Academie voor Technology, Innovation & Society Delft
Academie voor ICT & Media

Digitale Systeem Engineering 2

Week 3 – Testbenches (Part II)

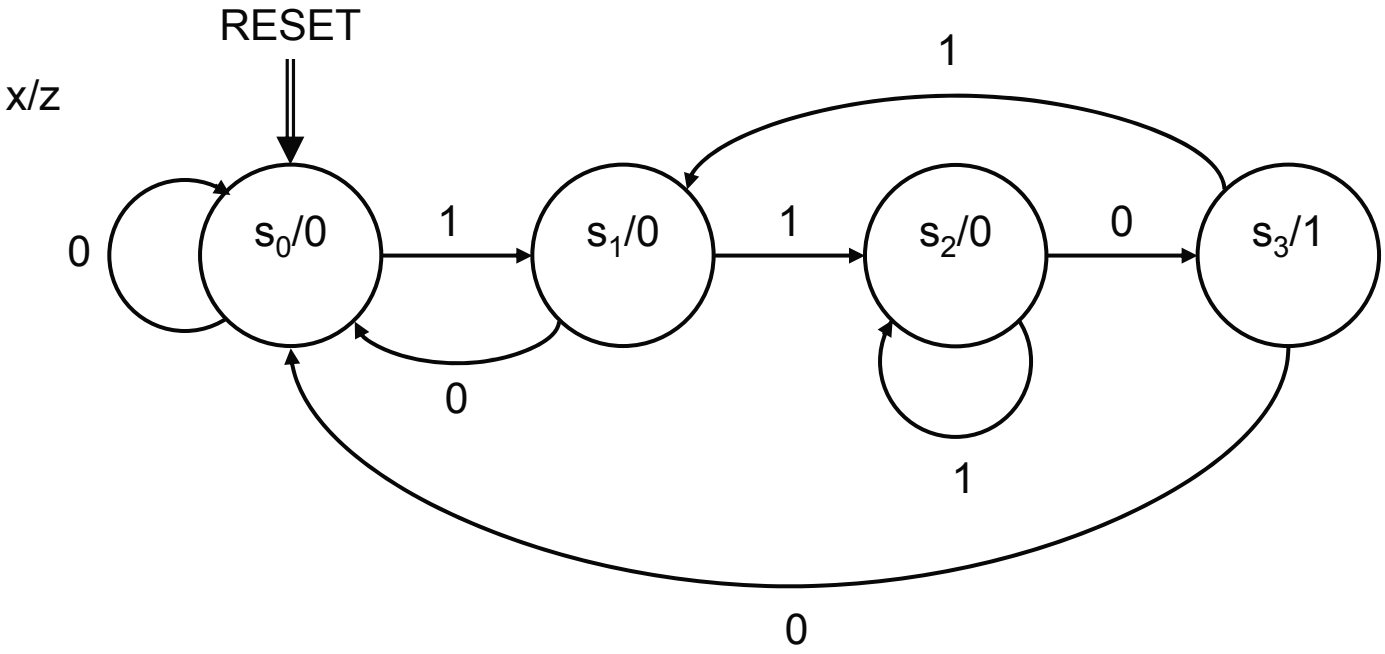
Jesse op den Brouw

DIGSE2/2016-2017

DE HAAGSE HOGESCHOOL

Herkenningsautomaat

- Ontwerp een machine voor het doorlopend herkennen van het patroon (of reeks) 110. De machine moet bij herkenning een 1 afgeven.



Herkenningsautomaat VHDL (1)

- De entity, architecture en toestandstype.

```
library ieee;
use ieee.std_logic_1164.all;

entity herk110 is
  port (clk    : in std_logic;
        areset : in std_logic;
        x      : in std_logic;
        z      : out std_logic);
end entity herk110;

architecture fsm of herk110 is
type state_type is (s0, s1, s2, s3);
signal current_state, next_state : state_type;

begin
```

Herkenningsautomaat VHDL (2)

- De combinatoriek voor zowel NSL als OL.

```
comb: process (current_state, x) is
begin
  z <= '0';           -- default output
  next_state <= s0;   -- default state
  case current_state is
    when s0 => if x = '1' then next_state <= s1; end if;
    when s1 => if x = '1' then next_state <= s2; end if;
    when s2 => if x = '0' then next_state <= s3;
                 else next_state <= s2; end if;
    when s3 => if x = '1' then next_state <= s1; end if;
               z <= '1';  -- bingo!
    when others => null;
  end case;
end process comb;
```

Herkenningsautomaat VHDL (3)

- Het toestandsregister met asynchrone reset.

```
  reg: process (clk, areset, next_state) is
  begin
    if areset = '1' then
      current_state <= s0;
    elsif rising_edge(clk) then
      current_state <= next_state;
    end if;
  end process reg;

end fsm;
```

Testbench herkenningsautomaat (1)

```
library ieee;
use ieee.std_logic_1164.all;

entity tb_herk110 is
end tb_herk110;

architecture testbench of tb_herk110 iS
constant Tperiod : time := 10 ns;
constant small_delay : time := 2 ns;
signal sim_clk : std_logic;      -- signals to trace
signal sim_areset : std_logic;
signal sim_x : std_logic;
signal sim_z : std_logic;

component herk110
  port (clk    : in std_logic;
        areset : in std_logic;
        x      : in std_logic;
        z      : out std_logic);
end component;
```

DE HAAGSE HOGESCHOOL

Testbench herkenningsautomaat (2)

```
begin
  dut : herk110
  port map (clk => sim_clk, areset => sim_areset,
            x => sim_x, z => sim_z);

  clockgen: process is  -- no sensitivity list
  begin
    sim_clk <= '0';     -- clock has 50% Duty Cyle
    wait for Tperiod/2;
    sim_clk <= '1';
    wait for Tperiod/2;
  end process clockgen;
```

Testbench herkenningsautomaat (3)

```
datagen: process is                -- no sensitivity list

begin
  sim_x <= '0';
  sim_areset <= '1';
  wait for 2*Tperiod;              -- wait for two clock cycles

  sim_areset <= '0';
  wait for 2*Tperiod;
  wait until sim_clk = '1';        -- synchronize on positive edge
  wait for small_delay;            -- and wait a bit

  sim_x <= '0';
  wait until sim_clk = '1';
  wait for small_delay;

  sim_x <= '1';
  wait until sim_clk = '1';
  wait for small_delay;
  ...
```

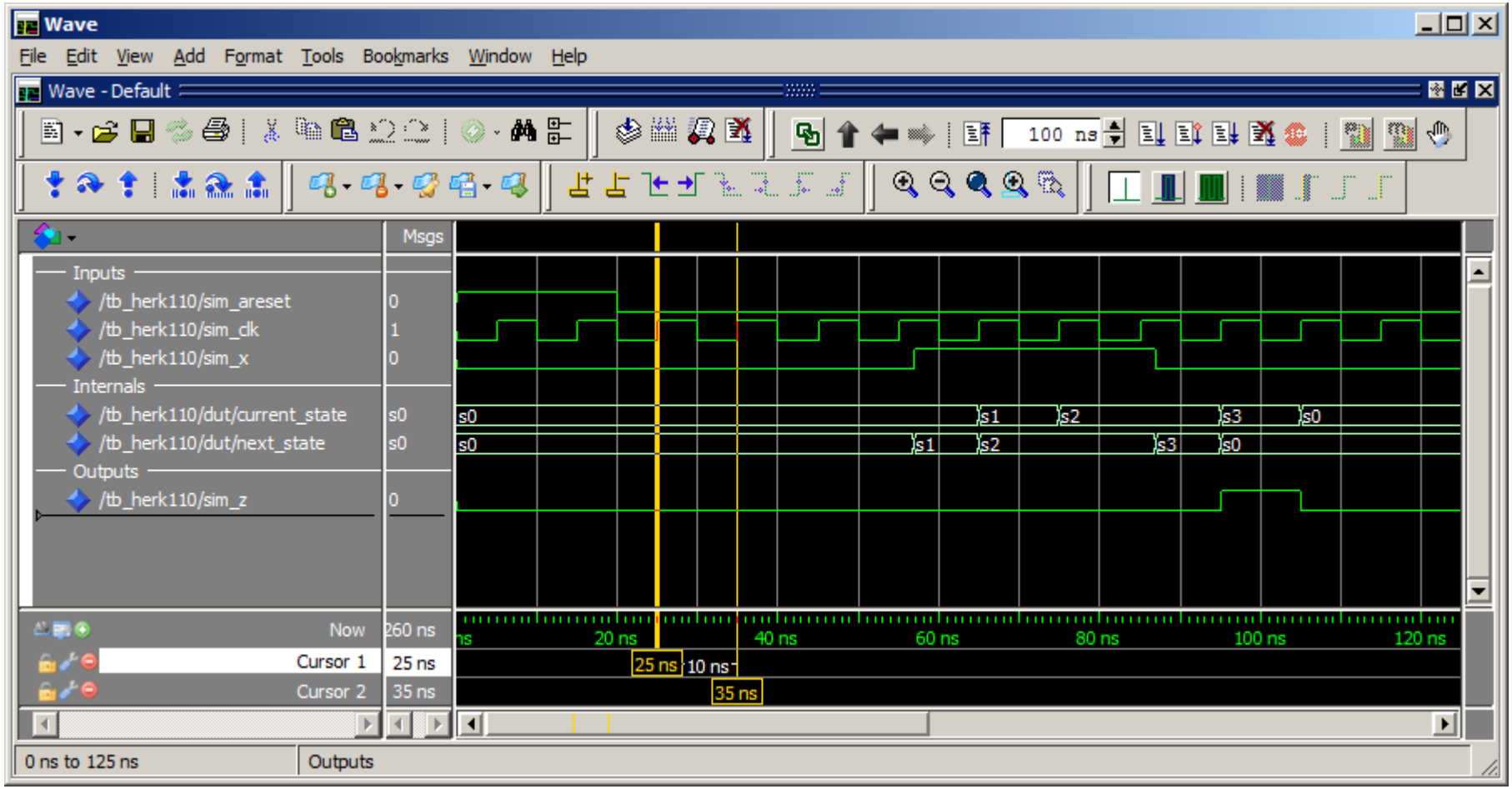
DE HAAGSE HOGESCHOOL

Testbench herkenningsautomaat (3)

```
    ...
    sim_x <= '1';
    wait until sim_clk = '1';
    wait for small_delay;

    sim_x <= '1';
    wait until sim_clk = '1';
    wait for small_delay;

    sim_x <= '0';
    wait until sim_clk = '1';
    wait for small_delay;

    wait;                    -- wait forever, end of simulation
  end process datagen;

end testbench;
```

DE HAAGSE HOGESCHOOL

Simulatieresultaat herkenningsautomaat



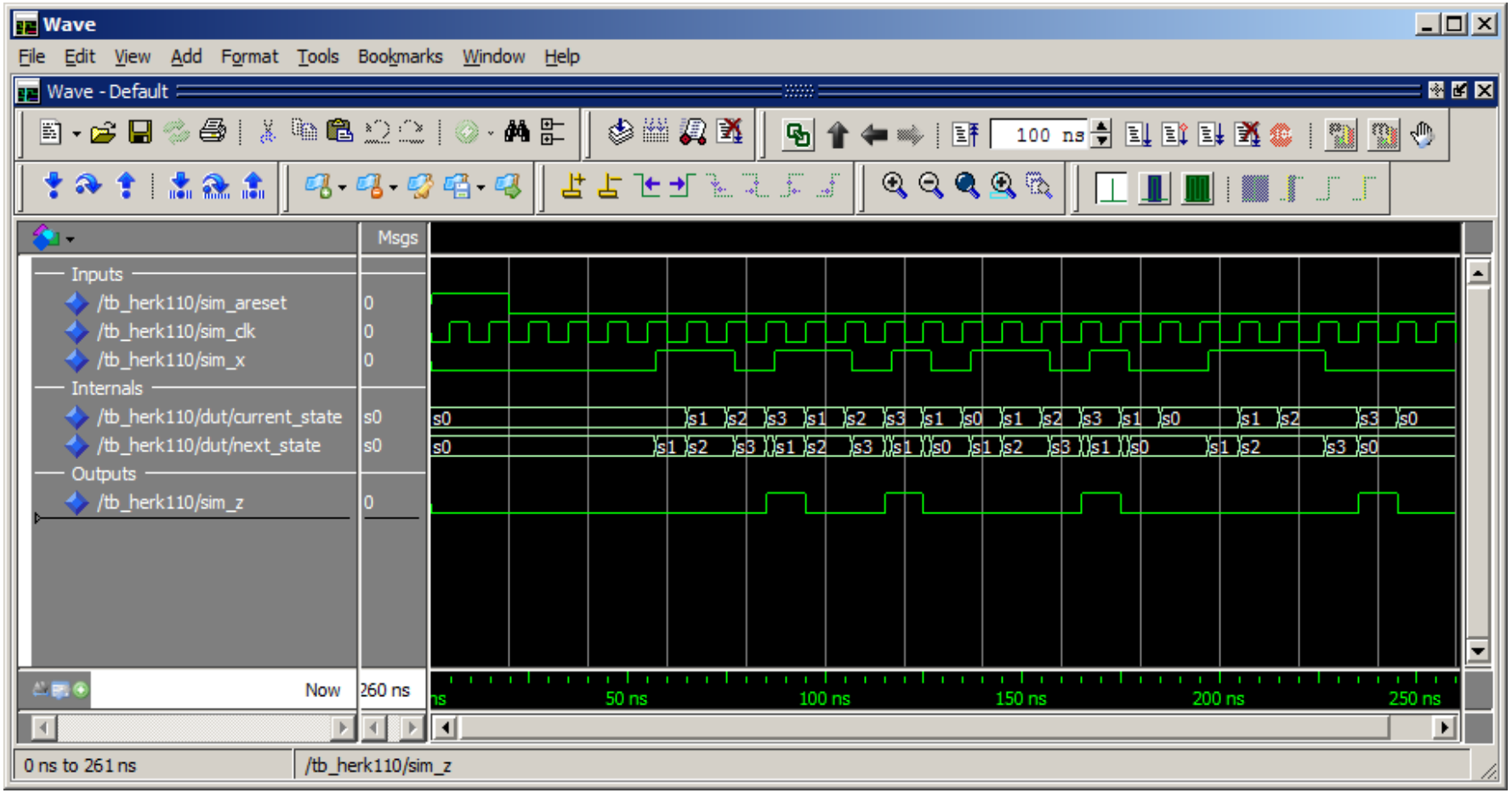
Alternatieve datagen

- VHDL is een taal met veel constructies (een rijke taal).
- Het is mogelijk om array's te gebruiken en lussen te construeren.
- Deze mogelijkheden kunnen goed ingezet worden bij testbenches.
- Een string (array van characters) wordt geladen met een bitpatroon.
- Eén voor één worden de bits toegekend aan de ingang van de toestandsmachine.

Alternatieve datagen

```
datagen: process is

-- the valuations for the FSM as string
constant x_i_s : std_logic_vector := "01101101011010011110";
variable x_i : std_logic_vector(x_i_s'range) := x_i_s;

begin

  -- reset omitted for clarity

  for index in x_i'range loop   -- loop through all input bits
    sim_x <= x_i(index);        -- assign to FSM input
    wait until sim_clk = '1';
    wait for small_delay;
  end loop;

  wait;
end process datagen;
```

Simulatieresultaat alternatieve datagen



Referenties

- http://www.mrc.uidaho.edu/mrc/people/jff/440/handouts/testbenches.pdf
- http://venus.ece.ndsu.nodak.edu/ece/academics/courses/ece375/f2005/notes/testbench.htm

- Fundamentals of Digital Logic with VHDL Design – Brown, 3rd Ed, 2008, ISBN 9780071268806
- Digital Design: Principles and Practices – John F. Wakerly, 4th Ed, 2006, ISBN 0-13-173349-4
- The Designer's Guide To VHDL – Peter J. Ashenden, 2nd Ed, 2002, ISBN 1-55860-674-2
- Digital System Design With VHDL – Mark Zwoliński, 2nd Ed, 2004, ISBN 0-13-039985-X

Academie voor Technology, Innovation & Society Delft

Academie voor ICT & Media

De Haagse Hogeschool, Delft
+31-15-2606311
J.E.J.opdenBrouw@hhs.nl
www.dehaagsehogeschool.nl

DE HAAGSE HOGESCHOOL