

---

# Concurrent Signal Assignments

## Module 3

# Concurrent Signal Assignment

---

- Signals that appear outside of a process
- Event-triggered, when an event (change in value) occurs on one of the signals in the expression
- Three types
  - concurrent signal assignment
  - conditional signal assignment
  - selected signal assignment

# Concurrent Signal Assignment (cont'd)

---

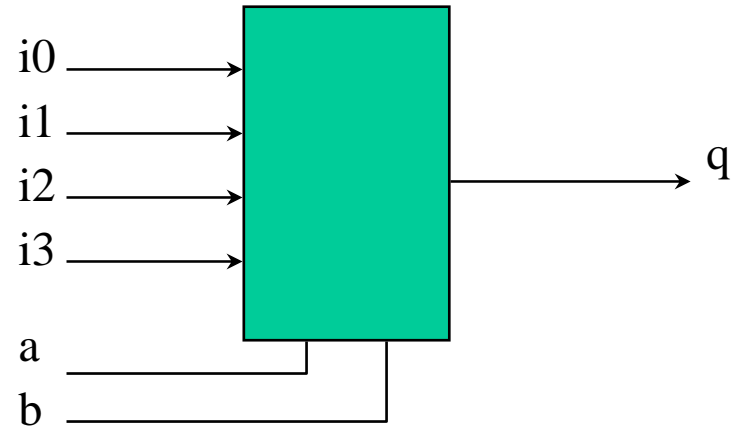
```
ARCHITECTURE example OF full_adder
BEGIN
    sum <= a XOR b XOR c;
    temp <= a AND b;
    cout <= temp AND c;
END example;
```

- Any time an event occurs on signals a, b, or c the concurrent signal assignments are re-executed.
- Signals in expression act like sensitivity list for process
- Equivalent process statement:

```
PROCESS (a, b, c)
BEGIN
    sum <= a XOR b XOR c;
END PROCESS;
```

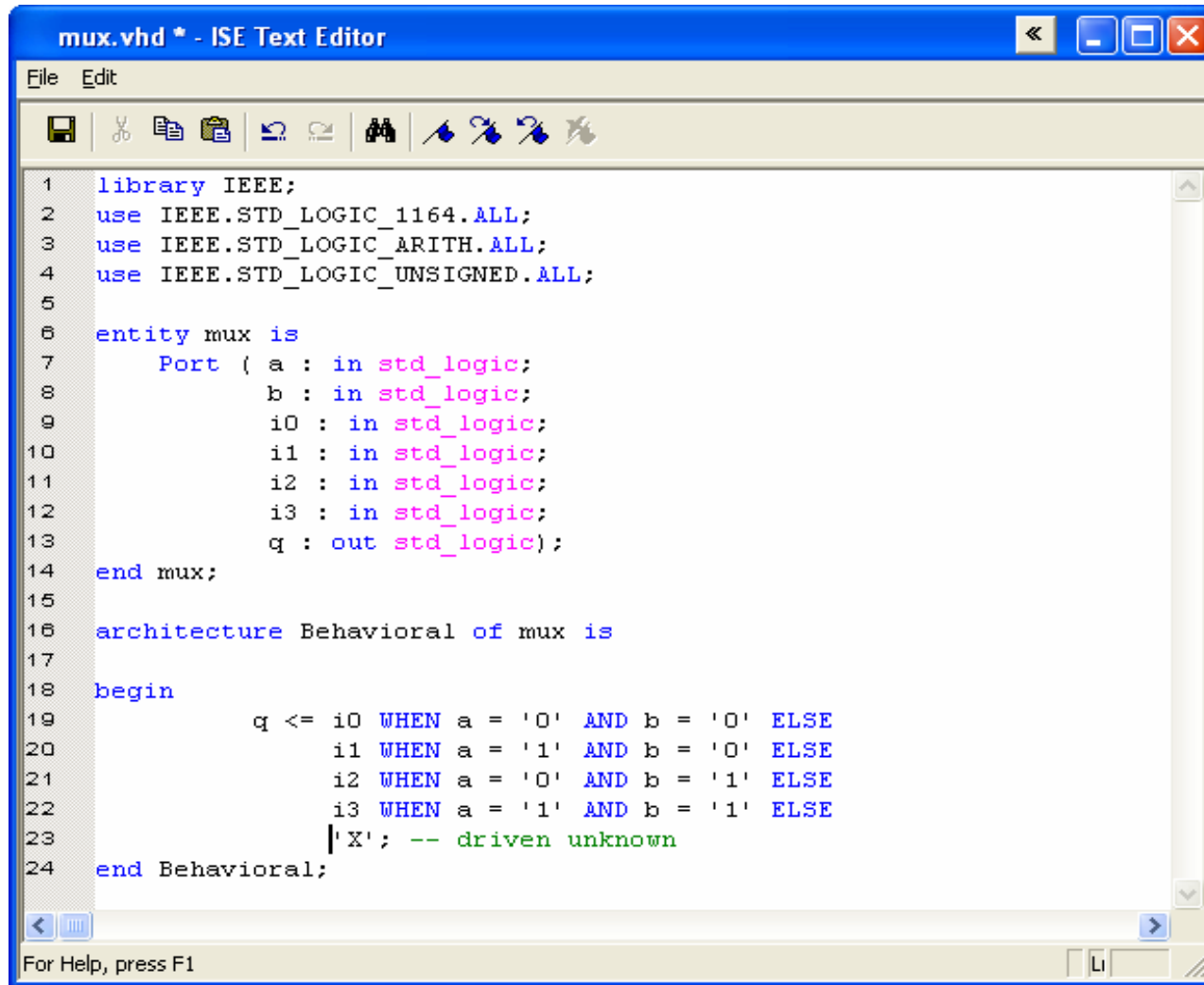
# Conditional Signal Assignment

- Selects different values for the target signal
  - priority associated with series of **WHEN .. ELSE**
- Similar to an IF statement
  - example multiplexer:



```
ARCHITECTURE example OF mux IS
BEGIN
    q <= i0 WHEN a = '0' AND b = '0' ELSE
        i1 WHEN a = '1' AND b = '0' ELSE
        i2 WHEN a = '0' AND b = '1' ELSE
        i3 WHEN a = '1' AND b = '1' ELSE
        'X';      -- driven unknown
END example;
```

# Complete VHDL



```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity mux is
7      Port ( a : in std_logic;
8            b : in std_logic;
9            i0 : in std_logic;
10           i1 : in std_logic;
11           i2 : in std_logic;
12           i3 : in std_logic;
13           q : out std_logic);
14 end mux;
15
16 architecture Behavioral of mux is
17
18 begin
19     q <= i0 WHEN a = '0' AND b = '0' ELSE
20         i1 WHEN a = '1' AND b = '0' ELSE
21         i2 WHEN a = '0' AND b = '1' ELSE
22         i3 WHEN a = '1' AND b = '1' ELSE
23         'X'; -- driven unknown
24 end Behavioral;
```

For Help, press F1

# Synthesize

The screenshot displays the Xilinx Project Navigator interface. The main window title is "Xilinx - Project Navigator - C:\ee574\mux\mux.isc - [mux.vhd]". The menu bar includes File, Edit, View, Project, Source, Process, Simulation, Window, and Help. The toolbar contains various icons for file operations and simulation. The "Sources in Project" pane shows a tree view with "mux.isc" and "xc3s200-4ft256" containing "mux-behavioral (mux.vhd)". The "Processes for Source: 'mux-behavioral'" pane shows a list of tasks: Edit Constraints (Text), Synthesize - XST (checked), View Synthesis Report, View RTL Schematic, View Technology Schematic, Check Syntax (checked), Generate Post-Synthesis Simulation, Implement Design, and Generate Programming File. The code editor shows the following VHDL code:

```
26 ---- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;
29
30 entity mux is
31     Port ( a : in std_logic;
32           b : in std_logic;
33           i0 : in std_logic;
34           i1 : in std_logic;
35           i2 : in std_logic;
36           i3 : in std_logic;
37           q : out std_logic);
38 end mux;
39
40 architecture Behavioral of mux is
41
42 begin
43     q <= i0 when a = '0' and b = '0' else
44         i1 when a = '1' and b = '0' else
45         i2 when a = '0' and b = '1' else
46         i3 when a = '1' and b = '1' else
47         'X';
48
49 end Behavioral;
50
```

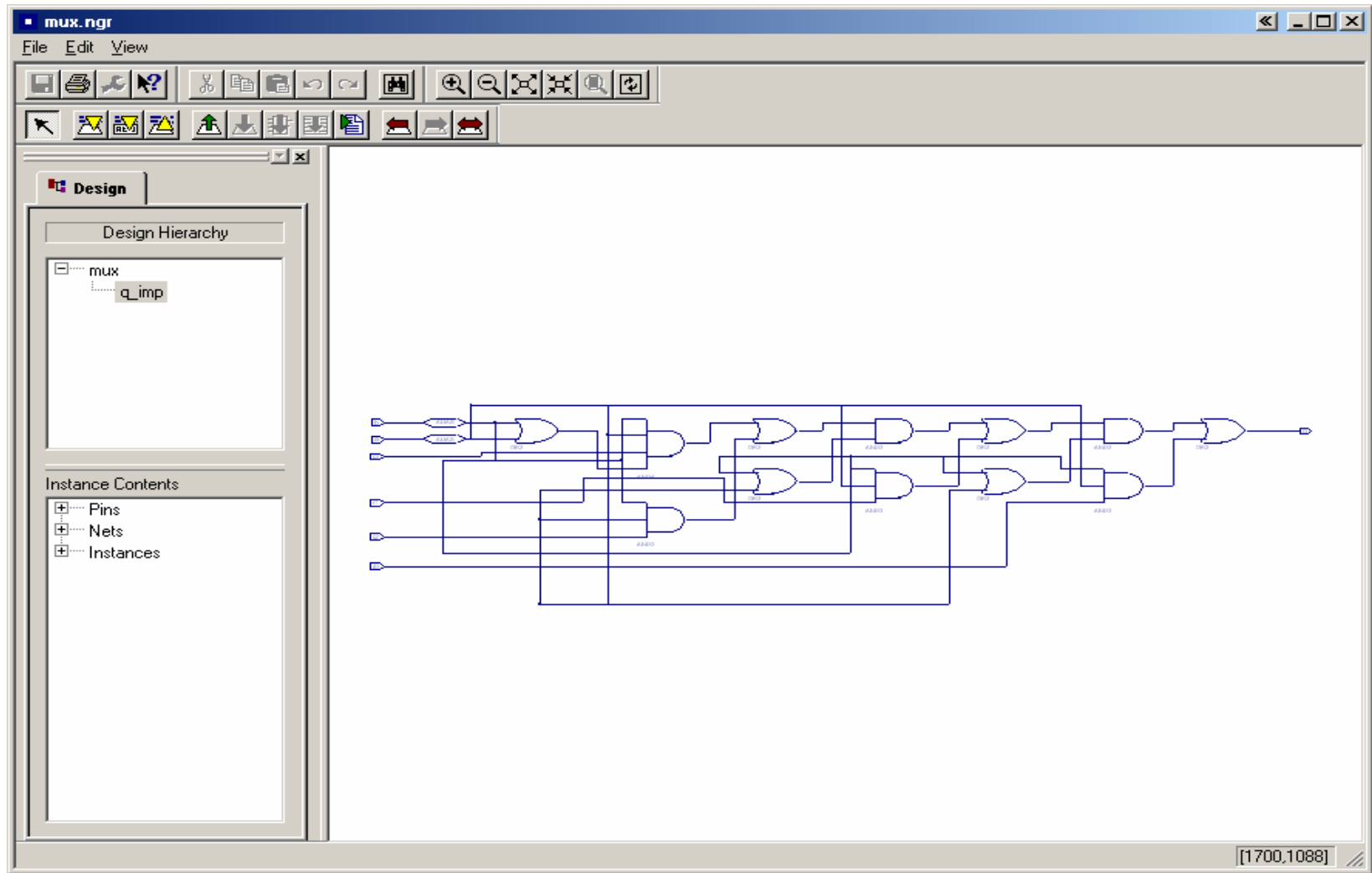
The Console pane at the bottom shows the following output:

```
No clock signals found in this design
Timing Summary:
Speed Grade: -4

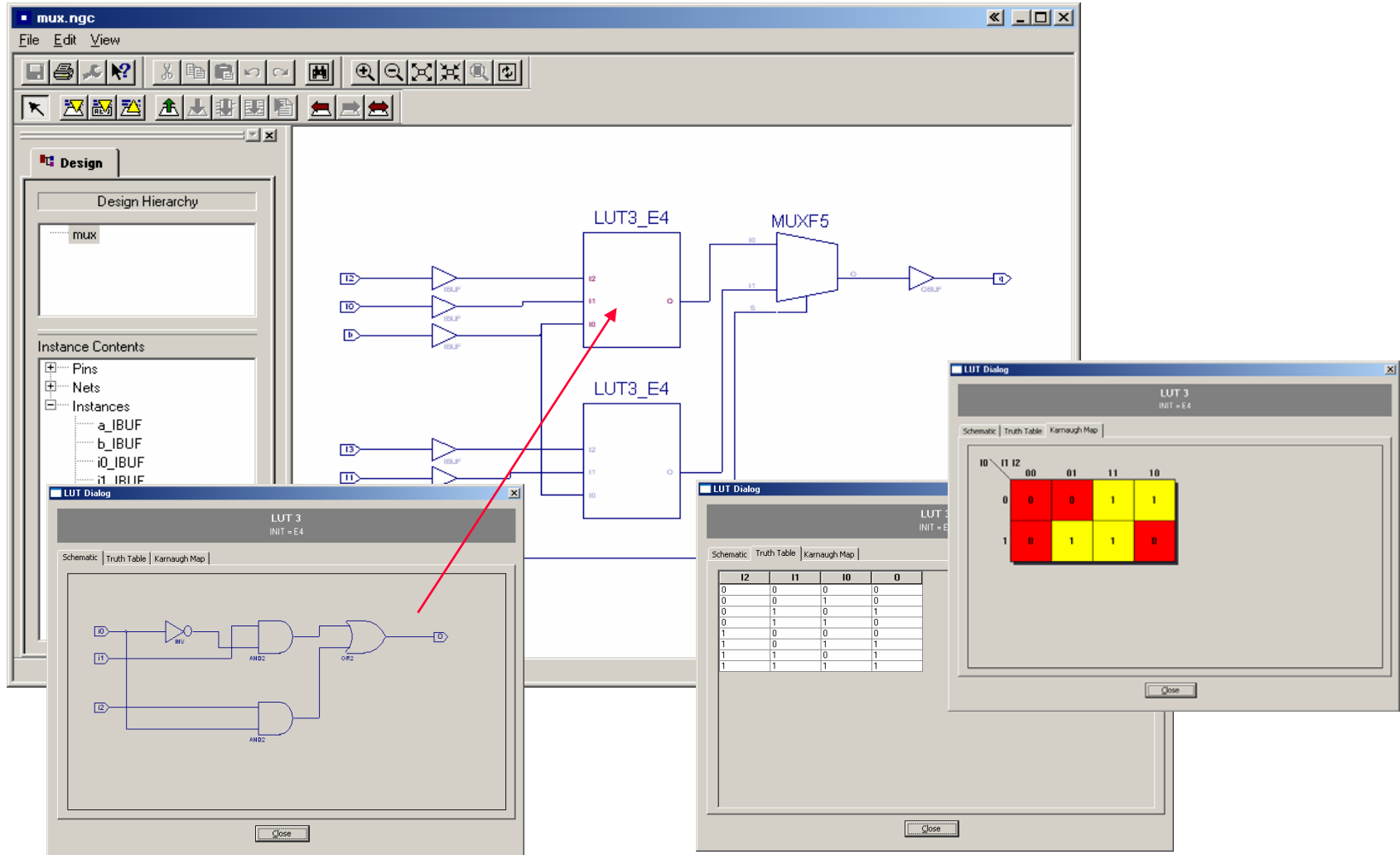
Minimum period: No path found
Minimum input arrival time before clock: No path found
Maximum output required time after clock: No path found
Maximum combinational path delay: 9.393ns
-----
```

The status bar at the bottom indicates "Process 'Synthesize - XST' is up to date." and "Ln 47 Col 9 | VHDL".

# Synthesis Results – RTL Schematic



# Synthesis Results – Technology Schematic



# FPGA Editor

The screenshot displays the Xilinx FPGA Editor interface. The main workspace shows a logic array with a grid of components. A window titled "Block1 - View Comp q\_OBUF at Site SLICE\_X10Y0" is open, showing a detailed logic diagram of the component. A "List1" window is also open, displaying a table of components. The status bar at the bottom shows the component name and site information.

	Name	Site	Type	#Pins	Hited
1	a	R5	IOB	1	no color
2	b	M7	IOB	1	no color
3	i0	N6	IOB	1	no color
4	i1	P6	IOB	1	no color
5	i2	R6	IOB	1	no color
	i3	M6	IOB	1	no color
	q	N5	IOB	1	no color
	q_OBUF	SLICE_X10Y0	SLICEL	8	no color

comp "q\_OBUF", site "SLICE\_X10Y0", type = SLICEL (RPM grid X18Y4)

# Selected Signal Assignment

---

- Selects different values for a target signal
  - no priority associated with input conditions
- Similar to a CASE statement

```
ARCHITECTURE example OF mux IS
BEGIN
    WITH sel SELECT
        q <= i0 WHEN "00",
            i1 WHEN "01",
            i2 WHEN "10",
            i3 WHEN "11",
            'X' WHEN OTHERS;
END example;
```

# Selected Signal Assignment (cont'd)

---

- Another example
  - using enumerated type

```
ARCHITECTURE example OF alu IS
    TYPE op_code IS (and_op, or_op, add_op, sub_op);
    SIGNAL inst : op_code;

BEGIN

    inst <=          -- inst gets some value;

    WITH inst SELECT
        result <= a AND b WHEN and_op,
        a OR b WHEN or_op,
        a + b WHEN add_op,
        a - b WHEN sub_op;

END example;
```

# Selected Signal Assignment (cont'd)

---

- Another example

```
ARCHITECTURE example OF sel IS
BEGIN
    WITH x SELECT
        z <= a WHEN 1 | 2,
            b WHEN 3 TO 6,
            c WHEN OTHERS;
END example;
```

# Selected Signal Assignment (cont'd)

---

- Equivalent Process Statement

```
ARCHITECTURE example OF sel IS
BEGIN
    PROCESS(x, a, b, c)
    BEGIN
        CASE x IS
            WHEN 1 | 2 =>
                z <= a;
            WHEN 3 TO 6 =>
                z <= b;
            WHEN OTHERS =>
                z <= c;
        END CASE;
    END PROCESS;
END example;
```

# Aggregates

---

- A set of comma separated elements enclosed in parenthesis
- Denotes the value of a composite type (array or record)
- Value specified by listing the value of each element

```
bus_a <= "10110111";
```

```
bus_b <= (OTHERS => '0');
```

```
bus_c <= (2 => '0', 6 => '0', OTHERS => '1');
```

# Encoder - Xilinx Implementation

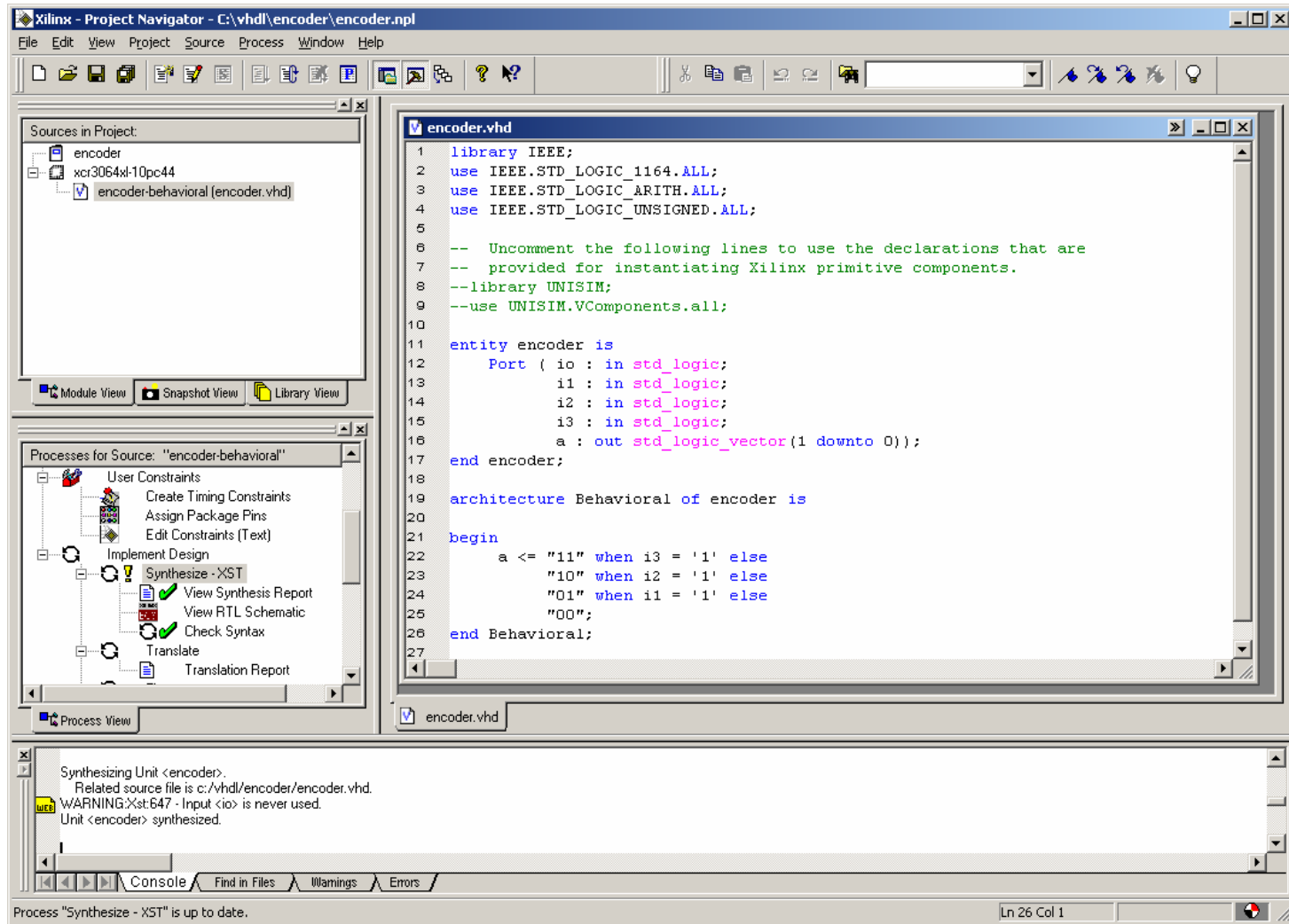
---

- Priority Encoder using conditional signal assignment
- Priority order determined by **WHEN .. ELSE** sequence
  - similar to if-then-else statement
  - (use selected signal assignment if no priority required)

```
ENTITY encoder IS      -- 4-line to 2-line priority encoder
    PORT(i0, i1, i2, i3 : IN std_logic;
          a : OUT std_logic_vector (1 DOWNTO 0));
END encoder;

ARCHITECTURE example OF encoder IS
BEGIN
    a <= "11" WHEN i3 = '1' ELSE
          "10" WHEN i2 = '1' ELSE
          "01" WHEN i1 = '1' ELSE
          "00";
END example;
```

# Synthesis results



The screenshot shows the Xilinx Project Navigator interface. The main window displays the source code for `encoder.vhd`. The code defines an entity `encoder` with three input ports (`io`, `i1`, `i2`) and one output port (`a`). The architecture `Behavioral` implements the logic for the output `a` based on the inputs.

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.ALL;
4 use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6 -- Uncomment the following lines to use the declarations that are
7 -- provided for instantiating Xilinx primitive components.
8 --library UNISIM;
9 --use UNISIM.VComponents.all;
10
11 entity encoder is
12     Port ( io : in std_logic;
13           i1 : in std_logic;
14           i2 : in std_logic;
15           i3 : in std_logic;
16           a : out std_logic_vector(1 downto 0));
17 end encoder;
18
19 architecture Behavioral of encoder is
20
21 begin
22     a <= "11" when i3 = '1' else
23         "10" when i2 = '1' else
24         "01" when i1 = '1' else
25         "00";
26 end Behavioral;
27
```

The console window at the bottom shows the synthesis progress and a warning:

```
Synthesizing Unit <encoder>.
Related source file is c:/vhdl/encoder/encoder.vhd.
WARNING:Xst:647 - Input <io> is never used.
Unit <encoder> synthesized.
```

Process "Synthesize - XST" is up to date.

# Synthesis results – RTL Schematic

