

Logique combinatoire

Sciences de l'ingénieur



```
#import "AutoPilotHelicopter.h"
#import "Background.h"
#import "Helicopter.h"
#import "Game.h"

enum {
    AUTOPILOTSTARTTIME = 2000,      /* ms */
    AUTOPILOTTAKEOVERTIME = 4000,  /* ms */
    AUTOPILOTHELICOPTERSPEED = (MAXVELX * 3) / 4
};

@implementation AutoPilotHelicopter
- (id)initInGame:(Game *)g {
    [super initInGame:g];
    autopilotTakeoverTime = [game updateTime] + AUTOPILOTSTARTTIME;
    return self;
}

- (void)setCommand:(NSInteger)cmd {
    autopilot = NO;
    autopilotTakeoverTime = [game updateTime] + AUTOPILOTTAKEOVERTIME;
    [super setCommand:cmd];
}
```



Logique combinatoire

Sciences de l'ingénieur



Représentation binaire

Un ordinateur est une machine qui manipule des chiffres binaires.

Une variable binaire ne peut prendre que deux valeurs : 0 ou 1.

Ces valeurs peuvent représenter : un interrupteur ouvert ou fermé, un transistor passant ou bloqué, la présence ou l'absence d'une tension...

Logique combinatoire

Sciences de l'ingénieur



Représentation binaire

Une variable E peut être complémentée, elle est alors notée \bar{E} ou $\neg E$ (E barre).

Compléter la table de vérité ci-contre :

E	\bar{E}
0	
1	

Logique combinatoire

Sciences de l'ingénieur



Fonction OUI (YES)

La sortie S est égale à la valeur de l'entrée.

Représentation graphique :



Équation logique : $S = e$

Table de vérité :

e	S
0	
1	

Logique combinatoire

Sciences de l'ingénieur



Fonction NON (NOT)

La sortie S est égale à la valeur inverse de l'entrée.

Représentation graphique :



Équation logique : $S = \bar{e}$

Table de vérité :

e	S
0	1
1	0

Logique combinatoire

Sciences de l'ingénieur

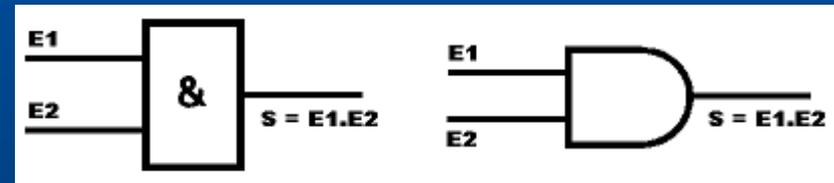


Fonction ET (AND)

La sortie S est vraie si toutes les entrées e_i sont vraies.

si $e_1 = 1$ ET $e_2 = 1$ alors $S = 1$

Représentation graphique :



Équation logique : $S = e_1.e_2$

Table de vérité :

e1	e2	S
0	0	0
0	1	0
1	0	0
1	1	1

Logique combinatoire

Sciences de l'ingénieur

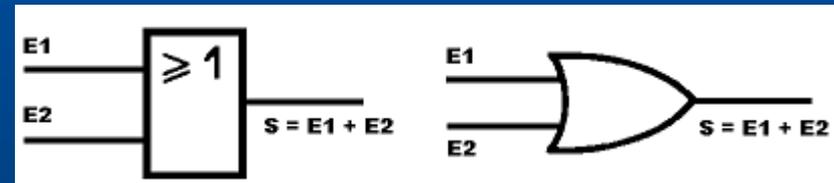


Fonction OU (OR)

La sortie S est vraie si au moins une des entrées e_i est vraie.

si $e_1 = 1$ OU $e_2 = 1$ alors $S = 1$

Représentation graphique :



Équation logique : $S = e_1 + e_2$

Table de vérité :

e_1	e_2	S
0	0	0
0	1	1
1	0	1
1	1	1

Logique combinatoire

Sciences de l'ingénieur

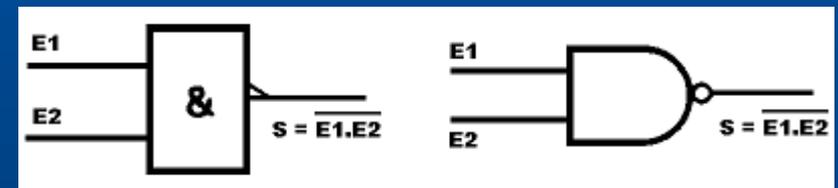


Fonction NON-ET (NAND)

La sortie S est vraie si au moins une des entrées e_i est fausse.

si $e1 = 0$ OU $e2 = 0$ alors $S = 1$

Représentation graphique :



Équation logique : $S = \neg(e1.e2)$

Table de vérité :

e1	e2	S
0	0	1
0	1	1
1	0	1
1	1	0

Logique combinatoire

Sciences de l'ingénieur

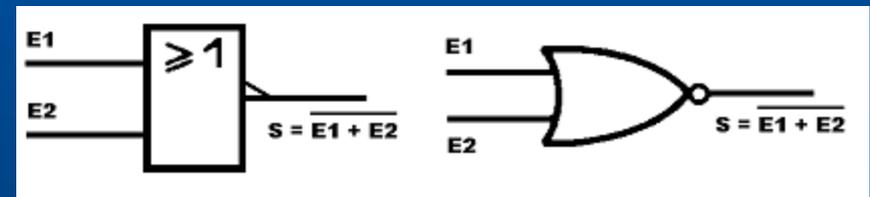


Fonction NON-OU (NOR)

La sortie S est vraie si toutes les entrées e_i sont fausses.

si $e_1 = 0$ ET $e_2 = 0$ alors $S = 1$

Représentation graphique :



Équation logique : $S = \neg(e_1 + e_2)$

Table de vérité :

e1	e2	S
0	0	1
0	1	0
1	0	0
1	1	0

Logique combinatoire

Sciences de l'ingénieur

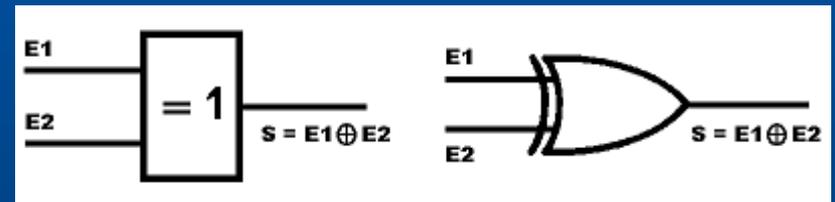


Fonction OU Exclusif (XOR)

La sortie S est vraie si toutes les entrées e_i ont des valeurs distinctes.

si $e1 \neq e2$ alors $S = 1$

Représentation graphique :



Équation logique : $S = e1 \oplus e2 = e1 \cdot \bar{e}2 + \bar{e}1 \cdot e2$

Table de vérité :

e1	e2	S
0	0	0
0	1	1
1	0	1
1	1	0

Logique combinatoire

Sciences de l'ingénieur

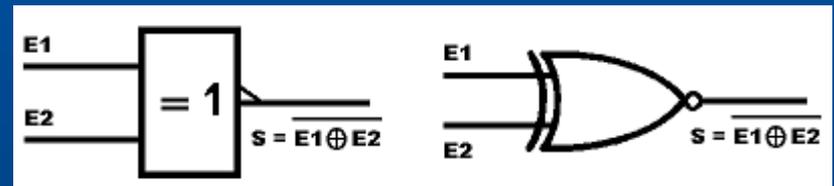


Fonction NON-OU Exclusif (XNOR)

La sortie S est vraie si toutes les entrées e_i sont identiques.

si $e1 = e2$ alors $S = 1$

Représentation graphique :



Équation logique : $S = \neg(e1 \oplus e2) = \bar{e}1.\bar{e}2 + e1.e2$

Table de vérité :

e1	e2	S
0	0	1
0	1	0
1	0	0
1	1	1

Logique combinatoire

Sciences de l'ingénieur



Propriétés des opérateurs

- involution (double complémentation)

$$\neg \bar{a} = a$$

- commutativité

$$a.b = b.a$$

$$a+b = b+a$$

$$a \oplus b = b \oplus a$$

- associativité

$$a.(b.c) = (a.b).c$$

$$a+(b+c) = (a+b)+c$$

$$a \oplus (b \oplus c) = (a \oplus b) \oplus c$$

- distributivité

$$a.(b+c) = a.b + a.c$$

$$a + b.c = (a+b).(a+c)$$

Logique combinatoire

Sciences de l'ingénieur



Exercices

$$a.1 =$$

$$a.0 =$$

$$a.a =$$

$$a.\bar{a} =$$

$$a+1 =$$

$$a+0 =$$

$$a+a =$$

$$a+\bar{a} =$$

$$a\oplus 1 =$$

$$a\oplus 0 =$$

$$a\oplus a =$$

$$a\oplus \bar{a} =$$

Logique combinatoire

Sciences de l'ingénieur



Élément neutre : $a.1 = a$ $a+0 = a$ $a\oplus 0 = a$

Élément absorbant : $a.0 = 0$ $a+1 = 1$

Complément : $a.\bar{a} = 0$ $a+\bar{a} = 1$ $a\oplus\bar{a} = 1$

Idempotence : $a.a = a$ $a+a = a$

$a\oplus 1 = \bar{a}$ $a\oplus a = 0$

Logique combinatoire

Sciences de l'ingénieur



Théorèmes de DE MORGAN*

$$\neg(ab) = \bar{a} + \bar{b}$$

$$\neg(a+b) = \bar{a}.\bar{b}$$

Exercice : simplifier les équations logiques suivantes

1. $a.(\bar{a}+b)$

2. $\bar{a}.b.c + a.b.c$

Exercice : démontrer $a + \bar{a}.b = a + b$

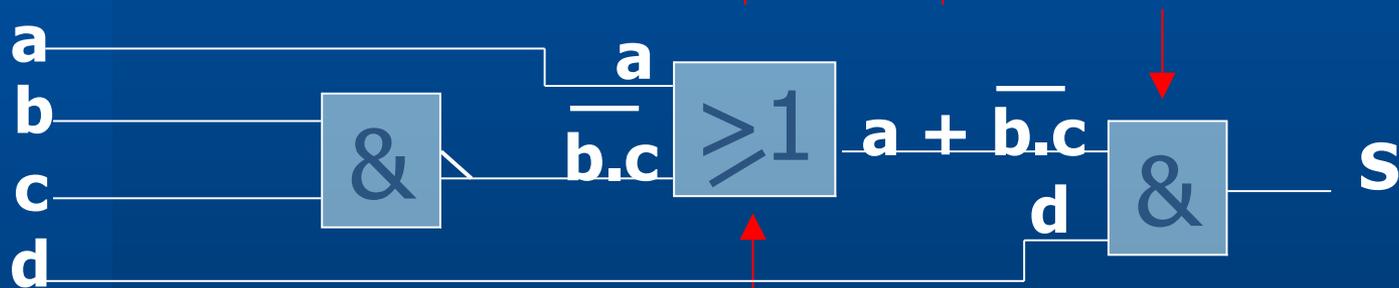
Logique combinatoire

Sciences de l'ingénieur



Création d'un logigramme

$$S = (a + \overline{b.c}).d$$



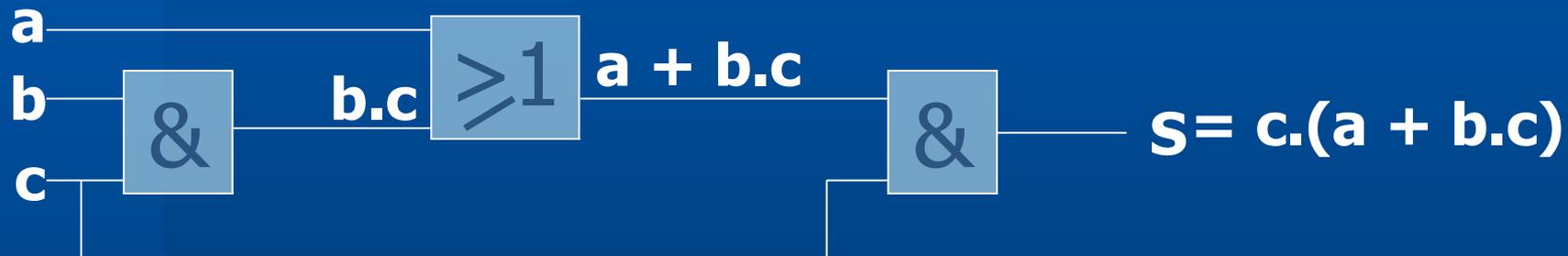
Règle de construction : Toujours partir de la sortie, rechercher l'opérateur logique qui sépare l'équation

Logique combinatoire

Sciences de l'ingénieur



Recherche d'équation



Simplification :

$$S = a.c + b.c.c$$
$$S = a.c + b.c$$
$$S = c. (a + b)$$

Logique combinatoire

Sciences de l'ingénieur



Exercice 1

- Donner l'équation logique de F à partir de la table de vérité ci-contre :
- Simplifier l'équation
- Produire le logigramme

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

Logique combinatoire

Sciences de l'ingénieur



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

$\bar{a}.\bar{b}.\bar{c}$

$\bar{a}.\bar{b}.c$

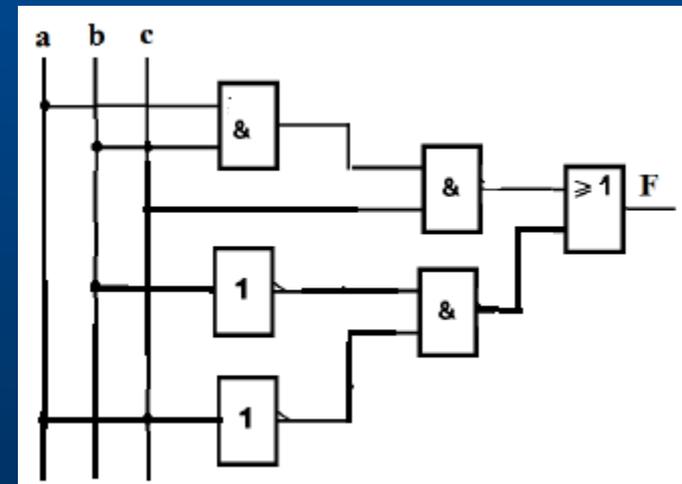
$a.b.c$

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

$$F = \bar{a}.\bar{b}.(c + \bar{c}) + a.b.c$$

$$F = \bar{a}.\bar{b}.1 + a.b.c$$

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



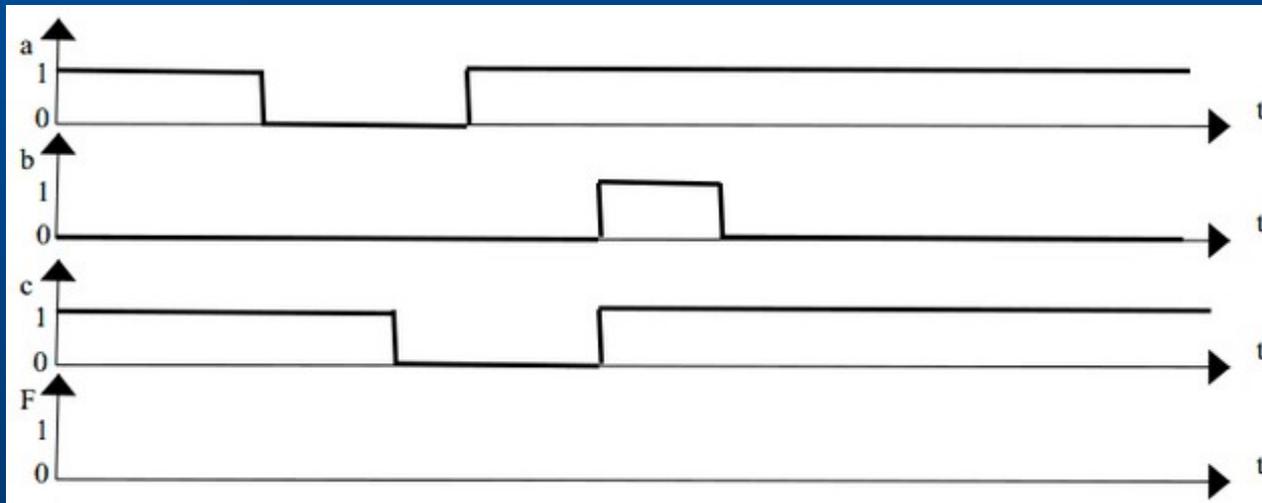
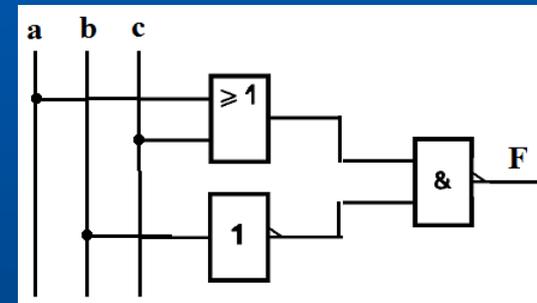
Logique combinatoire

Sciences de l'ingénieur



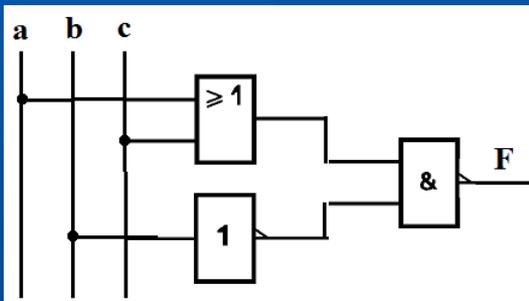
Exercice 2

- Donner l'équation de F
- Produire la table de vérité
- Compléter le chronogramme de F

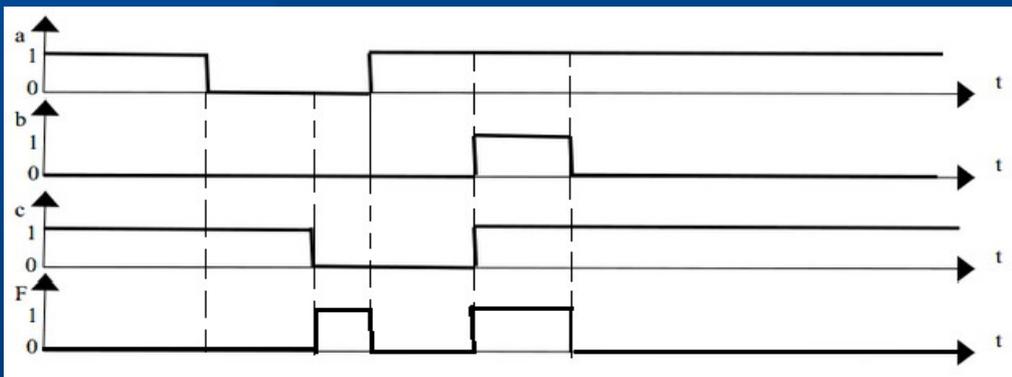


Logique combinatoire

Sciences de l'ingénieur



$$F = (a + c). \bar{b} = (a + c) + b$$



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