



# C5 – Délais, Delta Délais et variables

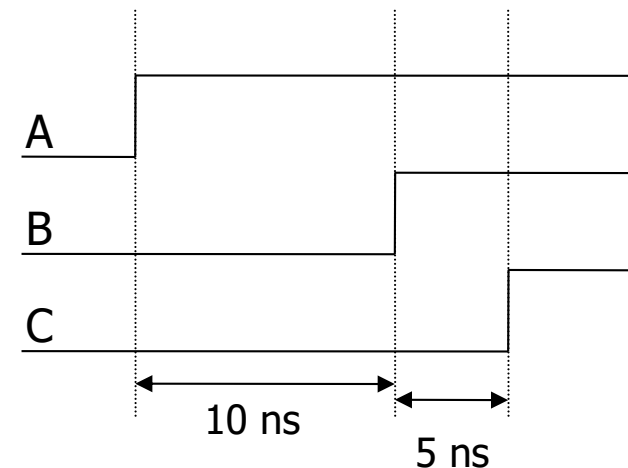
---

Yann DOUZE

# Les délais

chronogramme de A ,B et C ?

```
process (A,B)
begin
  B <= A after 10 ns;
  C <= B after 5 ns;
end process;
```

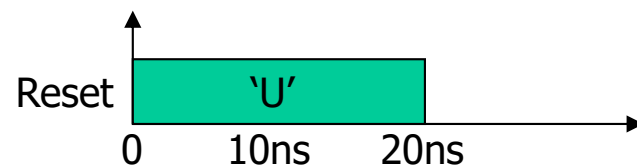


- Les délais sont utilisés pour :
  - Générer des stimuli pendant la simulation
  - Modéliser les retards dû à la technologie
- Les délais sont ignorés à la synthèse.

# Génération d'une impulsion (1)

```
process
begin
    reset <= '0';
    reset <= '1' after 10 ns;
    reset <= '0' after 20 ns;
    wait;
end process;
```

Mauvaise syntaxe !

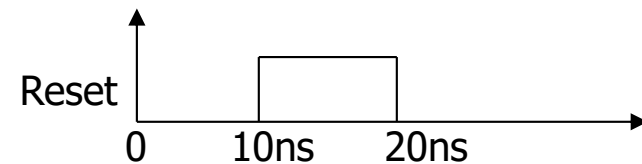


## Génération d'une impulsion (2)

```
Reset <= '0', '1' after 10 ns, '0' after 20 ns;
```

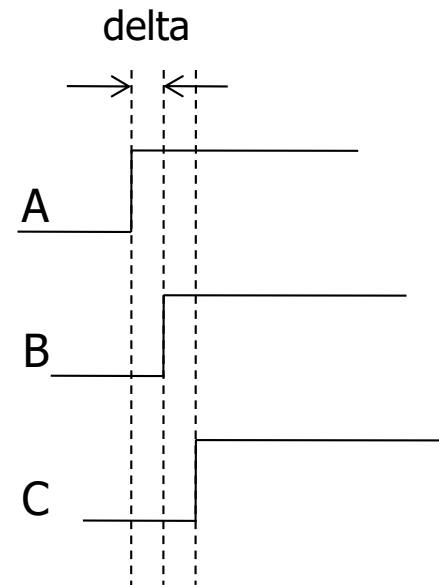
```
process  
begin  
    reset <= '0';  
    wait for 10 ns;  
    reset <= '1';  
    wait for 10 ns;  
    reset <= '0';  
    wait;  
end process;
```

Bonne syntaxe !



# Delta Délais

```
process (A,B)
begin
    B <= A;
    C <= B;
end process;
```

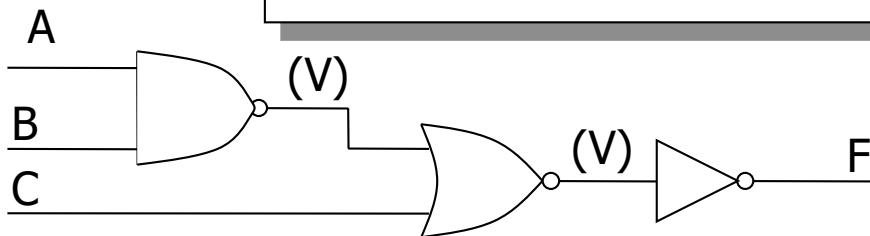


- En jargon VHDL, un délai delta = délai infinitésimal non nul
- Un signal prend sa nouvelle valeur après un délai delta.
- Le delta délais représente le temps de propagation
- Une variable prend sa nouvelle valeur immédiatement.

# Les Variables

- Les variables ne peuvent être déclarées et n'existent que dans un process.
- **L'affectation d'une variable est immédiate** : la valeur affectée à V à la première ligne peut directement être réutilisée à la deuxième.

```
process (A, B, C)
-- zone de déclaration d'une variable
  variable V: STD_LOGIC;
begin
  V := A nand B;
  V := V nor C;
  F <= not V;
end process;
```



Redessinez le schéma si on considère V comme un signal ?



# Exemple : parité impaire

```
Entity parite is
PORT ( a    : IN std_logic_vector(0 TO 3) ;
      s    : OUT std_logic );
END entity;
architecture behaviour of parite is
begin
    process(a)
        variable parite : std_logic ;
    begin
        parite := '1' ;
        FOR i in 0 to 3 LOOP
            if a(i) = '1' then
                parite := not parite;
            end if;
        END LOOP;
        s <= parite;
    end process;
END architecture;
```



# Exercice

```
process (A, S)
    variable V: STD_LOGIC;
begin
    V := A;
    S <= V;
    V := S;
    T <= V;
end process;
```

- Supposer que S et T vallent '0', et A change de l'état '0' à l'état '1'.





# Questions

---

1. Qu'elle est la valeur de S à la fin du process, avant le delta délai ?
2. Qu'elle est la valeur de V à la fin du process, avant le delta délai ?
3. Après l'exécution du process et après le delta délai, S et T prennent leurs nouvelles valeurs, que valent S et T ?
4. Après la première exécution du process et après le delta délai, que se passe-t-il ?

