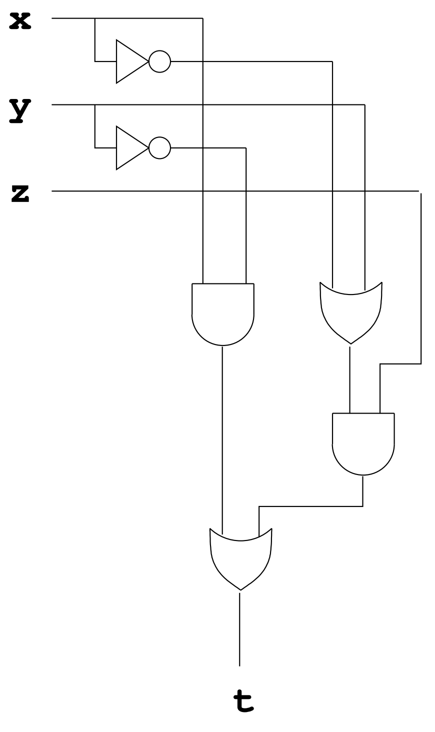
Exemple de construction directe

Réaliser un circuit combinatoire à partir de la formule:

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

Circuit:



Exemple où la méthode générale est inadaptée (Multiplexeur)

Circuit avec n entrées "adresses" et 2^n entrées "données"

L'adresse permet de choisir l'une des entrées "données"

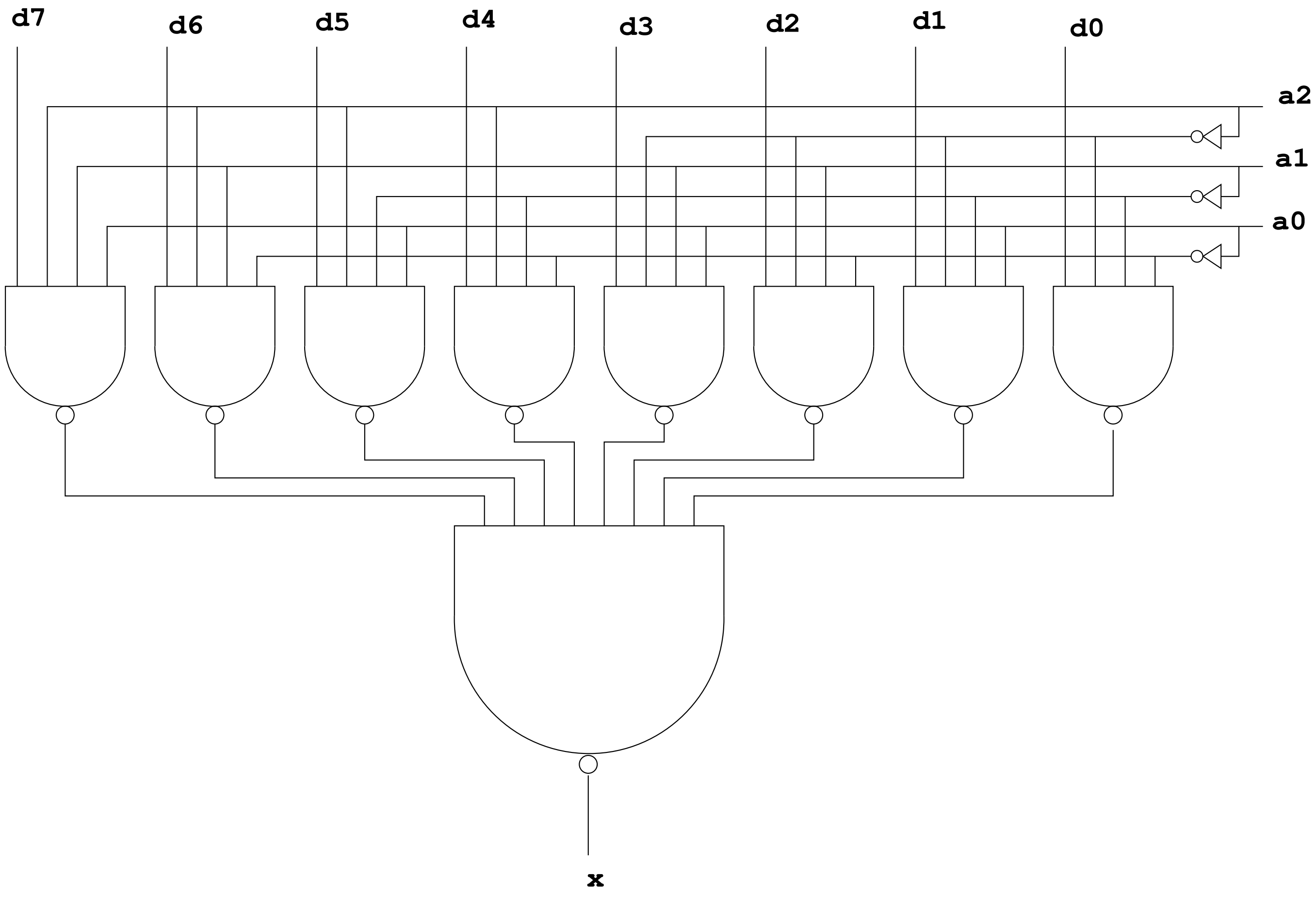
Table de vérité (abbreviée) (n = 3)

a2	a1	a0	d7	d6	d5	d4	d3	d2	d1	d0	out
0	0	0	–	–	–	–	–	–	–	c	c
0	0	1	–	–	–	–	–	–	c	–	c
0	1	0	–	–	–	–	–	c	–	–	c
0	1	1	–	–	–	–	c	–	–	–	c
1	0	0	–	–	–	c	–	–	–	–	c
1	0	1	–	–	c	–	–	–	–	–	c
1	1	0	–	c	–	–	–	–	–	–	c
1	1	1	c	–	–	–	–	–	–	–	c

La table de vérié non abbreviée a 2048 lignes dont 1024 avec out = 1

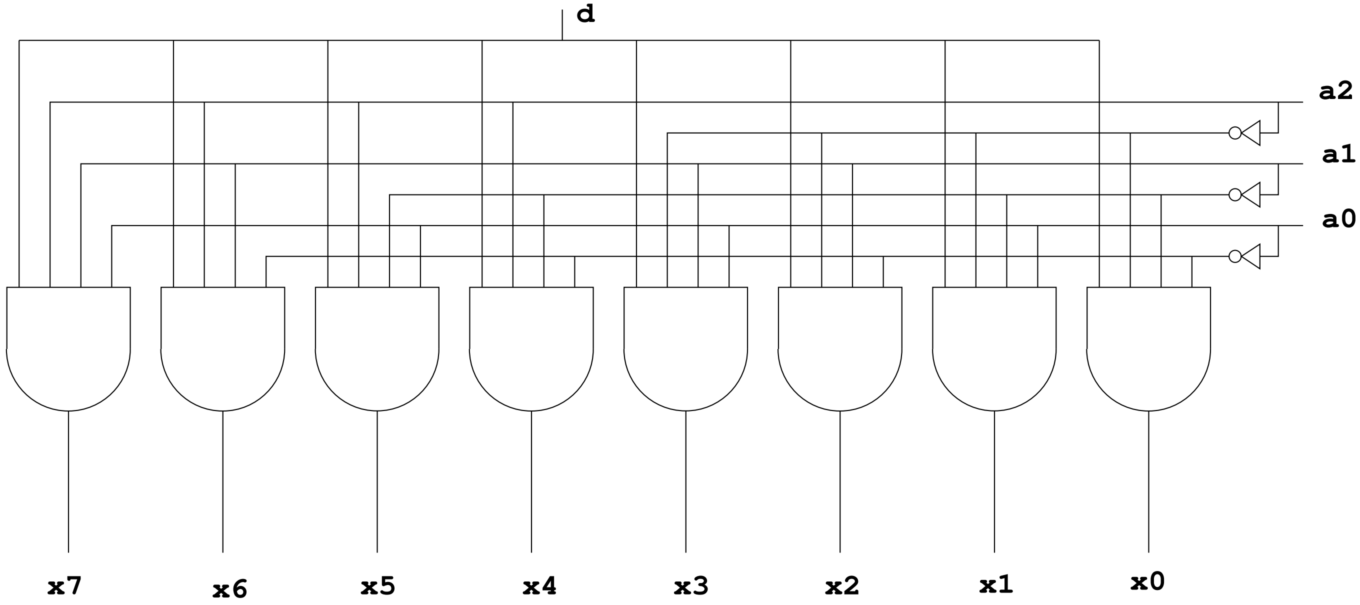
Donc le circuit construit avec la méthode générale a 1025 portes

Le circuit du multiplexeur



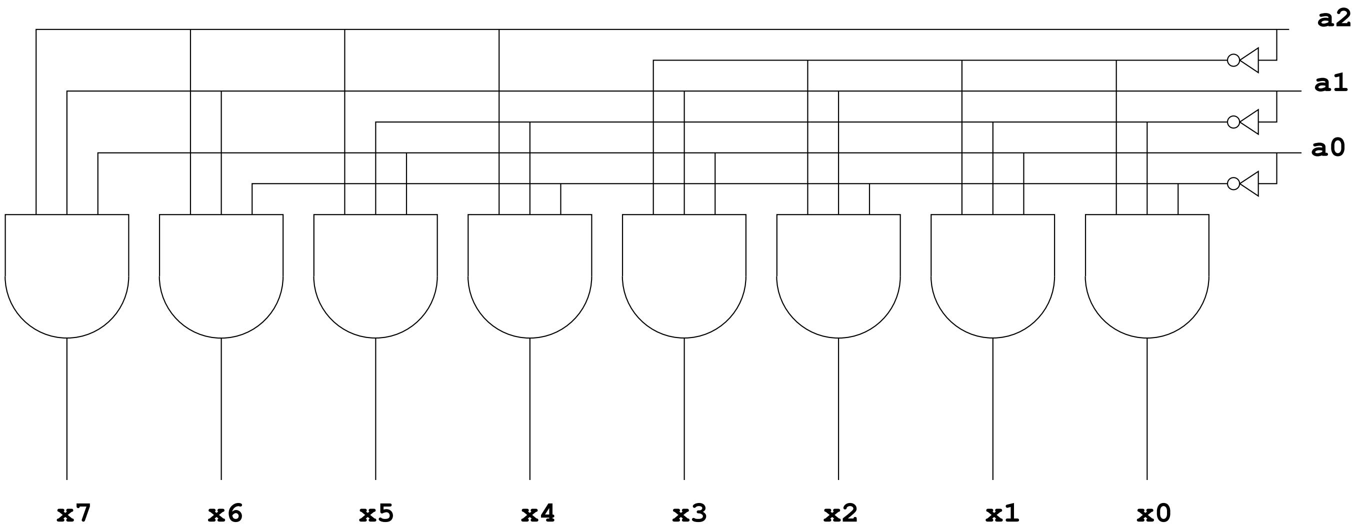
Démultiplexeur

a2	a1	a0	d	x7	x6	x5	x4	x3	x2	x1	x0
0	0	0	c	0	0	0	0	0	0	0	c
0	0	1	c	0	0	0	0	0	0	c	0
0	1	0	c	0	0	0	0	0	c	0	0
0	1	1	c	0	0	0	0	c	0	0	0
1	0	0	c	0	0	0	c	0	0	0	0
1	0	1	c	0	0	c	0	0	0	0	0
1	1	0	c	0	c	0	0	0	0	0	0
1	1	1	c	c	0	0	0	0	0	0	0



Décodeur

a2	a1	a0	x7	x6	x5	x4	x3	x2	x1	x0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



Arithmétique binaire

Représentation en base 10:

$$2034 = 2 * 10^3 + 0 * 10^2 + 3 * 10^1 + 4 * 10^0$$

Représentation en base 2:

$$11010 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$$

Algorithmes pour l'arithmétique binaire

Addition

$$
\begin{array}{cccccc}
 & \scriptstyle 1 & & \scriptstyle 1 & \scriptstyle 1 & \\
 & & 1 & 0 & 1 & 1 \\
+ & & 1 & 0 & 0 & 1 \\
\hline
 & 1 & 0 & 1 & 0 & 0
\end{array}
$$

Soustraction

$$
\begin{array}{ccccc}
 & & \scriptstyle \cancel{1}0 & \scriptstyle 10 & \\
 & \cancel{1} & 0 & 0 & 1 \\
- & & 1 & 1 & 0 \\
\hline
 & 0 & 0 & 1 & 1
\end{array}
$$

Multiplication

$$
\begin{array}{cccccccc}
 & & & & 1 & 1 & 0 & 1 \\
* & & & & & 1 & 0 & 1 \\
\hline
 & & & & 1 & 1 & 0 & 1 \\
 & & & 0 & 0 & 0 & 0 & \\
 & & 1 & 1 & 0 & 1 & & \\
\hline
 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

Division

$$
\begin{array}{cc|cccc}
 & & 0 & 1 & 1 & 0 \\
\cline{3-6}
1 & 0 & 1 & 1 & 0 & 1 \\
\cline{1-2}
 & & 1 & 0 & & \\
\cline{3-4}
 & & & 1 & 0 & \\
 & & & 1 & 0 & \\
\cline{4-5}
 & & & & 0 & 1
\end{array}
$$

Représentation de nombres négatifs

Il est possible d'utiliser une représentation avec signe + valeur absolue

Pour l'arithmétique il faut alors un circuit d'addition et un autre de soustraction

Il y a une meilleure solution qui ne nécessite qu'un circuit d'addition

Cette solution s'appele "complément à 2"

Pour la comprendre nous parlons de complément à 10 d'abord

Représentation de nombres négatifs
Complément à 10 avec précision infinie

Imaginons l'odomètre d'une voiture (ou d'un vélo) mais avec un nombre infine de roues.

Les nombres positifs sont représentés comme d'habitude

Les nombres négatifs avec un nombre infini de chiffres 9 à gauche

```
-1 = ...9999999
-2 = ...9999998
-3 - ...9999997
```

Supposons que nous avons un circuit d'addition de ce type de nombres

(Chaque nombre peut être représenté de façon finie)

C'est la représentation exacte utilisée par certains langage de programmation permettant la précision arbitraire des entiers (et des rationnels)

Représentation des nombres négatifs

Pour les nombres positifs de ce type, le circuit marche normalement

Pour additionner un nombre positif avec un nombre négatif, on fait comme si le nombre négatif était positif (et très grand)

```
... 0 0 0 3 4          (34)
... 9 9 9 9 3          (−7)
... 0 0 0 2 7          (27)
```

Pour additionner deux nombres négatifs, on fait la même chose

```
... 9 9 9 8 7          (−13)
... 9 9 9 9 3          (−7)
... 9 9 9 8 0          (−20)
```

Pour calculer la négation d'un nombre, il suffit de remplacer chaque chiffre c par (9 – c), puis finalement additioner 1 au résultat.

Représentation de nombres négatifs

Nous pouvons faire presque la même chose avec une précision finie

Mais il faut introduire la notion de débordement (overflow, underflow)

Si la roue la plus à gauche contient 0, 1, 2, 3 ou 4, alors un nombre positif
Si la roue la plus à gauche contient 5, 6, 7, 8 ou 9, alors un nombre négatif

Exemple d'addition:

```
2 3 3       2 3 3                  2 3 3
1 0 5       5 2 1   (-479)         9 9 5   (-5)
3 3 8       7 5 4   (-246)     1 2 2 8   (???)


2 3 3                     9 9 8   (-2)
3 2 1                     8 8 1   (-119)
5 5 4   (-446!!!)     1 8 7 9   (???)
```

Si le resultat de l'addition de deux nombres positifs est un nombre positif, alors le résultat est bon

Si le resultat de l'addition de deux nombres négatifs est un nombre négatif, alors le résultat est bon

Représentation de nombres négatifs

Si le résultat de l'addition d'un nombre positif et d'un nombre négatif est négatif, alors le résultat est bon

Si le résultat de l'addition d'un nombre positif et d'un nombre négatif contient un chiffre supplémentaire, alors le resultat (un supprimant le chiffre supplémentaire) est bon

Si le résultat de l'addition de deux nombres positifs est négatif, alors débordement

Si le résultat de l'addition de deux nombres négatifs contient un chiffre supplémentaire, alors débordement

Arithmétique binaire

La même représentation marche en base 2 (complément à 2)

Les nombres positifs ont un 0 à la première position

Les nombres négatifs ont un 1 à la première position

Pour calculer la négation d'un nombre, il faut remplacer chaque chiffre par son inverse, puis finalement additionner 1

Le débordement se détecte de la même facon

Représentation des nombres rationnels

Pas souvent en matériel, mais en logiciel, car nécessite la représentation des entiers avec une précision arbitraire

Représentation sous la forme de deux entiers, dont le premier éventuellement négatif

Les deux entiers n’ont pas de facteurs communs (représentation canonique)