

INF3500 : Conception et réalisation de systèmes numériques

Examen final

Hiver 2014

Durée: 2h30.

Pondération: 50%.

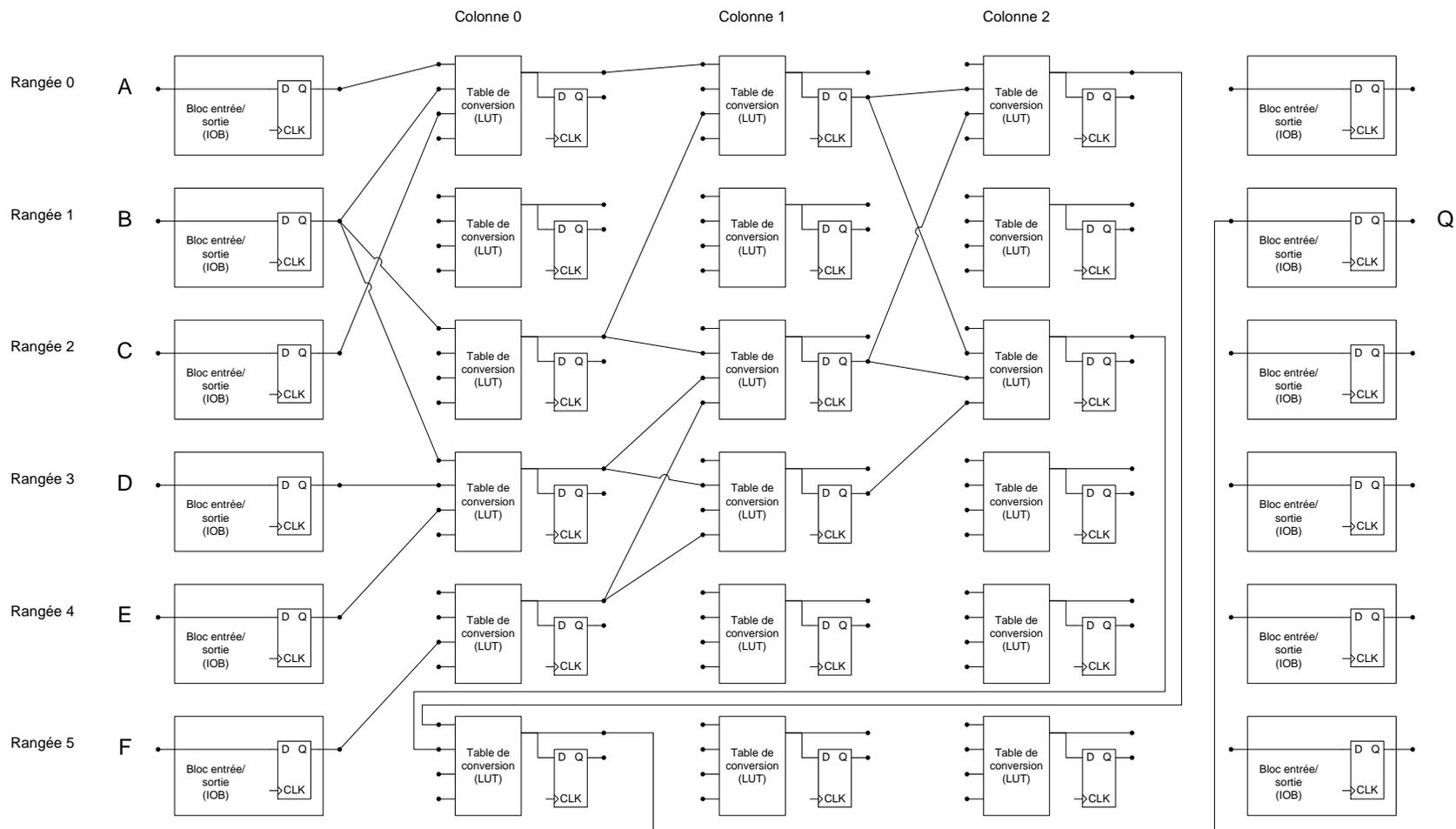
Documentation: Une feuille recto-verso 8.5"×11" ou A4 permise.

Calculatrice: Programmable permise.

Répondre sur le questionnaire et le remettre.

Directives particulières:

- Répondre à toutes les questions, la valeur de chaque question est indiquée.
 - Ne posez pas de questions. En cas de doute sur le sens d'une question, énoncez clairement toute supposition que vous faites.
-



Question 4. (8 points)

Considérez le code VHDL suivant. Au début de la simulation, le processus est lancé.

```

entity demoprocessus is
end demoprocessus;

architecture arch of demoprocessus is

signal s1 : integer := 100;
signal s2, s3, s4, s5 : integer := -100;

begin

                                process
                                variable v1 : integer := 2;
                                begin
                                    while (v1 >= 0) loop

                                        s2 <= v1 + 5;
                                        v1 := v1 - 1;
                                        s3 <= v1 + 5;

                                        s1 <= s1 + 1;
                                        s4 <= s3 + s2;

                                        wait for 5 ns;

                                        s3 <= s3 + v1;
                                        s4 <= s3 + s2;

                                        wait for 5 ns;

                                        s4 <= s3;
                                        s5 <= s3;

                                        wait for 0 ns;

                                    end loop;

                                    wait for 100 ns;

                                end process;

end arch;

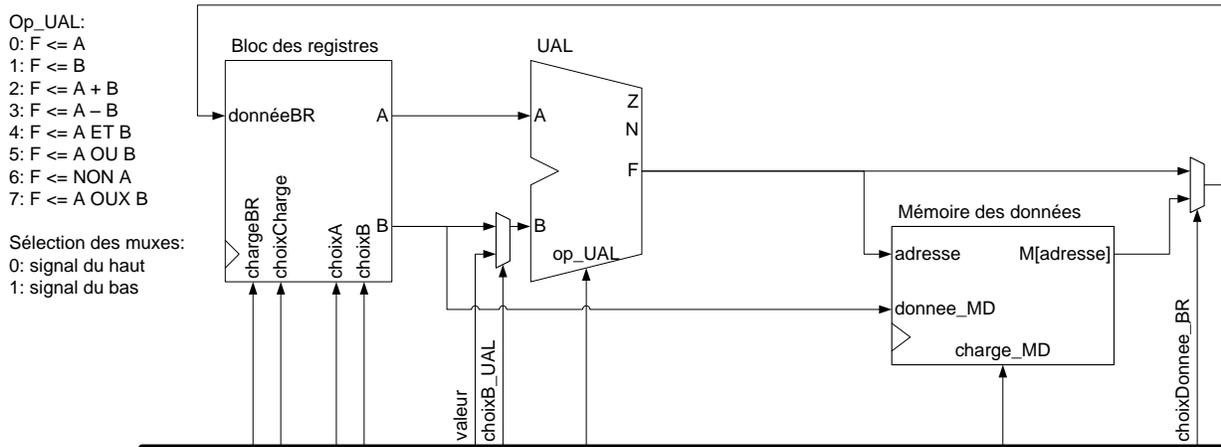
```

Complétez le chronogramme suivant montrant l'évolution de la variable V1 et des signaux S1, S2, S3 et S4 dans le temps.

	0 ns	5 ns	10 ns	15 ns	20 ns	25 ns	30 ns	35 ns	40 ns
V1									
S1									
S2									
S3									
S4									

Question 5. (8 points)

Considérez le chemin des données suivant pour un processeur à usage général.



a. (4 points) Donnez la valeur des signaux de contrôle pour effectuer les opérations suivantes.

opération	chargeBR	choixCharge	choixA	choixB	valeur	ChoixB_UAL	Op_UAL	Charge_MD	choixDonnee_BR
$R7 := R6 \text{ OUX } R7$									
$R14 := 79$									
$M[5 + R4] := R2$									
$R5 := M[31 + R5]$									

b. (2 points) Le chemin des données permet-il d'effectuer l'opération $MD[R0 + R2] := R4$?

Si oui, donnez les valeurs du mot de contrôle comme à la partie a. Si non, dites pourquoi et expliquez comment modifier le chemin des données afin de le pouvoir.

c. (2 points) Le chemin des données permet-il d'effectuer l'opération $R4 := MD[R0 + R2]$?

Si oui, donnez les valeurs du mot de contrôle comme à la partie a. Si non, dites pourquoi et expliquez comment modifier le chemin des données afin de le pouvoir.

Question 6. (10 points) Réponses brèves

a. (2 points) Expliquez l'effet de doubler la précision des calculs, par exemple en les faisant passer de 16 à 32 bits, sur les trois autres considérations d'implémentation d'un système numérique.

b. (2 points) On vous demande d'implémenter, sur un FPGA, un système numérique pour le traitement d'images dans un téléviseur HD. À part ce que vous trouvez à l'intérieur des blocs de logique configurable (CLB), nommez les deux composantes secondaires qui vous seront les plus utiles et dites pourquoi.

c. (2 points) Expliquez pourquoi les FPGA utilisés dans les laboratoires de ce cours doivent être reprogrammés quand le courant est coupé. Nommez deux technologies de programmation qui permettraient de régler ce problème.

d. (2 points) Expliquez deux problèmes qui peuvent survenir quand on interface deux systèmes numériques relativement asynchrones l'un par rapport à l'autre, et proposez une solution à chacun.

e. (2 points) Expliquez dans quels cas la division et le calcul du modulo sont supportés par les outils de synthèse présentement sur le marché, et pourquoi ces opérations ne le sont pas dans les autres cas.

INF3500 : Conception et réalisation de systèmes numériques
 Solutionnaire d'examen final
 Hiver 2014

Q1.

1. Combien de processeurs ?

En logique fixe, la fréquence d'horloge est suffisamment élevée pour qu'un seul processeur suffise.

Pour le FPGA HG, il faut deux processeurs en parallèle pour dépasser 650 Mhz, pour un total de 1050 blocs logiques. Donc une seule puce suffit.

Pour le FPGA BG, il faut 4 processeurs en parallèle pour dépasser 650 MHz, donc 2100 blocs. Il faut donc trois puces.

2. Quels coûts ?

Coûts LF : $\$95K + \$2K = \$97K$

Coûts FPGA HG: $\$50 \times 2K = \$100K$

Coûts FPGA BG : $\$15 \times 3 \times 2K = \$90K$

L'option FPGA bas de gamme est donc moins chère.

Cependant, étant donné qu'il faudra utiliser trois puces, l'option FPGA haut de gamme semble être bien meilleure. Elle serait aussi meilleure par rapport à la logique fixe à cause de la flexibilité qu'elle offre.

Q2.

a. Le chemin critique fait C1R2, C2R0, C0R5, IOB_Q.

Le délai du chemin critique est de $0.75 + 0.6 + 2.25 + 1.4 + 2.25 + 1.4 + 0.25 = 8.9$ ns, pour une fréquence maximale de 112 MHz.

b. Latence de 3 cycles, soit 26.7 ns.

c. Il faudrait faire un placement plus judicieux de façon à réduire le délai sur le chemin critique. On utiliserait la tranche C2R1 au lieu de C0R5, ce qui couperait 6 délais de routage vers C0R5 et 6 délais de routage vers l'IOB de sortie, pour un total de 2.4 ns de réduction.

Le nouveau chemin critique serait plutôt alors B, C0R2, C1R0.

Le délai critique serait $0.75 + 0.4 + 2.25 + 0.6 + 2.25 + 0.25 = 6.5$ ns, pour $f_{\max} = 154$ MHz. La latence serait inchangée.

Q3.

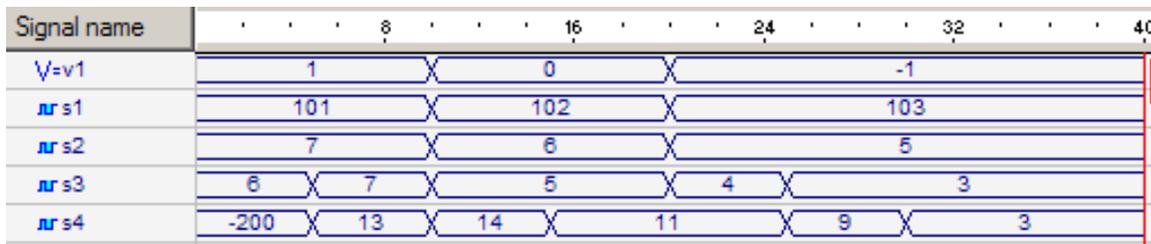
```

architecture arch of compteurSpecial is
begin

    process (clk, reset)
        variable c : unsigned(W - 1 downto 0);
        begin
            if (rising_edge(clk)) then
                if reset = '1' then
                    c := (others => '0');
                else
                    case mode is
                        when "00" =>
                        when "01" =>
                            c := D;
                        when "10" =>
                            c := c + delta;
                        when "11" =>
                            c := c - delta;
                        when others =>
                            c := (others => 'X');
                    end case;
                end if;
            end if;
            compte <= c;
        end process;
    end arch;

```

Q4.



Q5.

a.

opération	chargeBR	choixCharge	choixA	choixB	valeur	ChoixB_UAL	Op_UAL	Charge_MD	choixDon- nee_BR
R7 := R6 OUX R7	1	7	6	7	-	0	7	0	0
R14 := 79	1	14	-	-	79	1	1	0	0
M[5 + R4] := R2	0	-	4	2	5	1	2	1	-
R5 := M[31 + R5]	1	5	5	-	31	1	2	0	1

b. Non, c'est impossible présentement, parce qu'il faut utiliser les deux entrées de l'UAL pour calculer l'adresse, et on ne peut pas aiguiller de valeur à écrire du bloc des registres.

Il faudrait un bloc de registres à trois sorties. Cette sortie serait reliée au port `donnee_MD` de la mémoire. Une autre option serait de faire l'opération en deux cycles : $R3 := R0 + R2$; $MD[R3] := R4$.

c. Oui tout à fait. Les signaux seraient, dans l'ordre de la partie a., (1, 4, 0, 2, -, 0, 2, 0, 1).

Q6.

a. La taille du système et la puissance consommée vont augmenter, probablement d'un facteur de 2 pour les parties directement concernées par les calculs. Le taux de traitement va être réduit si on garde le reste du circuit pareil, c'est-à-dire sans utiliser de techniques de parallélisation ou de pipeline. Le facteur de ralentissement devrait être inférieur à 2 cependant.

b. On utilisera sûrement davantage les blocs de mémoire intégrée et les fonctions arithmétiques avancées. Les blocs de mémoire intégrée pourraient servir de tampons pour les images à traiter. Les fonctions arithmétiques, particulièrement la multiplication et la multiplication-accumulation seraient nécessaires pour le filtrage des images.

c. Le modèle de FPGA utilisé dans le cours est basé sur la technologie SRAM pour la programmation. C'est un type de mémoire volatile – il a besoin de tension électrique. Deux technologies de remplacement seraient les anti-fusibles et la mémoire flash.

d. Comme les deux systèmes ne partagent pas d'horloge, il y a une probabilité non nulle que les signaux transmis ne soient pas stables à l'intérieur de la période définie par t_{su} et t_h des bascules du système qui reçoit. Dans ce cas, les bascules de réception peuvent entrer dans un état métastable, ni 1 ni 0.

Pour réduire les chances que la métastabilité se propage au reste du système, on peut utiliser deux bascules en cascade, menées par l'horloge du système de réception.

Ensuite, les bascules peuvent se stabiliser sur une valeur qui ne correspond pas à la valeur entrée.

Pour réduire les conséquences de cette erreur, il faut gérer les échanges d'information avec un protocole de communication qui en prend compte.

e. La division et le modulo sont supportés quand le deuxième argument est une puissance de deux. Le résultat s'obtient alors par un décalage ou une sélection de bits, respectivement.

La division générale et le modulo général ne sont pas présentement synthétisable parce qu'il n'y a pas de bloc dédié à cette opération dans la plupart des technologies, dont les FPGA. Ces opérations peuvent être réalisées de différentes façons, par exemple par la multiplication par la réciproque du diviseur et par des circuits itératifs. Dans les deux cas, il y a beaucoup de choix de design à faire, incluant la complexité matérielle et la latence du circuit résultant. Les concepteurs de synthétiseurs préfèrent laisser ces choix à celles et ceux qui écrivent du code VHDL.