Goal

In this project you will solve a non-trivial design problem explicitly using the divide-and-conquer (D&C) approach. The main reason for using the D&C approach is the ease of the design process and cleanliness of the resulting design. A typical by-product is a design that also uses less hardware and may be faster.

Furthermore, the actual design will be implemented and simulated using Quartus using a hierarchical approach — design and simulate smaller components, use them to design and simulate larger components and so forth until the entire design is completed and simulated.

Design Problem

You need to design an 8-bit comparator for two 8-bit unsigned (non-negative) numbers $X$ and $Y$. The comparator outputs a 1 if $X \geq Y$ and 0 if $X < Y$.

Explicitly show the following in your design process:

1. The D&C breakup of the problem.
2. The “stitch-up” required at each level of the D&C tree.
3. The design of the basic/smallest unit(s) (sometimes you can have more than one type of basic unit as in the multi-bit shifter [HW 2] or the tree CLA adder [Lecture Notes on “Fast Adders”], and sometimes only one type of basic unit as in the ripple-carry and carry-select adders [Lecture Notes on “Fast Adders”]) resulting from the D&C breakup.

This design process should include the following for each type of basic unit: TT(s), K-maps/QM (use multi-function QM if a basic unit has more than one output bit and they are functions of the same set of input variables), the final logic expressions for each output bit, and the gate-level design (with gate sharing if any) using AND/OR/NOT gates.

4. The design of the entire 8-bit comparator using the basic unit and, if necessary, logic gates, as components.

5. Assuming a k-input gate has a delay of k units, determine the delay of your 8-bit comparator, showing clearly the critical path (max delay path) of your 8-bit comparator. Specify each delay component in the critical path (see for example the critical path delay calculation for the 64-bit tree CLA in Lecture Notes on “Fast Adders”). Writing down the final delay number is not enough; we need to know clearly how you arrived at your answer.

1400 points
Implementation and Simulation using Quartus

You are required to implement and simulate your gate-based design obtained above using the Quartus CAD software as specified below.

1. Unless there have been major problems with your current group, you need to do this project with the same group as in Project 1. If there have been such problems, and you wish to form a new group, please get in touch with the instructor latest by Thurs March 31.

2. Choose the schematic capture tool in Quartus to specify your design.

3. Device family to be used for the project is Cyclone which is selected by default in Quartus.

4. Design the basic unit(s) using gates, simulate it/them for correctness (generate your own inputs for this simulation; test exhaustively, i.e., using all possible input combinations), save it/them.

5. Using the above saved design(s) and regular gates, design the 8-bit comparator.

6. Perform simulations of the 8-bit comparator based on the input file provided by the TA.

7. You need to submit a clearly written project report detailing all your work including all the steps and results of the design process (discussed in the previous section), use of the schematic capture tool, simulation results of component(s) and the final design, and other findings, if any, and conclusions. You need to also clearly state in the project report the final task-distribution among the group members.

8. Students are encouraged to interact with each other, share ideas and ask for help from the instructor or the TA. Copying of the project between groups will, however, not be tolerated and if evidence of copying is detected the concerned groups will be reported for disciplinary action and will fail the course.