University of Tehran
Electrical and Computer Engineering Department

Computer Aided Design of Digital Systems (CAD)
Fall 99
Assignment Description

CA1: Review on logic design and introduction to FPGA bit-stream generation

In this assignment, you will review the concepts of programmable logic design. This assignment has three phases: (1) designing a combinational and sequential logic, (2) modeling the logic with Verilog, and (3) implementing the final design on FPGA-based programmable logic cells.

Phase 3

PROJECT DESCRIPTION

You have already designed the 4x4-bit multiplier in Phase 2 with gate-level Verilog. In Phase 3 you should synthesize the 4x4-bit multiplier on sample programmable logic cells of an FPGA. The logic cells are shown in Figure 1.

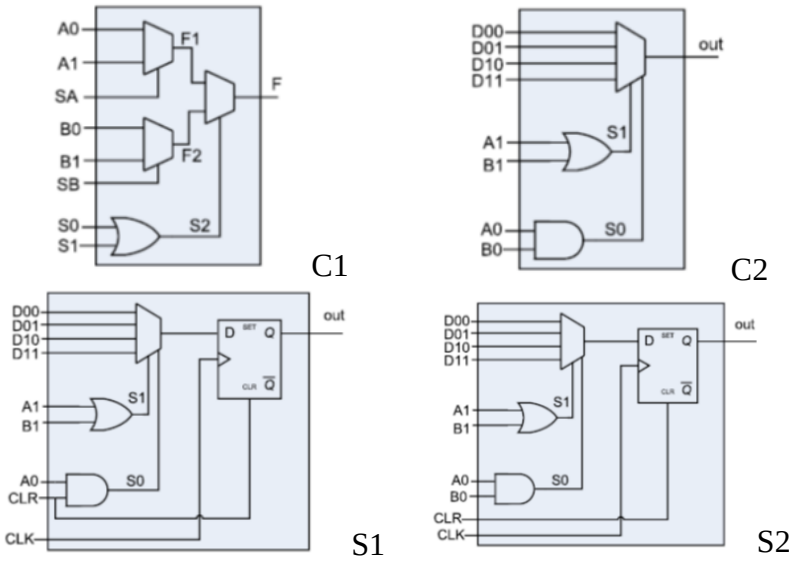

Figure 1. Programmable logic cells of Actel Logic Module.

In the design of target 4x4-bit Multiplier **you must only use C1, C2, S1 and S2** for designing the Datapath and the FSM (controller). You should also verify the functionality of the designed multiplier by a testbench considering all possible combinations of the two 4-bit inputs (256 combinations) by two for loops on the 4-bit inputs and then compare the output of your design with the golden_result=in1 * in2 and check if there is any mismatch between the results of our multiplier and the golden_result.

You could get the complete score of this phase if the multiplier output is functionally correct for all combinations of the inputs. However, this phase has a bonus score for groups that would optimize the circuit in terms of the area of the design. For calculating the total area of the design, simply stack up the area of all of the instantiated modules. Area of each module in terms of unit area is shown in the following table. To optimize the design, you should use as less modules as possible, therefore take advantage of the architecture of the modules and try to use the modules efficiently.

Module	C1	C2	S1	S2
Area	7	8	15	15

The bonuses of the top 5 high rank groups are calculated according to the following table.

Rank	1	2	3	4	5
Bonous	+15%	+8%	+4%	+2%	+1%

Deliverables:

1. The 4x4 multiplier code in Verilog (based on instantiating the introduced modules)
2. The calculated area of the design.
3. A testbench that applies all input combinations and compares the output of the code with the golden result.

Note:

- The inputs would change after the start pulse; therefore, you should locally register the inputs to protect them against probable changes in the inputs during multiply operation.
- This assignment can be done in groups (maximum 2 students)