Experiment 8

State Machines: A Read-Modify-Write Circuit

In this lab, we are going to design a data path and controller for a Read-Modify-Write Circuit.

Tasks

You are required to build a datapath and controller of a circuit that manipulates a certain bit value of a 8-bit number placed in a register and display the value on the seven-segments.

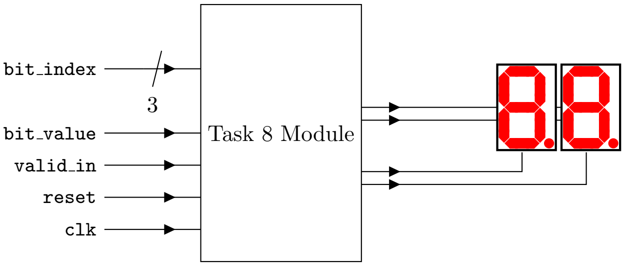

Fig. 8.1: Lab task modular circuit.

The user provides the index of the bit to be modified and the new bit value. On getting the valid-in signal, informing about the validity of the data, the read-modify-write process will initiate. The 8-bit (byte) data will be read from the register, which is then fed for modification. In the modify stage, the value of the bit at the bit-index given is manipulated to the given bit-value. **We do not have bit-wise access.** Rather a whole byte is read at a time. This modified value is then written back to the register. Remember that once the valid-in signal has been received, then until the entire process has been completed (a valid-out signal has been generated by the controller), no new input value will be considered.
Your datapath design will be comprise of a 8-bit register and a modifier circuit(Fig. 8.2).

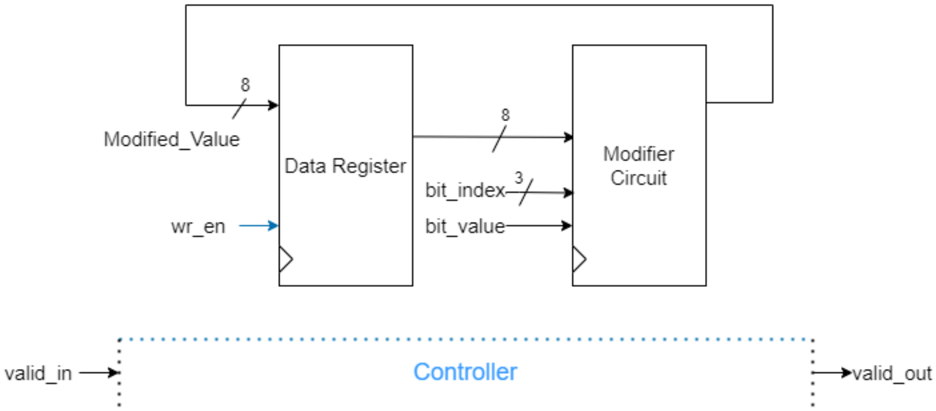

Fig. 8.2: Read-Modify-Write Modular Circuit

Requirement:

One of the elements of your modifier circuit will be a shift register for creating the mask for bit manipulation.

Note:

Use of latches are not allowed in the design of synchronous circuits for the lab. Synchronous circuits must be designed with flip-flops.

Deliverables

1. A report containing the following items:
   (a) State Transition Graph for the controller of Read-Modify-Write Circuit.
   (b) The designed datapath circuit along-with control signals. The design should also show the labels of wires, along with the label for the width of the line for multi-bit wires.
   (c) Implementation maximum combinational delay: Read the post implementation static timing report and identify the path with the maximum combinational delay?
   (d) Resource Utilization Report: Read the synthesis report and identify how many resources in the FPGA such as lookup tables (LUTs), input/output (IOs), etc., has been utilized.
2. Simulation of your designed module on QuestaSim. Show the simulation to the instructor.
3. FPGA Implementation: Synthesize the circuit for Nexys A7 FPGA available in the lab. Tie inputs to the switches and outputs to the seven-segment displays available on the board.

The collaboration between students is encouraged, but blind code sharing/copying is not allowed. If you are unable to explain anything in your code, it will be assumed you have copied it. **Copied code will be given ZERO marks.** So make sure you know every thing you have written in your code. We are least concerned about how you have learnt something as long as you have learnt it well.

Acknowledgments

The manual has been written by Dr. Ubaid Ullah Fayyaz and Miss Shehzeen Malik.