

# M1 Info – ARC - LAB2 Extension

Olivier HUREAU - Groupe 3

18/03/2020

## 1 Testbench : check frequency.

Soit une horloge avec une fréquence de 10MHZ. Alors il faut que l'horloge revienne à 1 toute les  $\frac{1}{10MHz} s$   
Soit  $\frac{1}{10^7 Hz} s = 10^{-7} = 100nanosecondes$   
La bonne implémentation est alors :

```
begin
clk <= '0';
wait for 50 ns;
clk <= '1';
wait for 50 ns;
end process;
```

## 2 Counter : the form of the description is not correct.

En reprennant la slide 125 il faut donc trois process pour modéliser le counter.

### 2.1 Calcul de l'état suivant

D'après le slide, ce process doit calculer le prochain état.

#### 2.1.1 Le code

Pour rappel le code est :

```
process (state, start, c)
begin
  case state is
    when IDLE =>
      if start = '1' then
        nextstate <= COUNTING;
      elsif start = '0' then
        nextstate <= IDLE;
      end if;
    when COUNTING =>
      if c < threshold then
        nextstate <= COUNTING;
      else
        nextstate <= IDLE;
      end if;
    end case;
  end process;
```

Figure 1: Process #1

#### 2.1.2 Vérification du code

**Sensitivity list** Tout les éléments pris en compte pour le calcul de l'état suivant sont : State, start et c. Alors la liste doit être (state, start,c).

**Calcul des états** Si l'on regarde le dessin de l'automate fournis :

- Dans l'état IDLE on reste dans cet état si start vaut '0', sinon si start vaut '1' on va dans l'état COUNTING. Ce qui est exprimé dans ce code
- Dans l'état COUNTING on reste dans cet état si c est j à threshold, sinon on va dans l'état IDLE. Ce qui est aussi exprimé dans ce code.

## 2.2 Mise à jour de l'état

D'après les slide, ce process exprime la mise à jour de la valeur du registre courant en synchronisation avec l'horloge. Ce process peut aussi prendre en considération le signal "reset".

### 2.2.1 Le code

Après avoir modifier le code tel que :

<pre>-- MISE A JOUR DU REGISTRE D'ETAT process(reset, clk, start) begin     -- RESET : asynchrone haut     if reset = '1' then         state &lt;= IDLE;     -- HORLOGE : front montant     elsif (clk'event and clk = '1') then         state &lt;= nextstate;         if ( (start = '1') and (state = IDLE) )then             state &lt;= COUNTING;         end if;     end if; end process;</pre>	<pre>35 -- MISE A JOUR DU REGISTRE D'ETAT 36 process(reset, clk, start) 37 begin 38     -- RESET : asynchrone haut 39     if reset = '1' then 40         state &lt;= IDLE; 41     -- HORLOGE : front montant 42     elsif (clk'event and clk = '1') then 43         state &lt;= nextstate; 44 +         -- Detecter un pic sur start 45 +         elsif ( (start = '1') and (state = IDLE) )then 46             state &lt;= COUNTING; 47 +         end if; 48     end if; 49 + 50 + 51 end process;</pre>
--	---

Figure 2: Process #2 Avant/après

### 2.2.2 Vérification du code

**Sensitivity list** Tout les éléments pris en compte pour le calcul de l'état suivant sont : reset clock et start. Je prend en compte ici start car je veux que la moindre impulsion sur le signal start puisse déclencher le compte à rebours et que pas uniquement lorsque l'horloge est sur un front montant. Je suis donc obligé de l'écrire ici. Néanmoins, j'avais fais une erreur d'imbrication de if/elsif précédement.

**Reset** Il y a bien une mise à de l'état courant si le reset est à sur un 1 (asynchronous reset HIGH)

**Front montant** L'état se met bien à jour si l'on se trouve sur un front montant de l'horloge.

## 2.3 Mise à jour de du compteur

Il n'y a pas d'indication dans le cours pour implémenter le counter mais d'après moi il faut un process qui s'en occupe.

### 2.3.1 Le code

Après avoir modifier le code tel que :

<pre>- process(start, clk, c, reset) begin     if(reset = '1') then         c &lt;= 0;     else         if (clk'event and clk = '1') then             if (state = IDLE and start = '0') then                 c &lt;= 0;             elsif ( state = IDLE and start = '1') then                 c &lt;= c + 1;             elsif (state = COUNTING and c &lt; threshold) then                 c &lt;= c + 1;             elsif (state = COUNTING and c &gt;= threshold) then                 c &lt;= 0;             end if;         end if;     end if; end process;</pre>	<pre>54 + process(start, clk, reset) 55 begin 56     if(reset = '1') then 57         c &lt;= 0; 58     else 59         if (clk'event and clk = '1') then 60             if (state = IDLE and start = '0') then 61                 c &lt;= 0; 62 +             c &lt;= 0; 63             elsif ( state = IDLE and start = '1') then 64                 c &lt;= c + 1; 65             elsif (state = COUNTING and c &lt; threshold) then 66                 c &lt;= c + 1; 67             elsif (state = COUNTING and c &gt;= threshold) then 68                 c &lt;= 0; 69             end if; 70         end if; 71     end if; 72 end process;</pre>
---	--

Figure 3: Process #3 Avant/après

J'avais "juste" oublié une ligne de code et un signal en trop dans la liste de sensibilité. C'est comme ça qu'une fusée peut exploser...

### 2.3.2 Vérification du code

**Sensitivity list** La valeur de c peut changer si et seulement si soit start, clk et reset d'où la list (start, clk, reset).

**Transitions** J'ai retranscrit les transitions de l'automate tel que les calculs de c ne se fasse uniquement une fois par cycle.

## 2.4 Exprime les fonctions de sorties

D'après les slides il faut un process qui exprime les fonctions de sorties (ou un ensemble de concurrent statement)

### 2.4.1 Le code

J'avais exprimé cela sous la forme :

```
aboveth <= '0' when c < threshold else '1';
```

car je n'ai pas réussi à trouver la comparaison booléenne exprimant  $\geq$   
Pour être sur de moi, j'ai transformé le tout en un process :

aboveth <= '0' when c < threshold else '1';	75	+	
	76	+	process(c)
	77	+	begin
	78	+	if (c >= threshold) then
	79	+	aboveth <= '1';
	80	+	else aboveth <= '0';
	81	+	end if;
	82	+	end process;

Figure 4: Process #4 Avant/après

### 2.4.2 Vérification du code

Je pense pas qu'il n'y ai vraiment grand chose à commenter.

## 3 Improve testbench for the system.

Il faut déjà changé le testbench avec la vrai valeur de l'horloge, je n'ai pas vraiment le temps de le faire...