Testability

verification

© Peter Veelaert
Dimitri Van Cauwelaert
Michaël Heyvaert

Testability

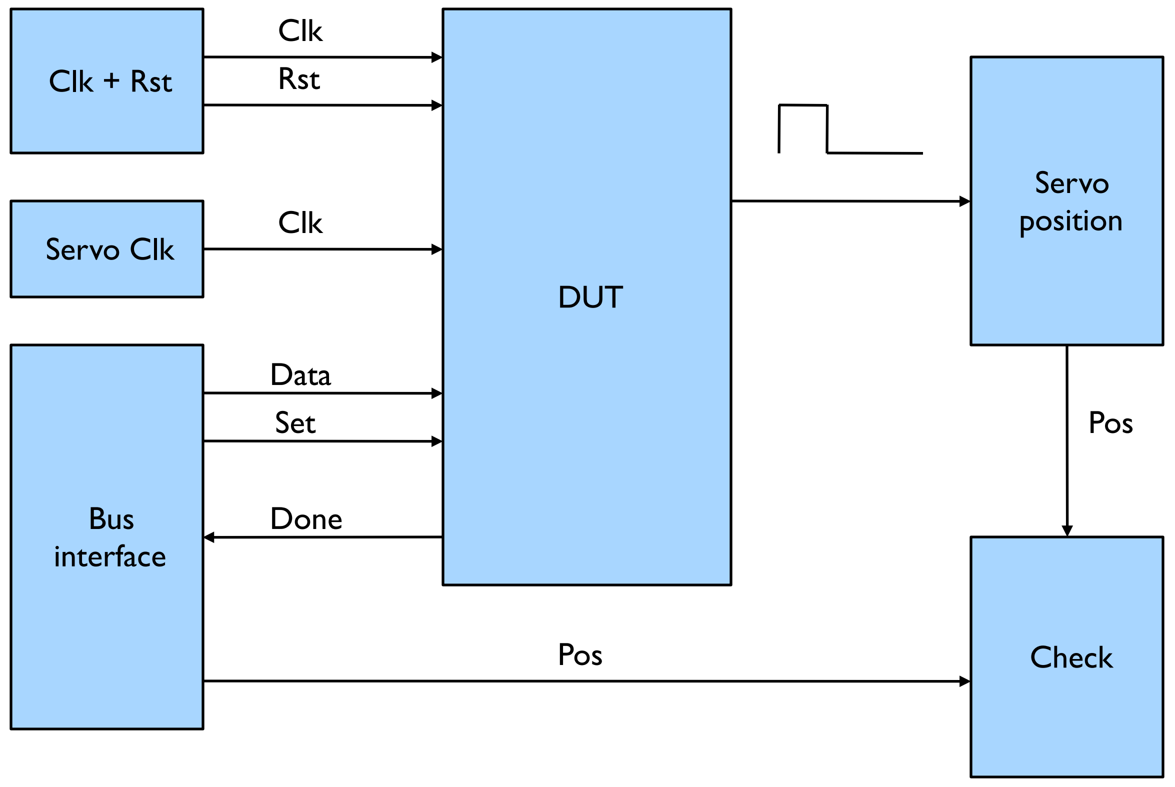
- Verification
  - Black box approach

  - Verify correct operation
    - Apply normal input signals
    - Check output

  - Check boundary conditions
    - What happens when a invalid input sequence is applied

Testability

- Testbench



Testability

- Testbench
  - Instantiate DUT (device under test)

```
dut: entity work.servo
    generic map(...)
    port map (....);
```

  - Generate clock and reset signals

```
clk_gen: process
begin
          clk <= '1';
          wait for 10ns;
          clk <= '0';
          wait for 10ns;
end process;
```

No sensitivity list when using wait... statements!

Testability

- Testbench
  - Generate input stimuli

```
input_gen: process
begin
        ...
        wait until rising_edge(clk);
        ...
        while x = '1' loop
                ...
        end loop;
      ...
        wait;
end process;
```

Wait until condition occurs

Loop constructs are allowed in testbenches

Last statement in testbench process: **wait** (otherwise the process repeats)

Testability

- Tips
  - Convert types to string

```
report “x = “ & integer'image(x)
report “x = “ & unsigned'image(x)
```

  - Check conditions with asserts

```
assert(done = '1');
assert(x = 14);
```

Assignment

- Minimum requirement
  - Generate clock and reset signals
  - Set servo position to 0..255, step 32
  - Get servo position from PWM signal
  - Check against set position
  - Do not use servo clock for PWM check!

- Additional tests
  - Generate incorrect bus signals
  - Test address check