Homework 1 – Energy-Delay Characterization

(Total points: 100)

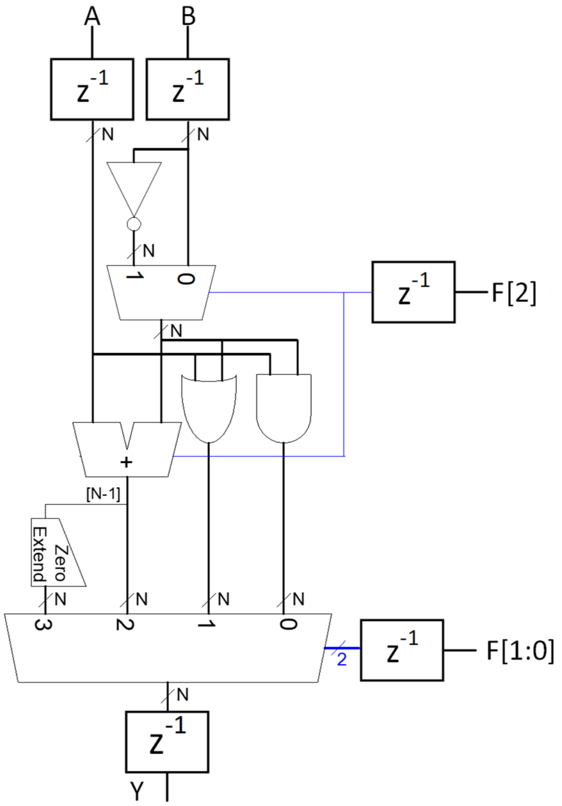
Objective

The objective of this assignment is to characterize the E-D curves of a 64-bit fixed-point ALU in 28nm CMOS technology using standard-cell-based design flow. Modern digital integrated circuit design uses CAD tools to optimize energy given a delay constraint. Because of the use of standard cell libraries, sizing, and threshold voltage (V_{th}), optimizations are performed with discrete-valued variables. For example, a NAND2 gate can have a gate size of x1, x2, x4, etc. Similarly, the gate in each size can have a V_{th} flavor of regular, low, and high. The CAD tools are able to optimize energy by choosing the best gate size and V_{th} flavor for each logic gate. Differently, supply voltage (V_{DD}) optimization CANNOT be performed by CAD tools automatically. This is because the timing/power libraries used by the tools are typically characterized at a specific process, voltage, and temperature (PVT) corner. As a result, V_{DD} can only be tuned globally with a designer's input by loading libraries characterized at different voltage corners at the design time.

While modern IC design flows based on standard-cell libraries and CAD tools simplifies circuit level optimization to a large extent, it also provides convenience and enables high productivity for designing VLSI system. The goal of this assignment is to characterize the E-D curves resulted from the circuit optimization performed by CAD tools in advanced CMOS technology node.

Tasks

1. Quickly prototype a simple 64-bit fixed-point ALU shown in the following figure using Simulink HDL Coder blocks. Create a testbench and verify the functionality of your design against a MATLAB reference model (referring to Tutorial 1, for example). Your testbench should randomly generate at least 50 samples of test data for each of the inputs (A, B, and F) for functional verification. Submit a screenshot showing the successful functional verification of the Simulink HDL Coder design.



$F_{2:0}$	Function
000	A & B
001	A | B
010	A + B
011	not used
100	A & ~B
101	A | ~B
110	A - B
111	Set Less Than (SLT)

2. Generate the RTL codes and HDL test bench automatically using the HDL Workflow Advisor with an ASIC target. Verify the functionality of the generated RTL design in HDL Workflow Advisor. Submit a screenshot showing the successful functional verification of the RTL codes.

3. Synthesize the design using the Design Compiler (DC) at 1V using a regular V_{th} (RVT) library only. To load RVT library only, you need to modify the 1V DC setup file by commenting out all the lines related to LVT or HVT libraries. Then, set increasingly tight timing constraints by modifying the generated “<title>_dc.sdc” file and re-run the synthesis to find out what the minimum delay (maximum frequency) of the design is regardless of energy consumption. Report the minimum delay found. For the rest tasks, you are going to take this particular design point as your reference point (min delay with standard V_{th}). The total energy consumption at this design point is E_{ref}.

4. **Characterize the E-D curve with sizing optimization only.** Synthesize the design with an delay increment (w.s.t min delay) of 5%, 10%, 15%, 20%, 25%, 30%, 40%, 50%, 75%, 100%, as well as 5x, 10x, 100x, 1000x of the min delay, respectively, at 1V using the RVT library only. A different timing constraint for synthesis can be set by modifying the generated “<project>_dc.sdc” file with a different clock period. Calculate the minimum total energy values (Energy=Power*Delay) for each of the delay constraint based on the results from the power reports.

5. **Characterize the E-D curve with sizing and V_{th} optimization.** Synthesize the design with the same set of delay constrains at 1V using all of the RVT, LVT, HVT libraries (the

default setup files load all V_{th} flavors). Calculate the minimum total energy values (Energy=Power*Delay) for each of the delay constraints based on the results from the power reports.

6. **Characterize the E-D curve with sizing and V_{DD} optimization.** Synthesize the design with the same set of delay constraints at 1V, 0.85V, and 0.75V, respectively, using the RVT library only. Calculate the minimum total energy values (Energy=Power*Delay) for each of the delay constraints based on the results from the power reports. You should take the minimum total energy values among all the V_{DD} cases **that meet timing** as the optimal energy at each delay point.
7. **Characterize the E-D curve with sizing, V_{th}, and V_{DD} optimization.** Synthesize the design with the same set of delay constraints at 1V, 0.85V, and 0.75V, respectively, using all of RVT, LVT, HVT libraries. Calculate the minimum total energy values (Energy=Power*Delay) for each of the delay constraints based on the results from the power reports. You should take the minimum total energy values among all the V_{DD} cases **that meet timing** as the optimal energy at each delay point.
8. Plot the total energy breakdown (in terms of dynamic and leakage energy) with respect to each delay value using a bar graph (MATLAB function “bar”) for each optimization performed in Problem 4-7 (one bar graph per problem). Normalize the energy numbers to E_{ref} (E_{Dyn}/E_{ref}, E_{Leak}/E_{ref}). For each bar graph, discuss why such an optimization setting derives such results based on energy-delay models. If you observe anything different from what we discuss in class, please also explain why.
9. Plot the E-D curves extracted from task 4-7 on a single plane as E/E_{ref} v.s. D/D_{min}, where D is (Delay Constraint – setup time slack). Plot the E-D curves with D/D_{min} on a logarithmic scale. Discuss the E-D curve results based on energy-delay models and the sensitivity theory. Identify which combination of variables is more effective for minimizing energy at different delay ranges. If you observe anything different from what we discuss in class, please explain why. Summarize what insights you gain from this study as future design guidelines.

Tips

- ALU design hints:
  - AND, OR, NOT operations are bitwise operations.
  - Use a Multiport Switch block under the Signal Routing library to implement a MUX.
  - To allow the adder block to have three input ports, specify the “List of signs” parameter as “+++”.
  - To implement the SLT logic, you need to zero-extend the most significant bit (MSB) of the adder output using Bit Slice and Bit Concat blocks under the Logic and Bit Operations library.

  - The inputs A and B and the output Y should use a fixed-point data type with one sign bit, an integer word length of 63 bits, and a fractional word length of 0 bit. The input F should use a fixed-point data type with no sign bit, an integer word length of 3 bits, and a fractional word length of 0 bit.
  - To keep the adder output at 64 bits, you can assign the data type of the adder block as “int64”.

- You should leverage the DC templates (<title>_dc.sdc, <title>_dc.tcl) provided in Tutorial 1 and write your own batch script to automate all of the synthesis runs and the result collection process as much as possible. Minimize the number of buttons you need to push. While writing your batch script, make sure to resynthesize the design from scratch for each delay constraint. You can either use a shell script with a for loop to launch dc_shell with a different setup to synthesize the design from scratch in each iteration. Or, you can add a for loop to the.tcl file to reload the synthesis settings and the Verilog file for synthesis from scratch in each iteration. (make sure to clear the cache with “remove_design -all” in DC at the beginning of each iteration to remove the libraries and synthesis results from the previous iterations).
- For a more accurate E-D curve characterization, **the actual delay values of the optimal design should be the delay constraint minus the setup time slack, if any.** Note that the reported dynamic power (P_{Dyn}= P_{switch}+ $P_{internal}$) is calculated by the CAD tool as P_{Dyn} = E_{Dyn}/(Delay Constraint). So the setup time slack is ignored in dynamic power estimation. For an accurate energy estimation, you should use **E_{Dyn} = P_{Dyn} * (Delay Constraint)** for dynamic energy, but **E_{Leak} = P_{Leak} * (Delay Constraint – setup time slack)** for leakage energy. **E_{total} = E_{Dyn} + E_{Leak}.**

Grading Rubrics

Inspection item	Points
Valid ALU design and testbench for successful functional verification against a MATLAB reference model	10
Successful generation and functional verification of RTL codes	10
Successful ASIC synthesis and valid reporting of the minimum delay	10
Four energy breakdown graphs showing the correct trends with valid discussions explaining the results	60
Combined E-D curve showing the correct trends with valid discussions explaining the results using sensitivity theory	10
Total	**100**

Due Date

See Canvas.

Instructions on Submission

- Submit the entire MATLAB work folder containing the Simulink model of your ALU design and test bench, as well as the project folders generated by the HDL Workflow Advisor containing the HDL codes and HDL test benches.
    - Include the MATLAB scripts for configuring the testbench, if any. Your submission should allow the grader to verify your design by running the Simulink test bench.
    - Please remove the large set of DC synthesis results from the submission. Instead, include the Excel sheet or MATLAB scripts containing the total power numbers extracted from the synthesis results and the calculation of total energy numbers.
- Submit your solution in PDF format.
- Put all of your files into a folder first and then compress the submission folder into a zip archive file named cen571-firstname_lastname-hw1.zip and submit it on Canvas.