

Nom :

Prénom :

Matricule :

INF3500 : Conception et réalisation de systèmes numériques

Examen intra #2 – 4 novembre 2021

Durée: 1 heure.

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

Pondération: 10%.

Calculatrice: Programmable permise.

Q1	
Q2	
Q3	
Q4	
Total	

Directives particulières:

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

Question 1. (2 points) (12 minutes)

Le problème du laboratoire 2 portait sur un vote à majorité qualifiée avec trois catégories de votants :

Dans le pays Polyanie, les changements constitutionnels nécessitent une majorité qualifiée dans trois groupes d'officiels de l'État, comme suit :

- la majorité simple (50% + 1) des juges de la cours suprême;
- la majorité aux deux tiers (2 / 3 + 1) des membres du sénat; et,
- la majorité aux deux tiers (2 / 3 + 1) des membres de l'assemblée des députés.

Par exemple, s'il y a 9 juges, 50 sénateurs et 300 députés, alors il faut au moins 5 juges ET 34 sénateurs ET 201 députés qui votent en faveur pour que le changement constitutionnel soit fait.

Donnez le nombre de vecteurs de tests nécessaires pour un test exhaustif et pour un test par partitionnement en classes. Supposez qu'il y a J juges, S sénateurs/trices et D député/es. Justifiez complètement votre réponse.

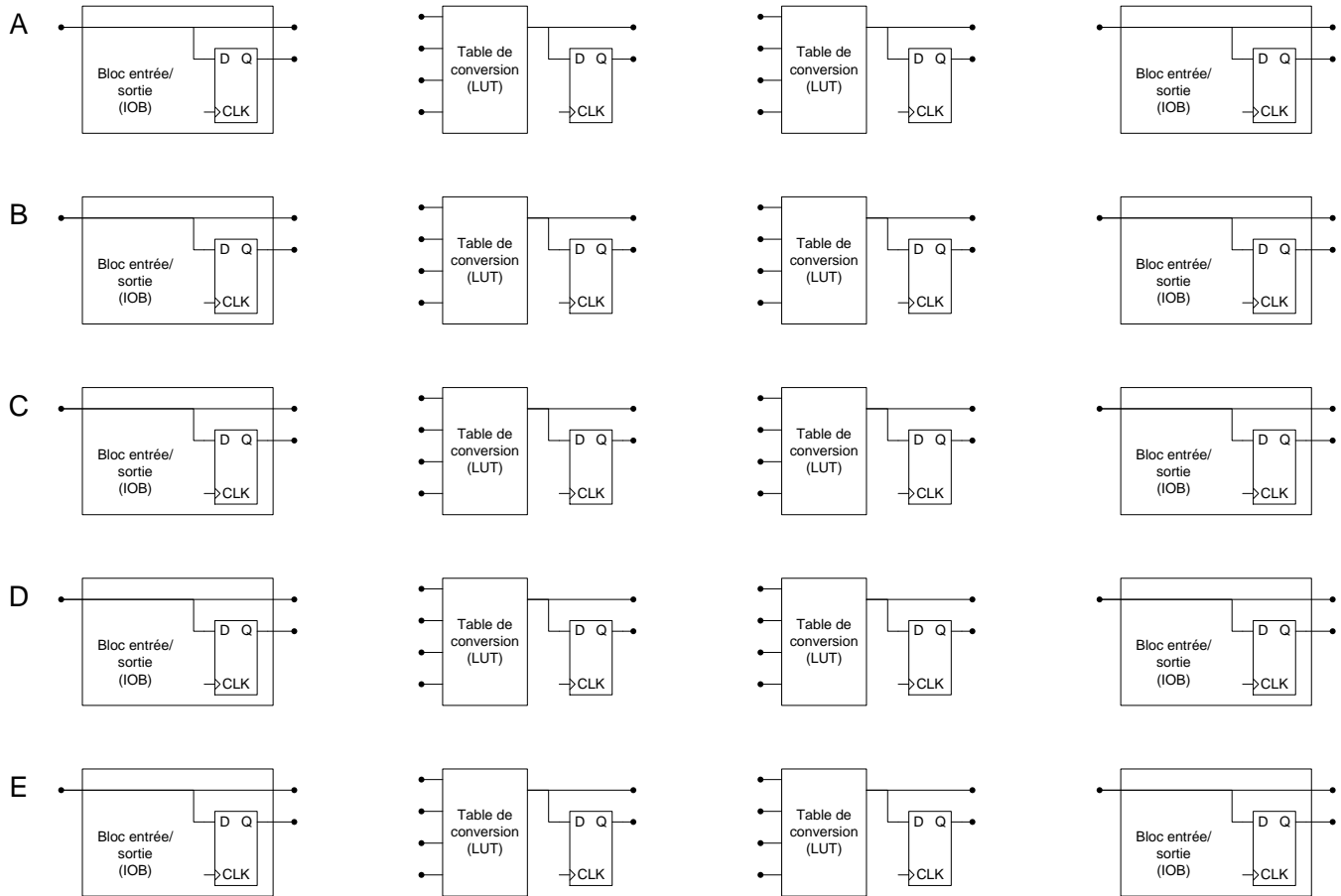
Approche	Nombre de vecteurs de tests
Test exhaustif	_____

Partitionnement en classes	_____

Question 2. (3 points) (18 minutes)

Considérez le code VHDL et le modèle de FPGA suivants. Montrez, sur le modèle du FPGA, un résultat possible de la synthèse et de l'implémentation de ce code. Indiquez directement sur le dessin où les signaux et ports de sortie se situent ainsi que les interconnexions entre les blocs. Les interconnexions peuvent contourner les blocs. Indiquez quand une bascule doit être utilisée. Indiquez par une équation la fonction logique réalisée par chaque LUT que vous utilisez. Respectez l'assignation donnée pour les ports d'entrée.

<pre> library ieee; use ieee.std_logic_1164.all; entity module10 is port (clk, A, B, C, D, E: in std_logic; X, Y, Z: out std_logic); end module10; architecture arch of module10 is signal F, G, H : std_logic; begin X <= not(A and B and E); Y <= G xor H; </pre>	<pre> process(clk) is begin if rising_edge(CLK) then F <= A and B and C and D; G <= F xor E; H <= B or C or D; end if; end process; process(A, B, C) begin if A = '1' then Z <= B or C; else Z <= B and C; end if; end process; </pre>
--	--



Question 3. (3 points : 18 minutes)

On peut calculer la racine carrée $X = \text{sqrt}(A)$ d'un nombre A par la méthode itérative de Newton.

À chaque itération k , on calcule : $X_{k+1} = (X_k + A / X_k) / 2$

et la valeur de X_k converge vers $\text{sqrt}(A)$ après quelques itérations.

Si A et X_k sont exprimés sur 16 et 12 bits, respectivement, on peut débiter les calculs avec l'approximation $X_0 = A / 32$.

Pour $A = 42871$, on obtiendrait la séquence montrée au tableau à droite en arrondissant les X_k à l'entier le plus proche à chaque étape.

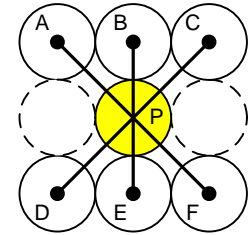
k	X_k
0	1340
1	686
2	374
3	244
4	210
5	207

Faites la conception d'un chemin des données correspondant à cet algorithme et donnez son diagramme. En plus des blocs fonctionnels habituels, vous pouvez utiliser un bloc pour la division d'un nombre de 16 bits par un nombre de 12 bits. Votre chemin des données doit faire 5 itérations comme montré dans l'exemple.

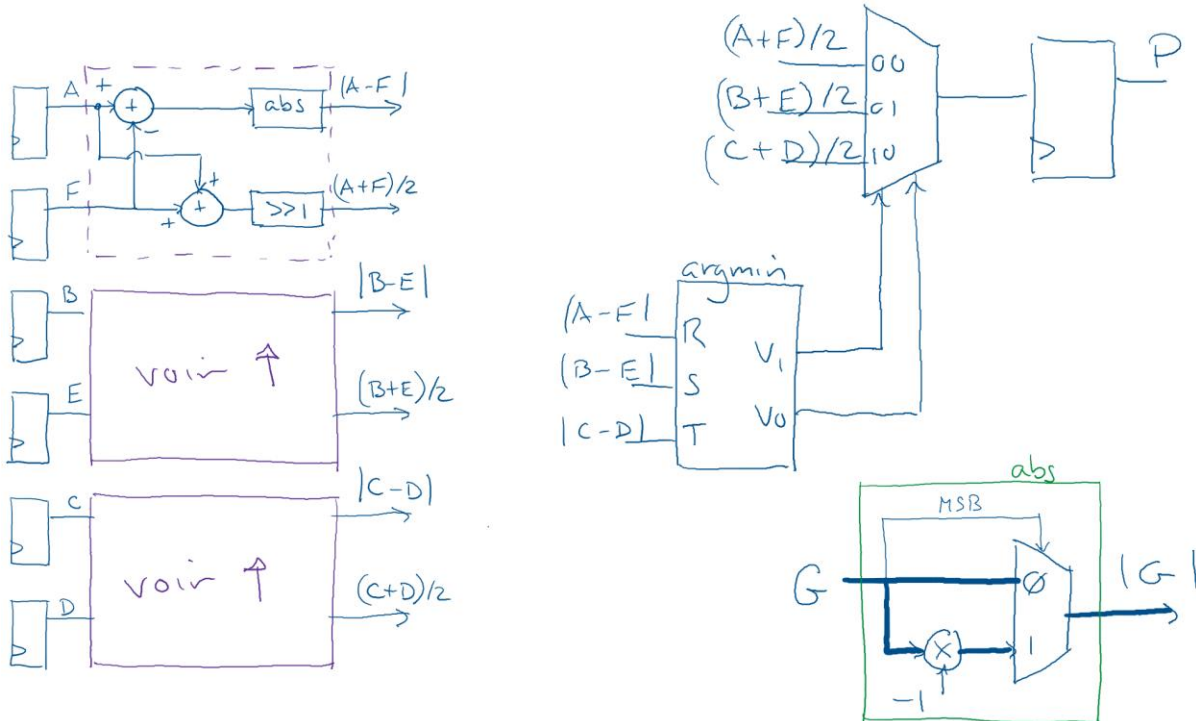
Question 4. (2 points : 12 minutes)

Considérez le chemin des données suivant pour l'algorithme ELA (Edge-based Line Average) qui double la résolution verticale d'une image en créant une nouvelle ligne de pixels entre chaque paire de lignes existantes.

Pour chaque paire de lignes, l'algorithme prend en entrée trois pixels de la ligne du haut (A, B, C) et trois pixels de la ligne du bas (D, E, F), et calcule un nouveau pixel P de la nouvelle ligne intermédiaire. Tous les pixels sont représentés en tons de gris sur 8 bits. L'algorithme calcule trois différences absolues : $|A - F|$, $|B - E|$ et $|C - D|$. La plus petite de ces trois valeurs indique la direction d'interpolation. La valeur de P est alors donnée par la moyenne des deux pixels le long de la direction d'interpolation.



Par exemple, si $|A - F|$ a la plus petite valeur, alors $P = (A + F) / 2$. Si $|B - E|$ a la plus petite valeur, alors $P = (B + E) / 2$, et de façon similaire pour $|C - D|$.



Estimez le nombre de bascules (FF) et de LUT nécessaires pour implémenter ce circuit dans un FPGA de la série 7 de Xilinx. **Supposez que le bloc argmin nécessite 26 LUT et aucune FF.** Justifiez complètement votre réponse.

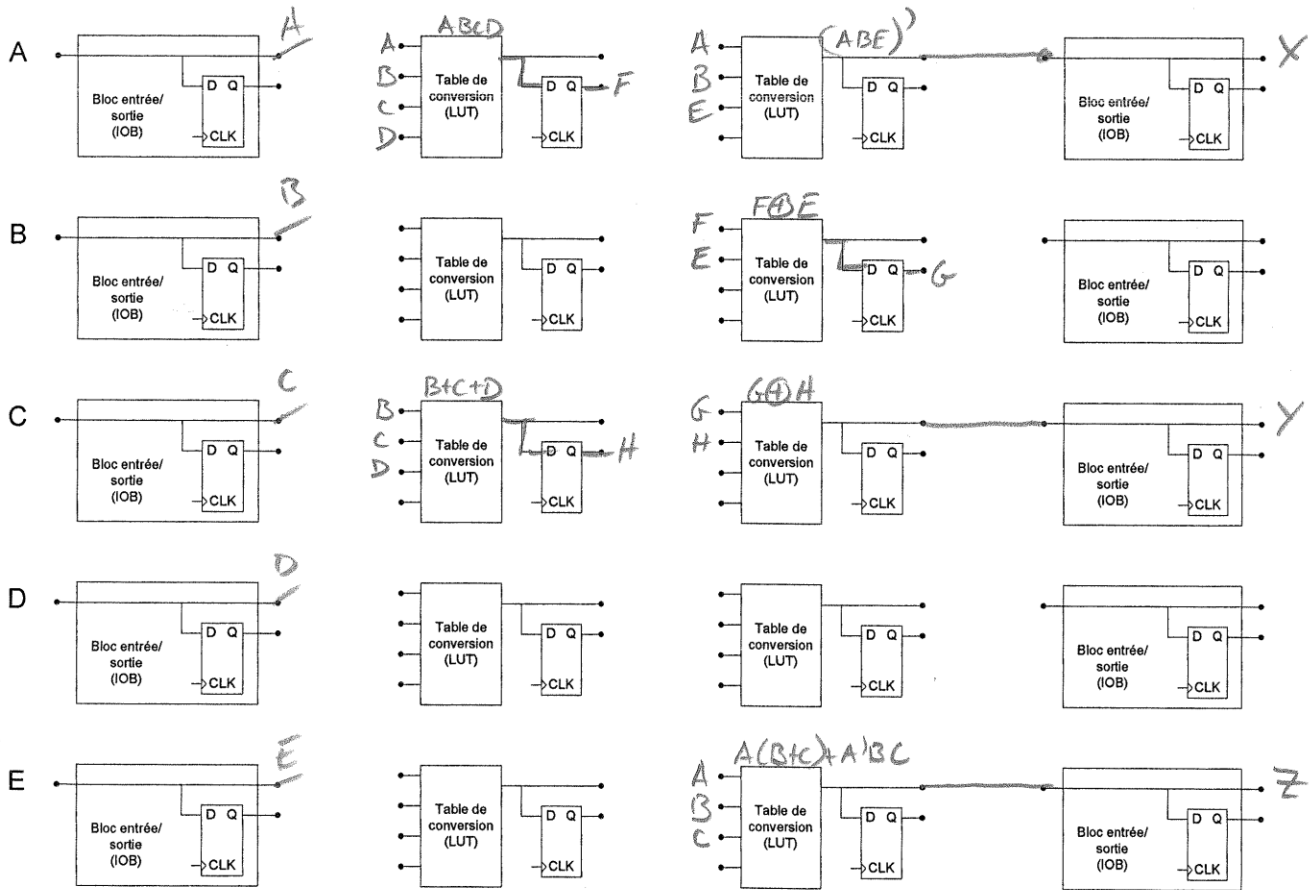
Type	Nombre	Justification
LUT		<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
FF		<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

Solutions

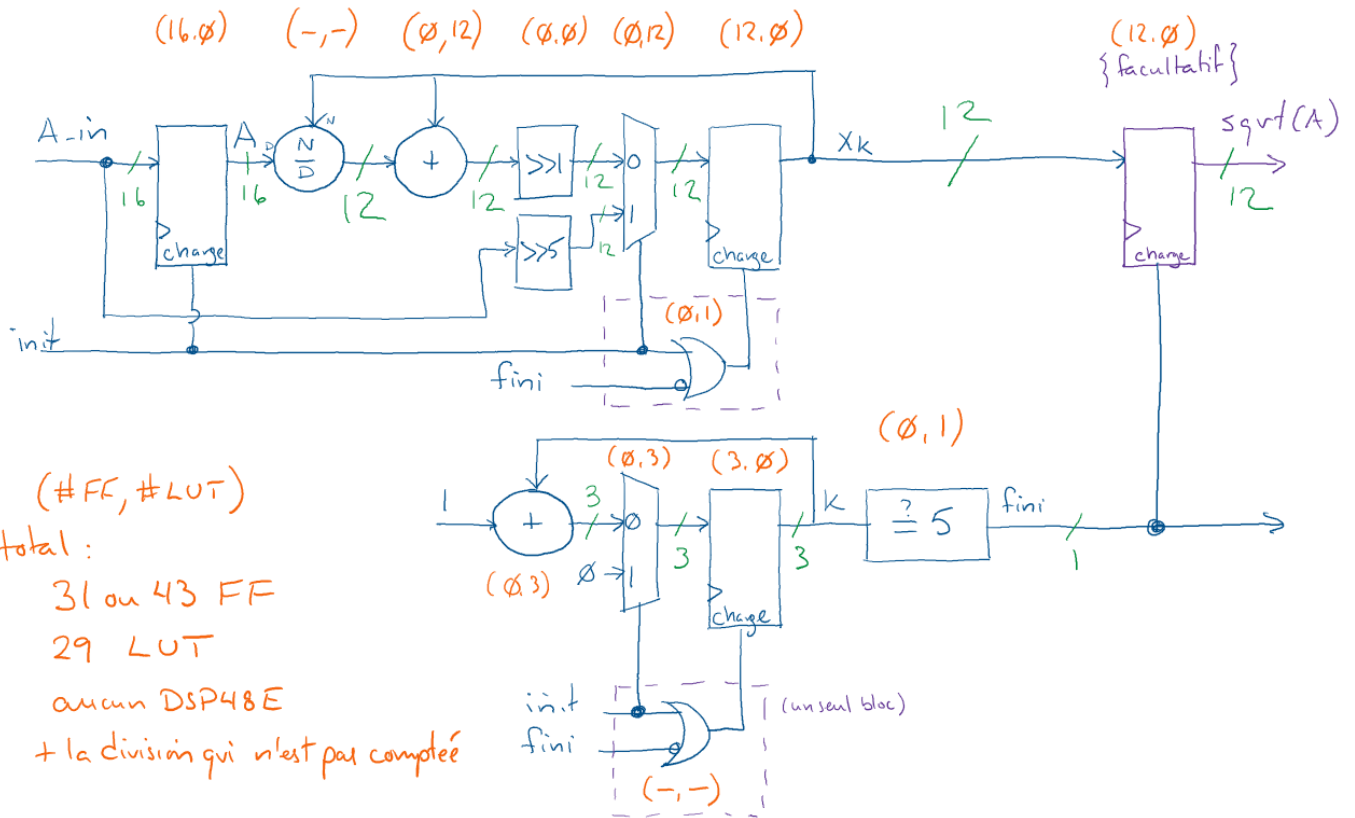
#1

Approche	Nombre de vecteurs de tests
Test exhaustif	<p>1. Si on considère qu'il y a trois entrées, chacune représentant le nombre de votes en faveur, alors il y a :</p> <p>$(J + 1) \times (S + 1) \times (D + 1)$ vecteurs de test.</p> <p>2. Si on considère qu'il y a $J + S + D$ entrées binaires, chacune représentant le vote d'une personne, alors il y a</p> <p>$2^{(J + S + D)}$ vecteurs de test.</p>
Partitionnement en classes	<p>1. Si on considère qu'il y a trois entrées, chacune représentant le nombre de votes en faveur, alors on peut faire deux classes pour chaque entrée.</p> <p>Juges : $\{0, 1, 2, \dots, J/2\}, \{J/2 + 1, J/2 + 2, \dots, J\}$ Sénateurs : $\{0, 1, 2, \dots, 2 \times S/3\}, \{2 \times S/3 + 1, 2 \times S/3 + 2, \dots, S\}$ Députés : $\{0, 1, 2, \dots, 2 \times D/3\}, \{2 \times D/3 + 1, 2 \times D/3 + 2, \dots, D\}$ Où les résultats des opérations avec division sont arrondis à l'entier inférieur, p. ex. 34.9 devient 34. Un test fort nécessiterait seulement $2 \times 2 \times 2 = 8$ vecteurs de test.</p> <p>Dans les faits, on voudrait tester aussi les cas de transition. On pourrait donc imaginer plutôt quatre classes pour chaque entrée :</p> <p>Juges : $\{0, 1, 2, \dots, J/2 - 1\}, \{J/2\}, \{J/2 + 1\}, \{J/2 + 2, J/2 + 3, \dots, J\}$ Sénateurs : $\{0, 1, 2, \dots, 2 \times S/3 - 1\}, \{2 \times S/3\}, \{2 \times S/3 + 1\}, \{2 \times S/3 + 2, 2 \times S/3 + 3, \dots, S\}$ Députés : $\{0, 1, 2, \dots, 2 \times D/3 - 1\}, \{2 \times D/3\}, \{2 \times D/3 + 1\}, \{2 \times D/3 + 2, \dots, D\}$ Où les résultats des opérations avec division sont arrondis à l'entier inférieur, p. ex. 34.02 devient 34. Un test fort nécessiterait alors $4 \times 4 \times 4 = 64$ vecteurs de test.</p> <p>2. Si on considère qu'il y a $J + S + D$ entrées binaires, chacune représentant le vote d'une personne, alors le test de partitionnement en classes ne s'applique pas. Pour chaque entrée, il n'y a que deux valeurs possibles, 0 et 1. On pourrait faire deux classes : $\{0\}$ et $\{1\}$. Le problème se réduit alors au test exhaustif et on n'a rien gagné.</p>

#2 Solution



#3 Solution



#4

Type	Nombre	Justification
LUT	106 ou 130	<p>Il y a 6 additions-soustractions sur 8 bits => $6 \times 8 = 48$ LUT. Le décalage est gratuit.</p> <p>option 1 : La fonction abs telle que montrée sur le diagramme nécessite 8 LUT pour l'inversion du signe, et 8 LUT pour le multiplexeur associé. Il y en a trois, donc $3 \times 16 = 48$ LUT.</p> <p>option 2 : La fonction abs nécessite seulement 8 LUT si on l'implémente avec un additionneur-soustracteur. Il y en a trois, donc $3 \times 8 = 24$ LUT.</p> <p>Le multiplexeur qui précède le registre P a 5 entrées, donc entre dans une seule LUT pour chaque bit. Les signaux sont sur 8 bits, donc il faut 8 LUT.</p> <p>Le bloc argmin nécessite 26 LUT (donnée du problème).</p> <p>Le total est $48 + \{24, 48\} + 8 + 26 = \{106, 130\}$ LUT.</p>
FF	56	Il y a 7 registres de 8 bits, A, B, C, D, E, F et P, donc il faut 56 FF.