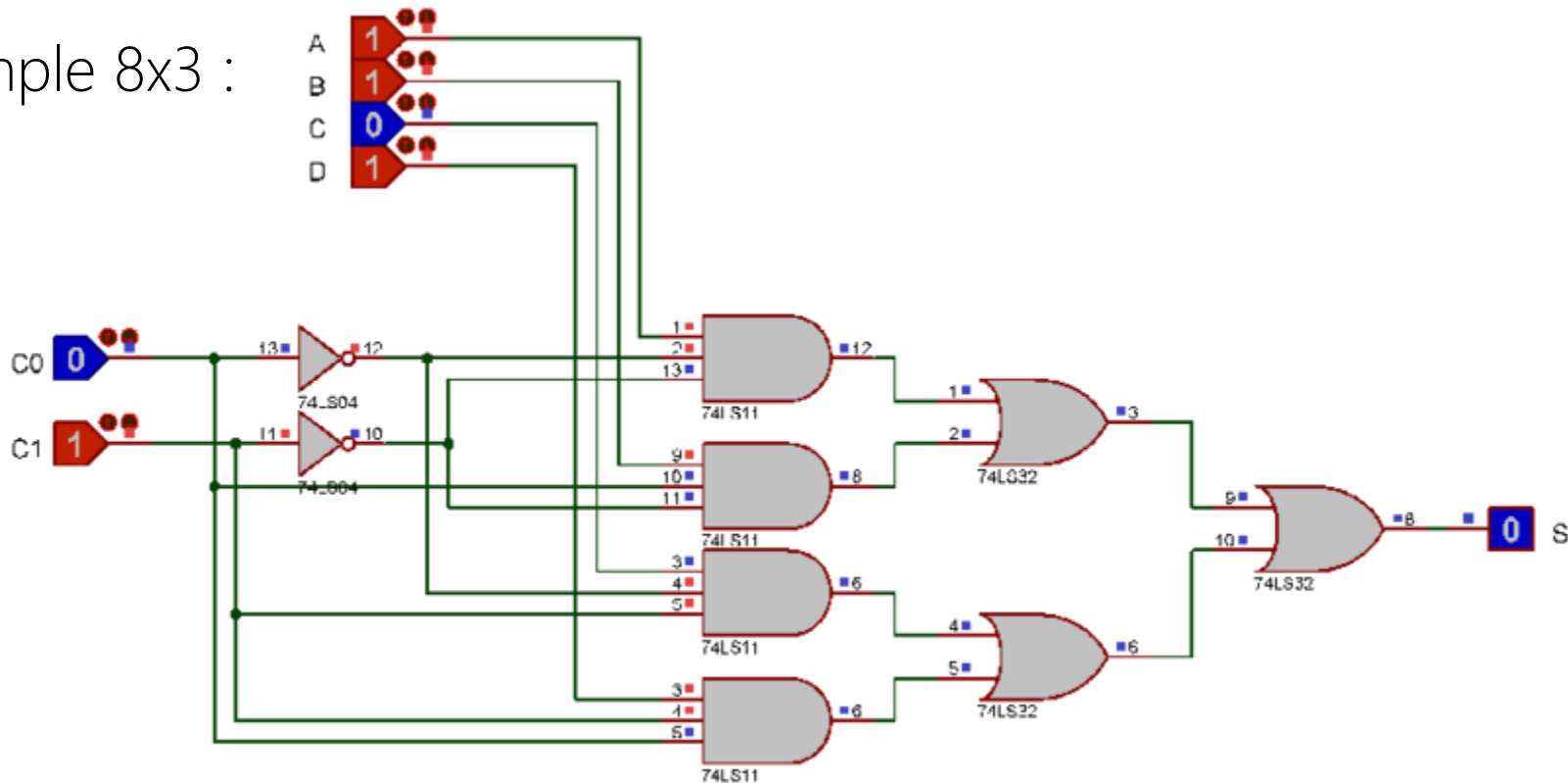
- Circuit booléens correspondant



MULTIPLEXEUR

- Formule générale :
 - 2^n entrées de données
 - n entrées de sélection

- Exemple 8x3 :



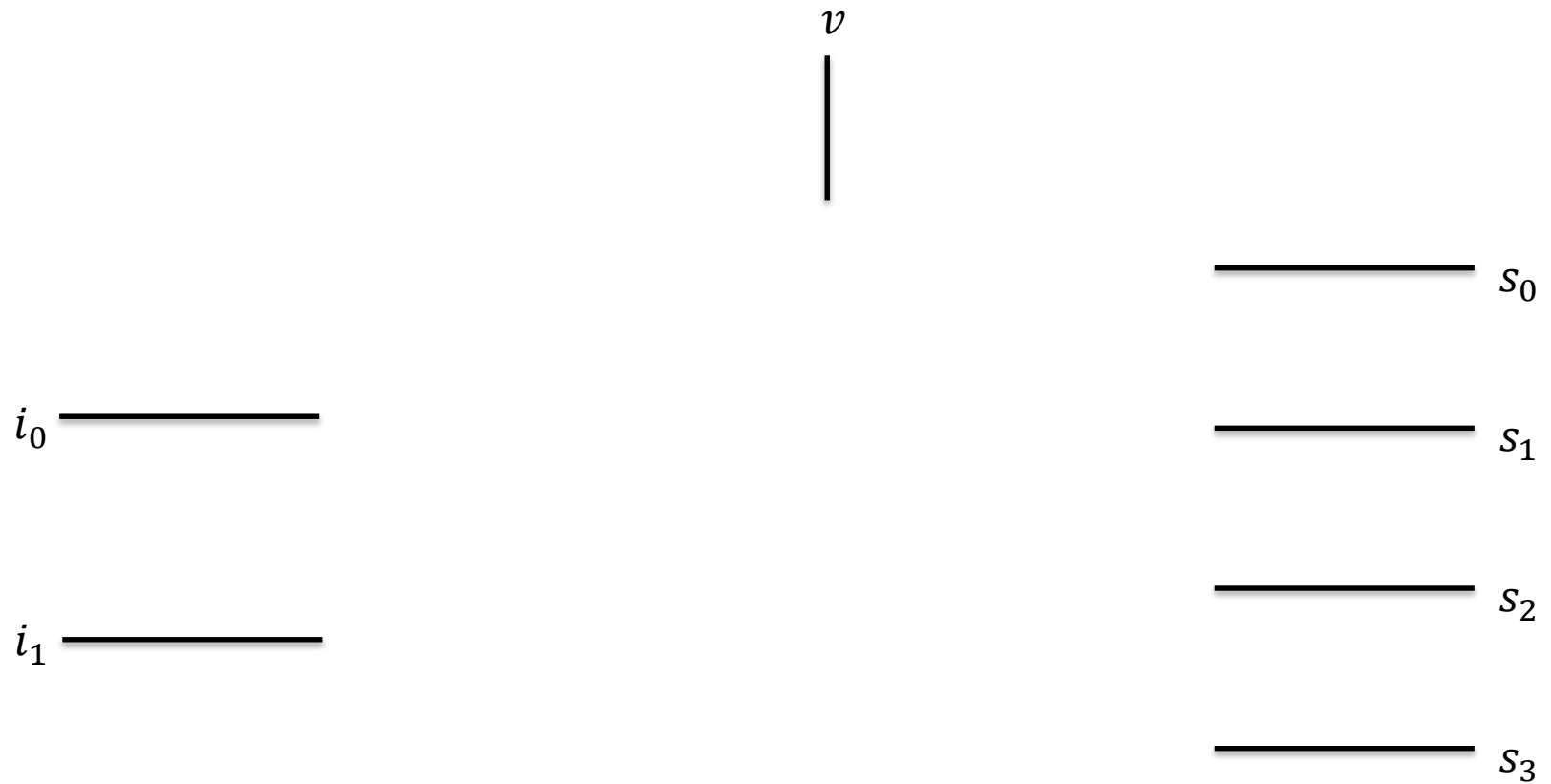
DÉMULTIPLIXEUR

- **Entrées et sorties :**
 - 2 entrées de sélection
 - Une entrée valeur
 - 4 sorties
- **Principe :**

Les valeurs de vérité des entrées de sélection définissent quelle est la sortie sur laquelle sera envoyé l'entrée valeur.

DÉMULTIPLIXEUR

- Circuit booléens correspondant



DÉMULTIPLIXEUR

- Formule générale :
 - n entrées de sélection
 - Une entrée valeur
 - 2^n sorties
- Cas particulier :
 - Si l'entrée de valeur a toujours la valeur 1, le circuit est appelé un décodeur.

PROCHAINE SÉANCE

Vendredi 29 mai

DEVOIR SUR TABLE

