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)

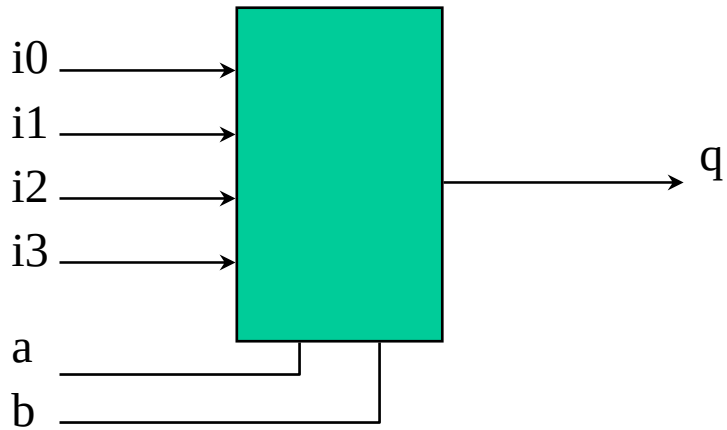
```
ARCHITECTURE example OF full_adder
BEGIN
    sum <= a XOR b XOR c;
    temp <= a AND b;
    cout <= temp AND c;
END example;
```

- 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

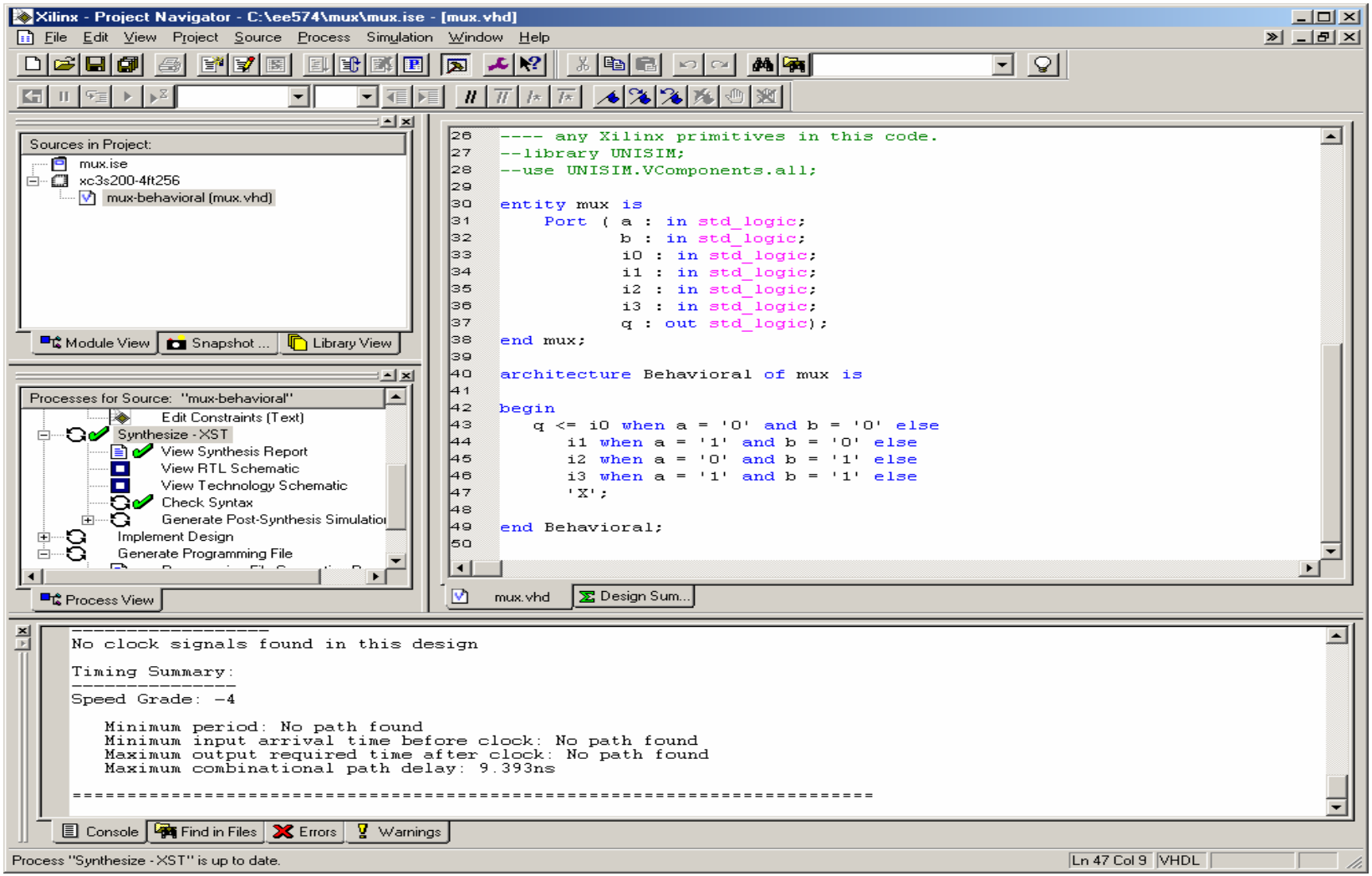
mux.vhd * - ISE Text Editor

File Edit

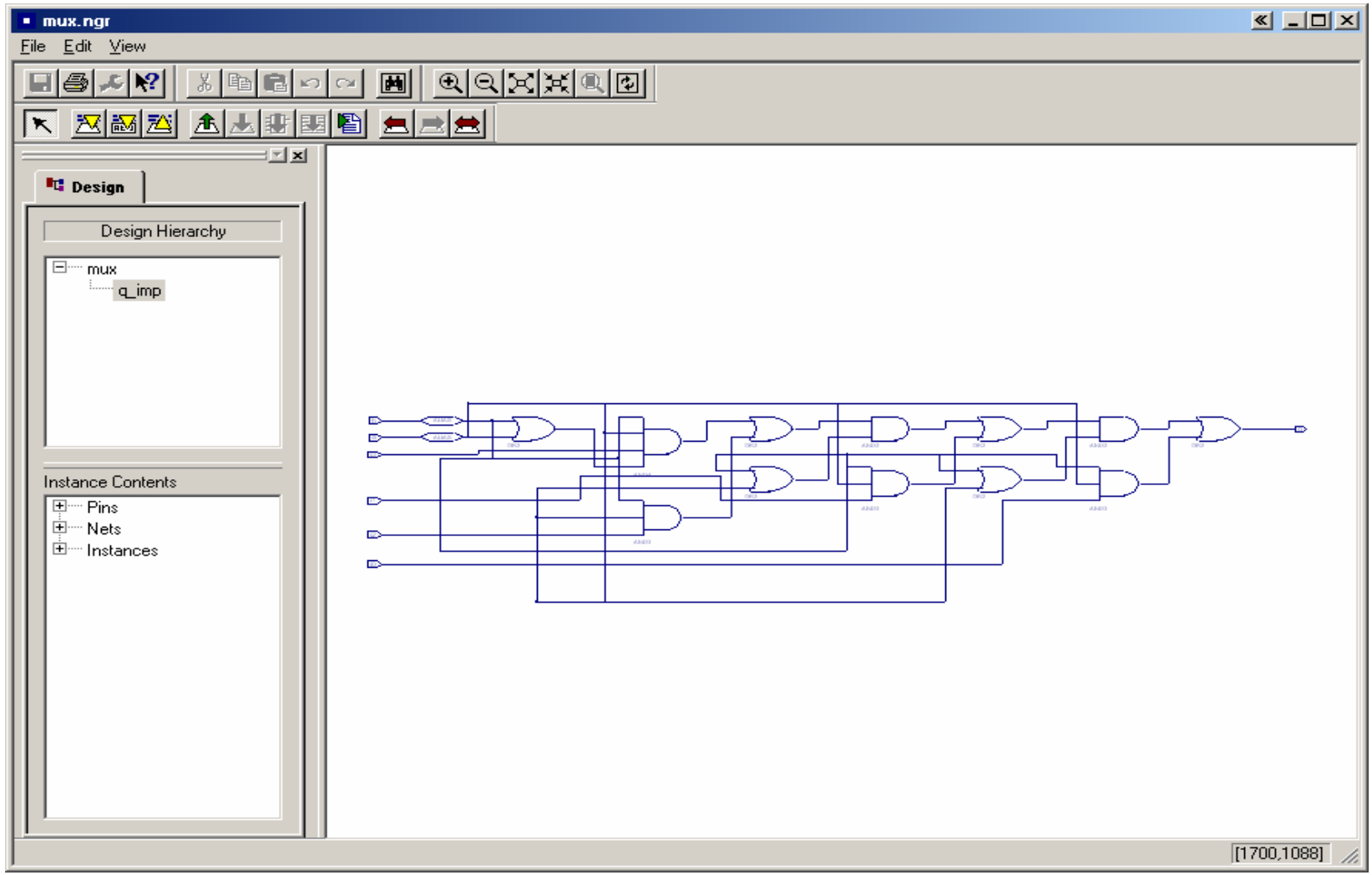
```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity mux is
7      Port ( a : in std_logic;
8             b : in std_logic;
9             i0 : in std_logic;
10            i1 : in std_logic;
11            i2 : in std_logic;
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

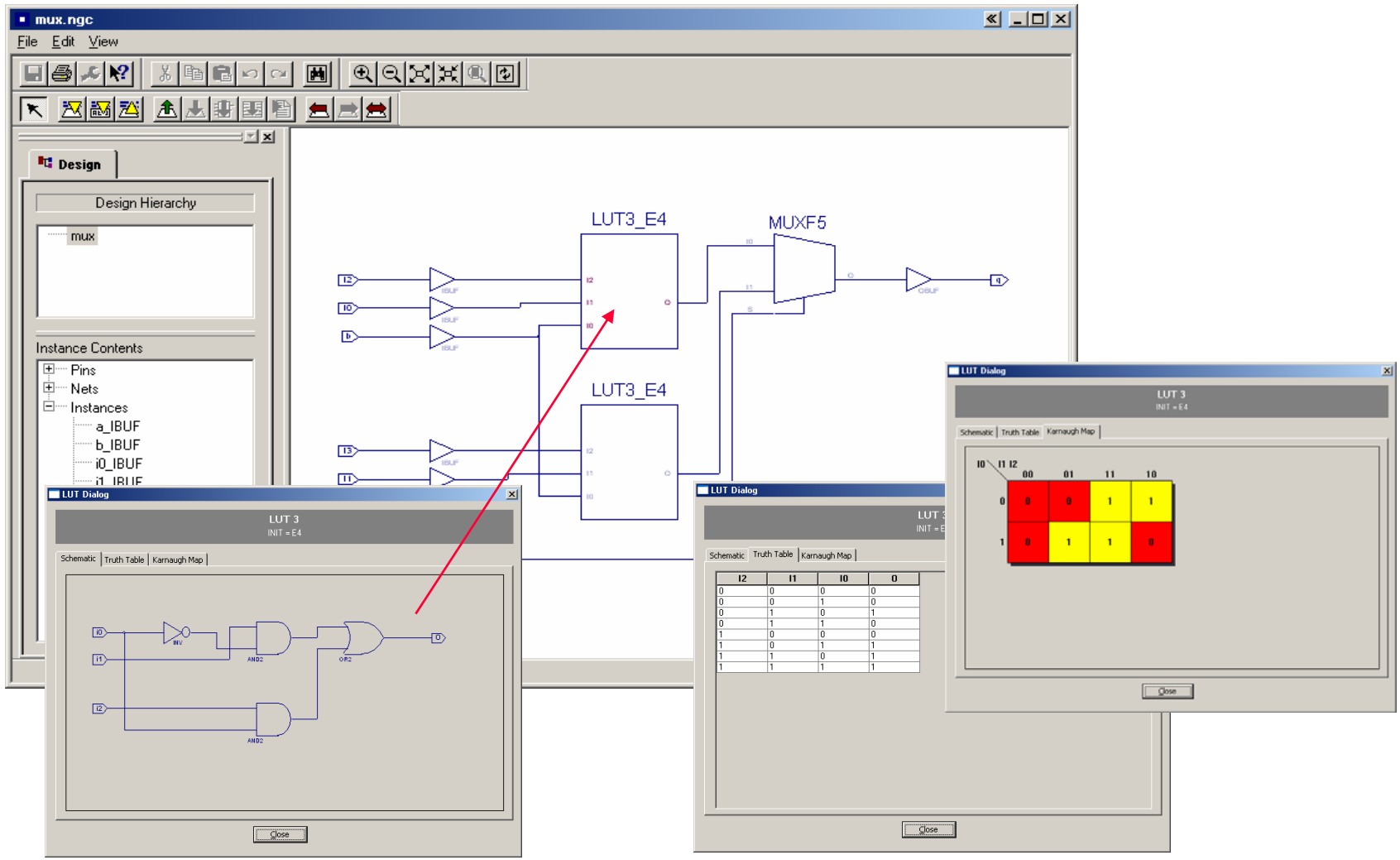
Synthesize


Xilinx - Project Navigator - C:\ee574\mux\mux.ise - [mux.vhd]
File Edit View Project Source Process Simulation Window Help
Sources in Project:
mux.ise
xc3s200-4ft256
mux-behavioral (mux.vhd)
Module View
Snapshot ...
Library View
Processes for Source: "mux-behavioral"
Edit Constraints (Text)
Synthesize - XST
View Synthesis Report
View RTL Schematic
View Technology Schematic
Check Syntax
Generate Post-Synthesis Simulation
Implement Design
Generate Programming File
Process View
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity mux is
Port (a : in std_logic;
b : in std_logic;
i0 : in std_logic;
i1 : in std_logic;
i2 : in std_logic;
i3 : in std_logic;
q : out std_logic);
end mux;
architecture Behavioral 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';
end Behavioral;
mux.vhd
Design Sum...
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
Console
Find in Files
Errors
Warnings
Process "Synthesize - XST" is up to date.
Ln 47 Col 9
VHDL

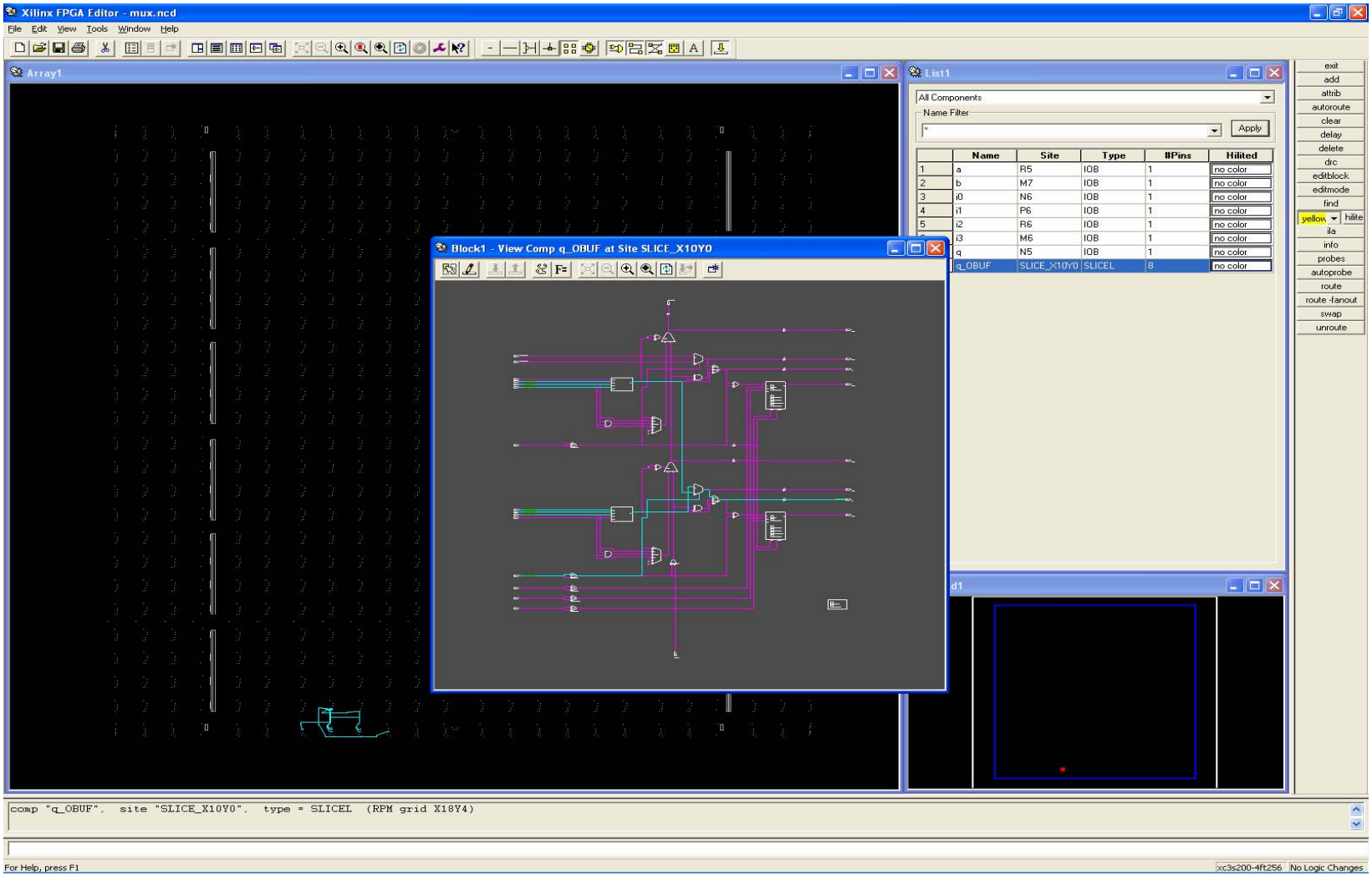
Synthesis Results – RTL Schematic



Synthesis Results – Technology Schematic



FPGA Editor


Xilinx FPGA Editor - mux.ncd
File Edit View Tools Window Help
Array1
List1
All Components
Name Filter
Apply
Name Site Type #Pins Hilited
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
exit
add
attrib
autoroute
clear
delay
delete
drc
editblock
editmode
find
yellow
hilite
ila
info
probes
autoprobe
route
route -fanout
swap
unroute
Block1 - View Comp q_OBUF at Site SLICE_X10Y0
d1
comp "q_OBUF", site "SLICE_X10Y0", type = SLICEL (RPM grid X18Y4)
For Help, press F1
xc3s200-4ft256
No Logic Changes

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’);
```

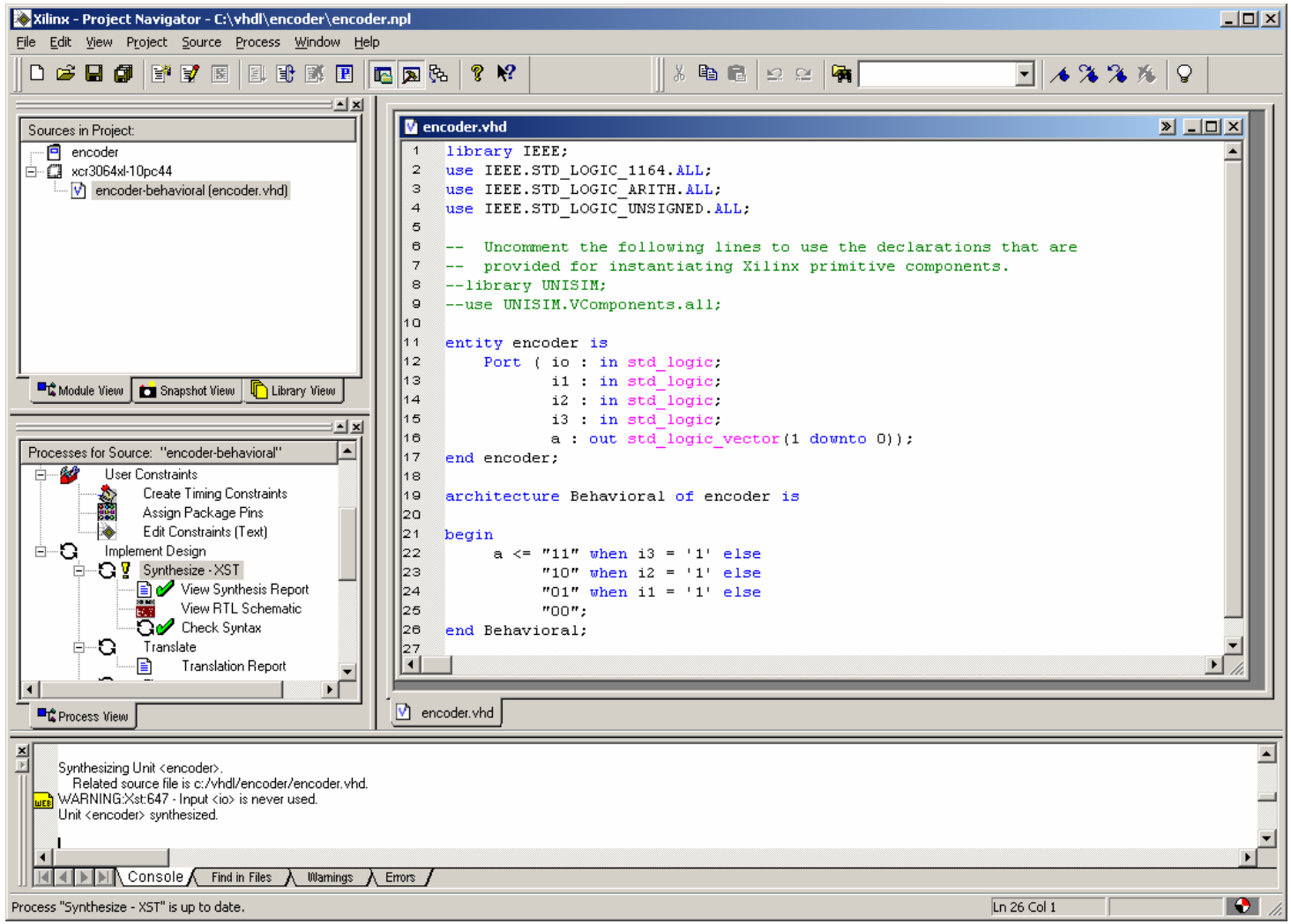
Encoder - Xilinx Implementation

- 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, i2, 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



Synthesis results – RTL Schematic