

INF3500 : Conception et réalisation de systèmes numériques

Examen intra #2 – jeudi 1^{er} novembre 2018

Durée: 1 heure.

Pondération: 10%.

Documentation: Une feuille recto verso 8.5”×11” ou A4 permise.

Calculatrice: Programmable permise.

Q1	
Q2	
Q3	
Q4	
Total	

Directives particulières:

- Ordinateurs interdits. Appareils mobiles interdits.
- Répondez à toutes les questions, la valeur de chaque question est indiquée.
- Répondez sur le questionnaire et remettez-le.
- Ne posez pas de questions. En cas de doute sur le sens d’une question, énoncez clairement vos suppositions.

Question 1. (2 points)

Considérez le code VHDL suivant.

```

library ieee;
use ieee.std_logic_1164.all;

entity module9 is
  port (
    clk: in std_logic;
    A, B, C: in std_logic;
    X, Y, Z: out std_logic
  );
end module9;

architecture arch of module9 is
  signal F, G : std_logic;
begin

  X <= F or A;

  process (clk) is
  begin
    if rising_edge(CLK) then
      F <= B and C;
      G <= B xor C;
    end if;
  end process;

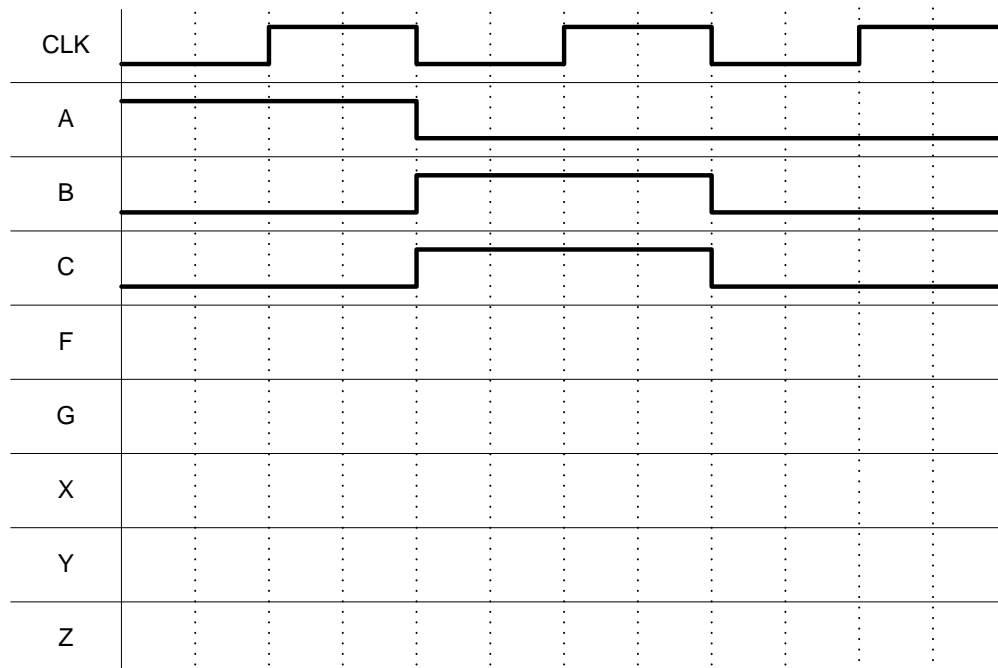
  process (F, G)
  begin
    Y <= F and G;
  end process;

  process (A, B, C)
  begin
    if C = '1' then
      Z <= A or B;
    end if;
  end process;

end arch;

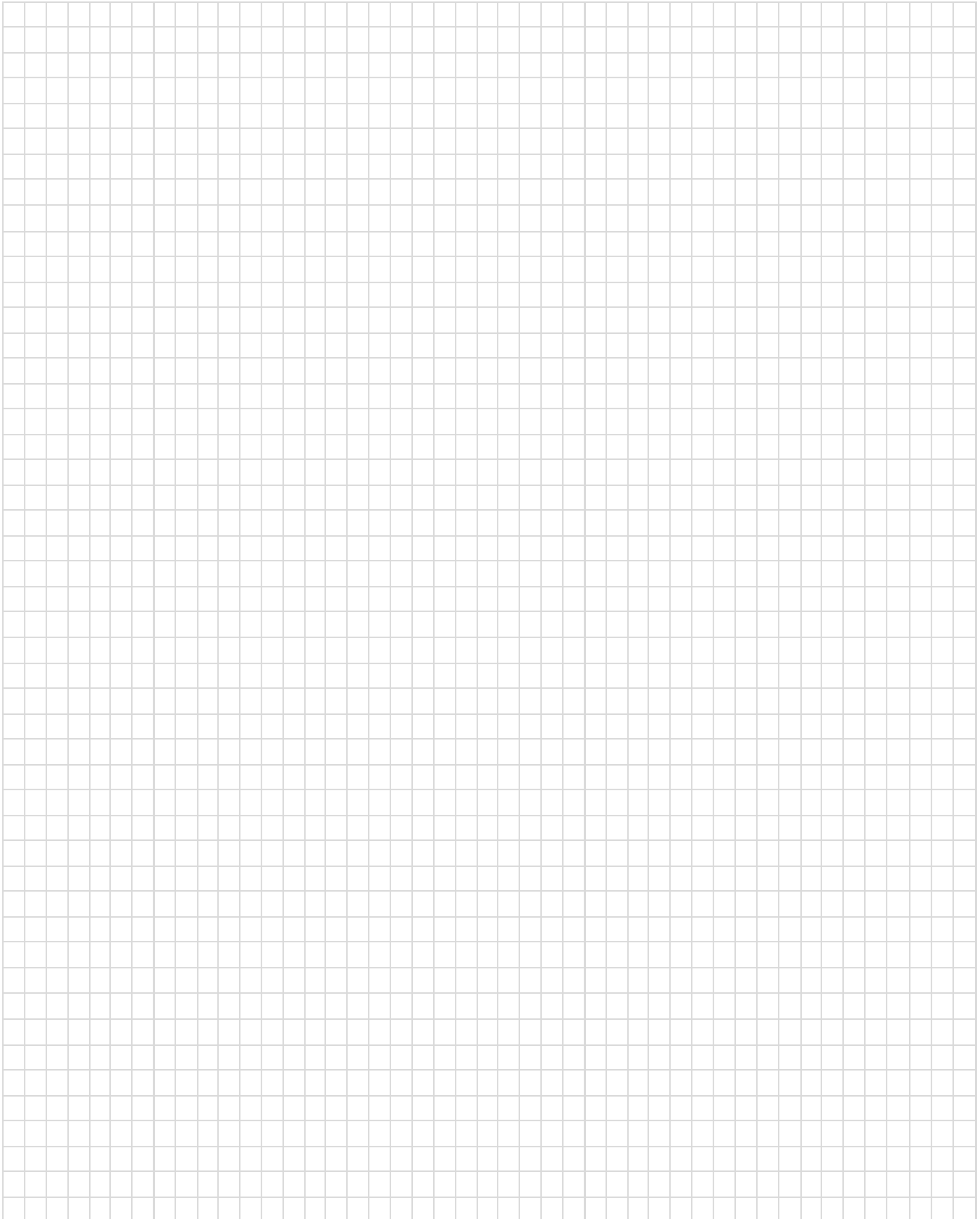
```

Complétez le chronogramme suivant pour ce module.



Question 2. (2 points)

Donnez un schéma d'éléments à mémoire et de portes logiques correspondant au modèle VHDL de la question 1.



Question 3. (4 points)

a. (3 points) La somme de N premiers nombres entiers peut être déterminée à l'aide d'un algorithme itératif exprimé par les trois micro-opérations suivantes. Les entrées du système sont $init$ et N_0 . Les sorties du système sont S et $fini$. Quand $init = 1$, on charge N_0 et on réinitialise la sortie $fini$. Le processus de calcul est lancé quand $init = 0$. La sortie $fini$ doit prendre la valeur 1 quand les calculs sont terminés, ce qui indique que la sortie S contient la somme finale. Donnez le diagramme d'un chemin des données correspondant à ces micro-opérations.

init : $N \leftarrow N_0$; init' ET $N > 0$: $N \leftarrow N - 1$;
 init : $S \leftarrow 0$; init' ET $N > 0$: $S \leftarrow S + N$;
 init : $fini \leftarrow 0$; init' ET $N = 0$: $fini \leftarrow 1$

b. (1 point) Estimez la quantité de bascules (FF) et de LUT nécessaires pour implémenter votre circuit de la question 3 dans un FPGA de la série 7 de Xilinx si N est exprimé sur 8 bits. Justifiez complètement votre réponse.

Type	Nombre	Justification
LUT		<hr/> <hr/> <hr/> <hr/>
FF		<hr/> <hr/> <hr/> <hr/>

Question 4. (2 points)

Le code VHDL suivant décrit l'entité d'un circuit qui effectue la multiplication $N1 \times N2$ de nombres à virgule flottante $N1 = (-1)^{s1} \times m1 \times 2^{e1}$ et $N2 = (-1)^{s2} \times m2 \times 2^{e2}$. Chaque nombre est représenté par un signe s de valeur binaire, d'un exposant e signé et d'une mantisse m non signée. Considérez le cas $E = 7$ et $M = 8$.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity mul_uvf is
  generic (
    E : natural := 7; -- nombre de bits des exposants
    M : natural := 8  -- nombre de bits des mantisses
  );
  port(
    CLK, reset : in std_logic;
    s1, s2 : in std_logic;           -- bits de signe des entrées
    sp : out std_logic;             -- bit de signe du produit
    e1, e2 : in signed(E - 1 downto 0); -- exposants
    ep : out signed(E - 1 downto 0); -- exposant du produit
    m1, m2 : in unsigned(M - 1 downto 0); -- mantisses
    mp : out unsigned(M - 1 downto 0)  -- mantisse du produit
  );
end mul_uvf;
```

a. (1.5 points) Pour chacune des entrées, donnez les valeurs à tester selon l'analyse des valeurs limites.

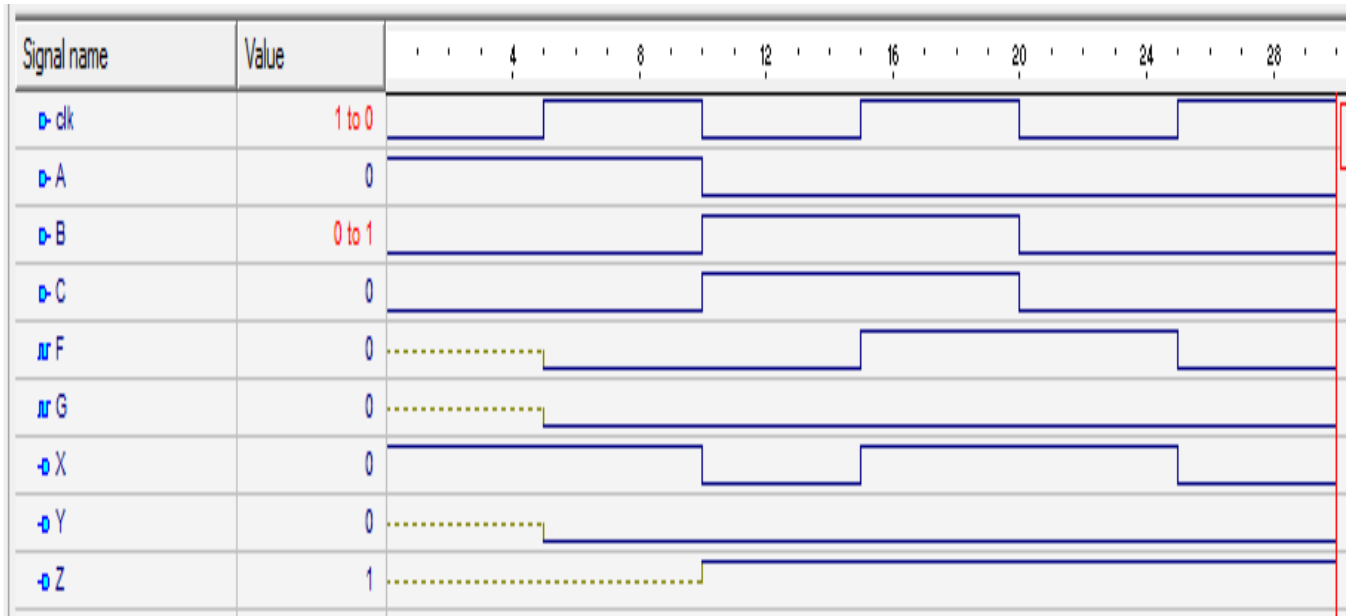
Entrées	Valeurs à tester selon l'analyse des valeurs limites
s1 et s2	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
e1 et e2	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
m1 et m2	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

b. (0.5 point) Combien de vecteurs de tests sont nécessaires pour effectuer la vérification selon l'analyse des valeurs limites?

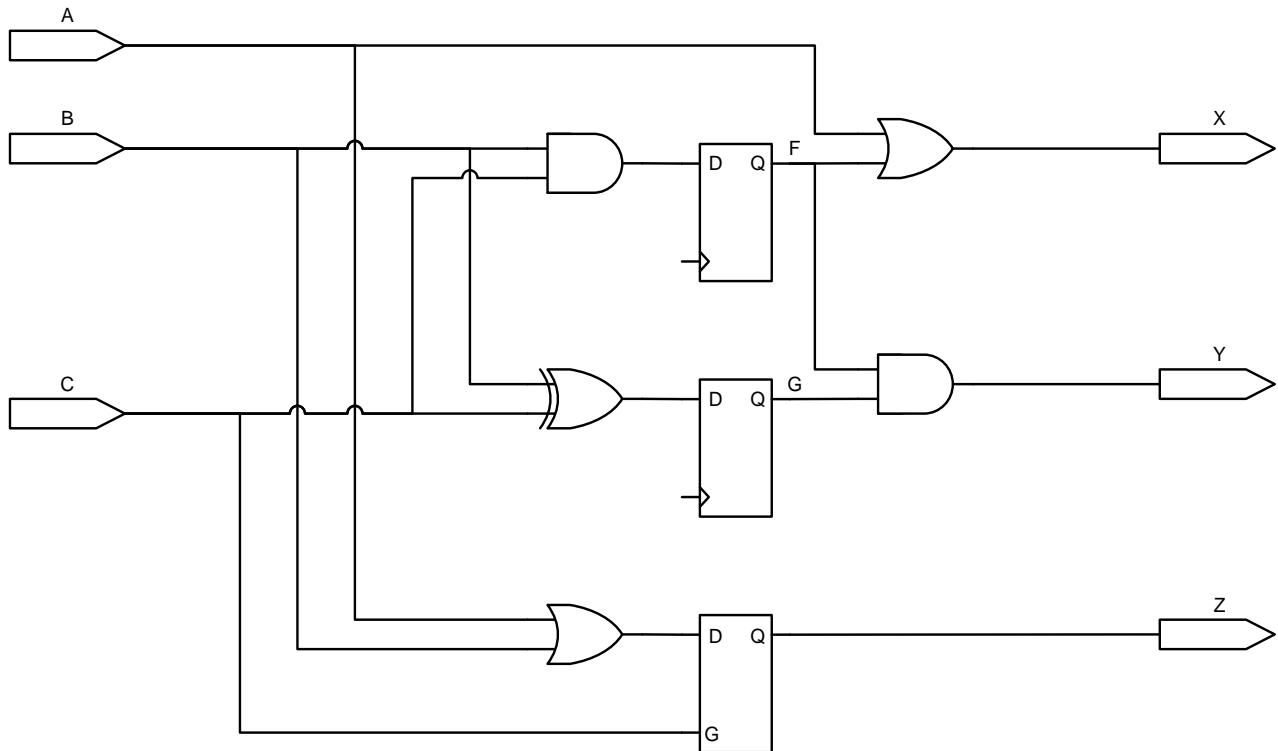
Votre réponse (un nombre)	Justification
	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

Solutions

Q1. chronogramme

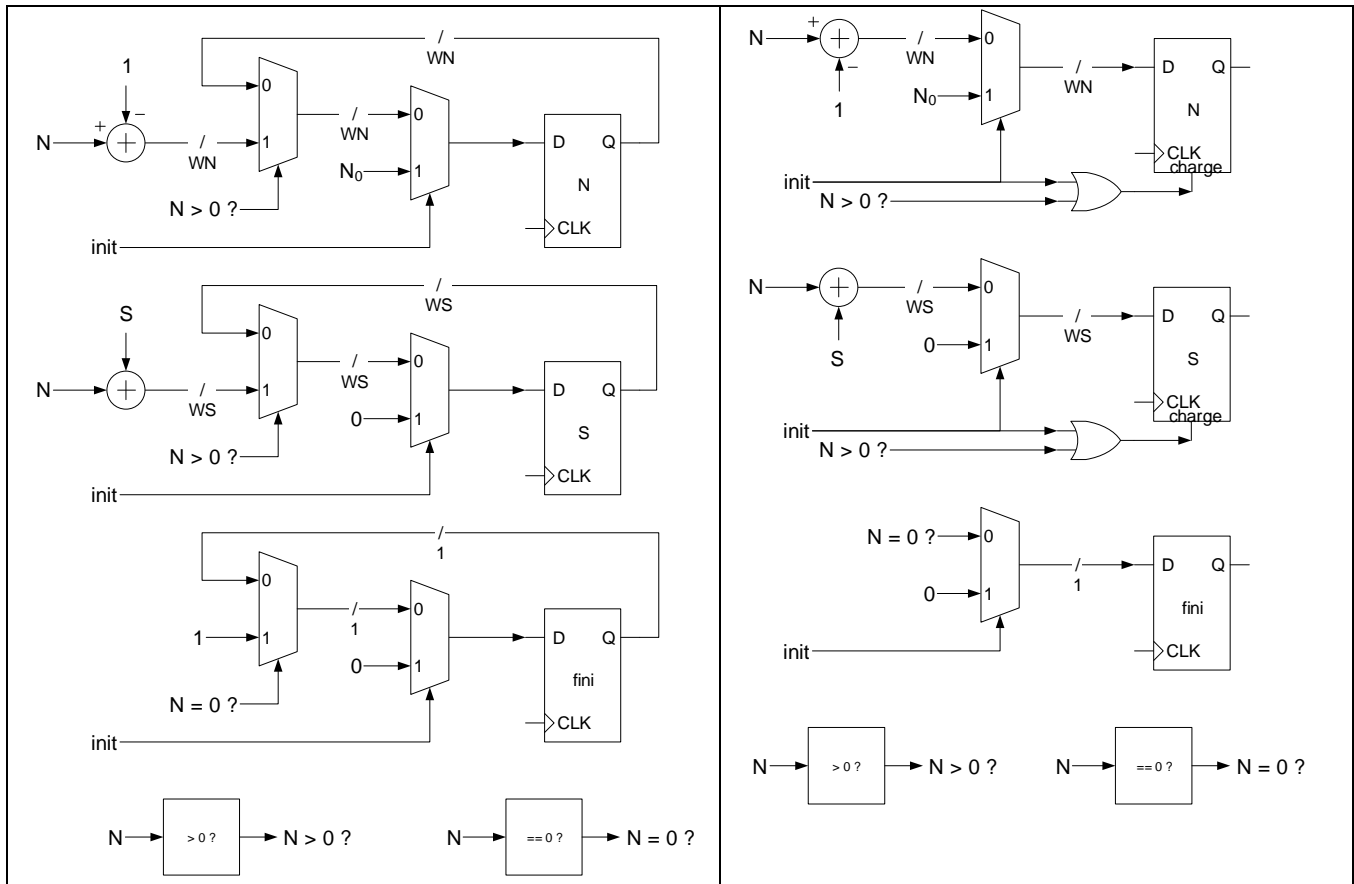


Q2 diagramme



Q3. micro-opérations

a. Réponses possibles



b. ressources

On a que N est exprimé sur 8 bits. Sa valeur maximale est donc 255. La somme S maximale est alors $255 + 254 + 253 + \dots + 1 = 256 \times 255 / 2 = 32640$, qui s'exprime sur 15 bits (puisque $2^{15} = 32768$).

On a donc besoin de 8 FF pour N, 15 FF pour S et 1 FF pour fini, total 24 FF.

La soustraction $N - 1$, sur 8 bits, nécessite 8 LUT. L'addition $N + S$, sur 15 bits, nécessite 15 LUT.

Dans la solution de gauche, les multiplexeurs de N peuvent être remplacés par un seul bloc à 5 entrées : N, $N - 1$, $N > 0 ?$, N_0 et init. Chacun peut donc entrer dans une seule LUT à 6 entrées. Pour les multiplexeurs de N on a alors 8 LUT. Par le même raisonnement, il faut 15 LUT pour les multiplexeurs de S, et une seule LUT pour les multiplexeurs de fini.

La comparaison avec 0 nécessite une porte NON-OU à 8 entrées, qui peut s'implémenter avec deux LUT seulement.

La comparaison $> 0 ?$ nécessite une porte OU à 8 entrées, qui peut s'implémenter avec deux LUT seulement. Dans les faits, on pourrait ne prendre qu'un seul de ces deux circuits (comme N est positif, si $N = 0$, alors N n'est pas $>= 0$, et inversement).

Le total est donc $(8 + 15) + (8 + 15 + 1) + 8 = 55$ LUT.

La solution de droite est un peu plus couteuse, avec une porte OU supplémentaire (partagée pour les registres N et S), donc 56 LUT. Dans la plupart des FPGA, les éléments à mémoire peuvent être configurés avec un signal de chargement ou non, sans coût supplémentaire.

Q4. vérification

Solutions :

a. Pour les entrées s1 et s2, {0, 1}.

Pour les entrées m1 et m2,

{00000000, 00000001, 11000000, 11111110, 11111111}

(si on respecte l'esprit de IEEE 754 :{1.0000000, 1.0000001, 1.1000000, 1.1111110, 1.1111111})

Pour les entrées e1 et e2, en entier : {-64, -63, 0, 62, 63};

en signed : {1000000, 1000001, 0000000, 0111110, 0111111}

Il faudrait prendre $5 \times 5 \times 5 \times 5 \times 2 \times 2 = 2500$ vecteurs de test.