

INF3500 : Conception et réalisation de systèmes numériques

Examen final

Hiver 2016

Durée: 2h30

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

Pondération: 50%

Calculatrice: Programmable permise.

Directives particulières:

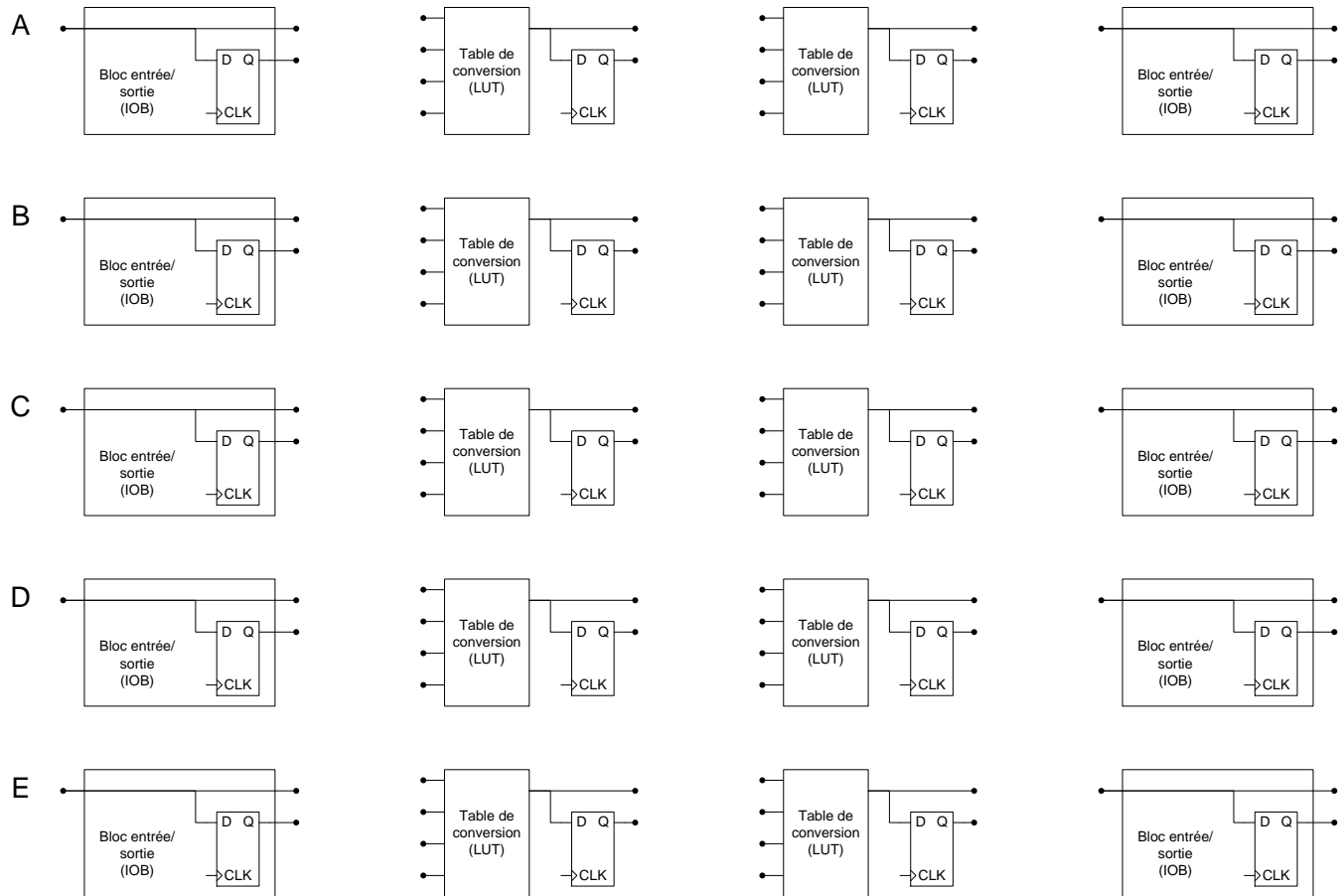
- Ordinateurs, tablettes, téléphones, caméras et autres appareils mobiles interdits.
 - Répondre à toutes les questions, la valeur de chaque question est indiquée.
 - Répondre sur le questionnaire et le remettre.
 - Ne posez pas de questions. En cas de doute sur le sens d'une question, énoncez clairement toute supposition que vous faites.
-

Question 2. (8 points)

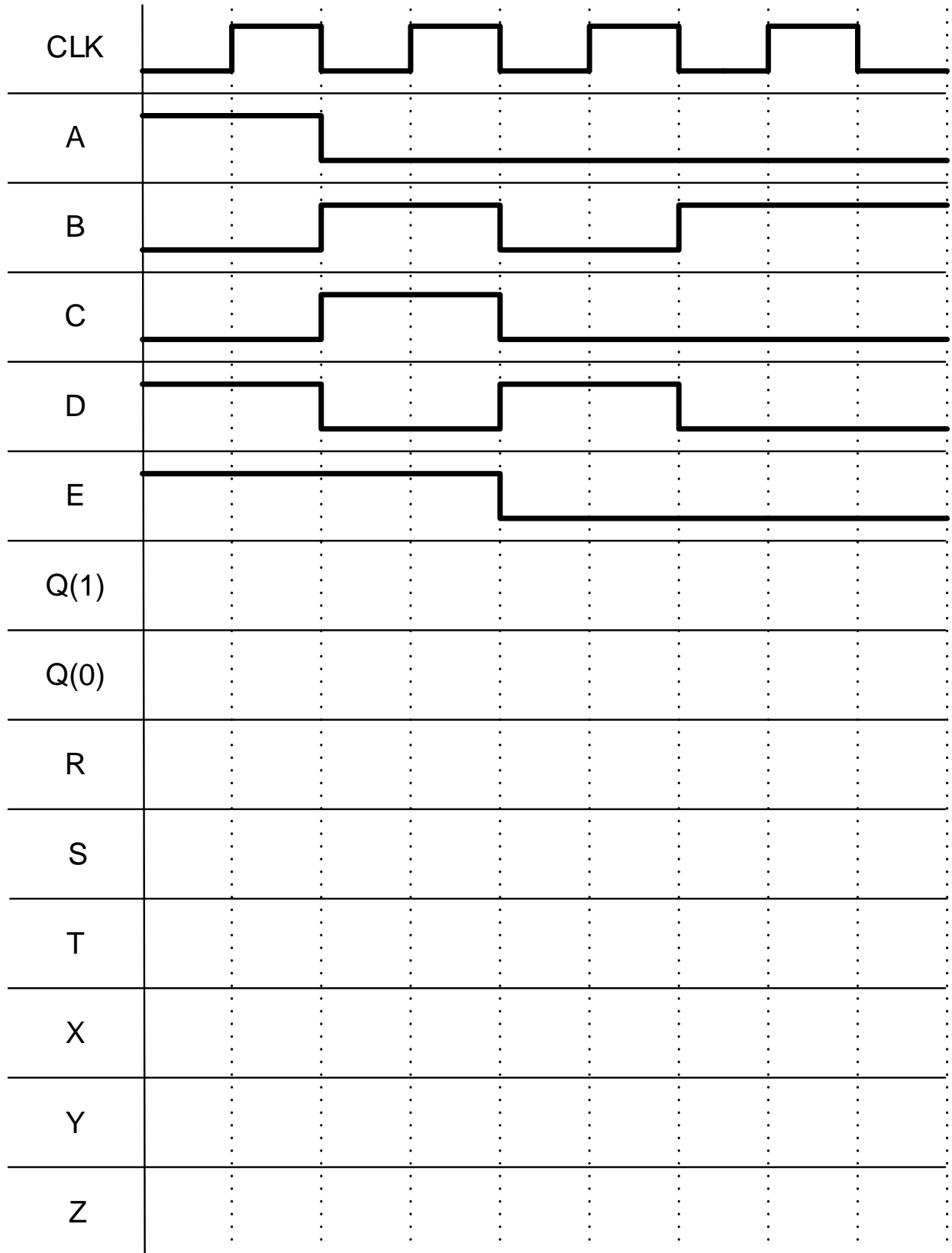
Considérez le code VHDL et le modèle de FPGA suivants.

a. (4 points) 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 chaque signal et port de sortie se situe ainsi que les interconnexions entre les blocs. Les interconnexions peuvent contourner les blocs. Indiquez quand une bascule doit être utilisée. Indiquez clairement, par une équation ou une porte logique, la fonction logique réalisée par chaque LUT utilisée.

<pre> library ieee; use ieee.std_logic_1164.all; entity module14 is port (clk, A, B, C, D, E: in std_logic; X, Y, Z: out std_logic); end module14; architecture arch of module14 is signal Q : std_logic_vector(1 downto 0) := "10"; signal R, S, T : std_logic := '1'; begin X <= T and Q(0); Z <= R xor S xor T; </pre>	<pre> with A select R <= B when '1', C when '0', '0' when others; process(D, E) begin S <= D and E; T <= D xor E; end process; process(clk) is begin if rising_edge(CLK) then Q(1) <= Q(1) or not(A) or S; Q(0) <= not(B and Q(1)); Y <= Q(1) xor Q(0); end if; end process; end arch; </pre>
--	---



b. (4 points) Complétez le chronogramme suivant pour ce code VHDL.



Temps (ns): 5 10 15 20 25 30 35 40 45

Question 3. (9 points)

On peut déterminer le plus grand facteur commun (PGFC) de deux nombres entiers de 16 bits A et B par un algorithme itératif exprimé par les trois micro-opérations suivantes. Par exemple, pour les nombres 42 et 56, pour lesquels le PGFC est 14, on aurait la séquence indiquée.

init : $A \leftarrow A_0$; init' ET $A > B$: $A \leftarrow A - B$;

init : $B \leftarrow B_0$; init' ET $A < B$: $B \leftarrow B - A$;

init : $fini \leftarrow 0$; init' ET $A = B$: $fini \leftarrow 1$

itération :	0	1	2	3
init :	1	0	0	0
A :	42	42	28	14
B :	56	14	14	14
fini :	0	0	0	1

Les signaux $init$, A_0 et B_0 sont les entrées du système. Les sorties du système sont A et $fini$. Quand $init = 1$, on charge A_0 et B_0 et on réinitialise la sortie $fini$. 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 la sortie A contient le PGFC.

a. (6 points) Donnez le diagramme d'un chemin des données correspondant à ces micro-opérations.

Question 6. (8 points)

a. (1 point) Expliquez pourquoi un ensemble de vecteurs de tests donnant une couverture de code de 100% n'est pas nécessairement suffisant pour vérifier adéquatement un module.

b. (1 point) Donnez deux qualités d'un bon ensemble de vecteurs de test.

c. (1 point) Donnez deux exemples de techniques de vérification de type « boîte noire. »

d. (1 point) Expliquez comment un double tampon peut permettre de synchroniser l'échange d'informations entre deux domaines d'horloge différents.

e. (1 point) Expliquez pourquoi la division générale n'est pas synthétisable par la plupart des synthétiseurs de code HDL ciblant des FPGA.

f. (1 point) Nommez et décrivez deux principes à suivre pour la conception et l'implémentation de circuits numériques sur FPGA.

g. (1 point) Donnez deux causes possibles du déphasage d'horloge.

h. (1 point) Donnez deux métriques pour comparer des solutions de placement d'un circuit sur un FPGA.

Solutions

Q1.

A. Analyse des coûts

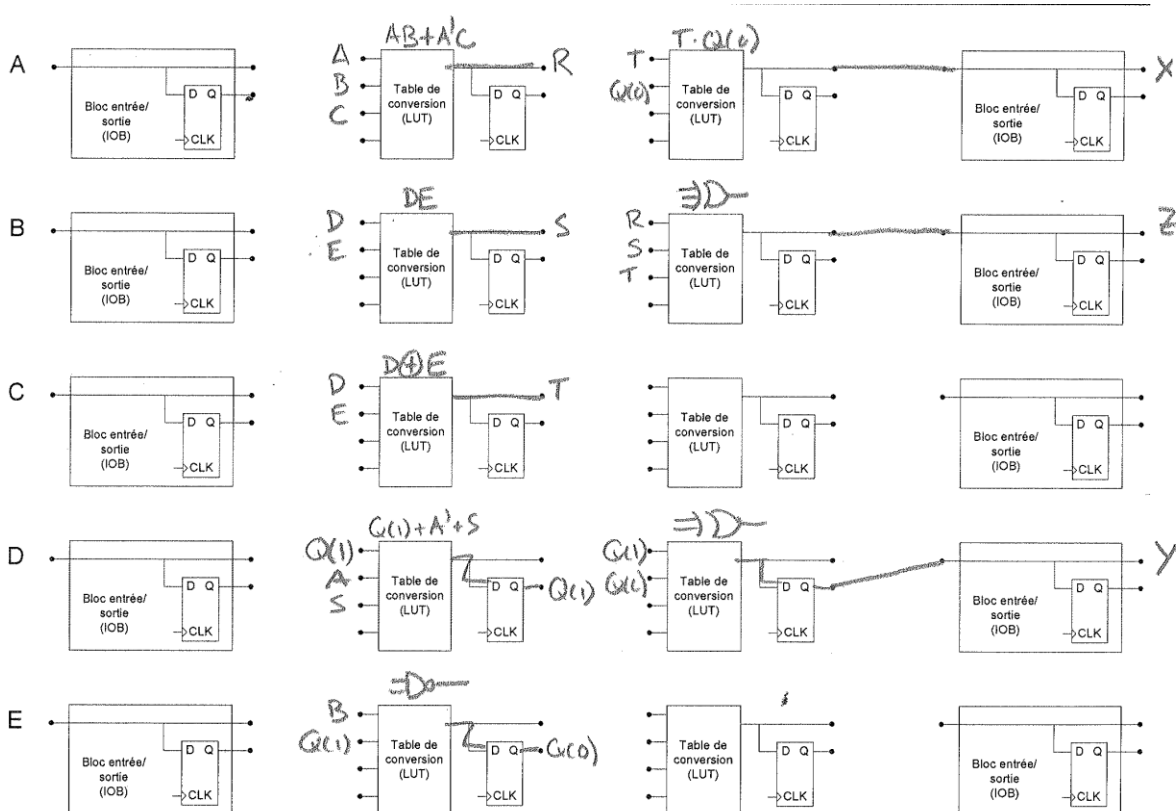
Quoi	FPGA	Logique fixe
Main d'oeuvre	$2 \times 6 / 12 \times \$100 \text{ k} / \text{an} = \100 k	$3 \times 6 / 12 \times \$100 \text{ k} / \text{an} = \150 k
Licenses	$2 \times \$5 \text{ k}$	$3 \times \$20 \text{ k}$
Autres frais	$\$0$	(fonderie) $\$250 \text{ k}$
Total des frais fixes	$\$110 \text{ k}$	$\$460 \text{ k}$
Par puce	$\$250$	$\$10$
Commande #1 : 100 systèmes	$\$110 \text{ k} + 100 \times \$0.25 \text{ k} = \$135 \text{ k}$	$\$460 \text{ k} + 100 \times \$0.01 \text{ k} = \$461 \text{ k}$
Commande #2 : 2000 systèmes	$\$110 \text{ k} + 2000 \times \$0.25 \text{ k} = \$610 \text{ k}$	$\$460 \text{ k} + 2000 \times \$0.01 \text{ k} = \$480 \text{ k}$

Le FPGA est préférable pour la commande de 100 unités, et la logique fixe est préférable pour la commande de 2000 unités.

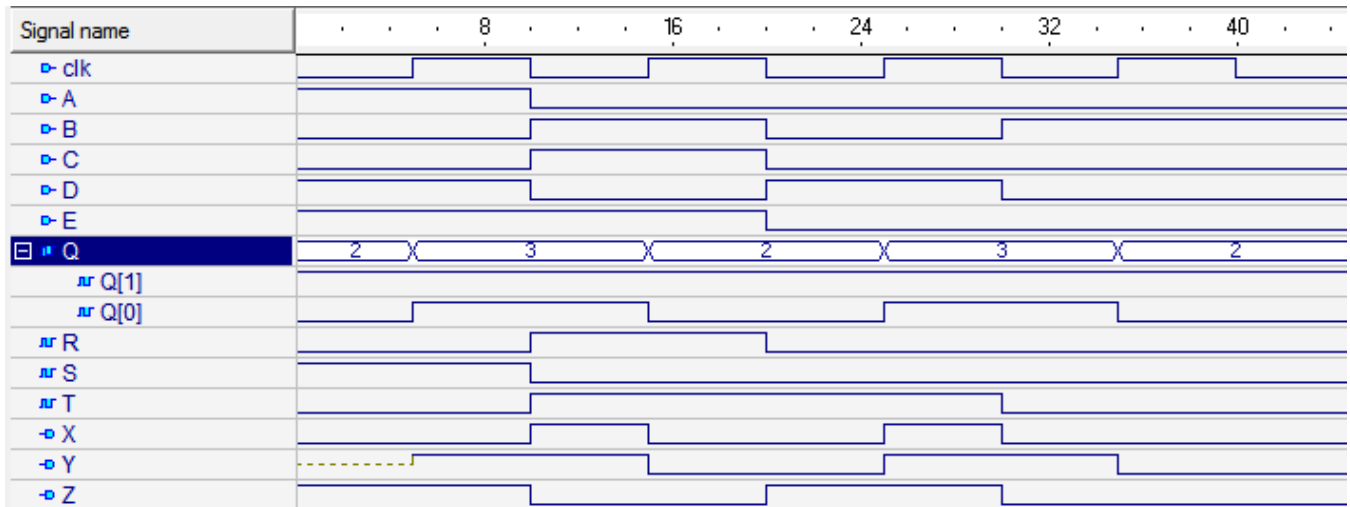
b. Le facteur temps peut être important aussi. La logique fixe a un délai de 4 mois. Les risques sont grands. Si on découvre un bogue, il faut attendre à nouveau 4 mois et repayer les frais fixes de fonderie pour faire une nouvelle puce. Avec le FPGA, les puces seraient disponibles dans un mois. Mais si un bogue était découvert, il ne serait pas nécessaire de faire une nouvelle commande, il faudrait seulement reprogrammer la puce.

La consommation d'énergie est plus élevée pour le FPGA que la logique fixe.

Q2. a.

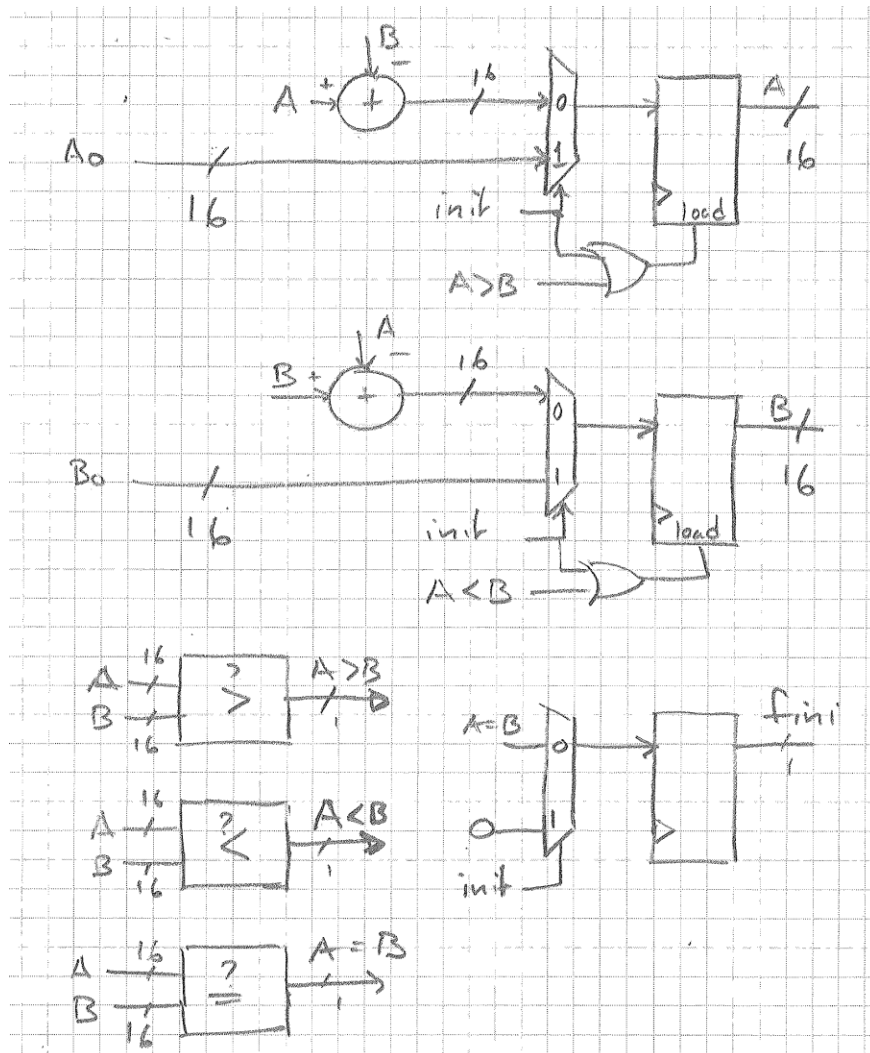


b.



Q3.

a. diagramme



b. ressources

LUT :

Soustractions : $2 \times 16 = 32$ LUTComparaisons (<, >, =) : $3 \times 16 = 48$ LUTMultiplexeurs : 2×16 LUT = 32 pour A et B, 1 LUT pour le mux de fini;

Portes logiques : 2 LUT pour les portes OU

Total : $32 + 48 + 32 + 1 + 2 = 115$ LUT environ

FF : il faut 33 registres : 16 pour A, 16 pour B et 1 pour fini

DSP48E : on n'utilise aucune tranche DSP48E pour ce circuit

Q4.

```

library IEEE;
use IEEE.std_logic_1164.all;
use ieee.numeric_std.all;

entity monnaie is
    port (
        clk, reset : in std_logic;
        charge, choix : in std_logic;
        montant : in unsigned(9 downto 0);
        fini : out std_logic;
        piece: out unsigned(7 downto 0)
    );
end monnaie;

architecture arch of monnaie is

begin

    process (clk, reset)
        variable aRendre : unsigned(9 downto 0);
    begin
        if reset = '0' then
            aRendre := to_unsigned(0, 10);
            piece <= to_unsigned(0, 8);
        elsif rising_edge(clk) then
            if charge = '1' then
                if choix = '1' then
                    aRendre := montant;
                else
                    if aRendre >= 200 then
                        aRendre := aRendre - 200;
                        piece <= to_unsigned(200, 8);
                    elsif aRendre >= 100 then
                        aRendre := aRendre - 100;
                        piece <= to_unsigned(100, 8);
                    elsif aRendre >= 25 then
                        aRendre := aRendre - 25;
                        piece <= to_unsigned(25, 8);
                    elsif aRendre >= 10 then
                        aRendre := aRendre - 10;
                        piece <= to_unsigned(10, 8);
                    elsif aRendre >= 5 then
                        aRendre := aRendre - 5;
                        piece <= to_unsigned(5, 8);
                    else
                        aRendre := aRendre - 0;
                        piece <= to_unsigned(0, 8);
                    end if;
                end if;
            end if;
            if aRendre < 5 then
                fini <= '1';
            else
                fini <= '0';
            end if;
        end process;

    end arch;

```

Q5.

a. Le chemin critique est F – C0R3 – C2R1 – C3R0 – R.

On a alors $T_{\min} = 0.4 + 3 \times 0.3 + 0.2 + 10 \times 0.25 = 4 \text{ ns}$, soit 250 MHz

b. 2 cycles, soit 8 ns

c. On veut $T_{\min} \leq 2 \text{ ns}$ pour un débit de 500×10^6 résultats par seconde.

Il faut pipeliner aux colonnes 0, 2 et 3.

Le chemin critique est alors de la colonne 2 à la colonne 3 :

$T_{\min} = 0.4 + 4 \times 0.25 + 0.3 + 0.2 = 1.9 \text{ ns}$

d. La latence est maintenant de 5 cycles, soit $5 \times 2 \text{ ns} = 10 \text{ ns}$.

e. Le signal Etemp serait placé dans un registre d'un bloc d'entrée, et le signal de sortie R dans un bloc de sortie. Il faut utiliser une table de conversion pour l'inversion. On place tout sur une seule rangée. On a alors $T_{\min} = 0.4 + 0.3 + 5 \times 0.25 + 0.2 = 2.15 \text{ ns}$, donc $f_{\max} = 465 \text{ MHz}$

Q6.

a. La couverture de code n'indique que si certaines situations ont été exercées ou non, sans égard à la fonctionnalité du système.

b. Efficace pour découvrir des bogues, c'est-à-dire que chaque vecteur de test vérifie plusieurs fonctionnalités en même temps, et donc que peu de vecteurs de tests sont nécessaires.

Identifie la source des bogues, pour aider à leur éradication.

Reproductible, donc il est facile de recréer le bogue.

Automatisé à l'aide d'un banc d'essai.

Exécutable dans un temps raisonnable

c. test exhaustif, test des valeurs limites, partitionnement en classes et test aléatoire

d. Un double tampon consiste de deux bascules en série. L'entrée asynchrone est appliquée à la première bascule. Il est possible que celle-ci entre dans un état métastable. Elle finira en pratique par se stabiliser sur un 1 ou un 0. La deuxième bascule permet d'accorder le plus long temps possible à la première bascule pour se stabiliser, soit une période d'horloge.

e. La division générale n'est pas synthétisable par choix des concepteurs de synthétiseurs. Puisque les FPGA ne possèdent pas de bloc implémentant la division, il faudrait choisir une architecture de diviseur en particulier et la réaliser avec un arrangement de LUT, FF, etc. Les concepteurs de synthétiseurs préfèrent laisser ce choix aux concepteurs de systèmes numériques qui écrivent du code HDL.

f. Quelques solutions possibles : conception synchrone, ne pas utiliser de loquets, signal d'initialisation global, routage spécial du signal d'horloge, exploitation de toutes les ressources de la puce, gestion manuelle de la disposition, arithmétique en virgule fixe plutôt que flottante.

g. 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*).

h. longueur totale des interconnexions, chemin le plus long, congestion des interconnexions.