1. Design of a 5-bit Adder/Subtractor – Description

Phase II of the project is the design of a 5-bit adder that generates the true and complimentary effective address bits that are fed to the decoder. The inputs to this adder are a 5-bit relative address and a 2-bit, 2’s complement offset address. Note that the complement versions of both of these inputs are also available. The 2-bit offset ranges from decimal -2 to 1, while the relative address ranges from decimal 0 to 31. A schematic indicating the inputs and outputs of the adder is shown in the figure below.

![5-bit Adder Block for Phase II](image)

2. Implementation

There are many different adder topologies that are used in modern digital circuits, each with their own tradeoffs between power and performance. The main goal of this phase of the project is to familiarize you with the underlying structure of adders – in phase three of the project you will have all the freedom you’d like to explore various adder circuits and topologies. For this phase, you will be designing a ripple-carry adder out of full-adders and half-adders, which in turn will be built from static combinational logic blocks. As was specified in the project phase 1 handout, the input capacitance of each of the adder/subtractor inputs must be less than 5 fF.
We have provided a library of standard cells that you are free to use in constructing your adder. Using the same process outlined in phase I for each of the cells, you can copy the standard cell library from the project directory `~ee141/fall/08/project/adder_blocks/`. The library includes many of the building blocks you are likely to need, including an XOR, XNOR, NAND2, NAND3, NOR2, NOR3, INV, and a full mirror adder (M_ADD).

The basic logic equations for half adders and full adders are shown below.

For a half adder:
Sum = A ⊕ B
Cout = A·B

For a full adder:
Sum = A ⊕ B ⊕ Cin
Cout = A·B + B·Cin + A·Cin

It is up to you to assemble (in both schematics and LVS/DRC-clean layout) the half-adders and full-adders into the 5-bit adder/subtractor using the given blocks (and whatever additional blocks you decide to design and lay out). Note that you are not required to optimize the gate sizes of your adder/subtractor – i.e., you can simply use the standard cells to implement the required logic functions. However, you should use the cells as efficiently as possible in order to minimize the delay and power of the adder. For example, you should make full use of the true and complement inputs to avoid adding inverters when you don’t need to. If you find that you’d like to have a larger width than that used in the standard cells for one of the gates, you can achieve this by placing two or more of the cells in parallel with each other. Note that if you do decide to explore optimizing the adder’s sizing, remember that the adder/subtractor will be driving the inputs of the decoder, and in phase 3 you will be free to change the input capacitance of the decoder. In other words, you only need to maintain an input capacitance of 5fF on the adder_rel and addr_off pins, but not on addr_eff – thus, the loading on the outputs of your adder/subtractor may not necessarily be 5fF.

3. Analysis and Simulation

Your primary goal in any IC design should be to ensure that the circuit you have designed functions as intended. Since the number of inputs to your adder/subtractor is relatively small, we
have provided to you a SPICE deck (~ee141/fall08/project/adder_blocks/check.sp) that you will use to exhaustively check that the output of your adder/subtractor is correct for all possible inputs. Please attend one of the discussion sessions for further instructions on how to use this deck to check your adder.

As you are assembling your adder/subtractor, you should keep in mind which input pattern will result in the worst case delay. Once you have finalized the assembly of your adder/subtractor and identified which gates are on the critical path, you should hand analyze the delay of this path. You should then extract the layout of your adder, and simulate it with this worst case pattern to find the delay. For this phase we will assume that the decoder input capacitance remains $5fF$, so you should remember to load the outputs of your adder/subtractor with $5fF$ capacitors (in order to simulate the load from the decoder). If there are any major discrepancies between your analysis and the simulation, you should explain the reasons for this in your report.

4. Report

The quality of your report is as important as the quality of your design. Be sure to provide all relevant information and eliminate unnecessary material. **Organization, conciseness, and completeness are of paramount importance.** Do not repeat information we already know. Use the templates provided on the web page. Make sure to fill in the cover page and use the correct units. Turn in the reports for each phase in the homework drop box.

4.1 Report for Phase II

The organization of the report should be based on the following outline:

- Cover page: Names, calculated delay, simulated delay.
- Page 1: Schematic of the 5-bit adder/subtractor with details of each of the blocks
- Page 2: Adder/Subtractor layout
- Page 3: Graph from the functionality check spice deck
- Page 4: Schematic showing the gates on the critical path and simulated delay.
- Page 5: Hand estimate of adder/subtractor delay, explanation of any discrepancies vs. sim.