

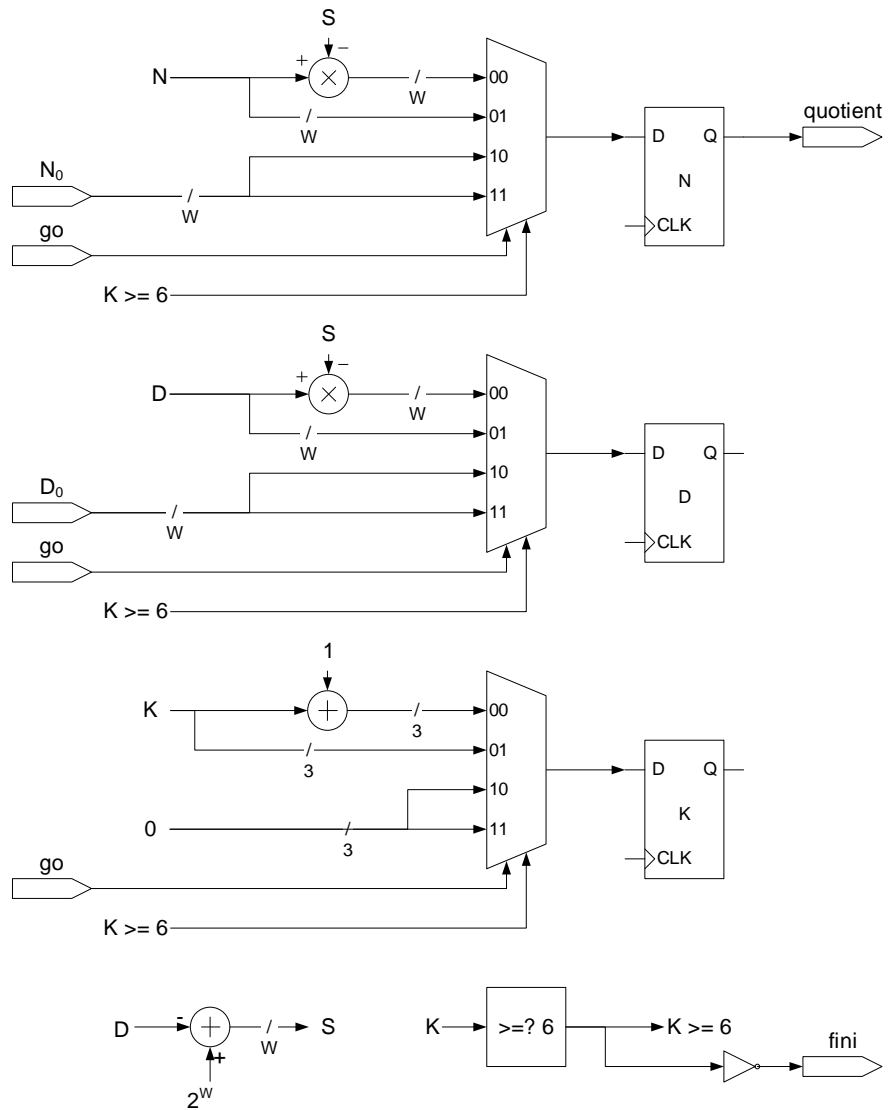
INF3500 : Conception et réalisation de systèmes numériques
 Examen intra #2 – 5 novembre 2020

Examen dans Moodle

Q1 conditions et déclaration

Q2. Division Goldschmidt

Considérez le diagramme du chemin des données suivant. Complétez l'architecture du code VHDL inclus avec cette question pour qu'elle corresponde à ce diagramme. Pour les multiplications, une normalisation est nécessaire. Utilisez donc la fonction qui est donnée dans la déclaration de l'architecture et non l'opérateur de multiplication (*) de VHDL. Commentez bien votre code; ajouter des annotations pour expliquer vos choix. Il n'est pas nécessaire de simuler votre code, ce qui compte c'est que le code corresponde bien au diagramme.



```

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

entity division_goldschmidt_2 is
    generic (W : positive := 10);
    port (
        clk, go : in std_logic;
        N0 : in unsigned(W - 1 downto 0);
        D0 : in unsigned(W - 1 downto 0);
        quotient : out unsigned(W - 1 downto 0);
        fini : out std_logic
    );
end;

architecture arch of division_goldschmidt_2 is

function mult(a, b : unsigned((W - 1) downto 0)) return unsigned is
begin
    return resize(a * b / 2 ** (W - 1), a'length);
end;
-- vos déclarations ici

begin
-- votre code ici
end;

```

Q3. et Q4 Division Goldschmidt, ressources

Considérez le diagramme de la question précédente. Estimez le nombre de ressources nécessaires en termes de tables de conversion (LUT), bascules (FF) et tranches DSP48 pour l'implémenter sur un FPGA de la série 7 de Xilinx. Supposez que $W = 16$ bits.

Arrondissez la quantité de ressources à l'entier immédiatement supérieur à tout nombre fractionnaire. Par exemple, pour 37.25 LUT, entrez la valeur 38. Ne tapez que des chiffres, pas d'autres caractères ni texte.

Justifiez complètement votre réponse.

Q5

Considérez le code VHDL suivant. Montrez comment l'implémenter sur le modèle de FPGA attaché à cette question. Une version .svg et une version .pdf sont également disponibles.

Vous pouvez éditer directement le diagramme en version électronique, ou reproduire uniquement les parties nécessaires sur papier et en remettre une photo. Indiquez clairement où chaque signal et port se situe, montrez bien quelles ressources vous utilisez et indiquez les fonctions logiques réalisées.

```

library ieee;

use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity compteur_deux_bits is
  port (
    clk, reset, go : in std_logic;
    c_out : out unsigned(1 downto 0);
    deux, trois : out std_logic
  );
end compteur_deux_bits;

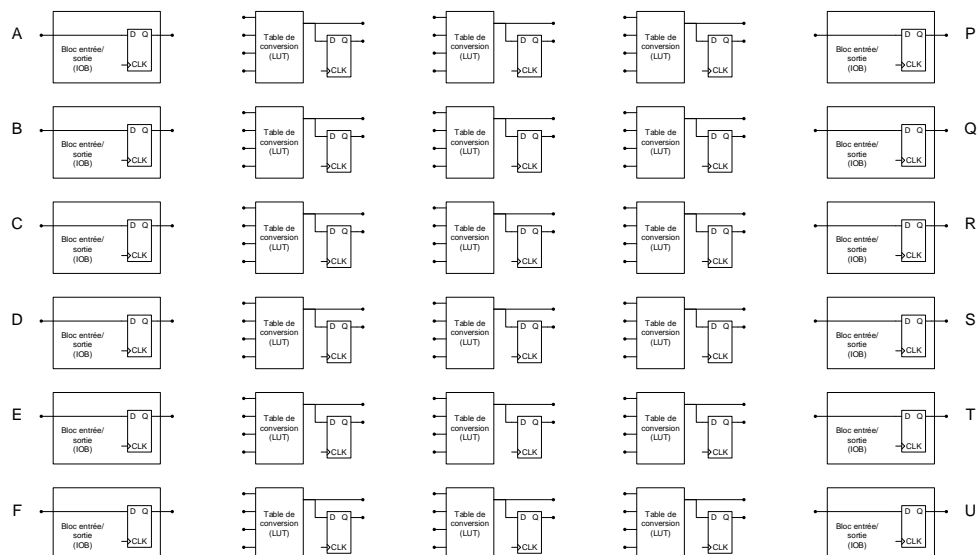
architecture arch of compteur_deux_bits is
  signal etat : unsigned(1 downto 0);
begin

  process(clk, reset) is
  begin
    if reset = '1' then
      etat <= to_unsigned(0, etat'length);
    elsif rising_edge(CLK) then
      etat(0) <= (not(go) and etat(0)) or (go and not(etat(0))) ;
      etat(1) <= (not(go) and etat(1))
                 or (etat(1) and not(etat(0)))
                 or (go and not(etat(1)) and etat(0)) ;
    end if;
  end process;

  c_out <= etat;
  deux <= etat(1) and not(etat(0));
  trois <= etat(1) and etat(0);

end arch;

```



Q6. D'accord ou pas d'accord ?

Dans le banc d'essai du laboratoire #3, quand on utilise des énoncés assert pour vérifier la sécurité du contrôleur de feux de circulation, en confirmant par exemple que des feux verts ne sont pas allumés dans deux directions différentes, il s'agit d'un test de boîte blanche puisqu'il faut comprendre le fonctionnement interne du contrôleur pour composer ce genre de test.

Donnez votre opinion et justifiez-la complètement. Votre réponse sera évaluée selon sa validité et la justesse, la pertinence et la qualité de vos arguments.

Q7. D'accord ou pas d'accord ?

On veut faire la vérification d'un microprocesseur spécialisé pour implémenter des algorithmes d'intelligence artificielle pour détecter des panneaux de circulation dans des images prises par une caméra frontale montée sur une voiture autonome. Le processeur contient 64 unités parallèles, chacune avec un multiplieur, un additionneur et un accumulateur de 32 bits. Chaque unité parallèle a aussi une mémoire temporaire de 128 mots de 32 bits. Le processeur a aussi une unité de contrôle qui contient 4 registres de 8 bits, et un compteur de programme de 32 bits. Pour un système d'une telle complexité, il est illusoire de penser à faire des tests de boîte noire puisqu'il faudrait beaucoup trop de vecteurs de test et qu'on ne pourrait pas les compléter dans un temps raisonnable.

Donnez votre opinion et justifiez-la complètement. Votre réponse sera évaluée selon sa validité et la justesse, la pertinence et la qualité de vos arguments.

Solutions

Q2.

```

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

entity division_goldschmidt_2 is
  generic (W : positive := 10);
  port (
    clk, go : in std_logic;
    N0 : in unsigned(W - 1 downto 0);
    D0 : in unsigned(W - 1 downto 0);
    quotient : out unsigned(W - 1 downto 0);
    fini, erreur : out std_logic
  );
end;

architecture arch of division_goldschmidt_2 is

function mult(a, b : unsigned((W - 1) downto 0)) return unsigned is
begin
  return resize(a * b / 2 ** (W - 1), a'length);
end;

signal N, D, S : unsigned(W - 1 downto 0);
signal K : unsigned(2 downto 0);

begin

  S <= 2 ** W - D;
  fini <= '1' when K >= 6 else '0';
  quotient <= N;

  process(all)
  begin
    if rising_edge(clk) then
      if go = '1' then
        N <= N0;
        D <= D0;
        K <= to_unsigned(0, K'length);
      elsif K < 6 then
        N <= mult(N, S);
        D <= mult(D, S);
        K <= K + 1;
      end if;
    end if;
  end process;
end;

```

Q3 réponse

LUT : 16 (mux N) + 16 (mux D) + 3 (mux K) + 16 (soustraction) + 1 (comparaison de K) + 1 (non) + 3 (incrément de K) = 56

FF : 16 + 16 + 3 = 35

DSP48 : 2

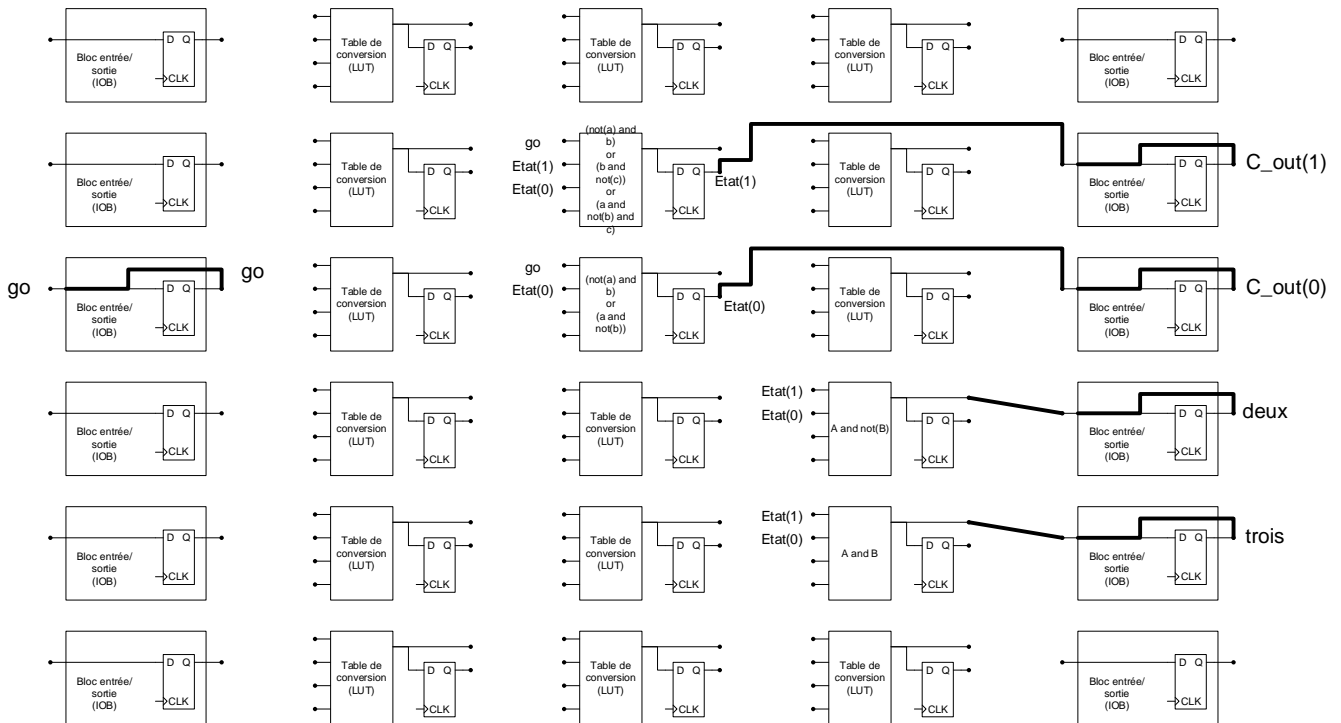
Q5

Voici une solution possible, il y en a plusieurs.

Les points importants sont :

- avoir deux bascules pour etat(1) et etat(0), dont les fonctions sont calculées chacune par une LUT;
- avoir deux sorties distinctes c_out(1) et c_out(0), reliées respectivement aux bascules qui entreposent les états etat(1) et etat(0);
- avoir deux sorties 'deux' et 'trois', reliées à des LUT sans passer par des bascules

** La solution donnée ici montre un contournement des bascules d'entrée et de sortie, ce qui est possible en pratique dans un FPGA mais qui n'était pas donné en option dans le diagramme. Il n'était donc pas possible d'implémenter exactement le code VHDL de la question. Ce fait a été tenu en compte dans la correction. **



Q6

Plutôt en désaccord.

Au contraire, le test qui est décrit ne suppose aucune connaissance du fonctionnement interne du contrôleur. Il faut uniquement connaître ses spécifications, par exemple que deux feux verts ne doivent pas être allumés en même temps. On n'a pas à connaître l'état du contrôleur ni comment il réagit aux entrées actuelles, ni les conditions sur lesquelles il se base pour déterminer l'affichage en cours des différents feux. On n'a pas à connaître aucun détail de l'implémentation du code. On ne regarde que les sorties du système par rapport au fonctionnement attendu. Il s'agit donc vraiment d'un test de boîte noire.

Q7

Plutôt en désaccord.

On peut très bien faire des tests de boîte noire basés sur les spécifications du microprocesseur. On n'a pas à faire nécessairement un test exhaustif, qui prendrait beaucoup trop de temps. On pourrait commencer par tester les fonctionnalités de base du processeur avec un programme simple. On pourrait faire des tests de valeurs limites, de partitionnement en classe ou encore des tests aléatoires. Finalement, on pourrait passer des images de test au processeur et comparer sa sortie avec une sortie de référence qu'on aurait calculée indépendamment à partir de la spécification du processeur.

On pourrait penser à faire des tests de boîte noire sur chacune des unités parallèles, en parallèle. Mais alors, ce ne seraient plus vraiment des tests de boîte noire, puisqu'on exploiterait nos connaissances de l'architecture du processeur.

On pourrait aussi faire des tests en boîte blanche puisqu'on a certaines connaissances de l'architecture du processeur, mais ce n'est pas le sujet de la question.