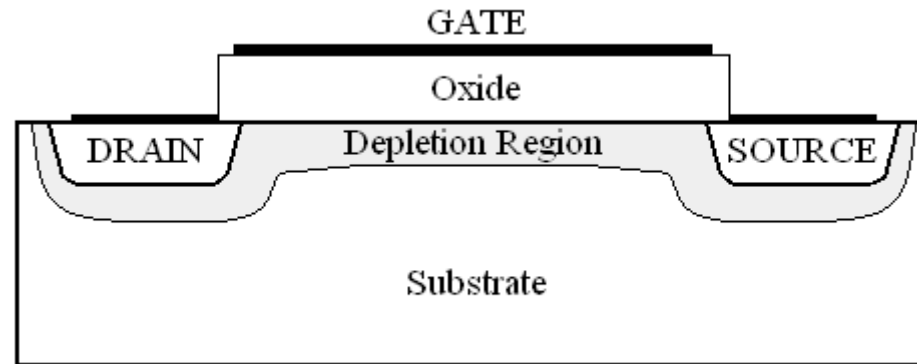


# Chapitre 4 : Circuits asynchrones

# Calculs

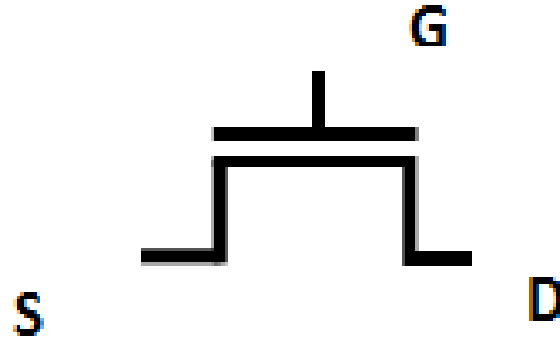
- Dans ce chapitre, nous allons étudier comment un ordinateur est capable d'effectuer des calculs.
- Etape 1 : fonctionnement du transistor
- Etape 2 : conception de portes logiques
- Etape 3 : assemblage de portes logiques
- Etape 4 : estimation des performances

# La transistor CMOS



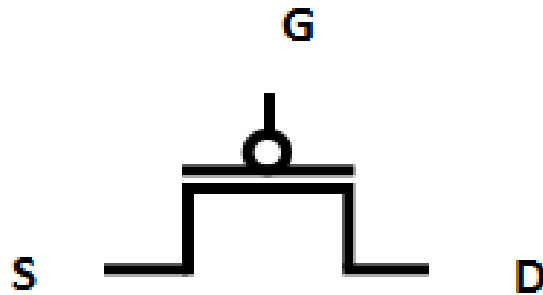
- 3 points importants : la source, le drain et la grille
- En fonction de la tension de la grille, il peut y avoir conduction électrique entre la source et le drain ou au contraire les 2 points peuvent être électriquement isolés.

# Transistor N



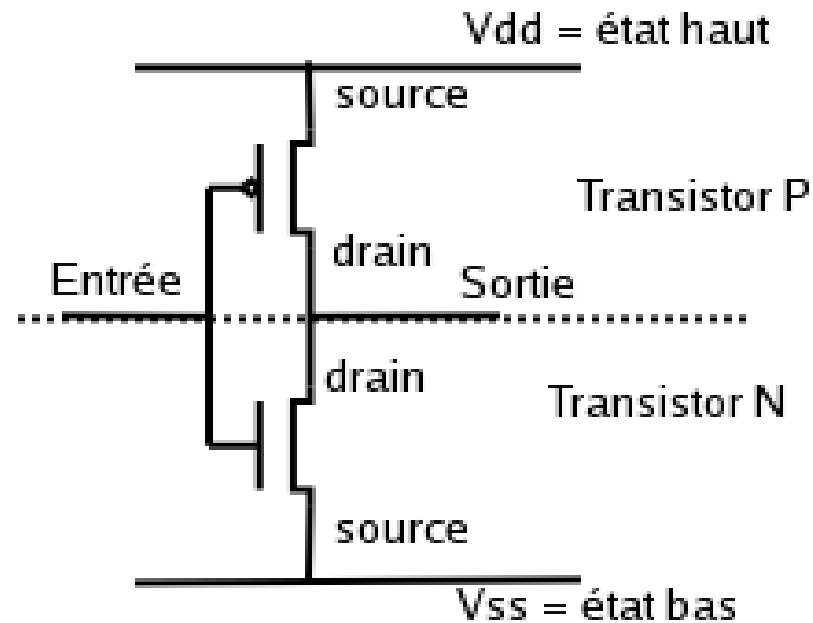
- Si  $G=0$  V ,S et D sont électriquement isolés
- Si  $G=1$  V, il y a conduction électrique entre S et D

# Transistor P



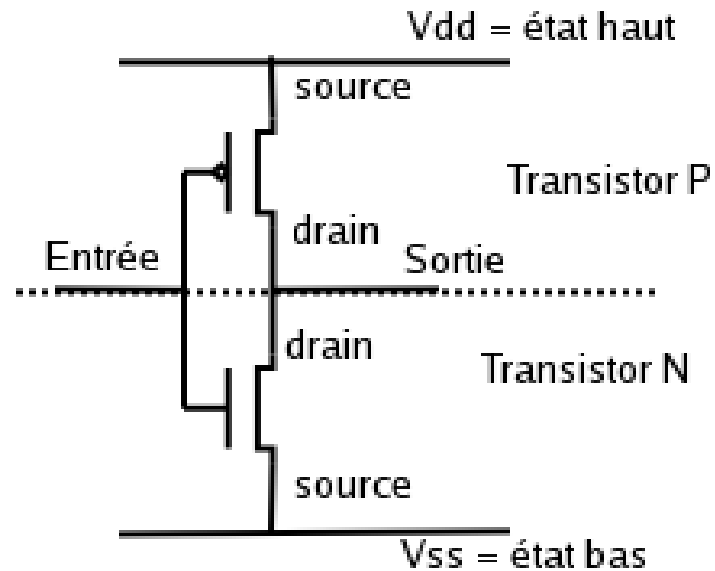
- Si  $G=0$  V , il y a conduction électrique entre S et D
- Si  $G=1$  V, S et D sont électriquement isolés

# Porte logique



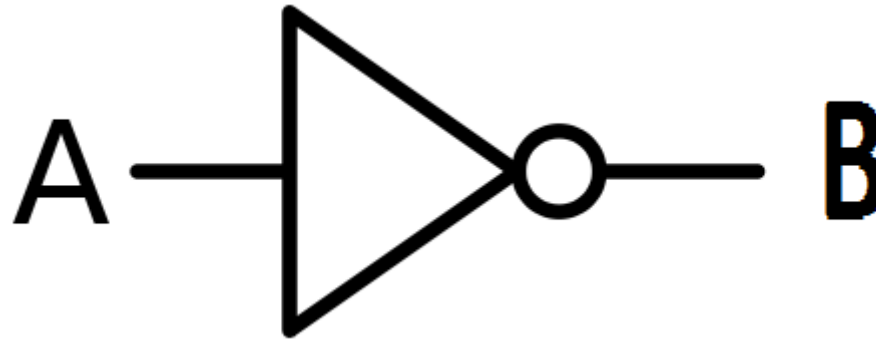
- Si  $E=0$  alors  $S=1$
- Si  $E=1$  alors  $S=0$

# Assemblage de transistors



- Si  $E=0$  alors  $S=1$
- Si  $E=1$  alors  $S=0$

# Porte logique NON

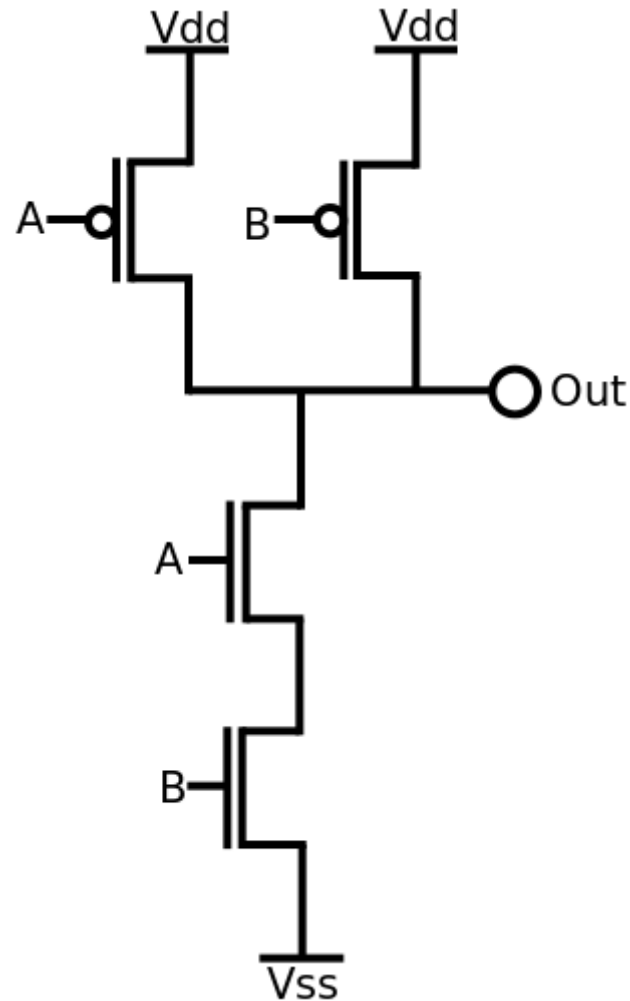


- Table de vérité

| A | B |
|---|---|
| 0 | 1 |
| 1 | 0 |

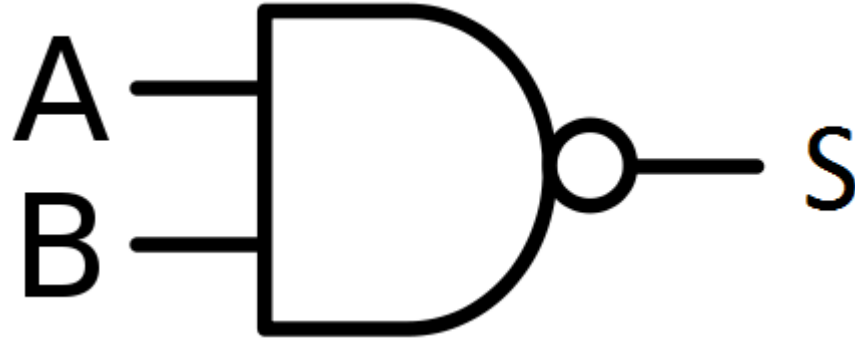


# Trouvez la table de vérité



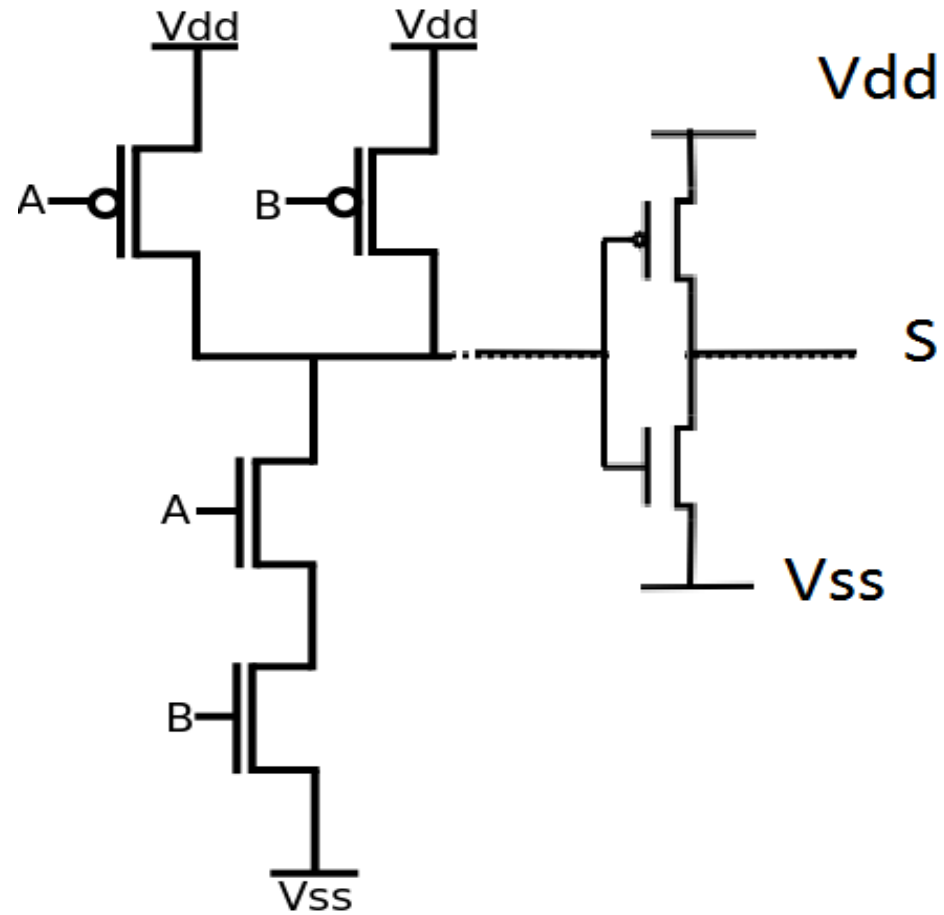
# Solution

- | A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



- C'est une porte NON -ET

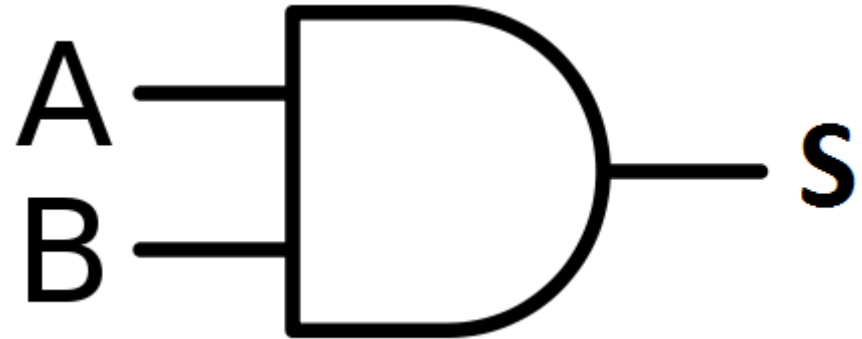
# Le ET logique



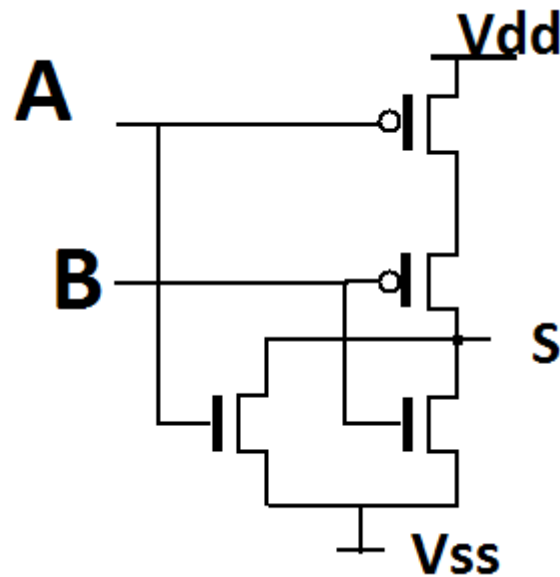
# Le ET logique

- Table de vérité

| AB  | S |
|-----|---|
| 0 0 | 0 |
| 0 1 | 0 |
| 1 0 | 0 |
| 1 1 | 1 |



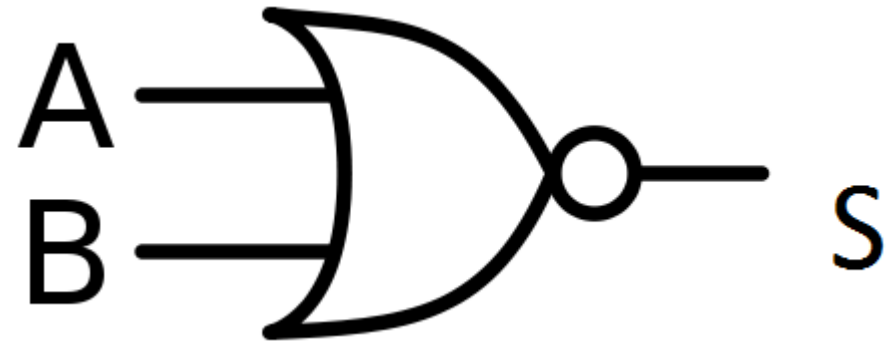
# Trouvez la table de vérité



# Solution

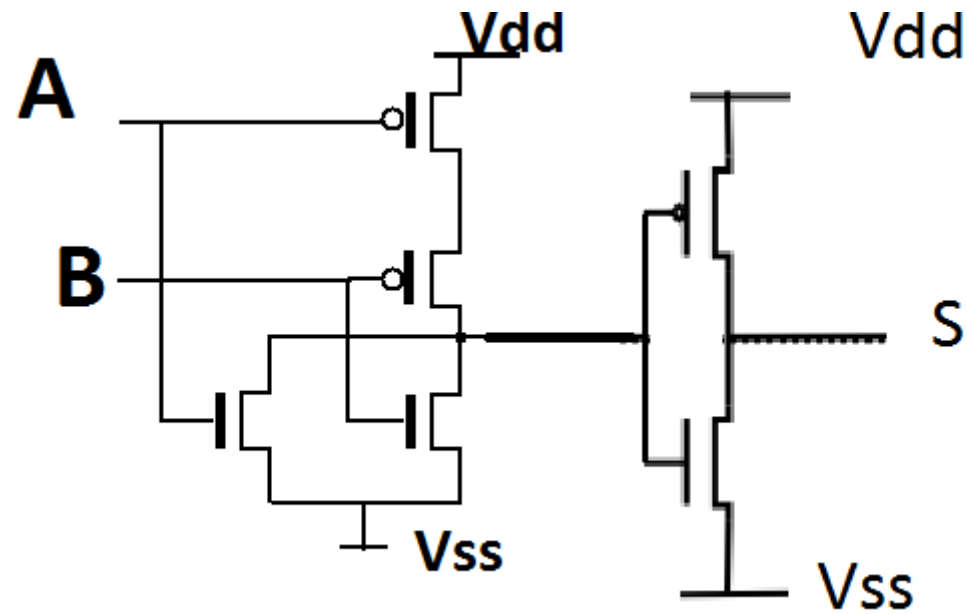
- Table de vérité

| A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



- NON OU logique

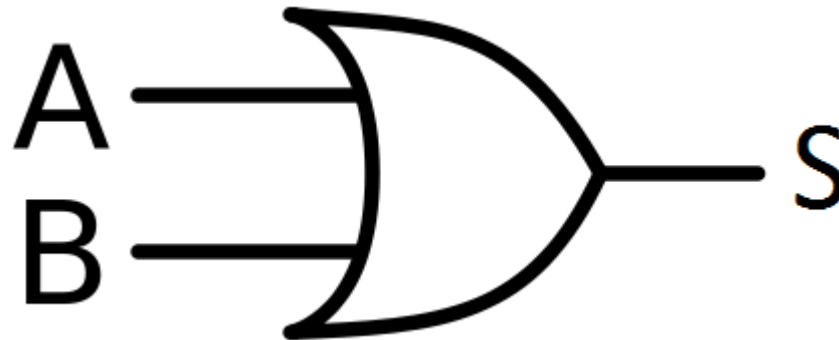
# OU logique



# OU logique

- Table de vérité

| AB  | S |
|-----|---|
| 0 0 | 0 |
| 0 1 | 0 |
| 1 0 | 0 |
| 1 1 | 1 |





# Assemblage de portes logiques

- En assemblant des portes logiques, on peut construire des circuits plus complexes
- La méthode de karnaugh permet de construire des circuits à partir de n'importe quelle table de vérité à base de portes ET, OU ET NON

# Karnaugh

- Proposez un circuit le plus simple possible ayant comme table de vérité

| ABC | S |
|-----|---|
| 000 | 1 |
| 001 | 1 |
| 010 | 0 |
| 011 | 0 |
| 100 | 0 |
| 101 | 1 |
| 110 | 1 |
| 111 | 1 |

# Etape 1

AB

|   |   | 00 | 01 | 11 | 10 |
|---|---|----|----|----|----|
| C | 0 | 1  | 0  | 1  | 0  |
|   | 1 | 1  | 0  | 1  | 1  |

# Etape 2

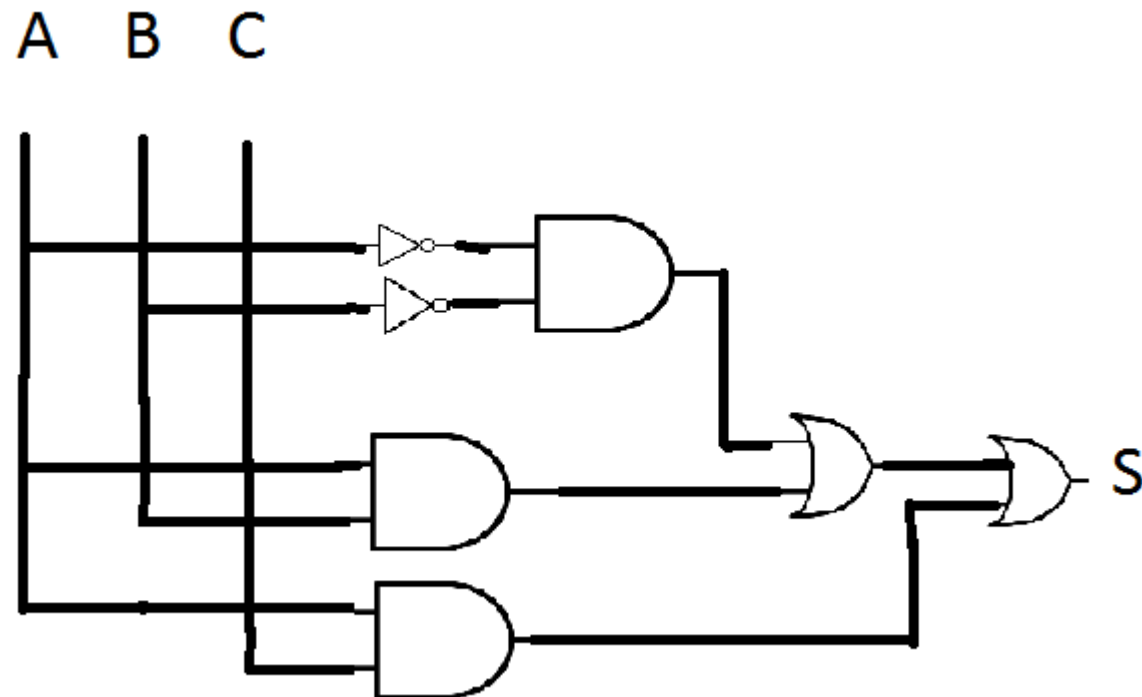
AB

|   |   | 00 | 01 | 11 | 10 |
|---|---|----|----|----|----|
| C | 0 | 1  | 0  | 1  | 0  |
|   | 1 | 1  | 0  | 1  | 1  |

- Etape 3

$$S = \overline{A} \overline{B} + A B + A C$$

- Etape 4



# Proposez des circuits synthétisant ces tables de vérité

ABC S

000 1

001 1

010 0

011 1

100 0

101 1

110 0

111 1

ABC S

000 1

001 1

010 0

011 0

100 0

101 1

110 1

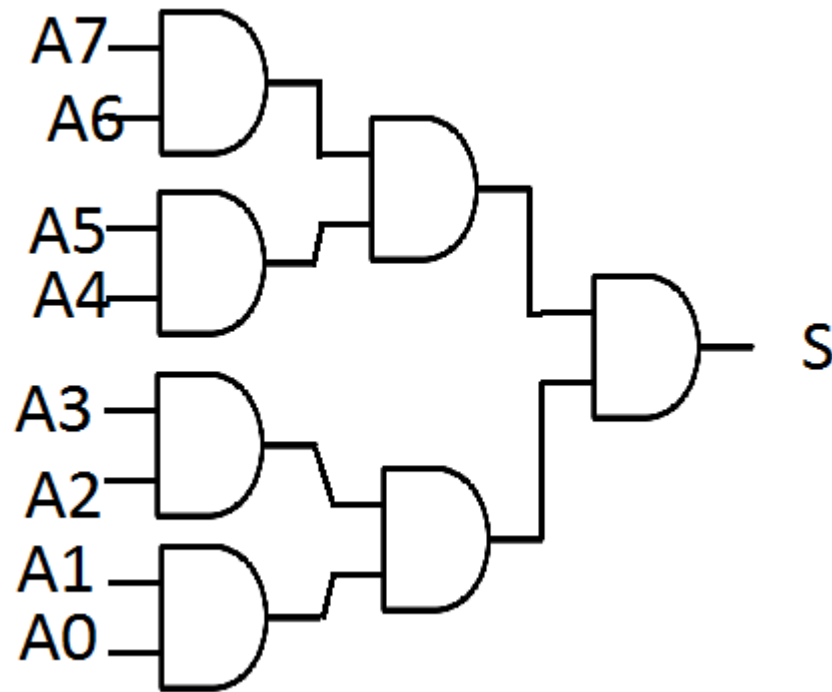
111 1

# Assemblage de portes logiques

- Proposer un assemblage de portes logiques permettant de créer un ET logique à 8 entrées.
- Le circuit aura 8 entrées A7 A6 A5 A4 A3 A2 A1 A0
- On utilisera un ET logique à 2 entrées comme porte logique de base.

# Solution

- ET logique à 8 entrées





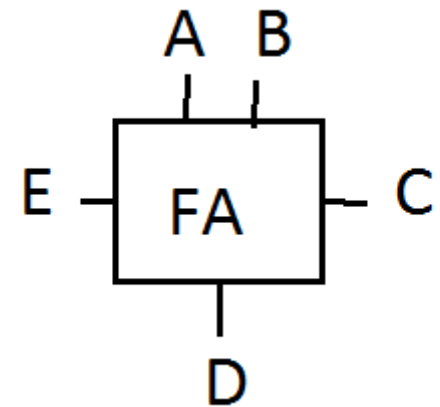
# Estimation du temps de calcul par la méthode du chemin critique

- On cherche le plus long chemin qui mène d'une entrée à une sortie.
- Dans notre dernier circuit, le chemin critique traverse 3 ET logique  
==> temps=3 temps(ET)

# La porte logique FA (Full Adder)

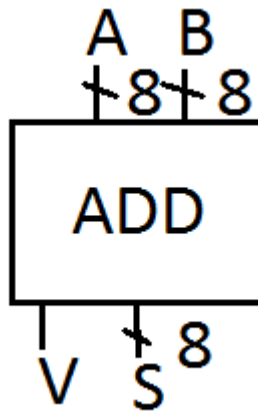
- 3 entrées A B et C et é sortie D et E
- Table de vérité

| A | B | C | E | D |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |



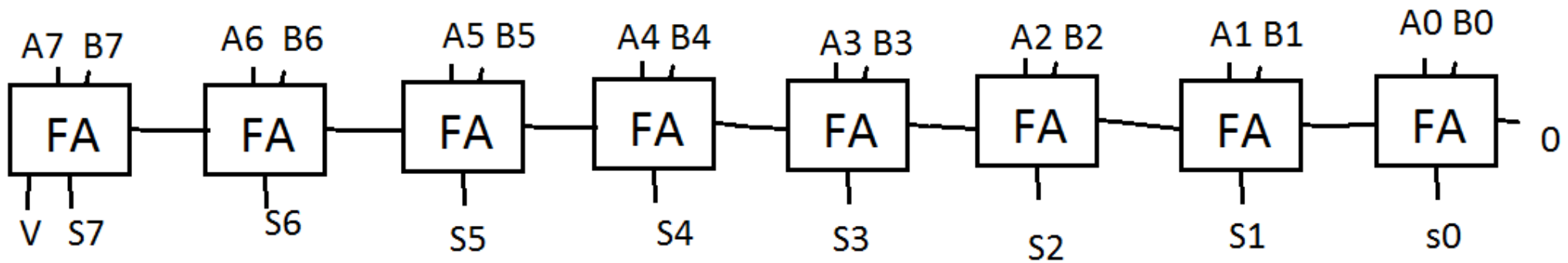
# Additionneur

- On veut réaliser un additionneur à 2 entrées A et B sur 8 bits et à une sortie S sur 8 bits et un bit V.
- Si  $V=1$ , il y a dépassement de capacité.  
Si  $V=0$ ,  $S=A+B$



# Solution

- Additionneur 8 bits



# Multiplexeur

- 3 entrées A B et C sur 1 bit et une sortie S sur 1 bit

Si  $C=0$ ,  $S=A$  sinon  $S=B$

- Table de vérité

ABC S

0 0 0 0

0 0 1 0

0 1 0 0

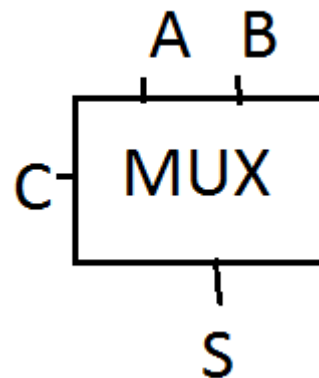
0 1 1 1

1 0 0 1

1 0 1 0

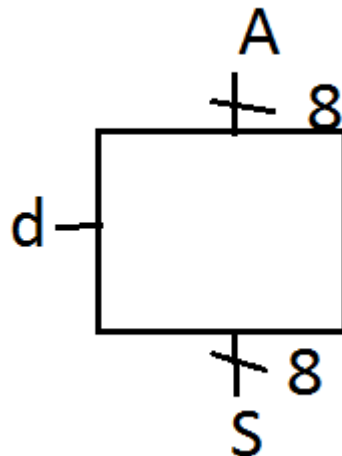
1 1 0 1

1 1 1 1



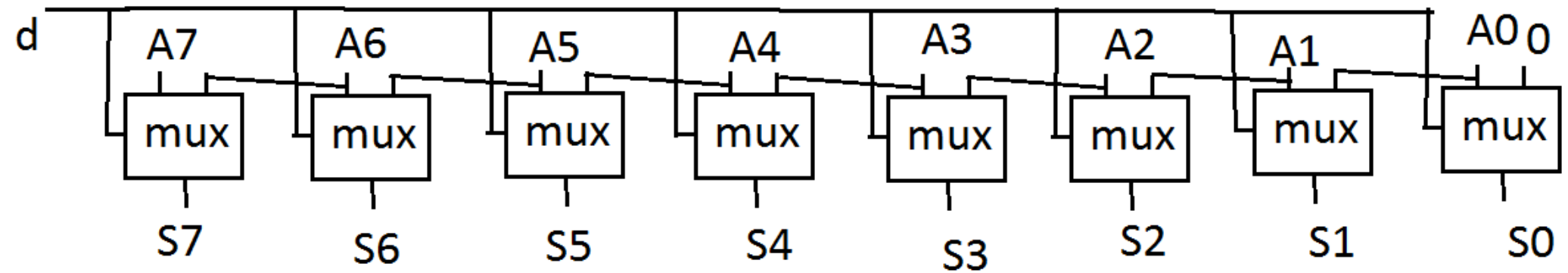
# Décaleur 8 bits

- On veut un circuit avec une entrée A sur 8 bits A7 A6 A5 A4 A3 A2 A1 A0 et une autre d sur 1 bit et une sortie S sur 8 bits S7 S6 S5 S4 S3 S2 S1 S0.
- On utilisera un multiplexeur comme porte logique de base.



# Solution

- Décaleur 8 bits



- Temps = 1 temps(MUX)

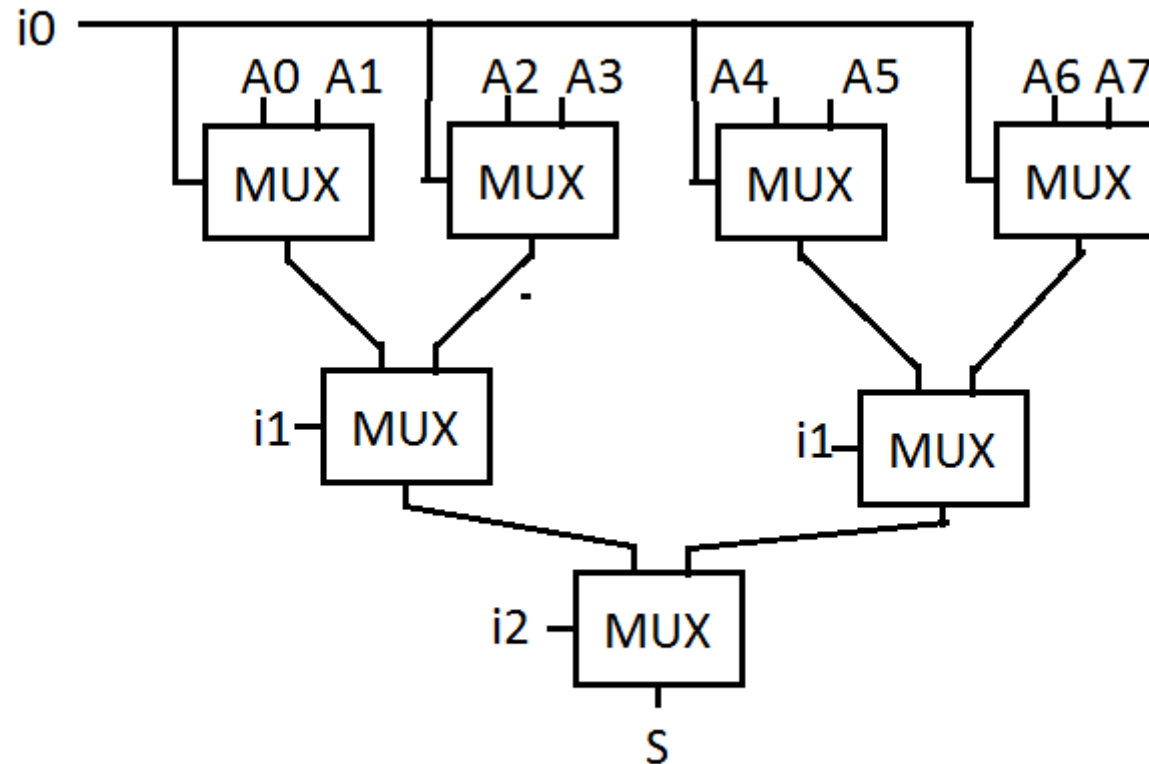
# Multiplexeur à 8 entrées

- On veut créer un circuit à 8 entrées  $A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7$  et une entrée  $i$  sur 3 bits  $i_2 i_1 i_0$  et une sortie  $S$  sur 1 bit
- $S$  est égal à une des 8 entrées  $A$   
 $S = A_i$
- On utilisera un multiplexeur 1 bit



# Solution

- Mutiplexeur 8 entrées



- Temps=8 temps(MUX)

# Conclusion

- Vous devez être capable de calculer la table de vérité d'un assemblage de transistors quelconques
- Vous devez connaître l'assemblage de transistors des portes logiques NON NON-ET NON-OU OU et ET.
- Vous devez connaître la méthode de Karnaugh pour synthétiser n'importe quelle porte logique à partir des portes ET à 2 entrées OU à 2 entrées et NON