Training Schedule

Lecture 1: VHDL Introduction

Lecture 2: Concurrent" and Sequential VHDL

Lecture 3: RTL Definition

Lecture 4: Signals and Types

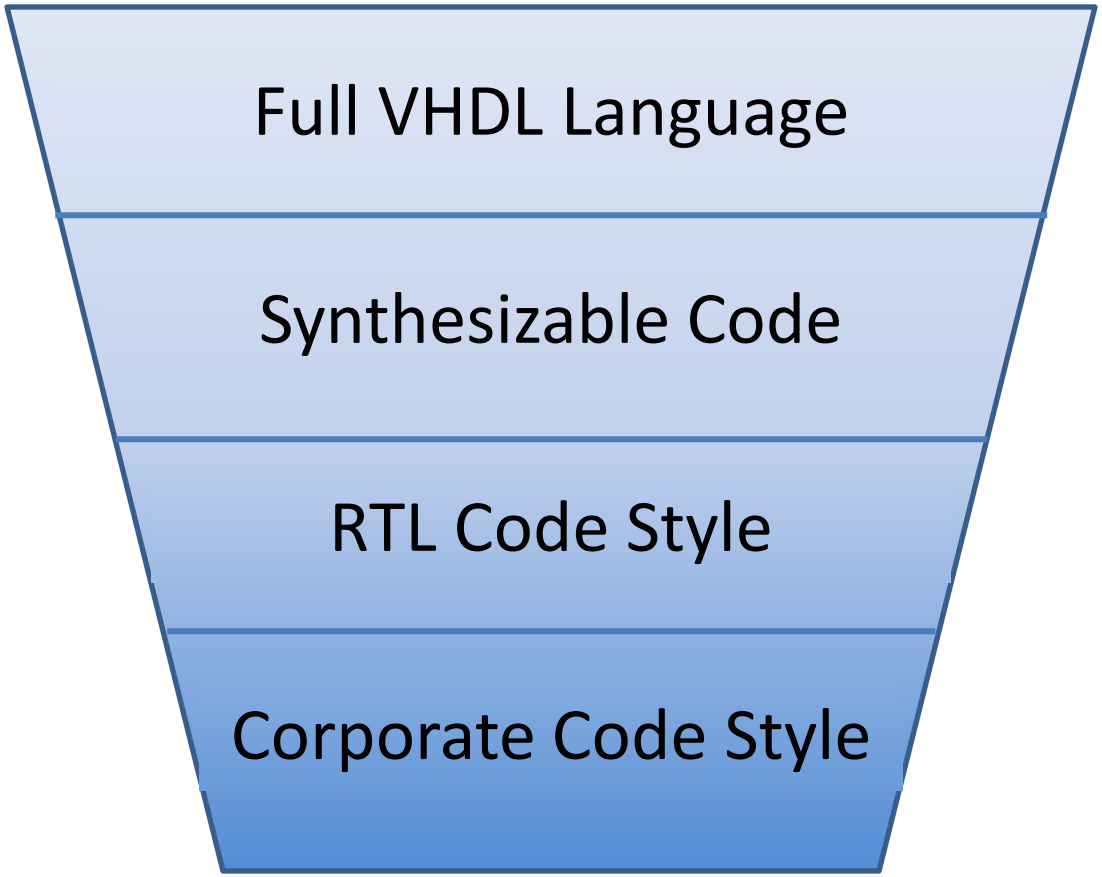
Lecture 5: VHDL Operators

Lecture 6: Synthesis VHDL Coding Styles

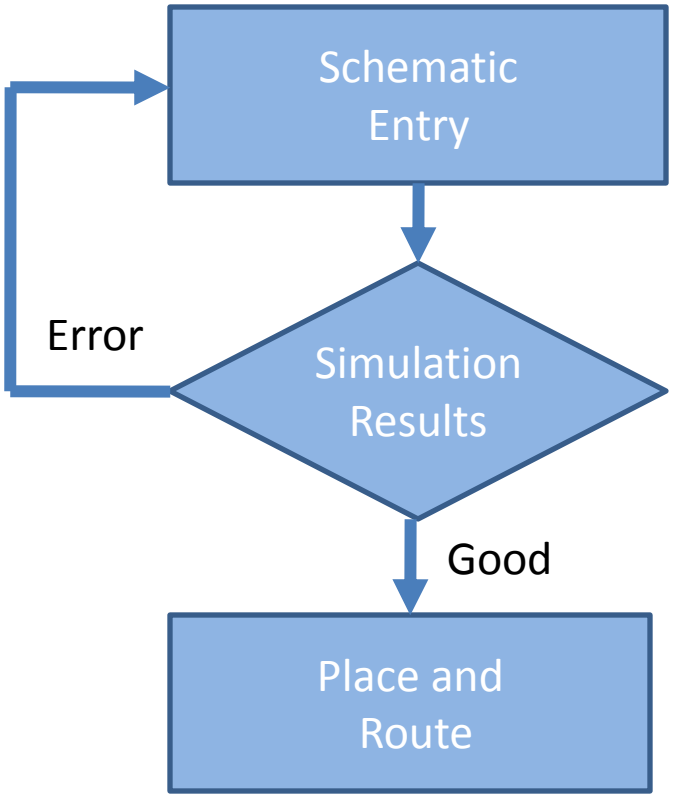
Lecture 7: Testbench Coding in VHDL

Lecture 8: Advanced VHDL

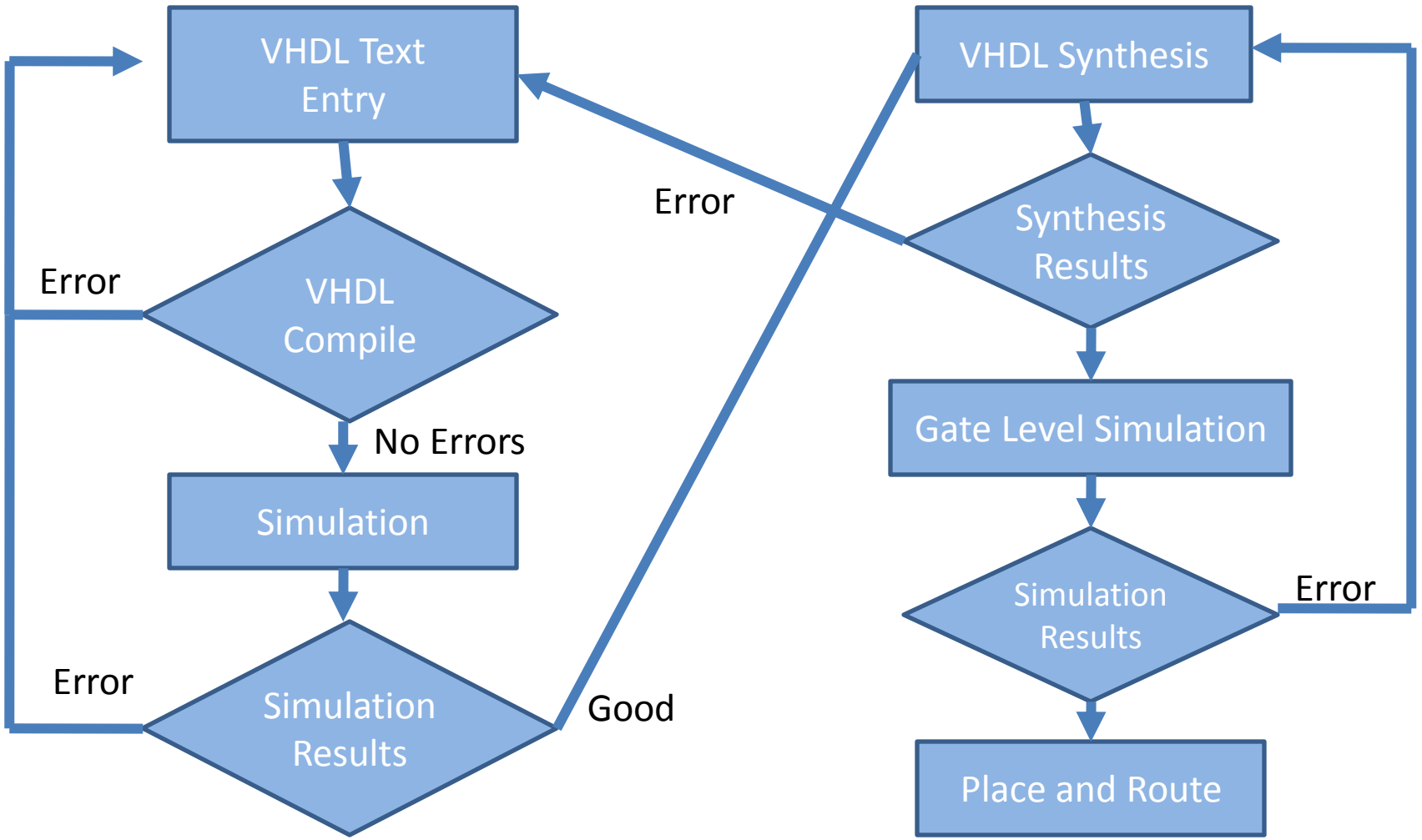
VHDL Language



Schematic Design Flow



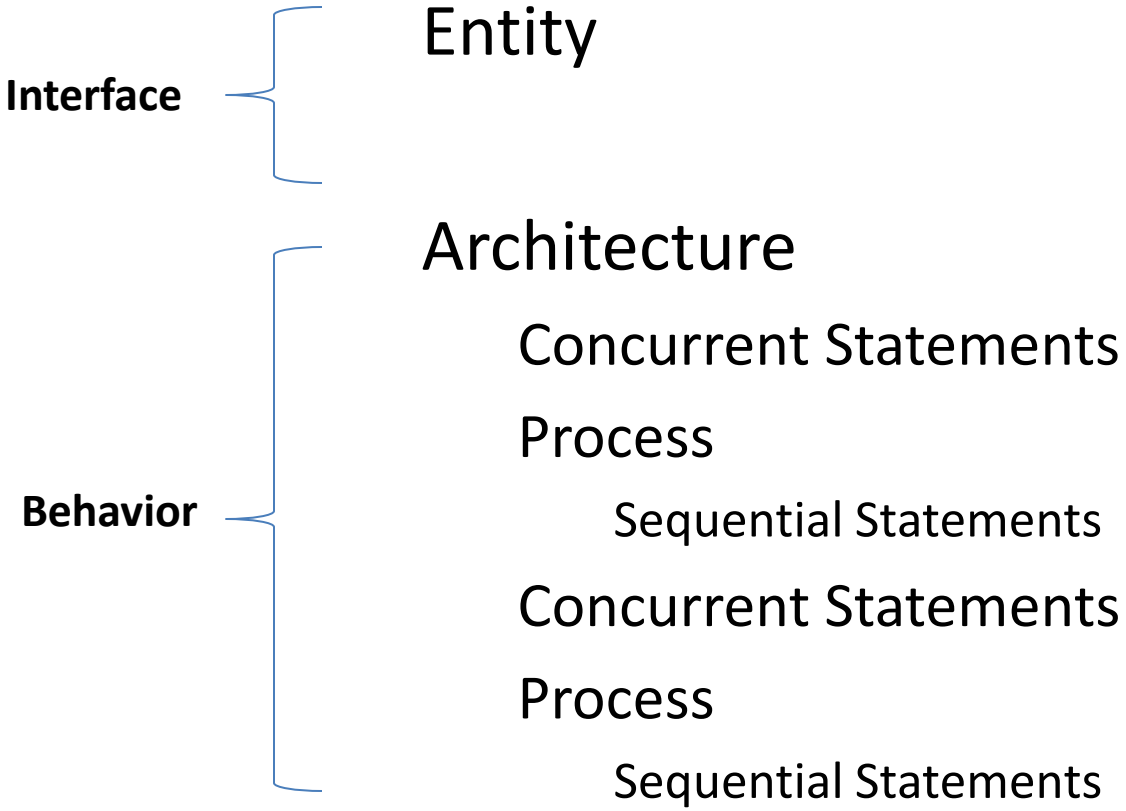
VHDL Design Flow



VHDL Structure

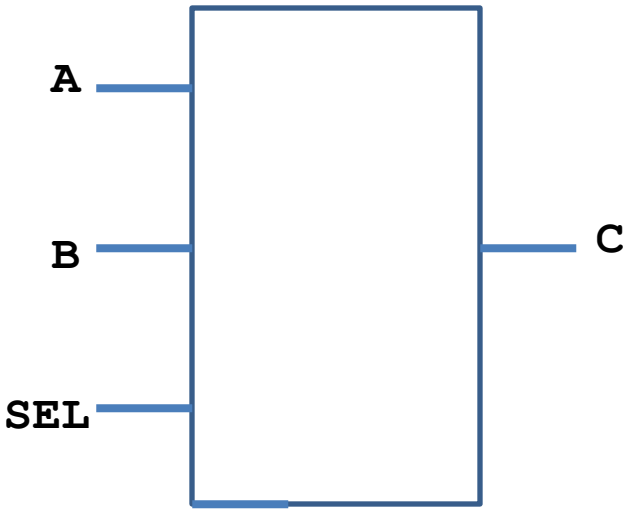
VHDL	Schematic
Interface	Symbol
Behavior	Logic
Configuration	Schematic Hierarchy

VHDL Structure (Typical)



Entity

```
entity MUX is
  port ( A   : in  Std_logic;
         B   : in  Std_logic;
         SEL : in  Std_logic;
         C   : out Std_logic
       );
end MUX
```

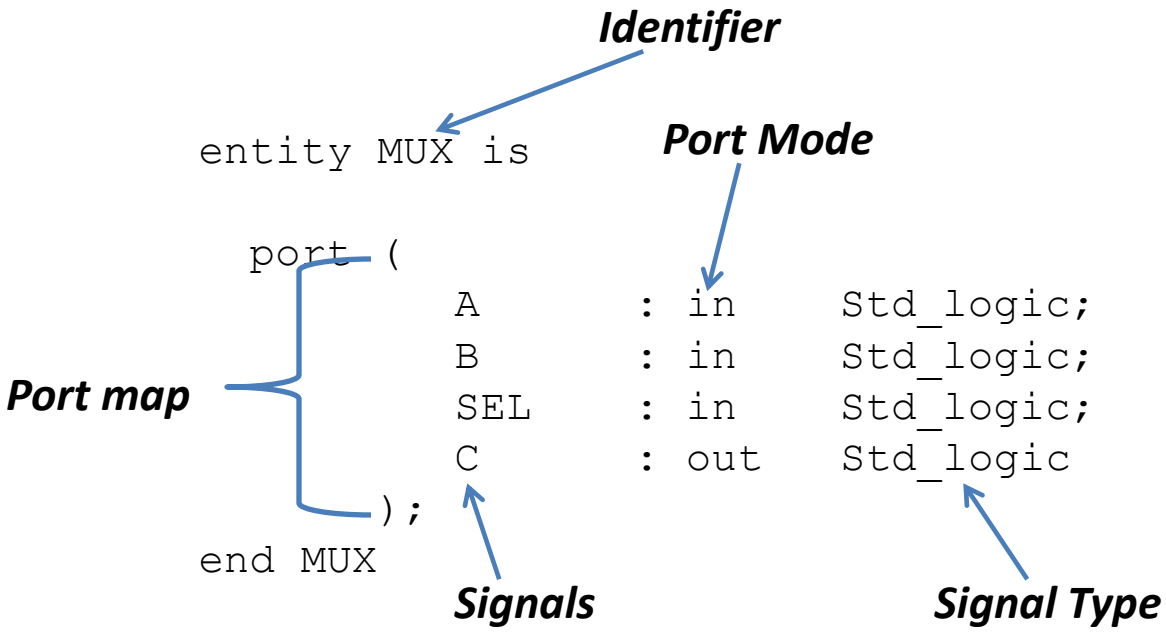


Entity Syntax

```
entity identifier is
   [port (port interface list);
end [entity] [identifier];
```

- Defines the Interface
- Is Associated with one or more Architectures

Entity Structure

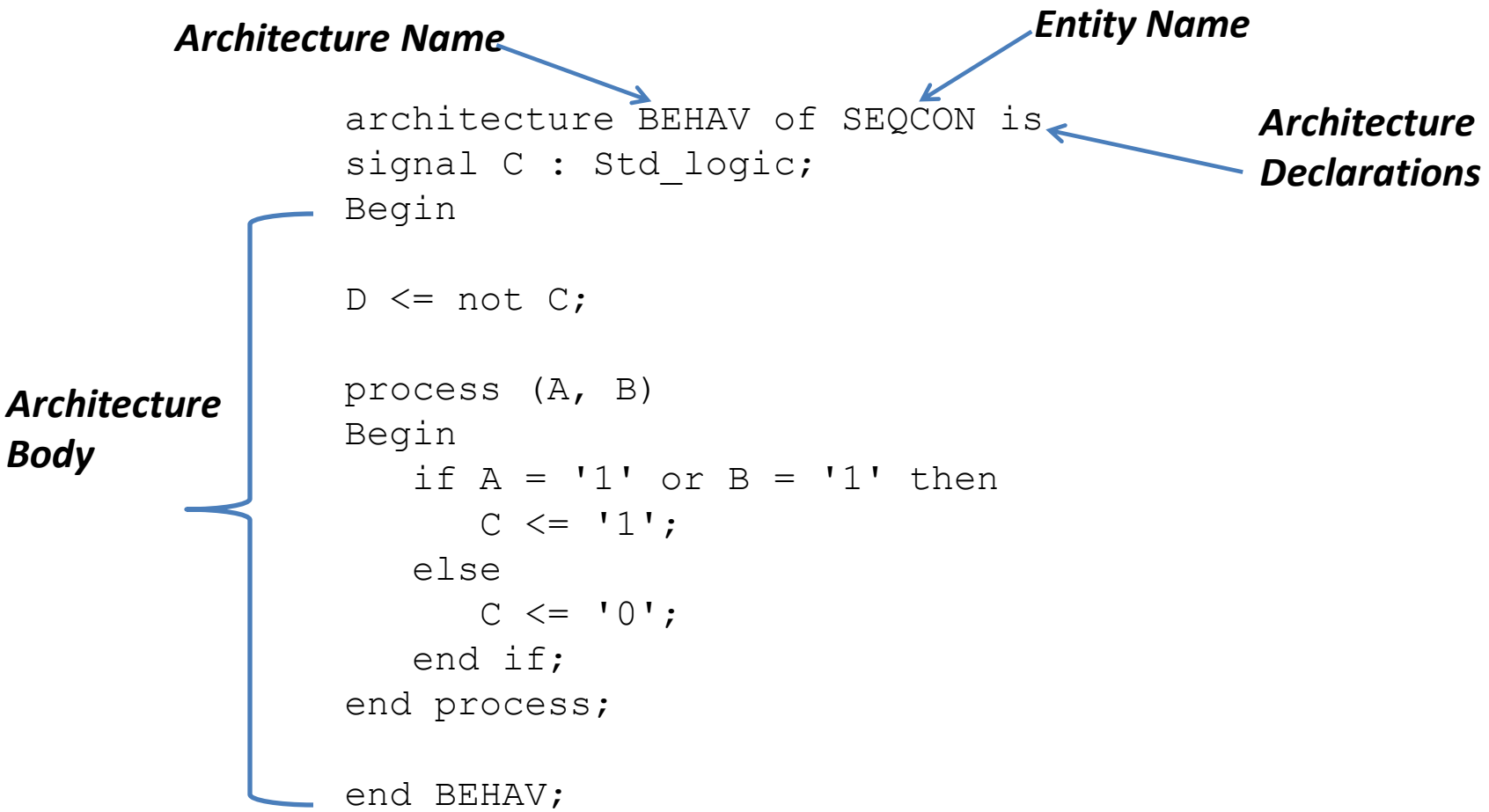


Architecture Syntax

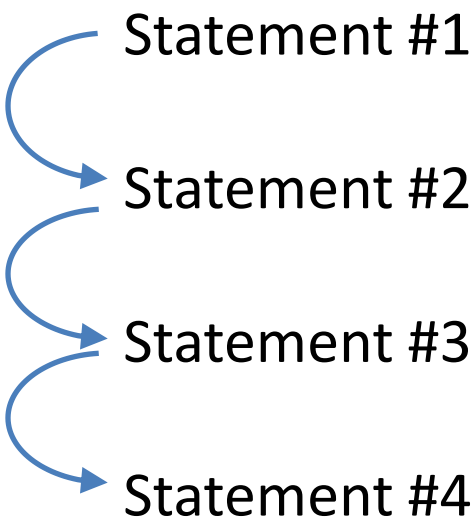
architecture identifier **of** entity-name **is**
begin
{architecture body}
end [architecture] [identifier];

- Behavior is defined
- Is Associated with a single Entity

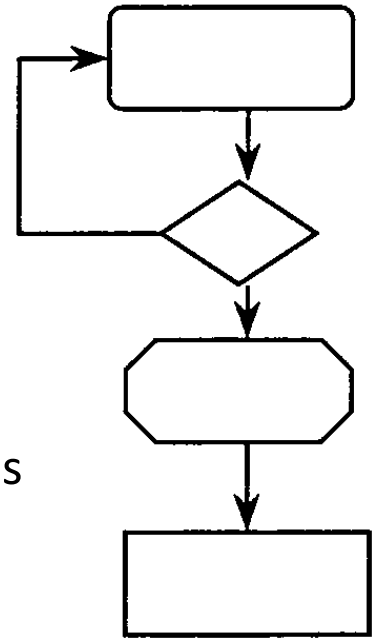
Architecture Structure



Sequential Definition



Flowcharts



- Statements executed sequentially
- "Software Model"

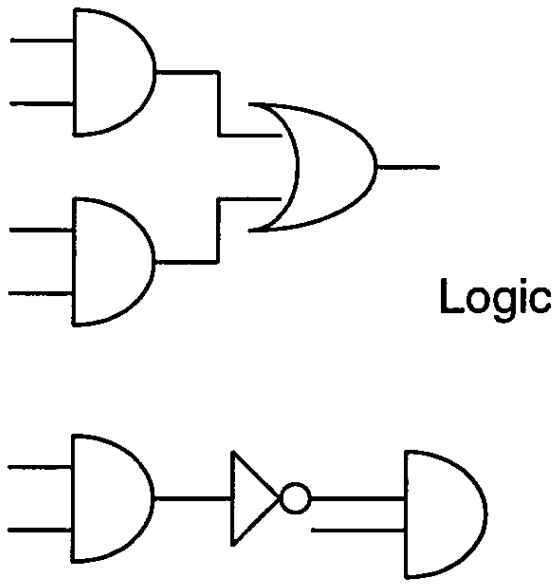
Concurrent Definition

Statement #1

Statement #2

Statement #3

Statement #4



- Statements executed simultaneously
- “Hardware Model"

Process Syntax

process_label: **process (**signal list**)**
begin
end process [process_label]**;**

- Contains Only Sequential Statements
- Executes Concurrently with other Processes

Sequential vs. Concurrent

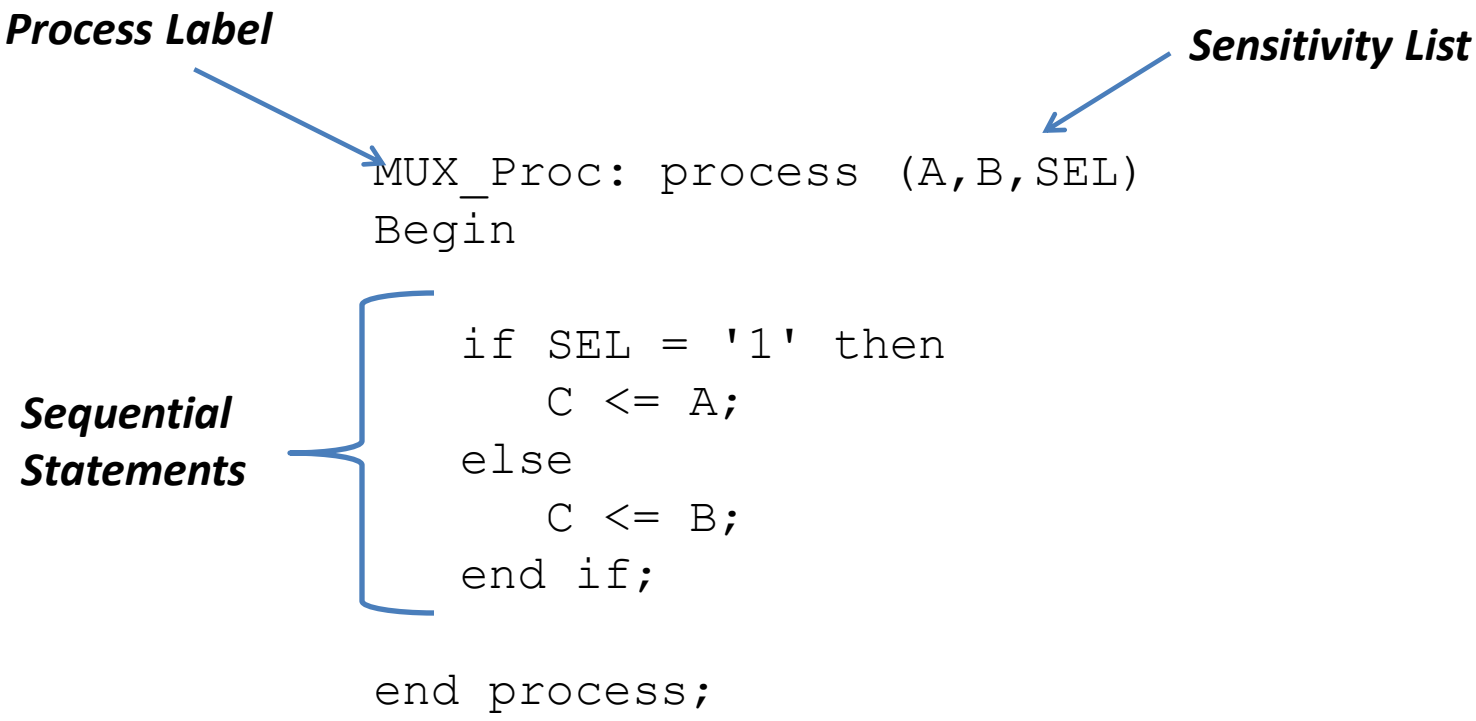
```
SEQ: process (A, B)
begin
   if A = '1' and B = '1' then
      C<='1';
   else
      C<='0';
   end if;
end process;
```

Sequential

```
C <= (A and B);
```

Concurrent

Process Structure



Process Characteristics

```
MUX_Proc: process (A,B,SEL)
Begin

   if SEL = '1' then
      C <= A;
   else
      C <= B;
   end if;

end process;
```

What happens when the sensitivity list is incomplete?

Process Rules

- Process executes when signal in sensitivity list changes
- Process contains only sequential statements
- Signal assignments occur at the "end process" statement
- Signals get the last executed assignment

Concurrent Example

```
entity CONEX is
port ( A   : in  Std_logic;
       B   : in  Std_logic;
       C   : out Std_logic);
end CONEX ;

architecture BEHAV of CONEX is
begin

   C <= A or B;

end behav;
```

What behavior does this code represent?

Ex1_Concurrent.vhd

Sequential Example

```
entity SEQEX is
port ( A   : in  Std_logic;
       B   : in  Std_logic;
       C   : out Std_logic);
end SEQEX ;

architecture BEHAV of SEQEX is
begin

process (A, B)
begin
   if A = '1' or B = '1' then
      C <= '1';
   else
      C <= '0';
   end if;

end process;

end behav;
```

What behavior does this code represent?

Ex1_Sequential.vhd

Concurrent & Sequential Example

```
entity SEQ2EX is
port ( A   : in  Std_logic;
       B   : in  Std_logic;
       D   : out Std_logic );
end SEQ2EX ;

architecture BEHAV of SEQ2EX is
signal C : Std_logic;
begin

D <= not C;

process (A, B)
begin
   if A '1' or B = ‘1’ then
      C <= '1';
   else
      C <= '0';
   end if;
end process;

end behav;
```

How do Concurrent and sequential statements interact?

Ex1_ConSeq.vhd

Concurrent vs. Sequential

```
architecture BEHAV1 of SEQCONA is

begin
   Z <= X and Y;
   Z <= X or Y;
end BEHAV1;
```

Concurrent

```
architecture BEHAV2 of SEQCONA is
Begin

process (X, Y)
Begin

   Z <= X and Y;
   Z <= X or Y;

end process;
end BEHAV2;
```

Sequential

Draw a block diagram for the logic represented by each architecture

Signal Types and Sizes

```
entity BADCODE is
port ( X_INT  : in  integer;
       Y_NIB  : in  Std_logic_vector(3 downto 0);
       Z_BYTE : out Std_logic_vector(7 downto 0);
end BADCODE;

architecture BEHAV of BADCODE is
Begin

Z_BYTE <= X_INT and Y_NIB;
Z_BYTE <= not Y_NIB;

end BEHAV;
```

Signal Types and Sizes must match

Ex1_SigTypes.vhd

VHDL Syntax

```
entity TEST is
port ( A    : in  Std_logic;
       B    : in  Std_logic;
       D    : out Std_logic;
end TEST ;

architecture RTL of ORGATE is

signal C : Std_logic;

begin

D <= not C;

process (A, B)
begin
   if A '1' or B = '1' then
      C <= '1';
   else
      C <= '0';
   end if;
end process;

end behav;
```

VHDL Language
Syntax is very strict

Find the errors in this code.