

---

# Banc d'essai pour un circuit séquentiel



Pierre Langlois

<http://creativecommons.org/licenses/by-nc-sa/2.5/ca/>

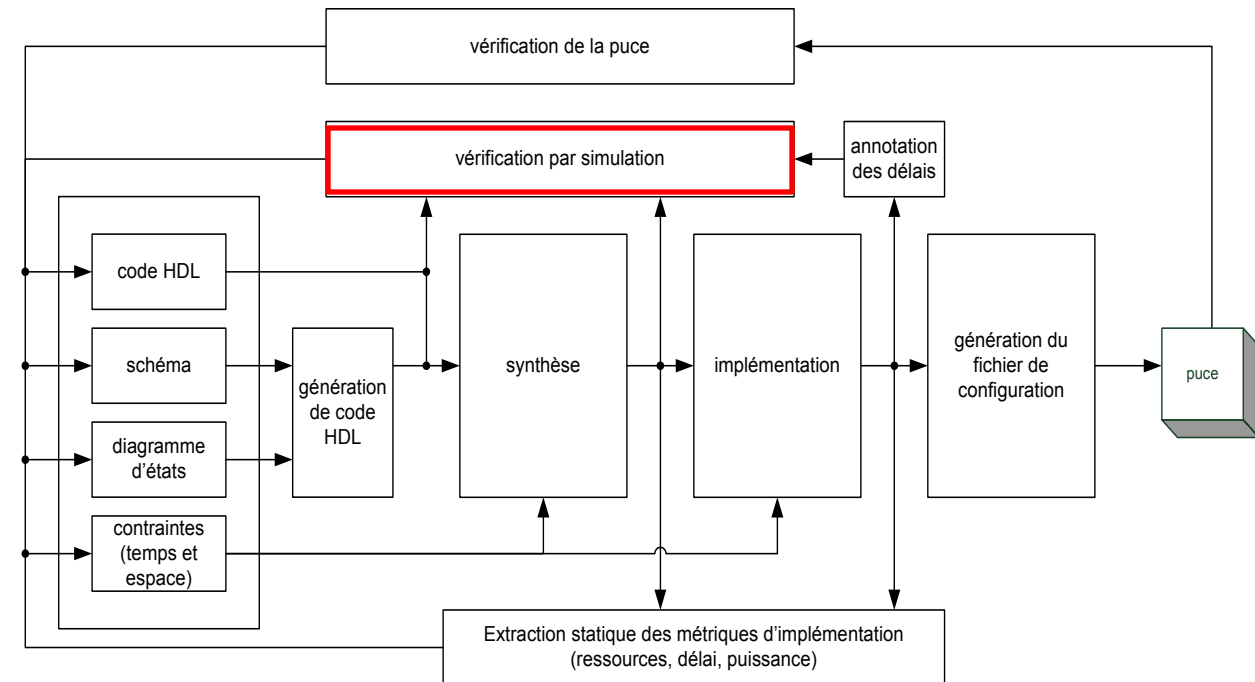
# Sujets de ce thème

---

- Structure d'un banc d'essai pour un circuit séquentiel.
- Générer un signal d'horloge et un signal de réinitialisation.
- Stimuler un circuit séquentiel avec un banc d'essai.

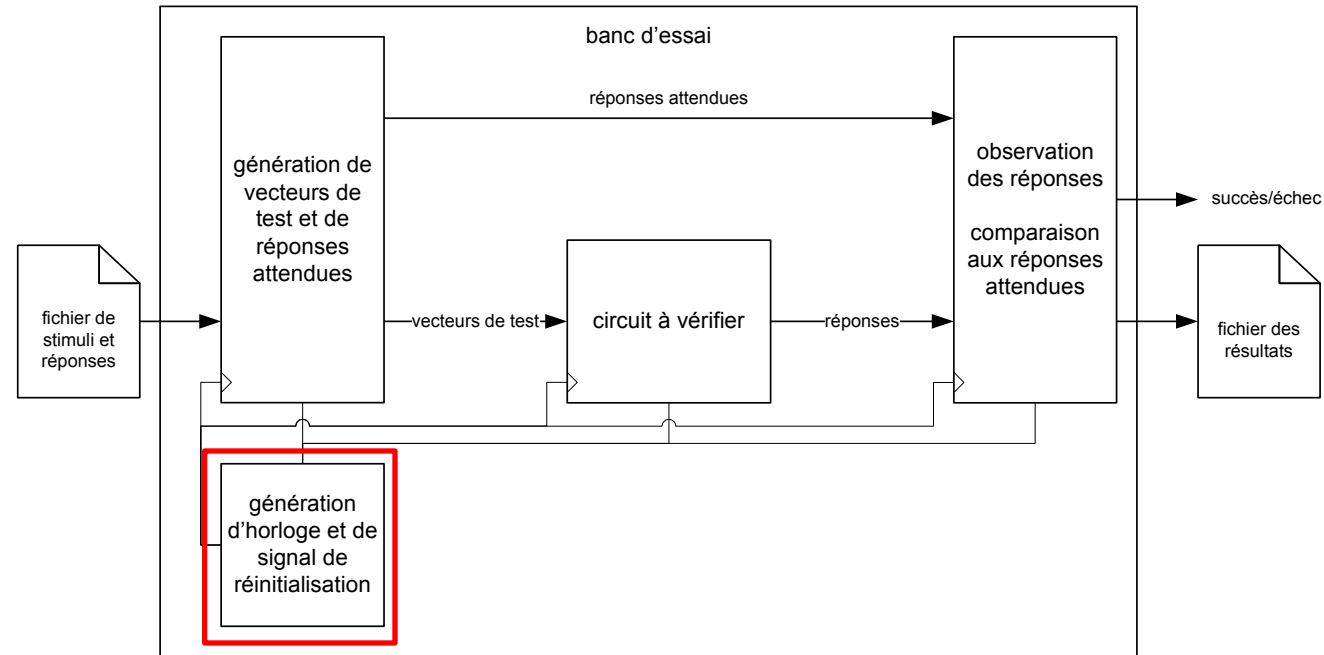
# La vérification d'un circuit

- La vérification a pour but de confirmer qu'un circuit rencontre bien ses spécifications.
- La vérification complète d'un circuit est normalement un problème très difficile.
- Dans l'industrie de la conception numérique, on considère en général que le processus de vérification nécessite autant d'efforts que le processus de conception lui-même.
- La vérification d'un circuit est un art qui repose sur la maîtrise de trois principes:
  - la compréhension de la spécification;
  - le contrôle des entrées et de signaux internes du circuit à vérifier; et,
  - l'observation des sorties, des signaux internes et de l'état du circuit à vérifier.



# Vérification par banc d'essai

- Un banc d'essai doit effectuer les tâches suivantes :
  - instancier le circuit à vérifier;
  - générer des vecteurs de test et les appliquer aux ports d'entrée du circuit;
  - [utile]: générer automatiquement des réponses attendues aux vecteurs de test;
  - [utile]: comparer les réponses du circuit aux réponses attendues, et indiquer toute différence entre les deux par une condition d'erreur;
  - [facultatif]: lire des stimuli d'un fichier et écrire les réponses dans un fichier.
  - + (pour circuit séquentiel): générer un signal d'horloge et un signal de réinitialisation



# Vérification par banc d'essai

- La génération d'un signal d'horloge périodique et d'un signal de réinitialisation peut être faite de façon très efficace par deux énoncés concurrents dans l'architecture du banc d'essai.
- L'horloge se voit assigner une valeur initiale dans sa déclaration, puis dans l'architecture elle bascule entre deux états à chaque demi-période.

```
-- partie du banc d'essai

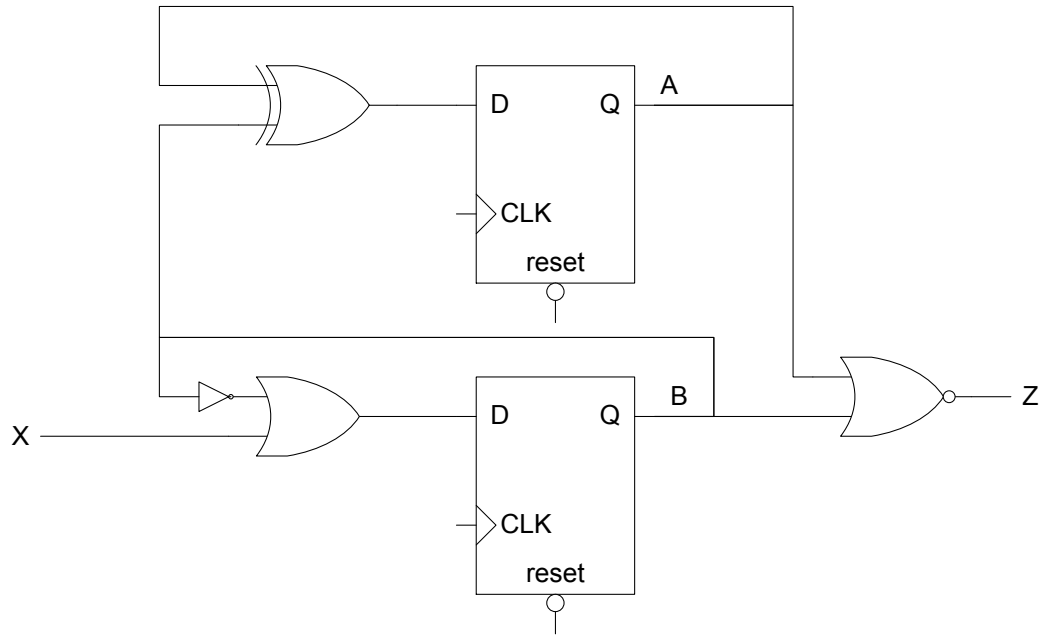
-- dans la partie déclarative de l'architecture
signal clk : std_logic := '0';
constant periode : time := 10 ns;

...

-- dans l'architecture
clk <= not clk after periode / 2;

reset <=
    '0' after 0 ns,
    '1' after periode / 4;
```

# Circuit séquentiel synchrone à vérifier



```
library IEEE;
use IEEE.std_logic_1164.all;

entity cctsequentielex1 is
    port (
        reset, CLK, X : in STD_LOGIC;
        Z : out STD_LOGIC
    );
end cctsequentielex1;

architecture arch1 of cctsequentielex1 is
    signal A, B : STD_LOGIC;
begin

    process(CLK, reset) is
    begin
        if (reset = '0') then
            A <= '0';
            B <= '0';
        elsif (rising_edge(CLK)) then
            A <= A xor B;
            B <= x or not(B);
        end if;
    end process;

    z <= not(A or B);

end arch1;
```

# Banc d'essai pour la vérification d'un circuit séquentiel

- Le processus est contrôlé par le signal d'horloge dans sa liste de sensibilité: pas d'énoncé `wait`.
- Les vecteurs de test sont encodés dans un tableau de constantes.
- Pas de vérification automatisée des réponses dans cet exemple.

```
architecture arch2 of cctsequentielex1TB is

signal reset, clk : std_logic := '0';
signal X, Z : std_logic;
constant periode : time := 10 ns;

begin

    clk <= not clk after periode / 2;

    reset <=
        '0' after 0 ns,
        '1' after 5 * periode / 4;

    -- instantiation du module à vérifier
    UUT : entity cctsequentielex1 (arch1) port map (reset, clk, X, Z);

    process (clk)
    constant vecteurX : std_logic_vector := "101010101";
    variable compte : natural range 0 to vecteurX'length := 0;
    begin
        if (rising_edge(clk)) then
            X <= vecteurX(compte);
            compte := compte + 1;
            if compte = vecteurX'length then
                report "simulation terminée" severity failure;
            end if;
        end if;
    end process;

end arch2;
```

# Vous devriez maintenant être capable de ...

---

- Utiliser la structure d'un banc d'essai pour stimuler un circuit séquentiel synchrone. (B3)
- Utiliser la modélisation des signaux d'horloge et de réinitialisation dans un banc d'essai. (B3)

Code	Niveau ( <a href="http://fr.wikipedia.org/wiki/Taxonomie_de_Bloom">http://fr.wikipedia.org/wiki/Taxonomie_de_Bloom</a> )
B1	Connaissance - mémoriser de l'information.
B2	Compréhension – interpréter l'information.
B3	Application – confronter les connaissances à des cas pratiques simples.
B4	Analyse – décomposer un problème, cas pratiques plus complexes.
B5	Synthèse – expression personnelle, cas pratiques plus complexes.