

1ère STI2D

Logique combinatoire



```
#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];
}
}
```



1ère STI2D

Logique combinatoire



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...

1ère STI2D

Logique combinatoire



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	

1ère STI2D

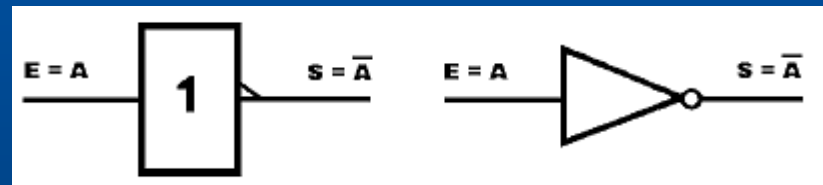
Logique combinatoire



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

1ère STI2D

Logique combinatoire

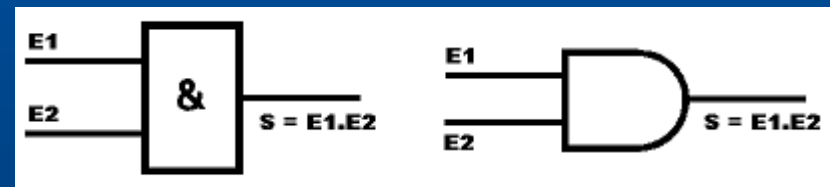


Fonction ET (AND)

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

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

Représentation graphique :



Équation logique : $S = e1.e2$

Table de vérité :

e1	e2	S
0	0	
0	1	
1	0	
1	1	

1ère STI2D

Logique combinatoire

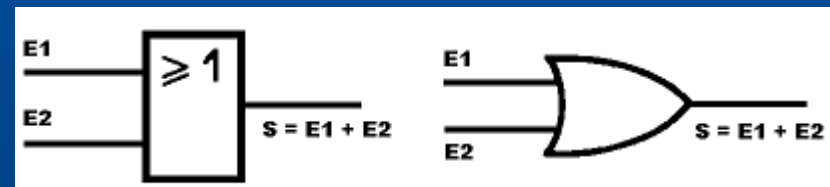


Fonction OU (OR)

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 :

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

1ère STI2D

Logique combinatoire



Fonction NON-ET (NAND)

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

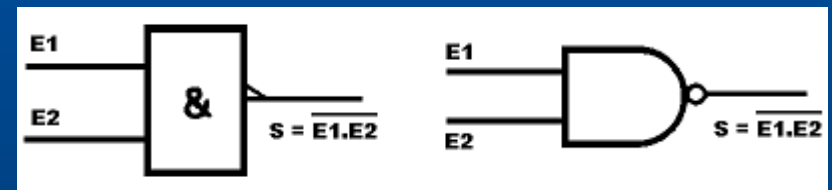
si $e1 = 0$ OU $e2 = 0$ alors $S = 1$

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



1ère STI2D

Logique combinatoire

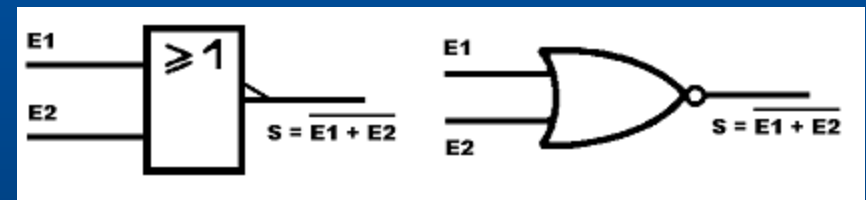


Fonction NON-OU (NOR)

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

1ère STI2D

Logique combinatoire



Fonctions logiques en langage C/C++

Les 3 fonctions principales

fonction	opérateur	test
NON	!	
ET	&	&&
OU		

Exemples

fonction	opérateur	test
NON	!a	
ET	a & b	a && b
OU	a b	a b
NON ET	!(a & b)	
NON OU	!(a b)	

1ère STI2D

Logique combinatoire



Exercice 1

A partir d'un certain seuil de luminosité, le programme d'un automate déclenche un moteur qui s'arrête lorsqu'il arrive en buté.

- a : seuil luminosité atteint
- b : arrivée en buté

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

$$F = a \cdot \bar{b}$$

$$F = \overline{\bar{a} + b}$$

1ère STI2D

Logique combinatoire



Exercice 2

En cas de vent trop violent, le moteur ne doit pas être déclenché même si la lumière est trop importante.

- a : seuil luminosité atteint
- b : arrivée en buté
- c : vent trop violent

$$F = a \cdot \bar{b} \cdot \bar{c}$$

$$F = \overline{a + b + c}$$

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

1ère STI2D

Logique combinatoire



Exercice 3

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

