FINAL PROJECT 4-BIT ALU DESIGN

Planning Stage Deadline: 23:55 on Sun 10 March, 2024
Implementation Stage and Exam Deadline: 23:55 on Thu 9th May, 2024

1. Overview

This project is aimed at providing you with a hands-on experience of defining, planning, and implementing a (relatively) large digital VLSI circuit. The process you will follow in the project is similar to the conventional process used for large-scale integrated circuits in the microelectronics industry. Due to time and practical constraints of a university course, we make some assumptions and simplifications, facilitating the planning stage.

2. Requirements

You are required to design a simplified 4-bit arithmetic logic unit (ALU) using Cadence Virtuoso and the gpdk045/gsclib045 technology. The project must be carried out in a few stages as described below, with a report required to be submitted for each stage.

3. Specifications

You are required to build a simplified 4-bit ALU with the following specifications.
Inputs:

Logical Inputs: The ALU must accept two inputs: A[3:0] and B[3:0]. Each input must be fed into registers, whose outputs connect to the arithmetic/logical part of the circuit.

Opcode: The ALU must accept one 5-bit opcode (the opcode must be registers), which determines the function to be performed by the ALU. You may decide on any opcode schema of your choice.

Outputs: The ALU must return one output Y[3:0]. This output must be supplied as an output of a register.

Feedback: At the input of the register storing A, there must be a multiplexer which can choose between a user-supplied input and the output Y from the previous clock cycle. The selection between the two must be made using the opcode.

Encoding: For arithmetic operations, A, B, and Y must represent integers using two's complement representation. Operations: Assume that $\langle X \rangle$ denotes the number represented by X[n -1 :0] in two's complement.

ADD: $\langle Y \rangle = \langle A \rangle + \langle B \rangle$

SUB: $\langle Y \rangle = \langle A \rangle - \langle B \rangle$

XOR: Y = (A) XOR (B)

BSR: Y = barrel-shift-right(A,i) (barrel shift right, by i positions), where $i \in \{1,2,3,4\}$ is encoded within the opcode

Supply voltages: Your circuit must use a single supply voltage (VDD), in addition to ground (as the reference zero voltage). The nominal supply voltage for the circuit should be assumed to be 1.2V.

Realization area: The realization area must not exceed 150% of the total area of the cells (i.e. the sum of the areas of all gates, registers, etc.).

Rise/fall times: The rise/fall times (10%–90%) at the inputs of any cell must not exceed 50ps

Adder Design: Your ALU must use a tree-based adder design as determined by algorithm 1.

Permitted libraries: You may use any components from gsclib045. You may use noConn from basic as dummy connections for unused circuit outputs, in order to suppress warnings about floating nets. You may also use other components from analogLib and basic, *for testing purposes only*.

Power consumption: The power consumption should be minimized as far as possible (upto reasonable limits, taking into account practical constraints). In particular, modules which are not required for the operation being performed should be "disabled".[1]

Algorithm 1 Adder Design Selection

```
Let x be the sum of the last digits of the IDs of all group members.
y ← (x mod 6)+1
if y = 1 then
    Your ALU must use a Brent-Kung adder.
else if y = 2 then
    Your ALU must use a Sklansky adder.
else if y = 3 then
    Your ALU must use a Kogge-Stone adder.
else if y = 4 then
    Your ALU must use a Han-Carlson adder.
else if y = 5 then
    Your ALU must use a Knowles [2,1,1,1] adder.
else if y = 6 then
    Your ALU must use a Ladner-Fishcer adder.
end if
See Chapter 11 in [WH10] (available on Moodle) for topologies of these adders
and explanations.
```

4. Stages of Execution

[1] For example, if the operation to be performed is XOR, blocks such as the adder should not perform their computation.

The project must be carried out in two stages: the planning stage and the implementation stage. A report must be prepared for each stage, as detailed below, and submitted by the respective deadlines. The final grade will be calculated taking both phases into account, with the planning stage counting towards 25% of the grade and the implementation stage counting towards 75%. The grades of each stage will be based on the quality of the design and explanations, taking into account all fundamentals covered during the semester.

4.1. Planning Stage. In the planning stage, you must

(1) sketch a block-level logical diagram of your circuit.
(2) estimate the number of cells (i.e. components from gsclib045)
(3) sketch the floor plan of your layout.
(4) estimate the area required for your layout.

All of these must be performed "on paper"[2] whilst examining cell areas in Virtuoso, but otherwise without significant use of software tools. These sketches, estimates, and all relevant explanations should be compiled into a report which should be submitted as a single PDF file on Moodle, by 23:55 on Sun 10 March, 2024.

4.2. Implementation Stage. In the implementation stage, you must[3]

(1) design a detailed logical schematic of your implementation, using pre-built cells from gsclib045
(2) measure the total delay time for the critical path, for supply voltages of $700\mathrm{mV}$ and $1.2\mathrm{V}$
(3) measure the maximal clock frequency (or equivalently the minimal clock period) for your circuit, for supply voltages of $700\mathrm{mV}$ and $1.2\mathrm{V}$.
(4) design a detailed, hierarchical floor plan for your layout.
(10) build the layout, splitting the work between group members using the hierarchical structure.
(11) perform DRC and LVS tests on your layout and ensure zero errors.
(12) perform parasitic extraction (QRC), and repeat all aforementioned tests with extracted parameters[4]

Based on these steps, you must compile a report which includes

(1) names, IDs, and university usernames of all group members
(2) name of the Virtuoso library which contains the project, the name of the person who owns this library, and the names of all cells you designed.
(3) exported images of all schematics and layouts, at various zoom levels as necessary
(4) all test results (waveforms, evaluated expression tables, pass/fail results, etc.)

---

[2] In this context, "paper" includes paper-like writing technologies such as word processors and digital tablets.

[3] Unless specified otherwise, assume nominal supply voltage ($1.2\mathrm{V}$) and maximal clock frequency.

[4] You may find it useful to save your waveform plotting settings as a template.

(5) confirmation of clean DRC and LVS for the complete layout
(6) confirmation of successful extraction (QRC) for the complete layout
(7) examples of the circuit functioning as expected for all operations, using digital bus waveforms.
(8) explanations and comments (including comparable examples where relevant) about
    (a) steps taken to optimize the performance of your circuit (including but not restricted to delay and power consumption)
    (b) the effect of the supply voltage on the maximal clock frequency of your circuit
    (c) the difference between schematic-based estimates of the critical delay and layout-based estimates of the same
(9) any other relevant data or explanations that you deem appropriate.

5. Submission and Miscellaneous Instructions

(1) The project must be performed in groups of four students. Groups of with fewer or more members are not allowed, unless explicitly approved by the course staff.
(2) Each group must select one member as the contact person. This member will be responsible for submitting the reports, and for most other communication.
(3) Each report should be submitted on Moodle, as a single PDF file. In addition, the reports should be stored on the micron servers in the home directory of the contact person, named alu_proj_report_1.pdf and alu_proj_report_2.pdf respectively.
(4) After uploading and saving the report as explained above, the contact person must send an email to Zvi Webb (webb@tauex.tau.ac.il), CCing all other group members, confirming the submission.
(5) The final grade will be determined based on both reports, and a short exam with all group members. The exam requires approximately half an hour and can be done either in-person or over Zoom. After submitting the final report, the contact person must coordinate with Zvi to set a time for this exam. All exams must be completed before the deadline (23:55 on Thu 9 May, 2024).
(6) Every group member must create a new Virtuoso library named <username>_final_proj, with <username> replaced by the username of the group member. All work done by a group member must be stored in this library *only*. Do not try to save work in other libraries, be they your own or those of other group members.
(7) The final circuit (i.e. the complete ALU design) must be stored in the library of the contact person.
(8) It is recommended to schedule a Zoom meeting with Zvi before submitting the first report, to verify that your designs are correct, relevant, and practical.
(9) The submission of the final report will be considered as partial fulfilment of the requirements for obtaining a final grade in the course.

(10) In case of general questions, use the project Q&A forum on Moodle.
(11) In case of technical issues related to the servers, design-level implementation, and submission, you may contact Zvi Webb (webb@tauex.tau.ac.il).
(12) In case of personal issues (i.e. ones which cannot be discussed in the public Moodle forum) including Virtuoso-related ones, you may contact any of the TAs.

6. Miscellaneous Tips

(1) As far as possible, use standard cells from gsclib045. Avoid building cells from the transistor level.
(2) Design your implementation in a hierarchical manner. This will simplify the work involved and will allow you to easily split the workload between group members.
(3) Use only Manhattan geometry for routing, i.e., use horizontal/vertical traces only, do not use diagonal traces.
(4) Split the workload in a way that all members can work in parallel.
(5) Keep your design simple and straightforward.
(6) Use edge-to-edge power rails, size your cells to have consistent height, and lay the cells out in placement rows, as shown in figs. 1 and 2.

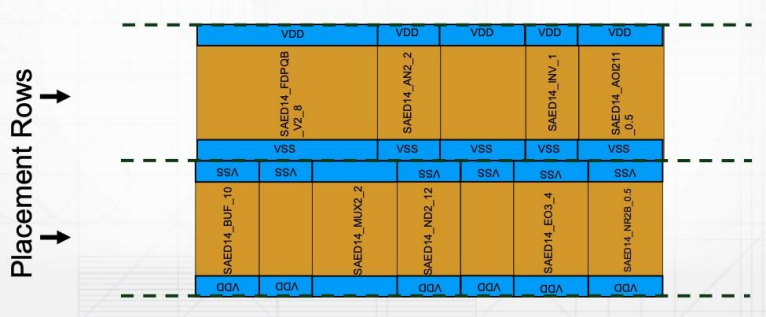

Figure 1. Structure of Placement Rows

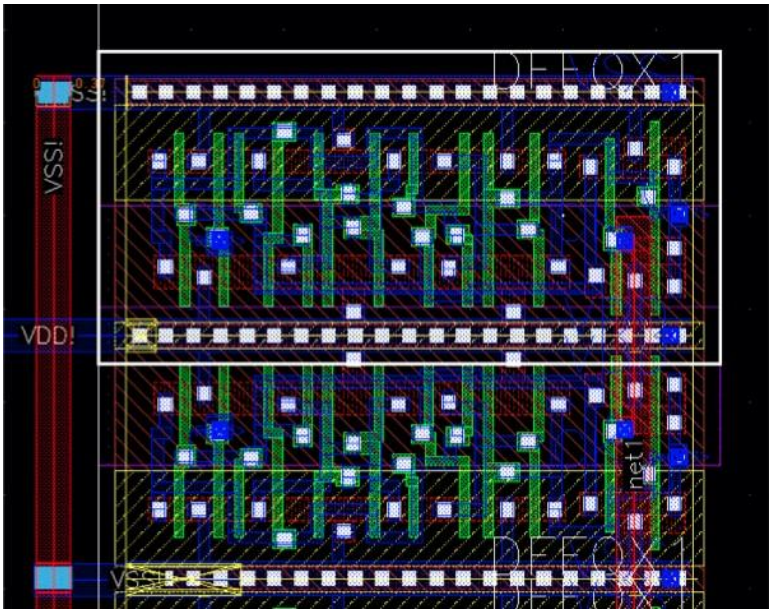

Figure 2. Example of Using Placement Rows

Using placement rows does not mean that *all* of your cells must be placed strictly in rows of equal height. If your cells are of similar heights, you can place them in rows of equal height. If the heights are significantly different, you should rather place them in a tiled manner, as shown in fig. 3.

(7) The AND2X1 cell in gsclib045 has a mismatch between its schematic and layout. In order to get around this, you can either use AND2X2 which has the same functionality and differs only in the transistor sizing, or you can create your own fixed version of AND2X1 using the following process.
   (a) Create a cell named AND2X1 in your own library.
   (b) Copy the schematic and layout of AND2X1 from gsclib045 into the corresponding views of AND2X1 in your own library.

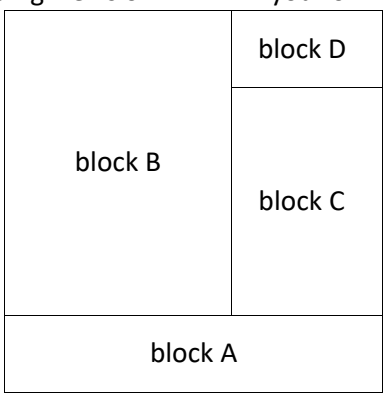

Figure 3. Example of Placement Using Tiling

   (c) In your schematic, delete
      (i) the wire connecting pin A to the gate of mn1
      (ii) the wire connecting pin B to the gate of mn0
   (d) In your schematic, DO NOT delete
      (i) the red label named VDD! ∗ next to the VDD pin
      (ii) the red label named VSS! ∗ next to the VSS pin
   (e) In your schematic, add
      (i) a wire connecting pin A to the gate of mn0
      (ii) a wire connecting pin B to the gate of mn1
   (f) When creating a symbol for your AND2X1, in the symbol creation dialog box, set “Exclude inherited connection pins” to “all”.
   (g) When running LVS, in the “LVS Run Submission Form”, go to “LVS Options” in the left sidebar. Then go to the “Comparison Options” tab, and set “compare port names” to off.
   (h) Use your implementation of AND2X1 instead of gsclib045’s implementation.

References

[WH10] Neil Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. 4th. USA: Addison-Wesley Publishing Company, 2010. isbn: 0321547748.