TEHRAN UNIVERSITY
Electrical and Computer Engineering Department
Digital System Design with Hardware Description Languages
Fall 1389
Homework 5 – Timed Logic Modeling with C++ Classes

A. Declare a wire that can carry timing and power consumption as well as logic values. Delay values are fixed per gate. Use 1 for inverters, 2 for 2-input gates, and 3 for 3-input gates. Power consumption is calculated by events on gate outputs that represent dynamic power consumption. Wires carry all event counts to the gates they are feeding. Total power consumption is calculated by accumulation of all events.

B. Using the wire discussed above, generate gate and flip-flop models necessary to develop basic RT level components such as counters, state machines and shift registers.

C. Using the above gates and flip-flops, design an 8-bit shift register with do-nothing (Mode=0), right-shift (Mode=1), left-shift (Mode=2), and parallel load (Mode=3) modes. The shift register has two mode lines (m_1, m_0), a *sri* (serial-right input), *sli* (serial left input), an 8-bit parallel data (*din)* input, an 8-bit parallel output (*pout*), and a *reset* and *clock.*

D. Develop a testbench in C++ that instantiates the above shift register and applies test data to it. Use a clock frequency that is slow enough to allow propagation of all values through the circuit gates.

E. Partition the shift register into an array of flip-flops and a combinational block (Huffman model). Write a C++ class that contains the complete functionality of the combinational part of the shift register. Develop the shift register model according to the partitioning described here. Use accumulation of gate delays from Part B to annotate the combinational logic of this part with the worst case delays.

F. Develop a testbench in C++ that instantiates the circuits of Parts C and D, and applies test data to the circuits, and compares the results. Test your circuit for all modes of operation. Perform a parallel loading and then start shifting right and left. Run the results and submit result files and your complete code.

G. Develop a templated class for the combinational part of Part E, via which the delay values specialize the class. Use this model in the testbench of Part F.