

Logic Synthesis

VHDL description

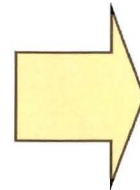
```
architecture MLU_DATAFLOW of MLU is
    signal A1:STD_LOGIC;
    signal B1:STD_LOGIC;
    signal Y1:STD_LOGIC;
    signal MUX_0, MUX_1, MUX_2, MUX_3: STD_LOGIC;

begin
    A1<=A when (NEG_A='0') else
        not A;
    B1<=B when (NEG_B='0') else
        not B;
    Y1<=Y1 when (NEG_Y='0') else
        not Y1;

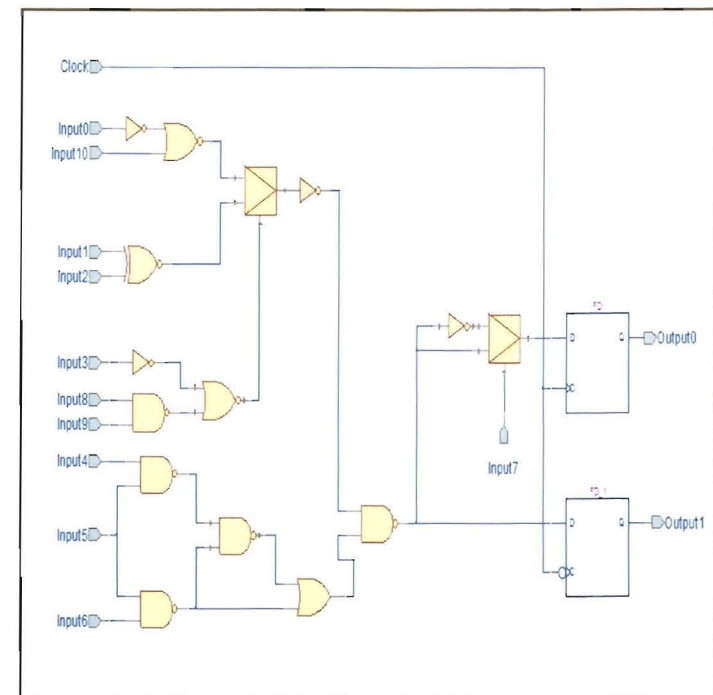
    MUX_0<=A1 and B1;
    MUX_1<=A1 or B1;
    MUX_2<=A1 xor B1;
    MUX_3<=A1 xnor B1;

    with (L1 & L0) select
        Y1<=MUX_0 when "00",
            MUX_1 when "01",
            MUX_2 when "10",
            MUX_3 when others;

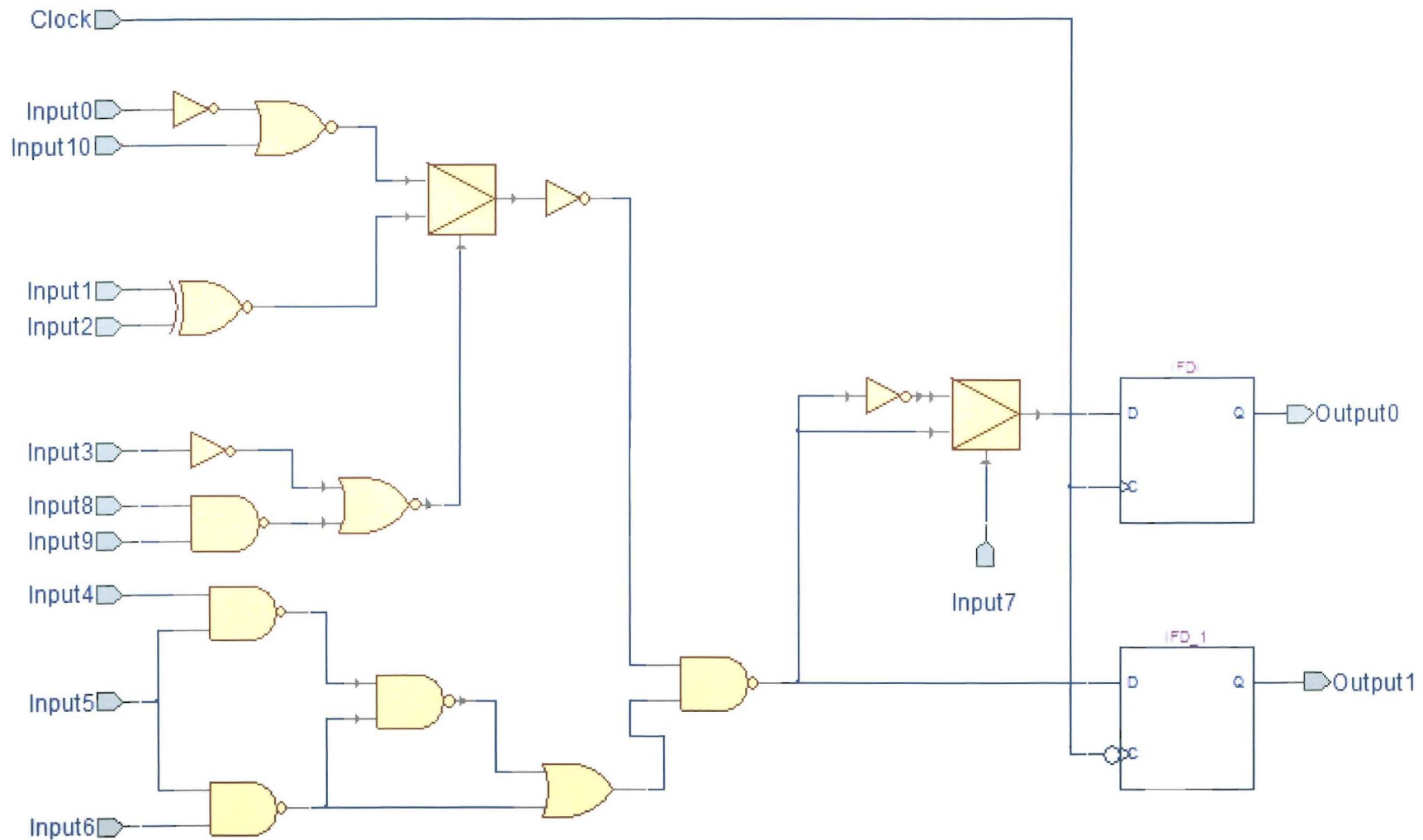
end MLU_DATAFLOW;
```



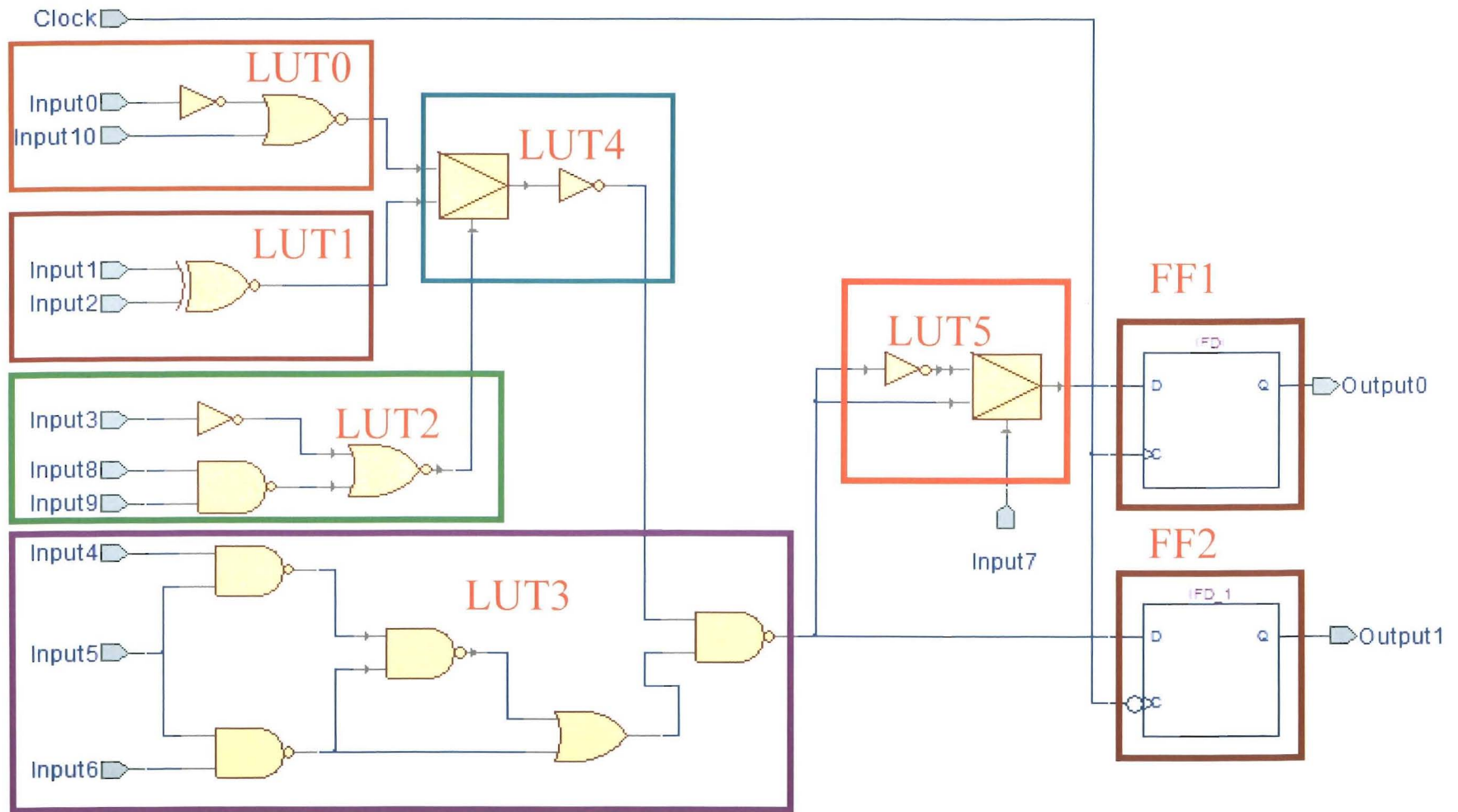
Circuit netlist



Circuit netlist



Mapping



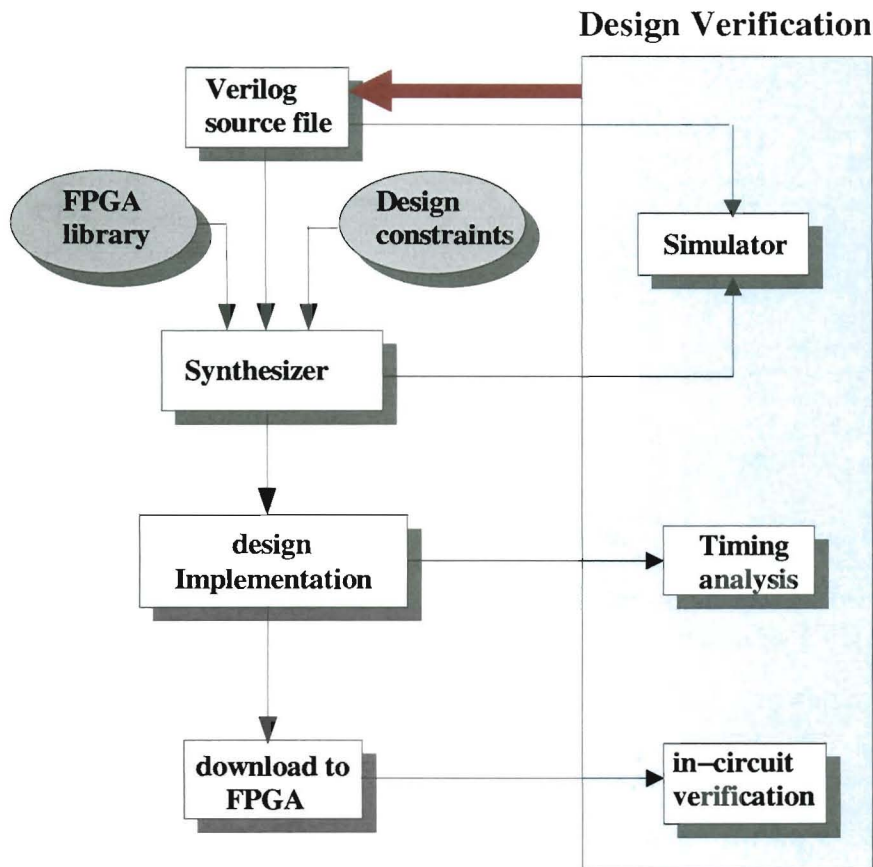
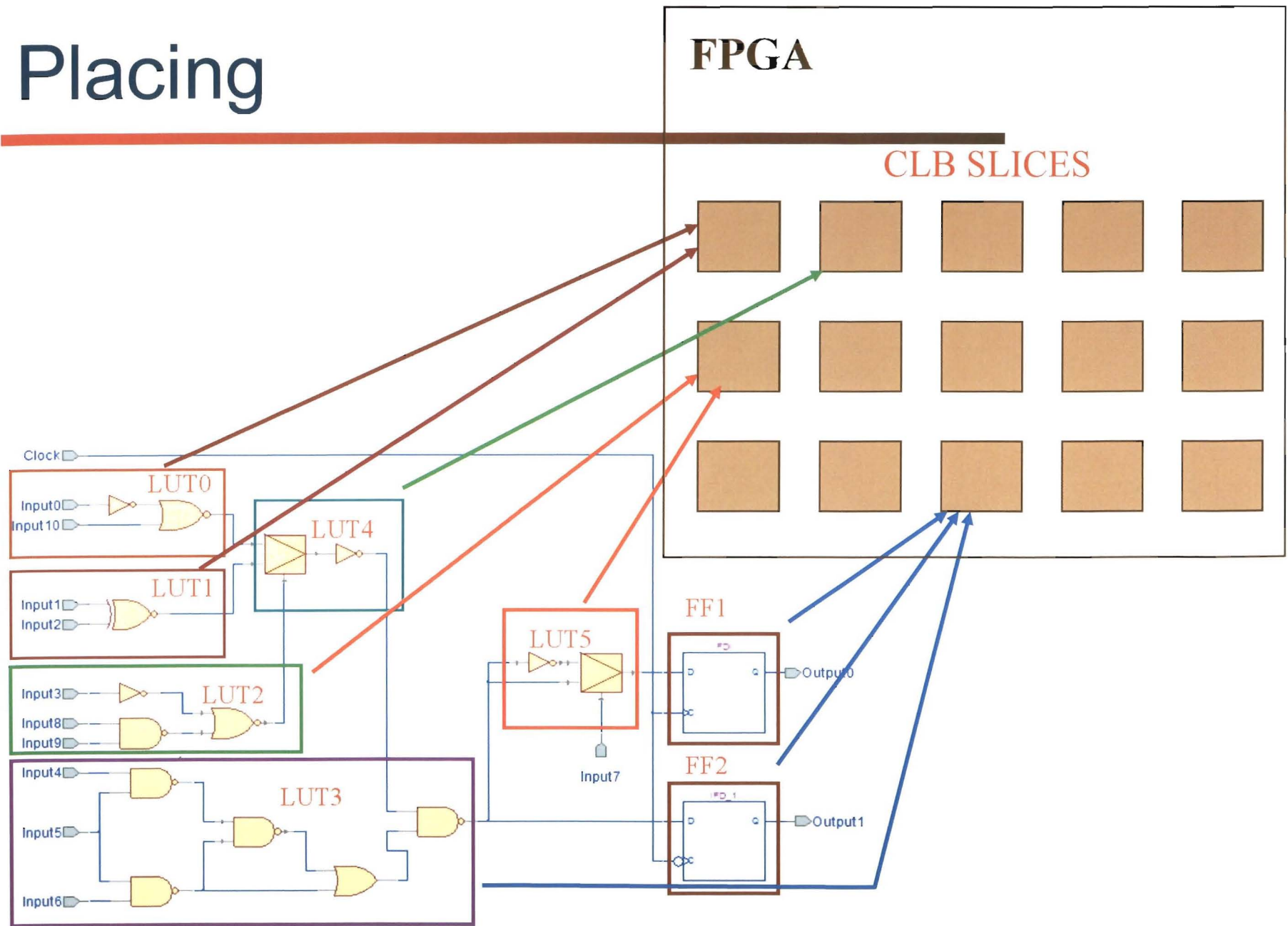


Figure 5.1. The FPGA design flow. Verilog is shown as the HDL, but VHDL could also be used. All tasks in the shading box perform design verification. The red arrow near the top means any failure during the verification must be handled by modifying the Verilog source code and then resynthesizing the design. Although FPGA synthesis is shown, the procedure is identical for ASIC design if the FPGA component library is replaced with a standard cell library.

Placing



Routing

FPGA

Programmable Connections

