VHDL

What is VHDL

- Very High Speed Integrated Circuit Hardware Description Language
- Language used for writing models of a system
- Circuit description at different level of abstraction
  - Behavioral
  - RTL
  - Structural
  - Gate level

Components of VHDL

- Entity – Interface definition of each design object
- Architecture – functionality of design entities
- Configuration – association of entities to architecture
- Package- library of data structure and subprograms
- Driver – source on a signal, signal may have drivers
- Bus – may have drivers turned off
- Attributes – current drive capability of a driver, predefined data about design objects
- Generic – required for passing information to an entity through parameters
- Process – basic unit of execution in VHDL

Entity

- Designs are expressed as entities
- Defines the Interface to communicate with other design objects

```
ENTITY mux IS
  PORT (a,b,c,d : IN BIT;
        s0, s1 : IN BIT;
        x: OUT BIT);
END mux;
```

Architecture

- Describes behavior of entity
- An entity can have many architectures: Behavioral or Structural

```
ARCHITECTURE dataflow OF mux IS
  SIGNAL select : INTEGER;
BEGIN
  select <= 0 WHEN s0 = '0' AND s1 = '0' ELSE
            1 WHEN s0 = '1' AND s1 = '0' ELSE
            2 WHEN s0 = '0' AND s1 = '1' ELSE
            3;
  x <= a AFTER  0.5 NS WHEN select = 0 ELSE
       b AFTER  0.5 NS WHEN select = 1 ELSE
       c AFTER  0.5 NS WHEN select = 2 ELSE
       d AFTER  0.5 NS;
END dataflow
```

Concurrent Signal Assignment

- No specified order of assignment statements
- Execution depends on an event occurring on the signal
- Event on signal : value of signal changes

```
  select <= 0 WHEN s0 = '0' AND s1 = '0' ELSE
            1 WHEN s0 = '1' AND s1 = '0' ELSE
            2 WHEN s0 = '0' AND s1 = '1' ELSE
            3;
  x <= a AFTER  0.5 NS WHEN select = 0 ELSE
       b AFTER  0.5 NS WHEN select = 1 ELSE
       c AFTER  0.5 NS WHEN select = 2 ELSE
       d AFTER  0.5 NS;
```

Event Scheduling

- AFTER clause
- Delaying the new value is called scheduling an event
- Event on signal : value of signal changes

```
x <= a AFTER  0.5 NS WHEN select = 0 ELSE
     b AFTER  0.5 NS WHEN select = 1 ELSE
     c AFTER  0.5 NS WHEN select = 2 ELSE
     d AFTER  0.5 NS;
```

Concurrency of Statements

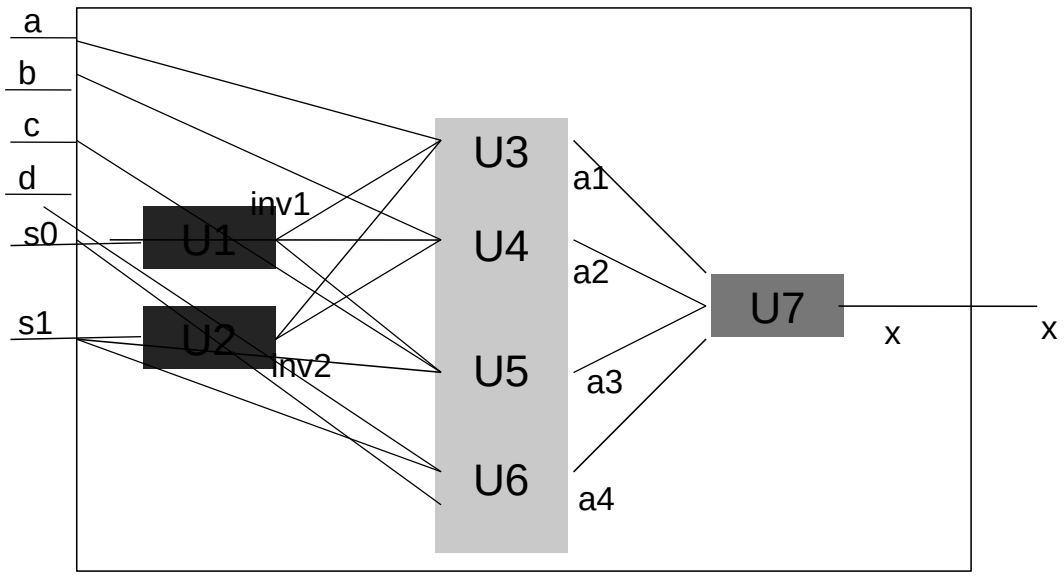
- First assignment statement gets executed when there is a change in s0 and s1
- Second : when value of select, a, b, c, d changes
- Right hand side of <=

```
select <= 0 WHEN s0 = '0' AND s1 = '0' ELSE
          1 WHEN s0 = '1' AND s1 = '0' ELSE
          2 WHEN s0 = '0' AND s1 = '1' ELSE
          3;
x <= a AFTER  0.5 NS WHEN select = 0 ELSE
     b AFTER  0.5 NS WHEN select = 1 ELSE
     c AFTER  0.5 NS WHEN select = 2 ELSE
     d AFTER  0.5 NS;
```

Behavioral Design

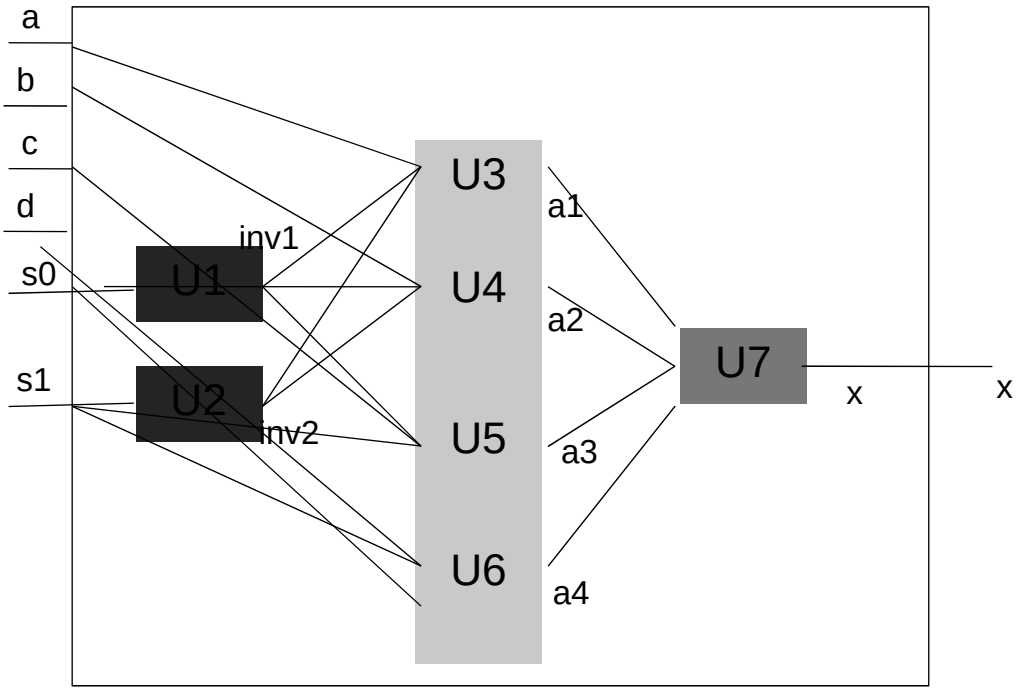
- Flow of data is expressed in behavioral design
- Not the structure of a design in terms of gates.
- Used for functionality test, to correctly specify the system requirements etc.
- Behavioral designs may not be synthesizable to target technology

MUX



Structural Design (Contd.)

```
BEGIN
U1: inv
    port map(s0,inv1);
U2: inv
port map (s1,inv2);
U3: and3
port map (a,inv1,inv2,a1);
U4: and3
port map(b,s0,inv1,a2);
U5: and3
port map(c,inv1,s1,a3);
U6: and3
port map(d,s0,s1,a4);
U7: or
port map(a2 => b, a1 => a, a4 => d, a3 => c, x => x);
END
```



Structural Design

- Architecture defined as connections of different basic / compound components
- Hierarchical structure

```
ARCHITECTURE netlist OF mux is
COMPONENT and3
PORT(a,b,c : IN BIT; x: OUT BIT;
END COMPONENT;
COMPONENT inv
PORT(a : IN BIT; d: OUT BIT;
END COMPONENT;
COMPONENT or4
PORT(a,b,c,d : IN BIT; x: OUT BIT;
END COMPONENT;
SIGNAL inv1,inv2,a1,a2,a3,a4 : BIT;
```

Sequential Behaviour

- Use of process statement

```
ARCHITECTURE seq OF mux IS
PROCESS(a,b,c,d,s0,s1) – sensitivity list
VARIABLE sel : INTEGER;
BEGIN
 IF s0 = '0' AND s1 = '0' THEN sel := 0;
 ELSIF  s0 = '1' AND s1 = '0' THEN sel := 1;
 ELSIF  s0 = '0' AND s1 = '1' THEN sel := 2;
ELSE sel := 3;
```

Sequential Behavior (Contd.)

```
CASE sel IS
WHEN 0 => x <= a;
WHEN 1 => x <= b;
WHEN 2 => x <= c;
WHEN OTHERS => x <= d;
END CASE;
END PROCESS;
END seq;
```

PROCESS statement

- Sensitivity list
- Process declarative part
- Statement part
- Events on process sensitivity list executes the process sequentially
- VARIABLES are only used in process statements
- Order of statement inside process – important

Architecture Selection

- Which architecture to model *mux?*
- *Structural :* if needs synthesis for final layout
- *Behavioral :* otherwise
- Better use concurrent style
- RTL VHDL : for synthesis- optimizes gate-level description

Architecture Selection : Configuration

- Maps component instantiation to entities
- Associate architecture with entity
- Only the new configuration is compiled in the library

```
CONFIGURATION muxconf OF mux IS
FOR netlist – architecture netlist for mux
 FOR U1, U2 : inv USE ENTITY WORK.myinv(v1);
 END FOR;
 FOR U3, U4,U5,U6 : and3 USE ENTITY WORK.myand(v1);
 END FOR;
FOR U7 : or4 USE ENTITY WORK.myor(v1);
 END FOR;
END FOR;
END muxconf
```

Behavioral Modeling

- Signal assignment a<= b
- Signal b is in sensitivity list of statement
- If different value, event scheduled for target signal
- No change in value – no event but transaction
- C <= a and b after 10ns
- Change in value ->Transaction-> schedules an event

Concurrent statement

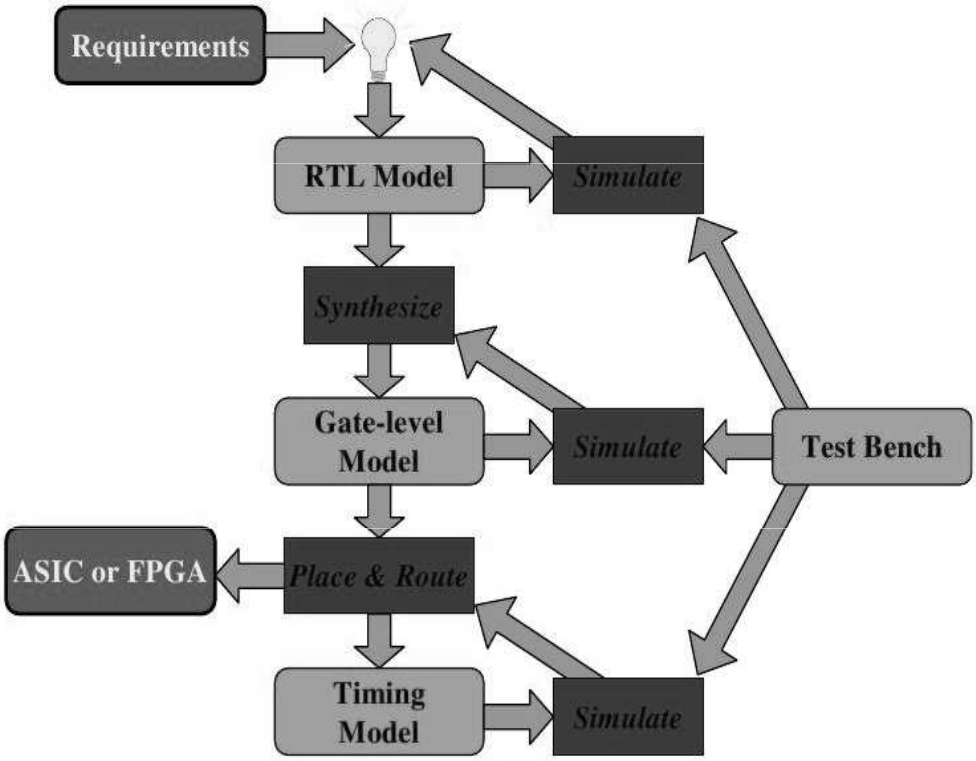
```
ARCHITECTURE arch OF mux IS
  SIGNAL sel : INTEGER;
BEGIN
 WITH sel SELECT
x <= a AFTER  0.5 NS WHEN 0 ,
        b AFTER  0.5 NS WHEN 1,
        c AFTER  0.5 NS WHEN 2 ,
        d AFTER  0.5 NS WHEN 3 ;

  sel <= 0 WHEN s0 = '0' AND s1 = '0' ELSE
              1 WHEN s0 = '1' AND s1 = '0' ELSE
              2 WHEN s0 = '0' AND s1 = '1' ELSE
              3;
```

Test Benches

- Testing a design by simulation
- Use simulator
- Inputs are sequence of test values
- Inputs are given in the form of waveforms
- Outputs are generated as waveforms

Design flow


Requirements
RTL Model
Simulate
Synthesize
Gate-level Model
Simulate
Test Bench
ASIC or FPGA
Place & Route
Timing Model
Simulate