Architecture et Microprocesseurs (AMP)

Responsable et Cours:	**Robert Strandh**
TD:	**Driss El Ouadghiri** **Ireneusz Tobor** **Géraud Sénizergues**
Planning:	**13 semaines** **2h de cours + 2h de TD + 4h de travail individuel par semaine**
Support de cours:	**Robert Strandh et Irène Durand** **Architecture de l'ordinateur pour informaticiens** **Transparents**

Web: http://dept−info.labri.u−bordeaux.fr/~strandh/Teaching/AMP/2001−2002/Dir.html

Objectifs et Contenu

Objectif: **Comprendre le fonctionnement interne des ordinateurs**

Contenu:

- **Circuits combinatoires et circuits séquentiels**
- **Circuits pour l'arithmétique binaire**
- **Logique à trois états et notion de bus**
- **Mémoires**
- **Instructions de base et microprogrammes**
- **Interruptions**
- **Protection et multiprogrammation**
- **Mémoire cache**
- **Mémoire virtuelle**

Circuits et Signaux

Chaque circuit a un certain nombre de "ports"

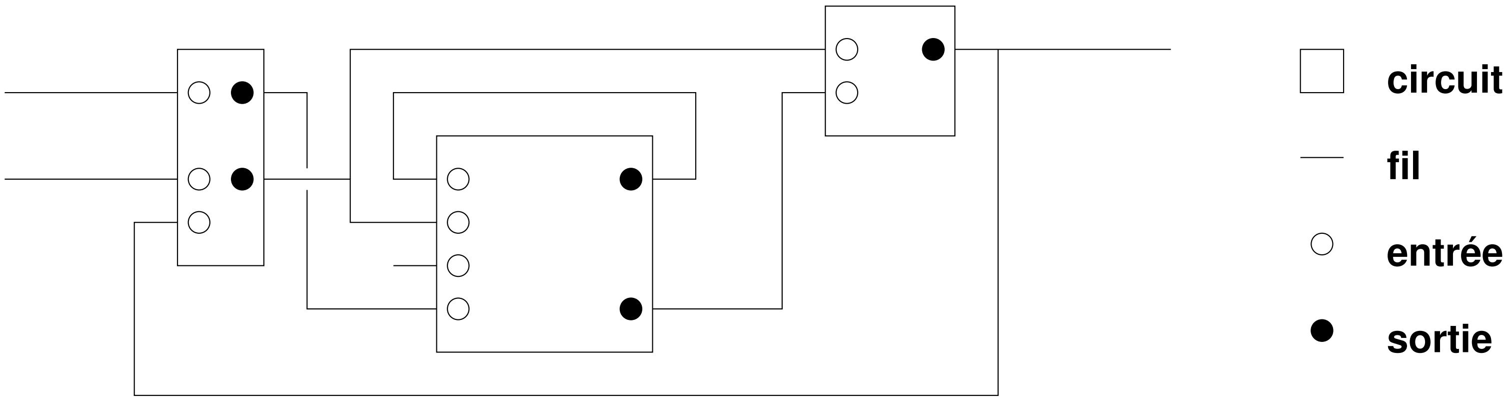
Un port est soit une entrée, soit une sortie
(cette restriction sera supprimée plus tard)

Les ports sont interconnectés par des "fils"

Les fils transmettent des signaux entre les circuits

Un fil est connecté à au plus une sortie et un nombre arbitraire d'entrées

La valeur d'un signal est soit 0 (zero, faux) ou 1 (un, vrai)



Circuits et Signaux (suite)

Les circuits sont souvent réalisés avec des transistors en silicium

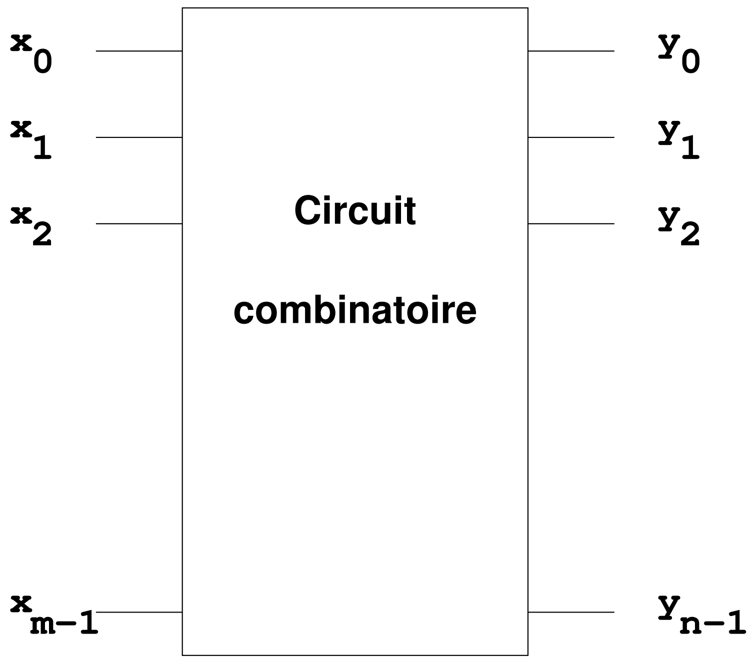
Les fils sont réalisés en aluminium ou cuivre

La valeur 0 d’un signal est souvent représentée par une tension de 0 V (environ)

La valeur 1 d’un signal est souvent représentée par une tension de 3 ou 5 V (environ)

Circuits combinatoires

Un circuit combinatoire est un circuit à m entrées et n sorties tel que la valeur des sorties dépend uniquement de celles des entrées.



Tables de vérité

Un circuit combinatoire peut être complètement décrit par une telle table

Montre la valeur de chaque sortie pour chaque combinaison des entrées

Exemple:

x	y	z	a	b
0	0	0	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

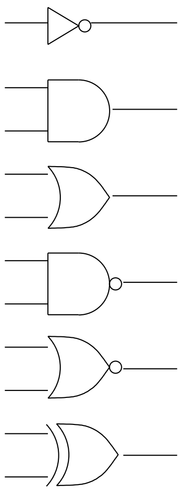
Portes logiques

Ce sont des circuits combinatoires avec les caractéristiques suivantes:

Une seule sortie

La fonction réalisée est régulière est simple

Nous allons traiter six type de portes logiques:



inverseur (ou porte "non")

porte "et"

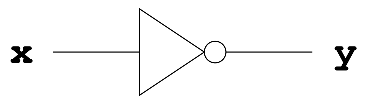
porte "ou"

porte "non−et"

porte "non−ou"

porte "ou exclusif"

Inverseur



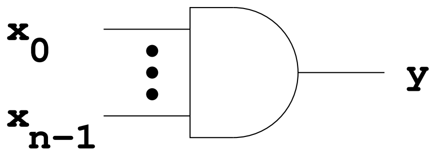
Cette porte a exactement une entrée et exactement une sortie

La valeur de la sortie est toujour l'inverse de celle de l'entrée

Voici la table de vérité de l'inverseur:

x	y
0	1
1	0

Porte "et"



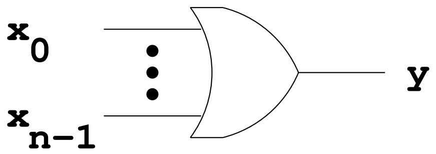
Cette porte a au moins deux entrées

La valeur de la sortie est 1 ssi la valeur de toutes les entrées est 1

Voici la table de vérité (deux entrées):

x_1	x_0	y
0	0	0
0	1	0
1	0	0
1	1	1

Porte "ou"



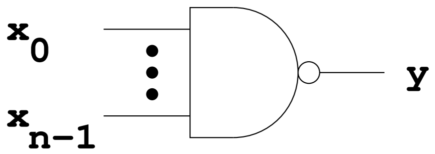
Cette porte a au moins deux entrées

La valeur de la sortie est 1 ssi la valeur d'au moins une entrée est 1

Voici la table de vérité (deux entrées):

x_1	x_0	y
0	0	0
0	1	1
1	0	1
1	1	1

Porte "non–et"



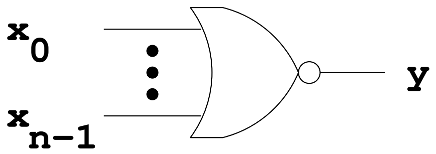
Cette porte a au moins deux entrées

La valeur de la sortie est 0 ssi la valeur de toutes les entrées est 1

Voici la table de vérité (deux entrées):

x_1	x_0	y
0	0	1
0	1	1
1	0	1
1	1	0

Porte "non−ou"



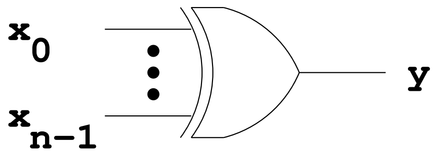
Cette porte a au moins deux entrées

La valeur de la sortie est 0 ssi la valeur d’au moins une entrée est 1

Voici la table de vérité (deux entrées):

x_1	x_0	y
0	0	1
0	1	0
1	0	0
1	1	0

Porte "ou exlusif"



Cette porte a au moins deux entrées

La valeur de la sortie est 1 ssi la valeur d'exactement une entrée est 1

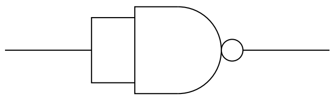
Voici la table de vérité (deux entrées):

x_1	x_0	y
0	0	0
0	1	1
1	0	1
1	1	0

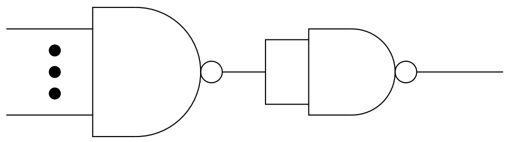
Simulation de portes

Il suffit d'avoir des portes "non−et" pour simuler les autres portes

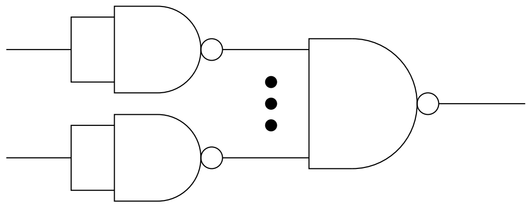
Pour simuler un inverseur, il suffit de brancher les deux entrées d'une porte "non−et" ensemble:



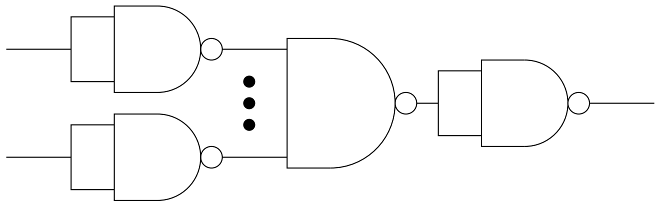
Pour simuler une porte "et" il suffit de brancher un inverseur à la sortie d'une porte "non−et":



Pour simuler une porte "ou" il suffit de brancher un inverseur à chaque entrée d'une porte "non−et":



Pour simuler une porte "non−ou" il suffit de brancher un inverseur à la sortie d'une porte "ou":



La porte "ou exclusif" est un peu plus difficile, mais possible elle aussi.

Construction de circuits combinatoires

Plusieurs critères possibles:

- **Le moins de transistors possible**
- **Le moins de consommation d'énergie possible**
- **Le plus rapide possible**

Spécification d'un circuit

- **Table de vérité**
- **Formule logique**

Spécification: formule logique

Constantes: 0, 1

Variables: x, y, z, ...

Opérateurs: $+$, $\overline{\ }$, $\cdot$

loi	forme et	forme ou
identité	$1x = x$	$0+x = x$
nullité	$0x = 0$	$1+x = 1$
idempotence	$xx = x$	$x+x = x$
inversion	$x\overline{x} = 0$	$x+\overline{x} = 1$
commutativité	$xy = yx$	$x+y = y+x$
associativité	$(xy)z = x(yz)$	$(x+y)+z = x+(y+z)$
distributivité	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$
absorption	$x(x+y) = x$	$x+xy = x$
De Morgan	$\overline{xy} = \overline{x}+\overline{y}$	$\overline{x+y} = \overline{x}\ \overline{y}$

Spécification: table de vérité

Pour toutes combinaisons des entées, donner la valeur de la sortie

x	y	z	out
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Souvent, une spécification complète n'est pas nécessaire

x	y	z	out
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	–
1	0	0	–
1	0	1	1
1	1	0	0
1	1	1	1

Abbreviation de la table de vérité

Souvent la valeur de la sortie est identique pour plusieurs combinaisons d'entrées:

x1	x2	x3	x4	x5	y
0	0	–	–	–	1
0	1	–	–	–	0
1	0	–	–	0	0
1	0	–	–	1	1
1	1	0	–	–	1
1	1	1	–	–	0

Il est possible d'avoir des valeurs de sortie non pécisées dans une table abbreviée:

x1	x2	x3	x4	x5	y
0	0	–	–	–	1
0	1	–	–	–	0
1	0	–	–	0	–
1	0	–	–	1	1
1	1	0	–	–	–
1	1	1	–	–	0

Méthode générale de construction d’un circuit combinatoire

Si la table de vérité n’existe pas, la construire

Première couche:

Pour chaque ligne dont la valeur est 1, mettre une porte non−et, avec une entrée normale pour un 1, et une entrée inversée pour un 0

Deuxième couche:

Mettre une porte non−et avec en entrée les sortie de la première couche

Exemple de construction générale

Réaliser un circuit combinatoire à partir de la formule:

$$t = x\overline{y} + z(\overline{x} + y)$$

Voici la table de vérité:

x	y	z	t
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Circuit: