

# COL216

# Computer Architecture

VHDL Guidelines  
21st February 2022

# How to design using VHDL?

- Knowing programming and VHDL syntax is enough
- Think in terms of VHDL constructs
- Any syntactically valid VHDL description is fine
- The tools should make correct sense out of it

X

# How to design using VHDL?

- Think in terms of circuit structure
- Describe it in VHDL
- Use well understood abstractions
- Do not lose sight of underlying hardware



# Design Styles

- An ARCHITECTURE can have -
  - Component instances (structural)
  - Processes (procedural)
  - Concurrent assignments (data flow)

These can be mixed but not advisable

- Structural hierarchy at the top
- Behavioural descriptions at the bottom

# Types of circuits

## ■ Combinational

- No storage / memory
- React instantly to change in inputs



## ■ Sequential (asynchronous)



- Storage / memory
- Level sensitive

## ■ Sequential (synchronous)



- Storage / memory
- Edge triggered

# Design descriptions in VHDL

## Combinational Circuits

- No cyclic dependence among Concurrent statements AND
- Every signal that is destination of some assignment, gets assigned under all conditions AND
- Every process is sensitive to all its sources

## Sequential Circuits

- Cyclic dependence among Concurrent statements OR
- Some signal that is destination of an assignment, doesn't get assigned under some condition OR
- Some process is not sensitive to some of its sources

# Semantics

- The whole design is a collection of concurrent processes
- Processes communicate with each other through signals
- No internal signals within processes, though the synthesizer may create some
- Ensure that each signal is driven by only one process (for the present)

# What does a process mean?

- Repeated computation of values and assignment to signals
- A process executes in
  - zero time
  - non-zero time (not to be used presently)
- A zero time process may represent
  - Either a combinational circuit
  - Or a Sequential circuit

# Signals and processes

- A signal is either an output of a gate or a flip-flop (or an array of these)
- Driven by a process of appropriate type (combinational or sequential)
  - no mix up
  - no latches created by synthesizer
- Each signal driven by a unique process
- A process may drive multiple signals

# Process describing a comb. circuit

- Sensitive to all its inputs
- Assigns to all outputs under all conditions

# Process describing a sequential circuit

- Sensitive to a clock
- All assignments to outputs on clock edge
- Optionally, sensitive to additional signal(s)
- Use this only for initialization
- Initialization with declaration is good only for simulation
- Use DFF description as template

# D Flip-flop with synch S/R

```
ARCHITECTURE synchronous OF DFFsr IS
BEGIN
  PROCESS (clk)
  BEGIN
    IF clk = '1' AND clk'EVENT THEN
      IF s = '1' THEN      q <= '1';
      ELSIF r = '1' THEN  q <= '0';
      ELSE                q <= d;
      END IF;
    END IF;
  END PROCESS;
END ARCHITECTURE synchronous;
```

# D Flip-flop with asynch S/R

```
ARCHITECTURE asynchronous OF DFFsr IS
BEGIN
  PROCESS (clk, s, r)
  BEGIN
    IF s = '1' THEN          q <= '1';
    ELSIF r = '1' THEN       q <= '0';
    ELSIF clk = '1' AND clk'EVENT THEN q <= d;
    END IF;
  END PROCESS;
END ARCHITECTURE asynchronous;
```

# Signal vs Variable

- Signals interconnect processes
- Variables are internal to processes
- A variable is updated instantly
- A signal is updated after a delay (the default is delta)

# Thanks