

**Conception des systèmes numériques (ELE140)**

**Devoir 2**

**Enseignant :** François Blanchard

**Groupe :** 01

**Remis par :**

Ourania Voyatzis (VOYO78260401)  
Jhermain Louis-Jean (LOUJ67360401)

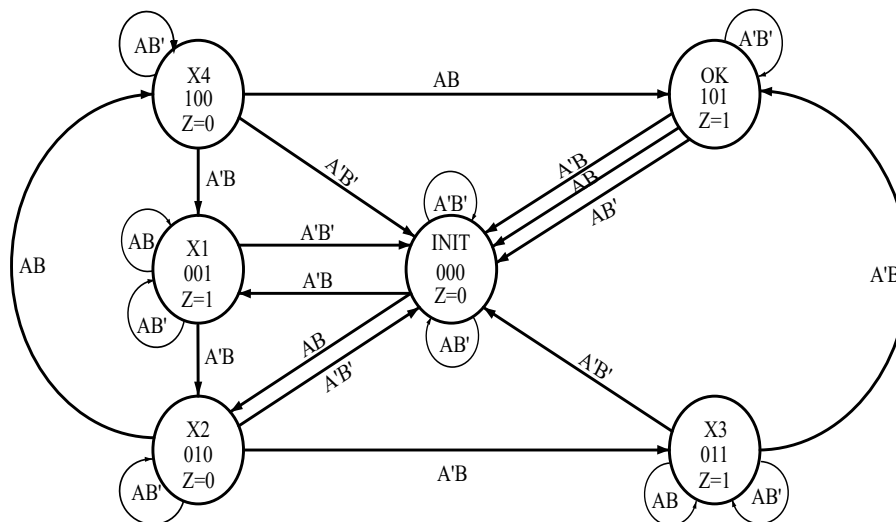
**Date :** 1<sup>er</sup> décembre 2025

## Q.1 a) Assignment des bascules de type « simplest » avec une gestion des coûts minimums

Assignment des Bascules : "simplest"

INIT	000
X1	001
X2	010
X3	011
X4	100
OK	101

### Diagramme d'états



### Table de transitions

Q2, Q1, Q0	AB				Z
	00	01	11	10	
000	000	001	010	000	0
001	000	010	001	001	1
010	000	011	100	010	0
011	000	101	011	011	1
100	000	001	101	100	0
101	101	000	000	000	1
	Q2*Q1*Q0*				

## Tables de Karnaugh des Bascules (Combinatoire)

### Q2\*

	$\overline{Q_2Q_1Q_0}$	$\overline{Q_2Q_1}Q_0$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1\overline{Q_0}$	$Q_2Q_1\overline{Q_0}$	$Q_2Q_1Q_0$	$Q_2\overline{Q_1}Q_0$	$Q_2\overline{Q_1}\overline{Q_0}$
$\overline{AB}$	0	0	0	0	X	X	1	0
$\overline{A}B$	0	0	1	0	X	X	0	0
$A\overline{B}$	0	0	0	1	X	X	0	1
$AB$	0	0	0	0	X	X	0	1

$$Q2^*(A, B, Q2, Q1, Q0) = A'B'Q2Q0 + A'BQ1Q0 + AQ2Q0' + ABQ1Q0'$$

### Q1\*

	$\overline{Q_2Q_1Q_0}$	$\overline{Q_2Q_1}Q_0$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1\overline{Q_0}$	$Q_2Q_1\overline{Q_0}$	$Q_2Q_1Q_0$	$Q_2\overline{Q_1}Q_0$	$Q_2\overline{Q_1}\overline{Q_0}$
$\overline{AB}$	0	0	0	0	X	X	0	0
$\overline{A}B$	0	1	0	1	X	X	0	0
$A\overline{B}$	1	0	1	0	X	X	0	0
$AB$	0	0	1	1	X	X	0	0

$$Q1^*(A, B, Q2, Q1, Q0) = A'BQ2'Q1'Q0 + A'BQ1Q0' + AB'Q1 + ABQ2'Q1'Q0' + AQ1Q0$$

### Q0\*

	$\overline{Q_2Q_1Q_0}$	$\overline{Q_2Q_1}Q_0$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1\overline{Q_0}$	$Q_2Q_1\overline{Q_0}$	$Q_2Q_1Q_0$	$Q_2\overline{Q_1}Q_0$	$Q_2\overline{Q_1}\overline{Q_0}$
$\overline{AB}$	0	0	0	0	X	X	1	0
$\overline{A}B$	1	0	1	1	X	X	0	1
$A\overline{B}$	0	1	1	0	X	X	0	1
$AB$	0	1	1	0	X	X	0	0

$$Q0^*(A, B, Q2, Q1, Q0) = A'B'Q2Q0 + A'BQ0' + AQ2'Q0 + BQ2Q0' + A'BQ1$$

## Q1 b) Tableau 2 (décomposé) avec une gestion des risques minimums

Table de transitions

	AB				Z
Q2 Q1 Q0	00	01	11	10	
111	111	110	010	111	0
011	111	010	011	011	1
010	111	001	000	010	0
001	111	100	001	001	1
000	111	011	100	000	0
100	100	111	111	111	1
110	111	111	111	111	0
101	111	111	111	111	0
	Q2*Q1*Q0*				

### Tables de Karnaugh des Bascules (Combinatoire) b)

Dans les états indéfinis (001,101), nous revenons à l'état initial (tout à 1) afin de minimiser les risques.

#### Q2\*

	$\overline{Q_2}Q_1\overline{Q_0}$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1\overline{Q_0}$	$\overline{Q_2}Q_1\overline{Q_0}$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1Q_0$
$\overline{AB}$	1	1	1	1	1	1	1	1
$\overline{AB}$	0	1	0	0	1	1	1	1
$AB$	1	0	0	0	1	0	1	1
$AB$	0	0	0	0	1	1	1	1

$$Q2^*(A, B, Q2, Q1, Q0) = A'B' + A'Q1'Q0 + A'Q2 + B'Q2 + ABQ1'Q0' + Q2Q1' + Q2Q0'$$

#### Q1\*

	$\overline{Q_2}Q_1\overline{Q_0}$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1\overline{Q_0}$	$\overline{Q_2}Q_1\overline{Q_0}$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1Q_0$	$\overline{Q_2}Q_1Q_0$
$\overline{AB}$	1	1	1	1	1	1	1	0
$\overline{AB}$	1	0	1	0	1	1	1	1
$AB$	0	0	1	0	1	1	1	1
$AB$	0	0	1	1	1	1	1	1

$$Q1^*(A, B, Q2, Q1, Q0) = Q1Q0 + B'Q1 + AQ2 + A'Q2'Q1'Q0' + A'B'Q0 + BQ2$$

**Q0\***

	$\overline{Q_2}\overline{Q_1}\overline{Q_0}$	$\overline{Q_2}\overline{Q_1}Q_0$	$\overline{Q_2}Q_1\overline{Q_0}$	$\overline{Q_2}Q_1Q_0$	$Q_2\overline{Q_1}\overline{Q_0}$	$Q_2\overline{Q_1}Q_0$	$Q_2Q_1\overline{Q_0}$	$Q_2Q_1Q_0$
$\overline{A}B$	1	1	1	1	1	1	1	0
$\overline{A}\overline{B}$	1	0	0	1	1	0	1	1
$AB$	0	1	1	0	1	0	1	1
$A\overline{B}$	0	1	1	0	1	1	1	1

$$Q0^*(A, B, Q2, Q1, Q0) = AQ2'Q0 + A'B'Q2' + Q2Q1'Q0 + B'Q2Q1 + A'BQ0' + AQ2Q0'$$

**Q1 c) Quel est le système (a ou b) qui comporte le moins de logique combinatoire ?**

Dans ce cas, le premier circuit A possède le moins de portes logiques.

**Q.2 a) a) Tracez le diagramme d'états en prenant soin de bien détailler votre démarche**

**Étape 1**

```

library ieee;
use ieee.std_logic_1164.all;
entity MEFDev2 is
port(
CLK, A, RESET, ENABLE : in std_logic;
Z : out std_logic);
end MEFDev2;
architecture MEFDev2_arch of MEFDev2 is
signal Q, QF: std_logic_vector(2 downto 0);
signal Q0, Q2, Q1 : std_logic;
begin
Q0 <= '1' when Q(2 downto 1) = "01" and A = '1' else
'1' when A = '0' and Q(2 downto 1) = "00" else
'1' when Q(2 downto 1) = "10" and A = '1' else
'1' when Q(2 downto 1) = "11" and A = '0' else
'0';
Z <= Q(1) and not(Q(2)) and Q(0);
process(CLK,RESET)
begin
if RESET = '1' then
tic <= "000";
elsif rising_edge (CLK) then
if ENABLE = '1' then
Q <= QF;
end if;
end if;
end process;
Q1 <= (not(A) and Q(0) and not(Q(2))) or (Q(1)
and not(Q(0))) or (Q(2) and not(Q(0)) and A);
QF <= Q2 & Q1 & Q0;
Q2 <= '0' when Q = "110" and A = '1' else
'0' when Q(2 downto 1) = "00" and A = '0' else
'0' when Q(1 downto 0) = "00" and A = '0' else
'0' when Q(2) = '0' and Q(0) = '1' else
'1';
end MEFDev2_arch;

```

J'ai décidé de renommer d'abord les variables pour rendre le code plus lisible.

## Étape 2 : Q2Q1Q0 et Z

À partir des équations ci-dessus, nous pouvons déduire les tableaux et équations suivants :

$$Q2^* = Q2Q0 + Q2'Q0'A + Q1Q0'A' + Q1'Q0'A$$

	$\overline{Q_0A}$	$\overline{Q_0}A$	$Q_0A$	$Q_0\overline{A}$
$\overline{Q_2Q_1}$	0	1	0	0
$Q_2\overline{Q_1}$	1	1	0	0
$Q_2Q_1\overline{Q_2Q_1}$	1	0	1	1
$Q_2Q_1Q_2Q_1$	0	1	1	1

$$Q1^* = Q2'Q0A' + Q1Q0' + Q2Q0'A$$

	$\overline{Q_0A}$	$\overline{Q_0}A$	$Q_0A$	$Q_0\overline{A}$
$\overline{Q_2Q_1}$	0	0	0	1
$Q_2\overline{Q_1}$	1	1	0	1
$Q_2Q_1\overline{Q_2Q_1}$	1	1	0	0
$Q_2Q_1Q_2Q_1$	0	1	0	0

$$Q0^* = Q2'Q1'A' + Q2'Q1A + Q2Q1A'$$

	$\overline{Q_0A}$	$\overline{Q_0}A$	$Q_0A$	$Q_0\overline{A}$
$\overline{Q_2Q_1}$	1	0	0	1
$Q_2\overline{Q_1}$	0	1	1	0
$Q_2Q_1\overline{Q_2Q_1}$	1	0	0	1
$Q_2Q_1Q_2Q_1$	0	0	0	0

$$Z = Q2'Q1Q0$$

	$\overline{Q_1Q_0}$	$\overline{Q_1}Q_0$	$Q_1Q_0$	$Q_1\overline{Q_0}$
$\overline{Q_2}$	0	0	0	0
$Q_2$	0	0	1	0

## Étape 3 : Table de transition

Q2*Q1*Q0*	A		
Q2Q1Q0	1	0	Z
000	001	100	0
001	011	000	0
010	110	111	0
011	010	001	1
100	000	010	0
101	100	100	0
110	111	110	0
111	101	100	0

## Étape 4 : Diagramme d'état

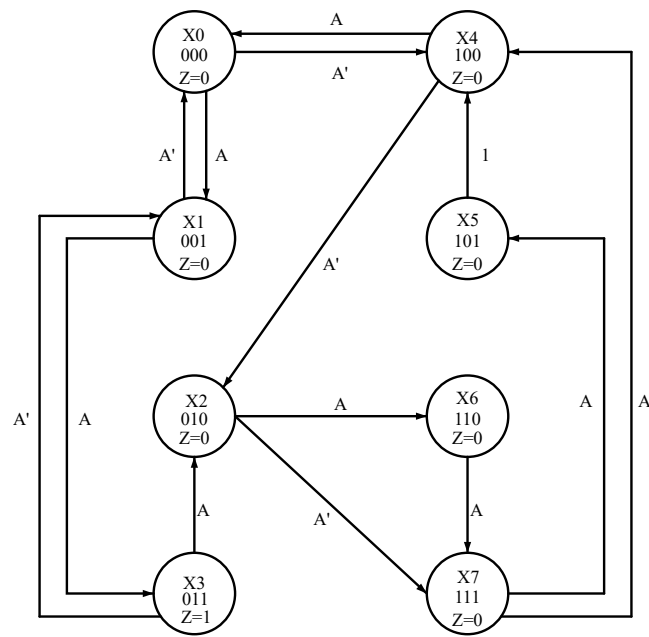


FIGURE 1

### Q2 b)

Il s'agit d'un circuit de Moore car les signaux d'entrée n'affectent pas directement les sorties de manière asynchrone.

### Q2 c) Commentez sur la qualité de ce code VHDL

Le code est très difficile à lire en raison des noms étranges ainsi que du fait que la logique combinatoire est divisée entre 2 blocs de code séparés, et ils ont utilisé inutilement des signaux binaires pour chaque bascule individuelle. Il y a peut-être encore plus de raisons pour lesquelles ce code ne suit pas les bonnes pratiques que j'ai peut-être manquées.