

Architetture e reti logiche

Esercitazioni VHDL

a.a. 2006/07

Testbench

Stefano Ferrari



UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI TECNOLOGIE DELL'INFORMAZIONE

Testbench

test bench = banco di prova

Per verificare che un componente di comporti secondo le specifiche di progetto, lo si sottopone ad una prova:

- si forzano in ingresso stimoli controllati;
- si verifica che le uscite corrispondano.

La descrizione degli stimoli e delle corrispondenti uscite arricchisce la documentazione del progetto.

Testbench (2)

Questo procedimento in VHDL è realizzabile mediante un simulatore VHDL.

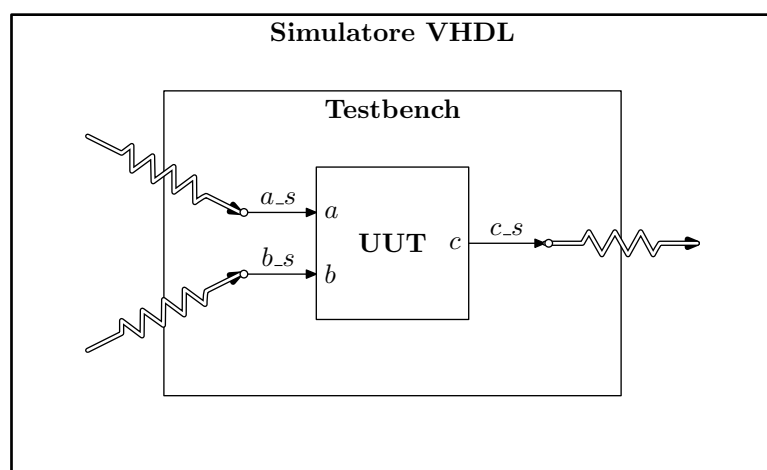
Bisogna predisporre una opportuna entità che descriva il test da eseguire:

- l'unità di **testbench**.

L'unità di testbench dovrà:

- inglobare l'entità sottoposta a verifica (**unit under test**, UUT);
- contenere una descrizione degli stimoli da applicare in ingresso alla *ut*.

Rappresentazione schematica



Caratteristiche di un testbench VHDL

- Assenza di porte di ingresso o di uscita.
- I segnali di ingresso possono essere fatti variare nel tempo, in modo da coprire più configurazioni.
- Se le combinazioni dei valori dei segnali di ingresso sono troppo numerose, il test viene limitato alle configurazioni più significative.
- I segnali possono essere descritti utilizzando istruzioni concorrenti (assegnamento con clausola `after`) o sequenziali (costrutti `wait` e `wait for` in istruzioni `process`).

Codifica VHDL

```

entity Testbench is
end Testbench;

architecture behav of Testbench is

    component UUT -- interfaccia dell'unita' sotto test
        port ( a, b: in bit;
              c   : out bit);
    end component;

    signal a_s, b_s, c_s: bit;

begin

```

(continua)

Codifica VHDL (2)

begin

(continua)

```
-- inclusione dell'unita' sotto test
test_unit: UUT port map (a_s, b_s, c_s);

-- descrizione dei segnali di stimolo
a_p: process -- andamento del segnale a_s
begin
    a_s <= '0';
    wait for 10 ns;
    a_s <= '1';
    wait for 10 ns;
end process;
```

(continua)

Codifica VHDL (3)

(continua)

```
b_p: process -- andamento del segnale b_s
begin
    b_s <= '0';
    wait for 20 ns;
    b_s <= '1';
    wait for 20 ns;
end process;
end behav;
```