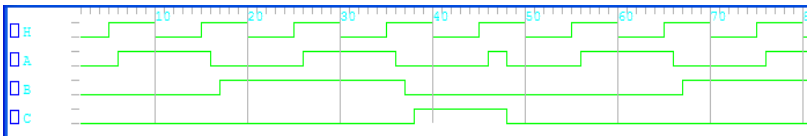
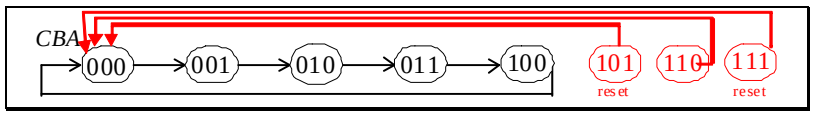
EXAMEN D'ELECTRONIQUE INTEGREE - CORRIGE

1. Compteur asynchrone avec Remise à Zéro (RAZ)

1.



2.



3. Auto-correcteur **OUI**

2. Analyse de système logique

A	B	C	D	FSvert	FSrouge	FSorange	PSvert	PSrouge
0	0	0	0	0	1	0	1	0
1	0	0	0	0	1	0	1	0
0	1	0	0	0	1	0	0	1
0	0	1	0	1	0	0	0	1
0	0	0	1	0	0	1	0	1

FSvert

AB \ CD	00	01	11	10
00	0	0	X	1
01	0	X	X	X
11	X	X	X	X
10	0	X	X	X

FSrouge

AB \ CD	00	01	11	10
00	1	0	X	0
01	1	X	X	X
11	X	X	X	X
10	1	X	X	X

FSorange

AB \ CD	00	01	11	10
00	0	1	X	0
01	0	X	X	X
11	X	X	X	X
10	0	X	X	X

PPvert

AB \ CD	00	01	11	10
00	0	0	X	1
01	0	X	X	X
11	X	X	X	X
10	0	X	X	X

PProuge

AB \ CD	00	01	11	10
00	1	1	X	0
01	1	X	X	X
11	X	X	X	X
10	1	X	X	X

1. FSvert = C FSrouge = $\overline{C}\,\overline{D}$ FSorange = D PPvert = C PProuge = $\overline{C}$

3. Synthèse de circuit logique

1.

Z : A \ BC	00	01	11	10
0	1	1	0	0
1	1	1	0	0

$Z = \overline{B}$

2. *Table des transitions (Synthèse) - (Horloge active)* **ABC = $Q_0Q_1Q_2$ = 000 → 001 → 011 → 111 → 110 → 100 → 000 → 001 …**

Transition $Q_{n-1} \rightarrow Q_n$	J	K
0 → 0	0	X
0 → 1	1	X
1 → 1	X	0
1 → 0	X	1

J_0 :

Q_0 \ Q_1Q_2	00	01	11	10
0	0	0	1	X
1	X	X	X	X

K_0 :

Q_0 \ Q_1Q_2	00	01	11	10
0	X	X	X	X
1	1	X	0	0

J_1 :

Q_0 \ Q_1Q_2	00	01	11	10
0	0	1	X	X
1	0	X	X	X

K_1 :

Q_0 \ Q_1Q_2	00	01	11	10
0	X	X	0	X
1	X	X	0	1

J_2 :

Q_0 \ Q_1Q_2	00	01	11	10
0	1	X	X	X
1	0	X	X	0

K_2 :

Q_0 \ Q_1Q_2	00	01	11	10
0	X	0	0	X
1	X	X	1	X

$J_0 = \overline{Q_1}$ $K_0 = \overline{Q_1}$

$J_1 = Q_2$ $K_1 = \overline{Q_2}$

$J_2 = \overline{Q_0}$ $K_2 = Q_0$

3. ***ABC*** **= $Q_0Q_1Q_2$ = 000 → 001 → 011 → 111 → 110 → 100 → 000 → 001 …**

A	*B*	*C*
0	0	0
0	0	1
0	1	1
1	1	1
1	1	0
1	0	0
0	0	0
…	…	…

On peut aisément observer que l'on voit apparaître, pour une ligne donnée n, au début de ligne suivante n+1, à droite, le complément de la valeur qui disparaît à gauche sur cette ligne n.

E = $\boxed{\overline{Q_0} = \overline{A}}$ $\qquad A_{n+1}B_{n+1}C_{n+1} = B_nC_n\overline{A}_n$

Le compteur est-il auto-correcteur ? *NON* Les états hors cycle 010 et 101 bouclent l'un sur l'autre : $\begin{cases} ABC = 010 \rightarrow 101 \\ ABC = 101 \rightarrow 010 \end{cases}$

4. Représentation des nombres en machine : somme et produit en flottant

1. X $\boxed{41\ 28\ 00\ 00_{(HEXA)}}$ Y $\boxed{41\ 50\ 00\ 00_{(HEXA)}}$

X = 10,5 = $2^k \cdot (1, \cdots)$ = 8 x 1,3125 = 2^3 x (1 + 0,**3125**) = $2^{130-127}$ x (1 + 0,**3125**)

s (1) = **0**
e (8) = 130 = **1000 0010**
m (23) = 0,3125 = $2^{-2} + 2^{-4}$ = **010 1000 0000 0000 0000 0000** = m_X
sem = **0100 0001 0010 1000 0000 0000 0000 0000 = 41 28 00 00$_{(H)}$**

Y = 13 = $2^k \cdot (1, \cdots)$ = 8 x 1,625 = 2^3 x (1 + 0,**625**) = $2^{130-127}$ x (1 + 0,**625**)

s (1) = **0**
e (8) = 130 = **1000 0010**
m (23) = 0,625 = $2^{-1} + 2^{-3}$ = **101 0000 0000 0000 0000 0000** = m_Y
sem = **0100 0001 0101 0000 0000 0000 0000 0000 = 41 50 00 00$_{(H)}$**

2. S = X + Y $\boxed{41\ BC\ 00\ 00_{(HEXA)}}$ Normalisation : *OUI*

X + Y = 10,5 + 13 =
1,010 1000 0000 0000 0000 0000 * $2^{130-127}$ (14)
\+ 1,101 0000 0000 0000 0000 0000 * $2^{130-127}$ (12)

10,111 1000 0000 0000 0000 0000 * $2^{130-127}$
= 1,011 1100 0000 0000 0000 0000 * $2^{131-127}$ Normalisation
soit pour le résultat : **X + Y**:
s (1) = **0**
e (8) = 131 = **1000 0011**
m (23) = **011 1100 0000 0000 0000 0000**
sem = **0100 0001 1011 1100 0000 0000 0000 0000 = 41 BC 00 00$_{(H)}$**
Vérification : A comparer au résultat **23,5 …**

3. M = XY $\boxed{43\ 08\ 80\ 00_{(HEXA)}}$ Normalisation : *OUI*

M = X x Y = 10,5 x 13 =
signe : **0** (xor entre les 2 signes des 2 opérandes)
exposant : **1000 0010 + 1000 0010 - (127)$_2$ = 1000 0010 + 1000 0010 – 0111 1111**
= 1000 0010 + 1000 0010 + (-127)$_{c2}$ = 1000 0010 + 1000 0010 + 1000 0001
= 1000 0101 (= 133$_{10}$) = 6 + 127 **(OK : 6 = 3 + 3)**
mantisse : Multiplication des mantisses m_X et m_Y :
rq: $(1 + m_X)(1 + m_Y) = 1 + m_X.m_Y + m_X + m_Y$ → **$m_Z = m_X.m_Y + m_X + m_Y$**

1,m_X = 1,**0101** 0…0 * $2^{130-127}$
x 1,m_Y = x 1,**101** 0…0 * $2^{130-127}$

10101
00000
10101
10101

1,m_M = 10,0010001 0…0 * $2^{133-127}$ = 1,**00010001** 0…0 * $2^{134-127}$ Normalisation
→ **m_M = 00010001** 0…0 → **m_M (23) = 000 1000 1000 0000 0000 0000**

→ *exposant* : **1000 0110 (= 134$_{10}$)**
s (1) : **0**
e (8) : **1000 0110** → **e = 134**
m (23): **000 1000 1000 0000 0000 0000**
sem = 0100 0011 0000 1000 1000 0000 0000 0000 = 43 08 80 00$_{(H)}$
Vérification : A comparer au résultat **136,5 …**

5. Pipelining

1. Indiquer le coût temps calcul, exprimé *en nombre de cycles*, du programme de la méthode 1 :

7 *cycles*

- l'instruction 2 engendre 0 bulle
- l'instruction 3 engendre 1 bulle
- l'instruction 4 engendre 2 bulles
- *Nombre de cycles = nombre d'instructions + nombre de bulles* = 4 + 3 = 7

2. Indiquer le coût temps calcul, exprimé *en nombre de cycles*, du programme de la méthode 2 :

8 *cycles*

- l'instruction 2 engendre 0 bulle
- l'instruction 3 engendre 2 bulles
- l'instruction 4 engendre 2 bulles
- *Nombre de cycles = nombre d'instructions + nombre de bulles* = 4 + 4 = 8

6. Mémoires programmables

Adresse :		*Donnée :*
	1.	$\sqrt{0} = 0$
	2.	$\sqrt{1} = 1$
	…	…
	256.	$\sqrt{256} = 16$

256 = 2^8 Adresses (→ 8 bits d'adresses) et Données sur 32 bits : il s'agit d'une mémoire morte **256x32**.

1. Nombre de bits d'adresse 8 Capacité (en octets) de la mémoire 1024 1 024 octets = 256 x 32 bits = 8 192 bits

2. Quel doit être le modulo du compteur ? 256 256 adresses (donc 256 états) à engendrer par le compteur.

3. Donnée à la dernière adresse = $\sqrt{256} = 16_{10} = 41\ 80\ 00\ 00_H$ $4180\,00\,00_H$

7. Microprogrammation

1. -3 0xFD est le code complément à 2 de -3.

2.

	Micro-instruction	R1	R2	IR	MDR	MAR	PC	Y	SP	Bus	
	Etat initial	indéfini	0x6F	0x7E	0x7E	0x07	0x08	0x08	indéfini	0x7E	
1	**PC out**									**0x08**	*Execute*
2	**MAR in**					**0x08**					
3	**INCR A**							**0x09**			
4	**Y out**									**0x09**	
5	**PC in**						**0x09**				
6	**Lecture; Attente**				**0xFD**						
7	**MDR out**									**0xFD**	
8	**Y in**							**0xFD**			
9	**R2 out**									**0x6F**	
10	**ADD**							**0x6C**			
11	**Y out**									**0x6C**	
12	**MAR in**					**0x6C**					
13	**Lecture; Attente**				**0x27**						
14	**MDR out**									**0x27**	
15	**R1 in**	**0x27**									

3. $0x27$
