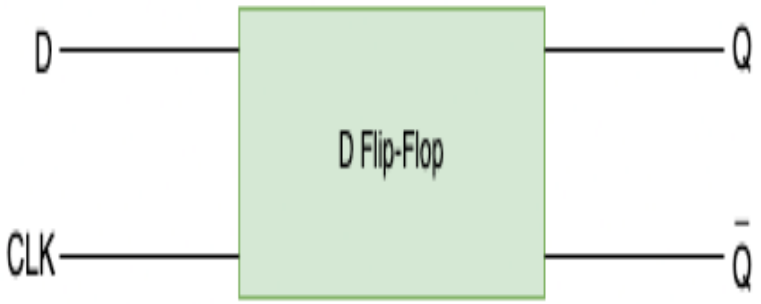
Course Name: Digital Hardware Design
Course Code: 17B1NEC741

VHDL-4

Dr. Arti Noor
Dean, Academic Affairs
Electronics and Communication Engineering,
Jaypee Institute of Information Technology, Noida

Behavioral Modeling

DFF



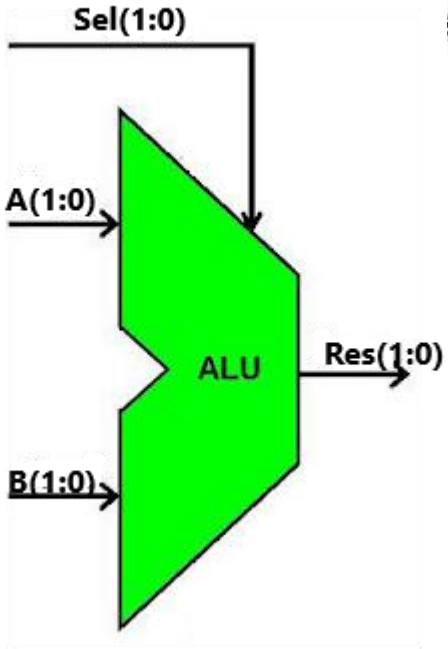
Clock	D	Q	Q’
↓ » 0	0	0	1
↑ » 1	0	0	1
↓ » 0	1	0	1
↑ » 1	1	1	0

Transition at +edge of clock

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

entity D_FF is
PORT( D,CLOCK: in std_logic;
Q: out std_logic);
end D_FF;

architecture behavioral of D_FF is
begin
process(CLOCK)
begin
if(CLOCK='1' and CLOCK'EVENT) then
Q <= D;
end if;
end process;
end behavioral;
```

ALU

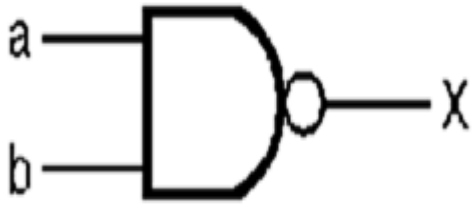
```
entity ALU is

port(    A:       in std_logic_vector(1 downto 0);
         B:       in std_logic_vector(1 downto 0);
         Sel:     in std_logic_vector(1 downto 0);
         Res:     out std_logic_vector(1 downto 0)
);

end ALU;

-------------------------------------------------

architecture behv of ALU is
begin

     process(A,B,Sel)
     begin

          -- use case statement to achieve
          -- different operations of ALU

          case Sel is
               when "00" =>
                    Res <= A + B;
               when "01" =>
                    Res <= A + (not B) + 1;
               when "10" =>
                    Res <= A and B;
               when "11" =>
                    Res <= A or B;
               when others =>
                    Res <= "XX";
          end case;

     end process;

end behv;
```



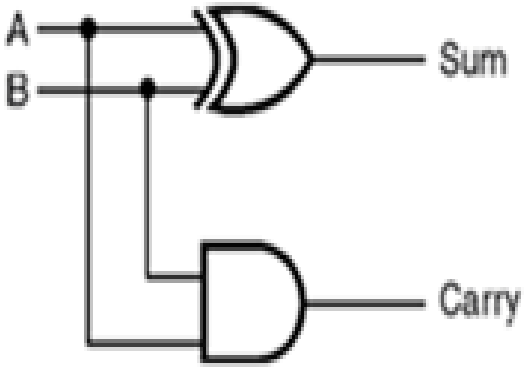
ALU Operation	Description
Addition	Res=A+B
Subtract	Res= A-B=A+(not B)+1
Bitwise OR	A or B
Bitwise AND	A and B

Dataflow Modeling

Dataflow Modeling




```
architecture DATAFLOW of NAND2 is
begin
 X <= a nand b;
end DATAFLOW;
```

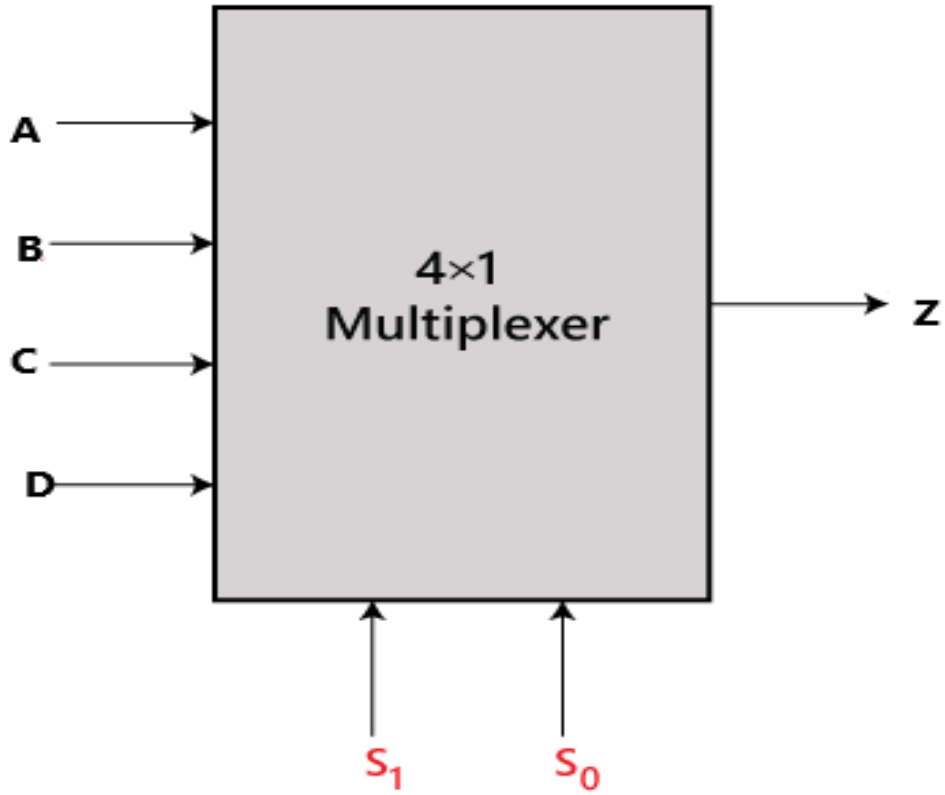



```
library ieee;
use ieee.std_logic_1164.all;
entity half_adder is
port (a, b: in std_logic;
sum, carry: out std_logic);
end half_adder;
architecture dataflow of Half-adder is
begin
sum <= A XOR B;
carry <= A AND B;
end dataflow;
```

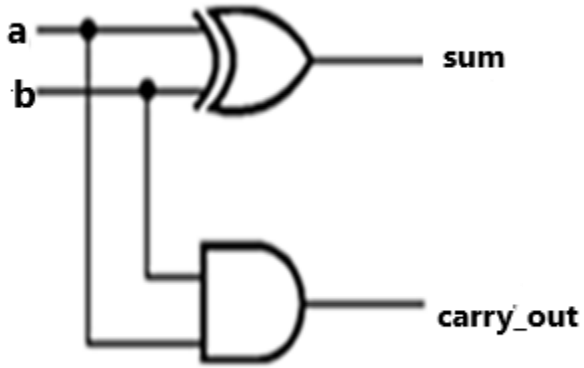
Dataflow Modeling

An example of a 4-to-1 multiplexer using conditional signal assignments:

```
entity MUX_4_1_Conc is
port (S1, S0, A, B, C, D: in std_logic;
        Z: out std_logic);
end MUX_4_1_Conc;

architecture dataflow_MUX41 of MUX_4_1_Conc is
Begin

Z <= A when S1='0' and S0='0' else
     B when S1='0' and S0='1' else
     C when S1='1' and S0='0' else
     D;
end dataflow_MUX41;
```



Dataflow Modeling vs. Behavioral Modeling




```
library ieee;
use ieee.std_logic_1164.all;
entity half_adder is
port (a, b: in std_logic;
sum, carry_out: out
std_logic);
end half_adder;
```

```
architecture dataflow of half_adder is
begin
sum <= a xor b;
carry_out <= a and b;
end dataflow;
```

a	b	sum	Carry-out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Same as b

Not of b

Same as b

```
architecture behavior of half_adder is
begin
ha: process (a, b)
begin
if a = '1' then
sum <= not b;
carry_out <= b;
else
sum <= b;
carry_out <= '0';
end if;
end process ha;
end behavior;
```

Structural modeling

Structural Modeling

- Structural modeling describes a circuit in terms of components and its interconnection
- It is very good to describe complex digital systems, though a set of components in a hierarchical fashion
- A structural description can best be compared to a schematic block diagram that can be described by the components and the interconnections.
- Constraint with structural modeling is that block diagram for the complete system must be known.

Structural Modeling

The following steps are involved for structural modeling.

a. Component declaration (list of components being used).
b. Component instantiation: Interconnection between components (Declare signals which define the nets that interconnect components).
c. Label multiple instances of the same component so that each instance is uniquely defined.

COMPONENT linked with, LIBRARY declarations, ENTITY, ARCHITECTURE. However, by declaring COMPONENT, it can be used within another system.

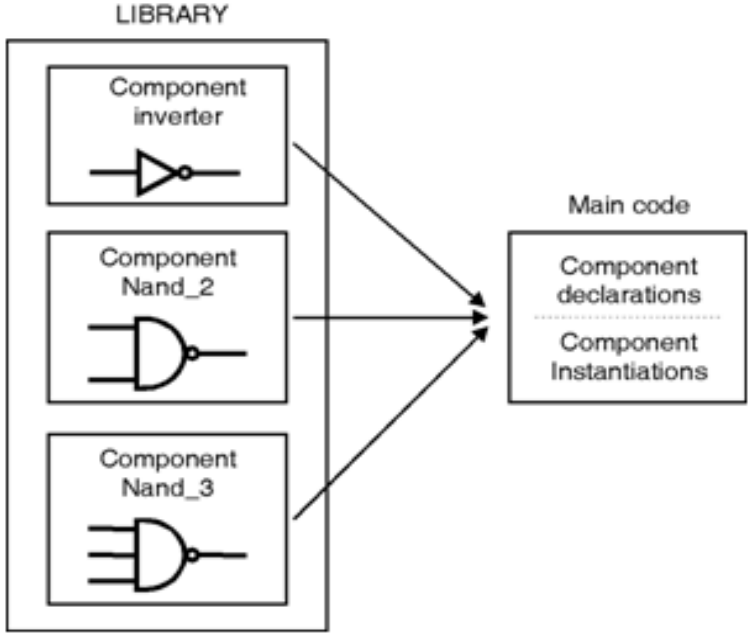
COMPONENT declaration :

```
COMPONENT component_name IS
PORT ( port_name : signal_mode signal_type;
       port_name : signal_mode signal_type; ...);
END COMPONENT;
```

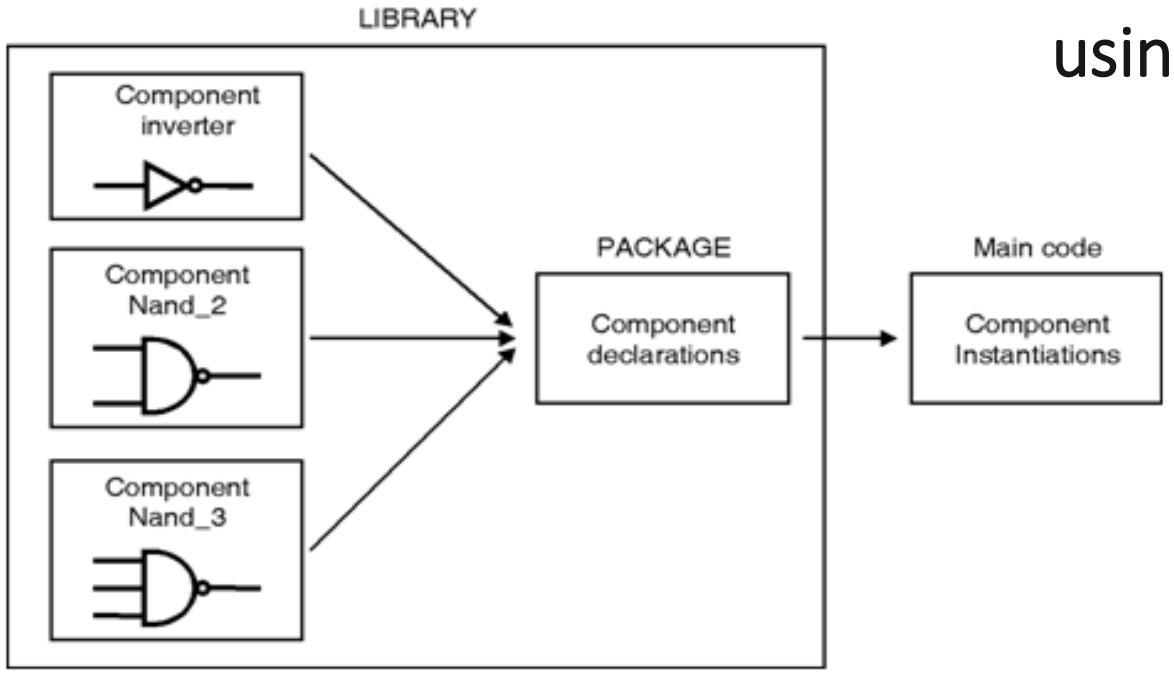
Declaration is similar to that of an ENTITY

COMPONENT instantiation :

label: component_name PORT MAP (port_list);



Declaration in main code



Declaration using package

Two ways to declare a COMPONENT.

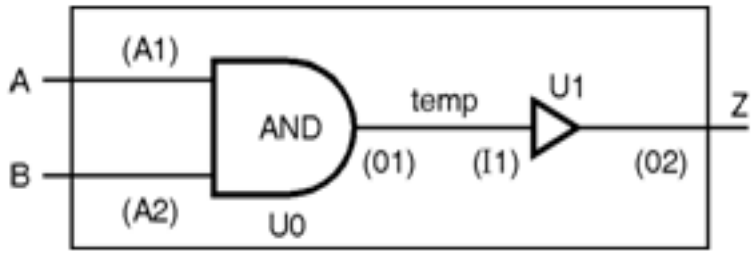
1) designed and placed in the destination LIBRARY
2) declared in the main code or declared using a PACKAGE.

Example 1 : Structural architecture for NAND2

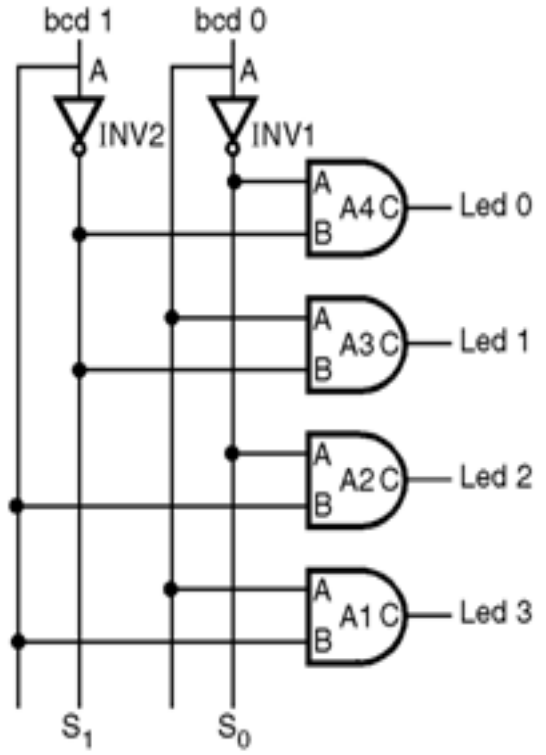
```
architecture STRUCTURAL of NAND2 is
 signal temp: BIT;                          -- (internal) signal declaration
 component AND2                             -- AND2 component declaration
  port (A1, A2: in BIT; O1: out BIT);
 end component;
 component INVERT                           -- INVERT component declaration
  port (I1: in BIT; O2: out BIT);
 end component;
 begin                                      --Architecture
U0: AND2 port map (I1 => A, I2 => B, O1 => temp);
U1: INVERT port map (I1 => temp, O2 => Z)
end STRUCTURAL;
```

Example 2 : Structure architecture of Decoder

The components Inverter and AND_Gate are instantiated under the names Inv1, Inv2, A1, A2, A3 and A4.
The connections among the components are realized by the use of signals S(0), S(1) declared in the architecture's declarative part.



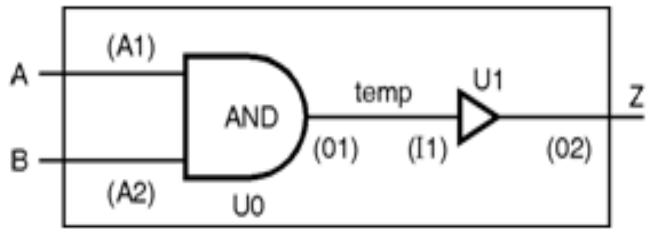
Example 1: NAND Gate



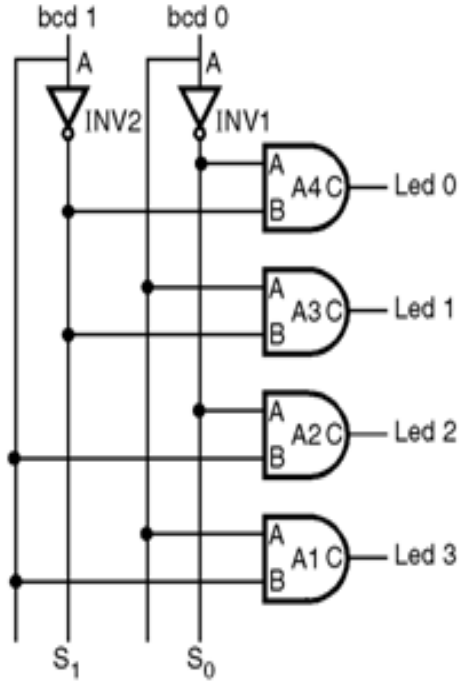
Example 2: Decoder

Example 2 : Structure architecture of Decoder

```
architecture Structural of Decoder is
signal S: Bit_Vector(0 to 1);
component AND_Gate
port(A,B:in Bit; C:out Bit);
end component;
component Inverter
port(A:in Bit; B:out Bit);
end component;

begin

Inv1:Inverter port map(A=>bcd(0), B=>S(0));
Inv2:Inverter port map(A=>bcd(1), B=>S(1));
A1:AND_Gate port map(A=>bcd(0), B=>bcd(1), C=>led(3));
A2:AND_Gate port map(A=>bcd(0), B=>S(1), C=>led(2));
A3:AND_Gate port map(A=>S(0), B=>bcd(1), C=>led(1));
A4:AND_Gate port map(A=>S(0), B=>S(1), C=>led(0));

end Structure;
```



Example 1: NAND Gate



Example 2: Decoder

- Order of the signal in the port command must be same as that in the entity declaration otherwise it will give syntax error.
- Example:

```
    Entity  xor1
    Port (a, b: in std_logic;
              c: out std_logic);
    end xor1;
```

and during component declaration its written as

```
component xor1
   Port ( b, a : in std_logic;
             c: out std_logic);
    end component;
-- error as in the entity a is declared before b.
```

- An alternative method is the positional association shown below,

  port map (*signal1, signal2,…signaln*);

in which the first port in the component declaration corresponds to the first signal, the second port to the second signal, etc.

- The signal position must be in the same order as the declared component's ports.

Example

```
component NAND2
 port (in1, in2: in std_logic;
 out1: out std_logic);
end component;

signal int1, int2, int3: std_logic;

architecture struct of EXAMPLE is
U1: NAND2 port map (A,B,int1);
U2: NAND2 port map (in1=>C, in2=>D, out1=>int2);
U3: NAND3 port map (in1=>int1, int2, Z);
```

Important tips about port-map: Unconnected inputs:

- Suppose for three input AND gate as a component and only two of its inputs are required for some application

```
Component and1
Port (a, b, c: in std_logic;  --  3 input AND gate.
        F       :  out std_logic);
End component;
Signal high : std_logic ; -- this is representing logic high
Begin
High<= '1';
U1: and1 port map ( x ,y ,high ,z );
```

If an input on a component is not in use, the signal should be connected to Vcc or Gnd such that overall functionality should not change.

c. Component Specification

- If the VHDL simulator is to find the declared component, use of the following command.

Syntax:

for: <component_name>use entity library. <entity name> (architecture name)

For u1: xor1 use entity work.xor1 (t1);

Practice Exercises

1. Write behavioural model for 1x8 **Demultiplexer** using case when statements.
2. Write VHDL code for SR latch in dataflow style.
3. For full adder circuit is shown below. Write structural model for the same.