Birla Institute of Technology and Science – Pilani
Pilani | K.K. Birla Goa

First Semester 2019-2020
CS G553-Reconfigurable Computing
Component: Regular (Closed Book)
Duration: 180 min Comprehensive Examination Max. Marks: 35

Instructions: Make suitable assumption and justify if required.

Q1. What are the objectives of decomposition and partitioning in library based technology mapping? **[01]**

Q2. Explain Boolean matching and structural matching with an example **[01]**

Q3. Name the preprocessing steps required for Chortle technology mapping algorithm? Explain their requirement with an example. **[01]**

Q4. What is configuration caching? **[01]**

Q5. Will it be possible to implement two 5-input one-output function using one ALTERA ALM? What are the constraints if it is possible? **[01]**

Q6. List any two drawbacks of KAMER online placement algorithm? **[01]**

Q7. Consider the design of a 3-input FPGA lookup table (LUT). An abstracted view of the 3-LUT is shown in figure-1. The ports, a, b, c, and y, are the data inputs and output, respectively. The input CIN stands for "configuration input", and CCLK stands for "configuration clock". The configuration clock is used at FPGA initialization time to shift in the LUT contents, one bit per clock cycle, over the CIN port. **[03+01]**

a) Draw a circuit diagram (logic) for the internal implementation of the 3-LUT, including configuration loading. Remember to label all inputs and outputs.

b) Imagine now that the 3-LUT is programmed to implement the following function: ~a&b&c. Label the appropriate nodes in your circuit diagram with the correct configuration values.

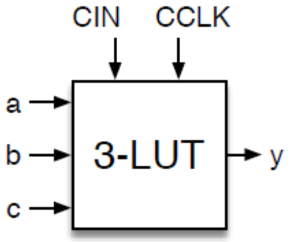

Figure-1

Q8. Imagine you are given a "mystery" FPGA programmed to perform a particular set of functions. We know the following about the FPGA:

1. The device contains a collection of N-LUTs (the value of N is part of the mystery). The LUTs are numbered 0, 1, 2, ... Each LUT in the FPGA has the same number of inputs (same N).
2. Each LUT has an output labeled yi, where i is the LUT number, and inputs labeled xi_j, where i is the LUT number and j is the input number.
3. For programming, the LUTs are connected in a shift register. They are programmed with a configuration bit-stream shifted in from one LUT to the next.
4. LUT 0 is programmed as an inverter, therefore y0 = NOT(x0_0). Futhermore, LUT 0 is the only LUT programmed as an inverter.

5. We were able to recover a fragment of the bit stream, shown here: 0xE9AC96017F88FF55.

a) How many LUTs would the above bit stream program?

b) For each of the LUTs, draw a circuit using simple logic gates to represent the function it is programmed to implement. Label all inputs and outputs. **[09]**

Q9. Neatly sketch the circuit that corresponds to the following Verilog code: **[04]**

```
module foo (CE, X, CLK, RST, OUT);
      input CE, CLK, X, RST;
      output OUT;
      reg [3:0] Q1;
      reg [3:0] Q2;
      assign OUT = ^Q1;
      always @ (posedge CLK)
            if (RST) begin
                  Q1 <= 4'h5;
                  Q2 <= 0;
            end
            else
                  if (CE) begin
                        if (X) Q2 <= Q1<<1;
                        else Q2 <= Q1;
                        Q1 <= Q2;
                  end
endmodule // foo
```

Q10. The combinational part of a digital function (Y) given below is to be implemented on a FPGA. The library elements are LUTs with 5 inputs and one output. Complete the required number of LUTs by applying the Chortle-crf LUT-Technology mapping algorithm. Show the output during all the stages of the algorithm. Y=AB+CD+DE+FGH+IJKL+MN+OP **[04]**

Q11. For the function given below, 1, 2, …., 16 represent the operation nodes v1, v2,….,v16 respectively.

$$x = \left((a \overset{1}{+} b) \overset{8}{\times} (c \overset{2}{\times} d)\right) \overset{14}{-} \left((c \overset{3}{+} d) \overset{9}{\times} (e \overset{4}{+} f)\right),$$

$$y = \left((c \overset{2}{\times} d) \overset{10}{+} (e \overset{4}{+} f)\right) \overset{15}{+} \left((e \overset{5}{-} f) \overset{11}{-} (g \overset{6}{\times} h)\right) \text{ and}$$

$$z = \left((a \overset{7}{-} b) \overset{12}{+} (g \overset{6}{\times} h)\right) \overset{16}{\times} \left((a \overset{7}{-} b) \overset{13}{\times} (e \overset{5}{-} f)\right)$$

Assume that multiplication require 100 clock delay, add/sub require 50 clock delay, data transmission delay is negligible.

a) If one multiplier and two adder/subtractor are available, schedule using list scheduling with the priority as depth of the node. **[02]**

b) For a FPGA with 520 LUTs multiplication, addition and subtraction consumes 200, 100 and 120 LUTs respectively. Use list scheduling for temporal partitioning the above function and also find the quality of the partitioning. **[04+02]**