



POLYTECHNIQUE
MONTRÉAL

Questionnaire examen final

INF3500

Sigle du cours

Identification de l'étudiant(e)		
Nom :	Prénom :	
Signature :	Matricule :	Groupe :

Sigle et titre du cours			
INF3500 – Conception et réalisation de systèmes numériques			
Professeur	Téléphone	Groupe	Trimestre
André Baron, Chargé de cours Guy Bois, Coordonnateur		Tous	2024-1
Jour	Date	Durée	Heures
Mercredi	24 avril 2024	2 h 30	13 h 30
Documentation	Calculatrice	Outils électroniques	
<input type="checkbox"/> Aucune <input type="checkbox"/> Toute <input checked="" type="checkbox"/> Voir directives particulières	<input type="checkbox"/> Aucune <input checked="" type="checkbox"/> Toutes <input type="checkbox"/> Non programmable (AEP)	Les appareils électroniques personnels sont interdits.	
Directives particulières			
<ul style="list-style-type: none"> Le professeur ne répondra à aucune question durant cet examen. Si vous estimez que vous ne pouvez pas répondre à une question pour diverses raisons, veuillez le justifier puis passer à la question suivante. Une feuille de notes recto verso 8.5"x11" ou A4 permise. Répondre à toutes les questions, la valeur de chaque question est indiquée. Répondre sur le questionnaire et le remettre. Assurez-vous que le crayon que vous utilisez soit suffisamment lisible pour le correcteur. 			
Important	Cet examen contient <input type="text" value="6"/> questions sur un total de <input type="text" value="12"/> pages (incluant cette page).		
	La pondération de cet examen est de <input type="text" value="50"/> %		
	Vous devez répondre sur : <input checked="" type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input type="checkbox"/> les deux		
Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non			

L'étudiant doit honorer l'engagement pris lors de la signature du code de conduite.

Q1 (10 points)

Q1.1 Considérez le code VHDL suivant pour un module combinatoire et son banc de test associé.

```

library ieee;
use ieee.std_logic_1164.all;

entity add3 is
  port (x      : in  std_logic;
        y      : in  std_logic;
        cin    : in  std_logic;
        s      : out std_logic;
        cout   : out std_logic);
end add3;

architecture ex of add3 is

  signal t1 : std_logic;
  signal t2 : std_logic;
  signal t3 : std_logic;

begin

  t1 <= x xor y;
  t2 <= x and y;
  t3 <= cin and t1;
  s   <= t1 xor cin;
  cout <= t3 or t2;

end ex ;
    
```

```

library ieee;
use ieee.std_logic_1164.all;

entity add3_TB is
end add3_TB;

architecture ex of add3_TB is

  signal x, y, cin, s, cout : std_logic;

begin

  UUT : entity work.add3(ex)
    port map (x, y, cin, s, cout);

  x <= '1' after 0 ns, '0' after 10 ns ;
  y <= not (x) ;
  cin <= x ;

end ex;
    
```

Donnez la valeur de tous les ports et signaux internes en fonction du temps, en tenant compte des délais deltas.
Toutes les lignes du tableau pourraient ne pas être requises.

Temps	Delta	x	y	cin	t1	t2	t3	s	cout
0 ns	0	U	U	U	U	U	U	U	U
0 ns	1								
0 ns	2								
0 ns	3								
0 ns	4								
0 ns	5								
10 ns	0								
10 ns	1								
10 ns	2								
10 ns	3								
10 ns	4								
10 ns	5								

Q1.2 Pour chacun des cas suivants, indiquez la valeur de la variable V et du signal S suite à l'exécution du *process*.

```
architecture cas1 of sigvar is
  signal S : std_logic ;
begin
  process
    variable V : std_logic := '0' ;
  begin
    V := not V ;
    S <= not V ;
    report "V = " & to_string(V);
    report "S = " & to_string(S);
    wait ;
  end process;
end cas1 ;
```

V = _____

S = _____

```
architecture cas2 of sigvar is
  signal S : integer range -3 to 9 ;
begin
  process
    variable V : positive ;
  begin
    for i in 0 to 8 loop
      S <= S + 1;
    end loop ;
    wait for 10 ns ;
    report "V = " & to_string(V);
    report "S = " & to_string(S);
    wait ;
  end process;
end cas2 ;
```

V = _____

S = _____

```
architecture cas3 of sigvar is
  signal S : unsigned (7 downto 0) := (0 => '1', others => '0') ;
begin
  process
    variable V : unsigned (7 downto 0) ;
  begin
    V := (others => '0') ;
    for i in 2 downto 0 loop
      V := V + 1 ;
      S <= S + V ;
      wait for 10 ns ;
    end loop ;
    report "V = " & to_string(V);
    report "S = " & to_string(S);
    wait ;
  end process;
end cas3 ;
```

V = _____

S = _____

Q2 (12 points)

Considérez le circuit suivant :

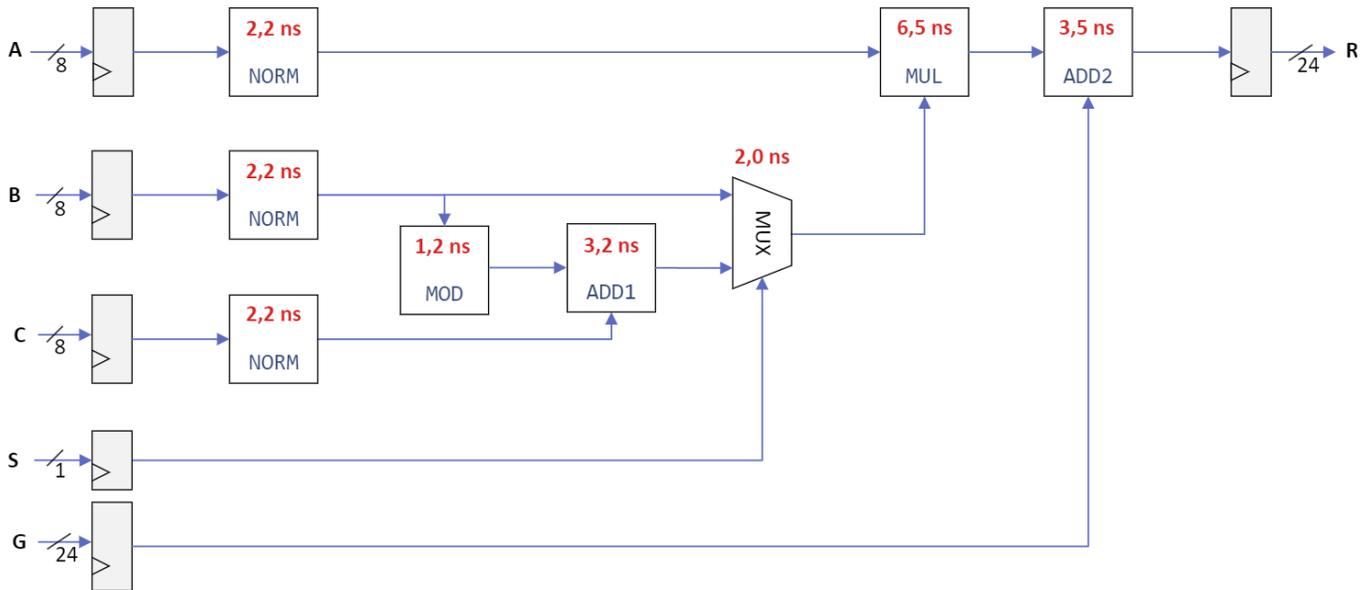


Figure 1

Les bascules ont un délai intrinsèque t_d de 0,4 ns, un temps de préparation t_{su} de 0,3 ns et un temps de maintien t_h de 0,2 ns. Les délais t_{comb} de chacun des blocs combinatoires sont indiqués en rouge sur le schéma. Les délais des interconnexions t_{fil} sont estimés à 0,1 pour chacun des fils.

Pour les questions 2.1, 2.2 et 2.3, assumez que les horloges de tous les registres sont parfaitement en phase (aucun déphasage).

Q2.1 Identifiez le chemin critique du circuit directement sur le diagramme de la Figure 1. Donnez la période minimale d’horloge en ns, la fréquence maximale d’opération en MHz, la latence du circuit en ns, et le débit du circuit en résultats par seconde (un résultat est défini comme une nouvelle valeur de sortie R). Montrez vos calculs.

Tmin		ns
Fréquence maximale		MHz
Latence		ns
Débit		Résultats/Seconde

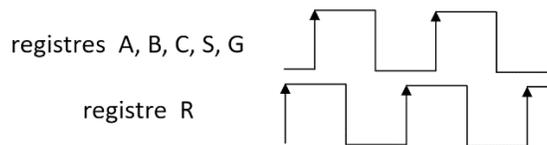
Q2.2 Modifiez le circuit en ajoutant un étage de pipeline afin d’obtenir un débit minimum de 90×10^6 Résultats/Seconde (90 MR/sec). Montrez clairement sur le diagramme de la Figure 1 où vous insérez des registres de pipeline. Indiquez le nouveau chemin critique du circuit, et calculez la nouvelle fréquence maximale d’opération. Donnez le débit obtenu ainsi que la nouvelle latence. Montrez vos calculs.

Tmin cible	<input type="text"/>	ns
Tmin obtenue	<input type="text"/>	ns
Fréquence maximale	<input type="text"/>	MHz
Latence	<input type="text"/>	ns
Débit atteint	<input type="text"/>	Résultats/Seconde

Q2.3 Supposez que vous n’avez pas la possibilité de modifier les blocs combinatoires du circuit de la Figure 1. Calculez la fréquence maximale d’opération qu’il est possible d’atteindre en ajoutant des étages de pipeline. Justifiez votre réponse. N’annotez pas le diagramme.

Fréquence maximale d’opération MHz

Q2.4 Supposons un décalage d’horloge tel qu’illustré ci-dessous (l’horloge alimentant le registre de sortie R est en avance par rapport à l’horloge alimentant les registres d’entrées). Décrivez de façon qualitative l’effet sur la fréquence maximale d’opération du circuit.



Q3 (6 points)

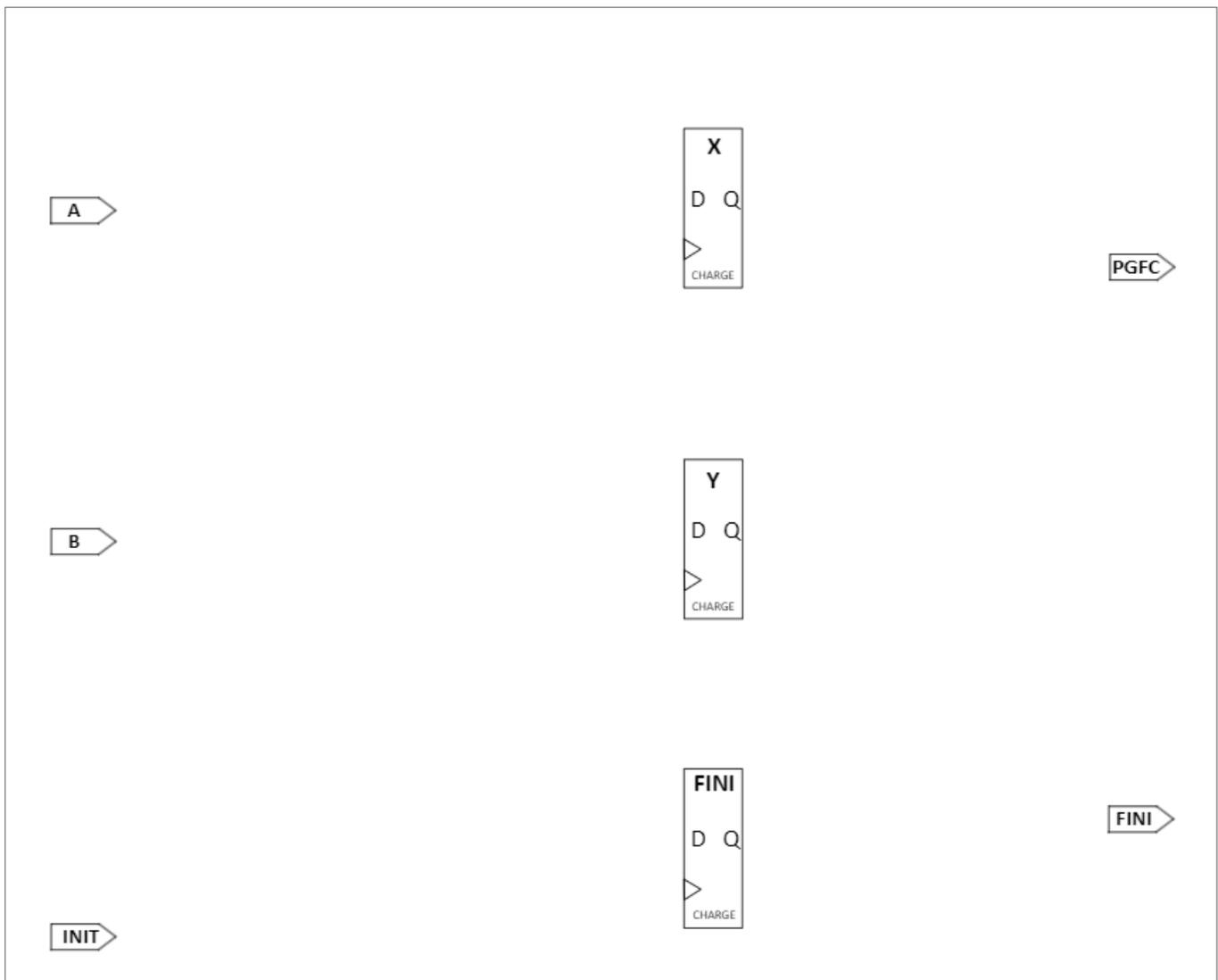
On peut déterminer le plus grand facteur commun (PGFC) de deux nombres entiers A et B par un algorithme itératif exprimé par les micro-opérations suivantes :

```

init : X ← A      ; (init' ET X > Y) : X ← X - Y ;
init : Y ← B      ; (init' ET Y > X) : Y ← Y - X ;
init : fini ← 0   ; (init' ET (X = Y OU X = 1 OU Y = 1)) : fini ← 1 ;
PGFC ← min (X, Y) ;
    
```

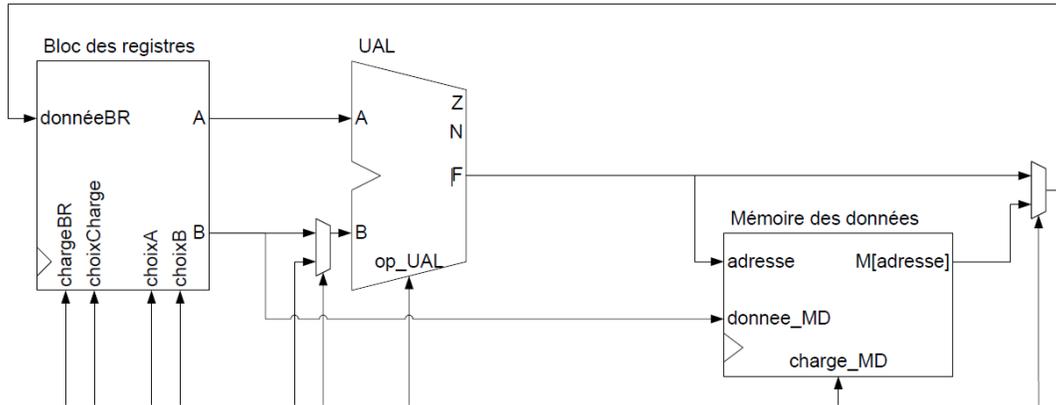
Les signaux *init*, *A* et *B* sont les entrées du système. Les sorties du système sont *PGFC* et *fini*. Quand *init* = 1, on charge les valeurs *A* et *B* dans les registres internes *X* et *Y* respectivement, et on réinitialise la sortie *fini* à 0. Quand *init* = 0, le processus de calcul est lancé. Quand les calculs sont terminés, la sortie *fini* doit prendre la valeur 1, ce qui indique que les calculs sont terminés. *PGFC* est une sortie combinatoire qui prend la plus petite des valeurs *X* ou *Y*.

Le diagramme suivant montre les entrées, sorties et registres du système. Complétez le diagramme de chemin des données qui implémente ce circuit pour des entiers *A* et *B* de 16 bits. Les registres ont un signal de contrôle de charge qui peut être utilisé ou non (optionnel).



Q4 (6 points)

Considérez le chemin des données du processeur PolyRISC.



Supposez que le bloc des registres comporte 4 registres de 16 bits, et que l’UAL peut effectuer les 8 opérations suivantes : A, B, A + B, A – B, A ET B, A OU B, NON A, A OUX B. Supposez que la mémoire des données comporte 256 mots de 16 bits.

Estimez combien de ressources seraient nécessaires pour implémenter ce circuit sur un FPGA de la série 7 de Xilinx. Énoncez clairement toutes vos suppositions et justifiez complètement votre réponse. Donnez votre réponse en termes de LUT, de bascules, de tranches DSP48 et de bits de mémoire Block RAM (BRAM).

LUT	_____ _____ _____ _____ _____ _____ _____
-----	---

Bascules D (FF)	_____ _____ _____
-----------------	-------------------------

DSP48	_____ _____
-------	----------------

BRAM	_____ _____
------	----------------

Q5 (8 points)

Considérez le circuit illustré a la Figure 2 et la déclaration d’entité correspondante.

```
entity efq5 is
port (
  clk : in std_logic ;
  rst : in std_logic ;
  a : in unsigned (15 downto 0);
  b : in unsigned (15 downto 0);
  op : in std_logic_vector (1 downto 0);
  r : out unsigned (15 downto 0));
end efq5 ;
```

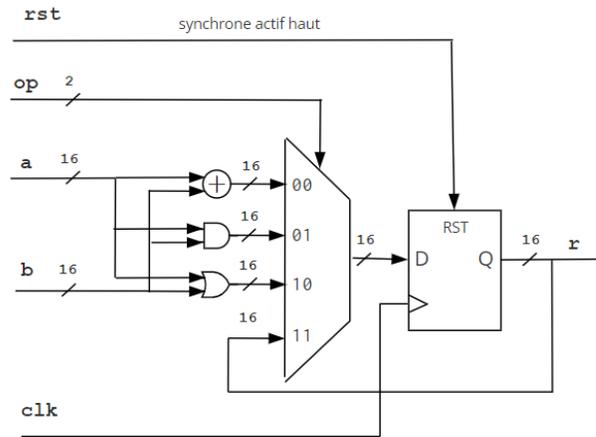


Figure 2

Q5.1 Donnez le code VHDL pour une architecture du circuit illustré a la Figure 2.
Donnez votre réponse à la page suivante.

Q5.2 Considérez le circuit de la Figure 2.
Pour l’entrée « op », on propose le partitionnement en classe suivant : {{00}, {01}, {10}, {11}}.
Proposez un partitionnement en classes pour les entrées « a » et « b », et donnez la taille minimale de l’ensemble de vecteurs de test pour effectuer un test fort sur ce circuit.

Q5.3 Pour une bascule D, expliquez la différence entre une initialisation de type asynchrone et une initialisation de type synchrone.

Q6 (8 points)

Q6.1 Que contient la liste de sensibilité d'un process VHDL?

Q6.2 Combien de transistors sont requis pour implémenter la fonction ou-exclusif (OUX) à deux entrées en CMOS? Justifiez votre réponse.

Q6.3 Donnez deux causes possibles de déphasage d'horloge.

Q6.4 Expliquez pourquoi une implémentation en Logique fixe convient mieux à la production de masse à très haut volume qu'une implémentation en FPGA.

CETTE PAGE NE SERA PAS CORRIGÉE. À UTILISER COMME BROUILLON.

CETTE PAGE NE SERA PAS CORRIGÉE. À UTILISER COMME BROUILLON.

INF3500 202401 EF Solutions

Q1

Q1.1

Time	Delta	x	y	cin	r1	r2	r3	s	cout
0 ps	0	U	U	U	U	U	U	U	U
0 ps	1	1	U	U	U	U	U	U	U
0 ps	2	1	0	1	U	U	U	U	U
0 ps	3	1	0	1	1	0	U	U	U
0 ps	4	1	0	1	1	0	1	0	U
0 ps	5	1	0	1	1	0	1	0	1
10000 ps	0	0	0	1	1	0	1	0	1
10000 ps	1	0	1	0	0	0	1	0	1
10000 ps	2	0	1	0	1	0	0	0	1
10000 ps	3	0	1	0	1	0	0	1	0

Q1.2

Cas1

EXECUTION:: NOTE : V = 1

EXECUTION:: NOTE : S = U

Cas2

EXECUTION:: NOTE : V = 1

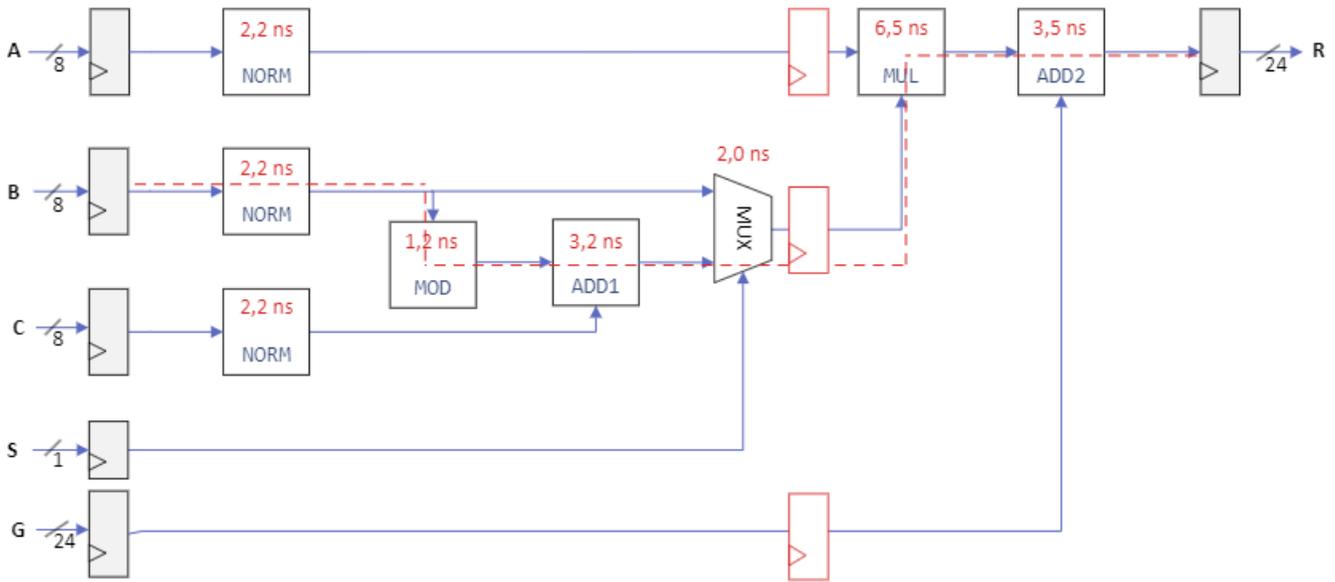
EXECUTION:: NOTE : S = -2

Cas3

EXECUTION:: NOTE : V = 00000011

EXECUTION:: NOTE : S = 00000111

Q2



Q2.1

$T_{min} = 0,4 + 2,2 + 1,2 + 3,2 + 2,0 + 6,5 + 3,5 + 0,3 + 7 \times 0,1 = 20 \text{ ns}$

$F_{max} = 50 \text{ Mhz}$

Latence = 20 ns

Débit = $50 \times 10^6 \text{ R/sec}$

Q2.2

$T_{min \text{ cible}} = 1 / 90 \times 10^6 \text{ R/sec} = 11,1 \text{ ns}$

$T_{min} = 0,4 + 6,5 + 3,5 + 0,3 + 3 \times 0,1 = 11 \text{ ns}$

$F_{max} = 90,9 \text{ Mhz}$

Latence = 22 ns

Débit = $90,9 \times 10^6 \text{ R/sec}$

Q2.3

Chemin critique : MUL

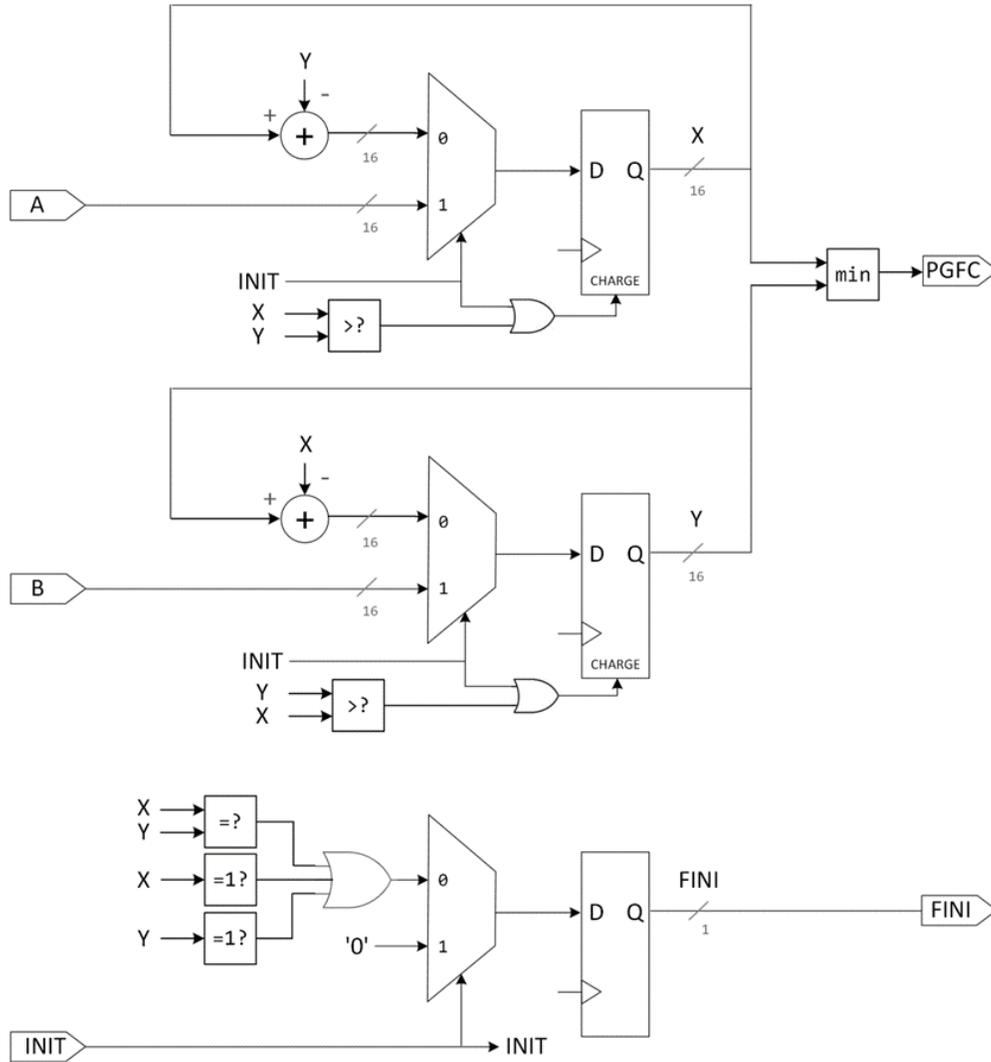
$T_{min} = 0,4 + 6,5 + 0,3 + 2 \times 0,1 = 7,4 \text{ ns}$

$F_{max} = 135,1 \text{ Mhz}$

Q2.4

F_{max} diminue car il faut ajouter tcs au T_{min}

Q3



Q4

LUT

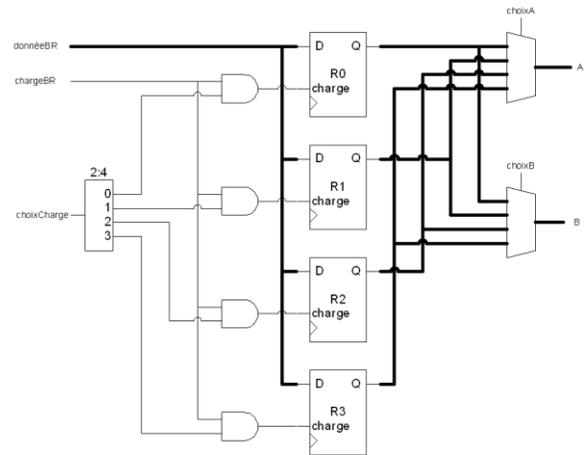
Bloc des registres

Les deux multiplexeurs de sortie du bloc de registres ont chacun 6 entrées : un bit par registre et deux bits pour le choix de la sortie. Donc il faut 1 LUT par bit pour chacun des multiplexeurs, donc 16 LUT chacun, donc 32 LUT.

Il faut 4 LUT pour les 4 portes ET.

Le décodeur a deux entrées et 4 sorties, il faut 4 LUT.

Total Bloc des registres : **40 LUT.**



UAL

Les fonctions + et – peuvent être combinées, il faut 1 LUT par bit ajouté/soustrait, donc 16 LUT.

Les 6 fonctions logiques A, B, ET, OU, NON et OUX ont 5 entrées : les bits Ai et Bi, et 3 bits pour choisir quelle opération est faite. Donc il faut 16 LUT pour ces fonctions.

Il faut choisir entre une somme ou une fonction logique pour chaque bit, donc 16 autres LUT.

Mux UAL_B : 16 LUT

Total UAL : environ **64 LUT.**

Mux BR : **16 LUT.**

TOTAL LUT : 40 + 64 + 16 = 120 LUT.

Bascules

Bloc des registres

Il faut $4 \times 16 =$ **64 bascules.**

BRAM

Il y a 256 mots de 16 bits, donc il faut **4096 bits** de mémoire en Block RAM.

DSP48

Aucun

Q5

Q5.1

```
architecture cp of cp2q3_2 is
    signal q : unsigned (15 downto 0) ;
begin
    process (clk)
    begin
        if (rising_edge(clk)) then
            if (rst = '1') then
                q <= (others=>'0');
            else
                case op is
                    when "00" => q <= a + b ;
                    when "01" => q <= a and b ;
                    when "10" => q <= a or b ;
                    when "11" => q <= q ;
                    when others => q <= (others=>'X');
                end case ;
            end if ;
        end if ;
    end process;

    r <= q ;
end cp ;
```

Q5.2

Partitionnement en classes pour a et b : $\{-2^{15}\}$, $\{-2^{15}-1 \dots -1\}$, $\{0\}$, $\{1 \dots 2^{15}-2\}$, $\{2^{15}-1\}$ – Autres réponses possibles

On a donc $4*5*5 = 100$ vecteurs de tests.

Q5.3

Initialisation asynchrone est indépendante de l'horloge, condition placée avant le test transition de l'horloge, signal d'initialisation requis dans la liste de sensibilité du process VHDL.

Initialisation synchrone est dépendante de l'horloge, condition placée après le test transition de l'horloge, signal d'initialisation non requis dans la liste de sensibilité du process VHDL.

Q6

Q6.1

La liste de sensibilité d'un process VHDL spécifie la liste des signaux qui peuvent entraîner l'exécution du process. Un process n'est exécuté que si un évènement (changement d'état) survient sur un de plusieurs signaux spécifiés dans sa liste de sensibilité.

Q6.2

Dans une implémentation traditionnelle, on a $F = A'B + B'A = (AB + A'B)'$.

Il faut 4 transistors pour inverser A et B.

Ensuite il faut 8 transistors pour réaliser la fonction.

Donc 8 ou 12 transistors, selon la disponibilité de A' et de B' en entrée.

Q6.3

Signaux d'horloge avec des chemins de longueur différente, signal d'horloge avec un débalancement de la charge du signal, contrôle du signal d'horloge avec de la logique combinatoire (*clock gating*).

Q6.4

FPGA : Cout fixe peu élevé mais cout variable élevé

Logique fixe : Cout fixe élevé mais cout variable peu élevé