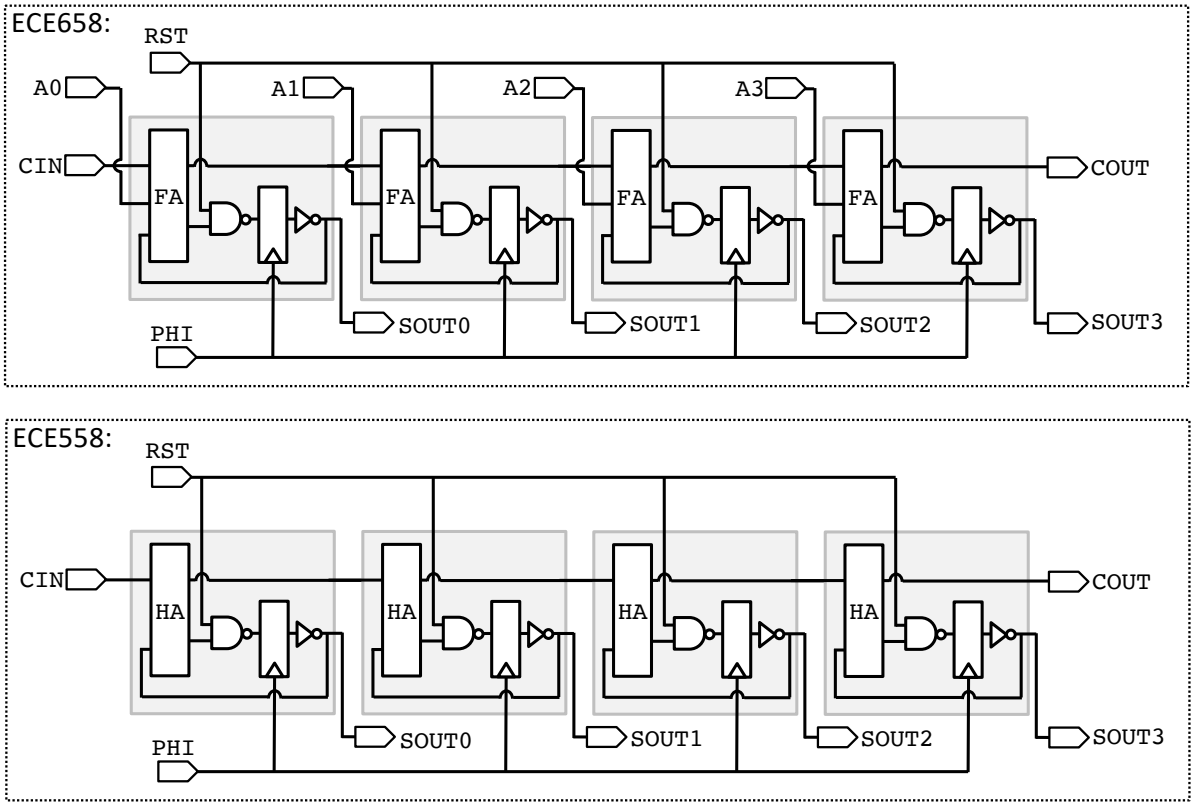
# ECE 558/658 VLSI Design -- Fall 2021
# Lab 3: Design and Analysis of 4-bit Accumulator using Bitsliced Layout

In this lab you will create a 4-bit accumulator circuit. A fully functional lab 2 is a prerequisite for completing this lab. A 4-bit accumulator is a sequential circuit that in each clock cycle updates a sum by adding to it the current value of the input. Therefore, the sum stored in the accumulator represents the sum of the inputs since reset; overflows are ignored.

As you have seen in the previous assignment, a 1-bit accumulator consists of a full adder and a resettable register. A 4-bit accumulator, built from four 1-bit accumulators, consists of a 4-bit adder and a resettable 4-bit register. Its 7 inputs are PHI, RST, {A3, A2, A1, A0}, and CIN. Its 5 outputs are {SOUT3, SOUT2, SOUT1, SOUT0} and COUT. The A and SOUT values each represent 4-bit words, in which A3 and SOUT3 are the most significant bits. The adder computes the sum of {A3, A2, A1, A0}, {SOUT3, SOUT2, SOUT1, SOUT0}, and CIN, and generates carry out COUT and a new 4-bit sum that gets stored to flip-flops on the positive edge of the clock. The braces { } in the above description are for readability of the 4-bit vectors and are not part of the signal names.

Students in ECE558 are using half adders instead of full adders. Their circuits are similar to the full adder accumulator, except that only CIN is being added to the sum in each cycle. In other words, the half-adder version of the accumulator computes the sum of all the valuations on CIN since reset.




In labs 1 and 2, you've designed your circuit at transistor level and created its layout; this style of custom design requires high engineering effort and is used sparingly. Lab 3 is the final lab that will use this style of custom design. In lab 4, you will create a similar accumulator by using design automation to translate a design from hardware description language through synthesis and placement and routing into a layout made of standard cells from a library.

Follow instructions carefully. Be sure to include each required item (indicated by **SUBMIT**) in your report. You must also explain what you did and why; images alone are not sufficient.

**Step 1: Schematic** [10pts]

Draw a schematic for the entire 4-bit accumulator using the schematic editor tool (Cadence). Capture a screen image of the editor window. Use the following approach.

- Create a symbol view for the bitslice accumulator that you created in lab 2. Symbol views enable hierarchical schematics by hiding lower levels of detail (such as the interconnections of transistors). Symbol views are often simply rectangles with input and output pins, although many logic gates (inverter, NAND, etc.) and macro functions (mux, ALU, etc.) have customary shapes which you have drawn by hand many times.
- Instantiate the bitslice symbol four times to create the top-level 4-bit accumulator schematic. No individual transistors should be visible.
- The RST inputs of all four bitslices should be connected. Likewise for PHI.
- The carry out of each bitslice should connect to the carry in of the next bitslice to form a ripple carry chain. The carry out from the final bitslice does not connect to anything.

**SUBMIT:** Image of your 4-bit accumulator schematic.

**Step 2: Validate schematic using HSPICE .VEC** [25pts]

Apply a test sequence that is based on the 8 digits of your student ID number. You should reset the flip flops near the start of the simulation, and then in the next 8 cycles will update the sum as follows.

- If you are in ECE658, convert each digit of the ID to a 4-bit value, and sum the 8 values by applying them on the {A3,A2,A1,A0} inputs of your circuit. Don't worry if the sum overflows during the sequence.
- If you are in ECE558, you should do the same, but using only the least significant bit of each digit and applying it on CIN.

**SUBMIT:** A table that shows your input sequence. Each row of the table corresponds to one cycle and shows the value of inputs, reset, current state of flip flops, outputs, and next state of flip flops. An example of the start of such a table is shown below, where D and Q are the next and current state of the flip flops. Students in ECE558 will not have the A inputs in their table, but it will be otherwise similar.

| RST | A3 | A2 | A1 | A0 | CIN | Q3 | Q2 | Q1 | Q0 | SOUT3 | SOUT2 | SOUT1 | SOUT0 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | x | x | x | x | x | x | x | x |
| 1 | | | | | | | | | | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

**SUBMIT:** An image of the simulator output showing all the waveforms for the inputs and outputs to 4-bit accumulator. Annotate waveform to show that the inputs and outputs in each cycle match your table. Annotate the waveform to show which clock edge is resetting the sum at the beginning of the simulation.

**Step 3: Design a layout for your accumulator** [10pts]

Design a layout using Cadence Layout XL/L. Your layout must consist of four instances of the bitslice layout from Lab 2 connected by abutment; no additional wiring may be used. Any changes you need to make should be made in the original bitslice layout so that you can instantiate it four times to get a clean and functional layout with no additional drawing. Do not "fix" the layout at the 4-bit hierarchical level. In the

Cadence layout editor you can change the bitslice layout and have the change be automatically reflected in the 4-bit layout without having to re-instantiate.

- At bitslice boundaries, the Vdd and Gnd rails and the phi and reset signals should become connected automatically by abutment. Similarly, the carry out and carry in of adjacent bitslices should connect.
- Label the input and output pins at the 4-bit level. This is the only "new" content in your top-level layout.
- If the aspect ratio of your bitslice layout was Y:X = 4:1, the aspect ratio of your 4-bit accumulator should be 1:1 (square).

**SUBMIT:** An image of your layout, with the total height and width annotated.
**SUBMIT:** Describe any changes you had to make to the bitslice layout from Lab 2.

**Step 4: Testing** [15pts]

As you've done in the prior labs, run DRC to make sure your layout has no violations and LVS to make sure that it matches the schematic. Use Calibre PEX to extract a netlist with the parasitic capacitances and resistances from your layout. Reapply your test sequence from step 2 to check that your design works correctly with the parasitics included.

**SUBMIT:** Screen capture of DRC, LVS results showing no violations.
**SUBMIT:** An image of the simulator showing inputs and outputs (waveforms).

**Step 5: Determine Maximum Clock Frequency** [25pts]

Try to identify the critical (slowest) timing path between flip-flops in your circuit, which will limit the clock frequency of your accumulator. Be sure to consider transitions that can propagate through multiple bitslices. Develop a test sequence that exercises the suspected critical path. Use HSPICE to apply this sequence to the extracted netlist from step 4 and determine by trial and error the maximum clock frequency. Assume that the clock period can be set with 20 ps precision. As in lab 2, rise and fall times of all inputs, including the clock, must not exceed 10% of clock period as the clock period is changed.

**SUBMIT:** A description of the critical path, and discussion of why you believe it is the critical path.
**SUBMIT:** The test sequence which exercises the critical path. Report the sequence as a table, as in step 2.
**SUBMIT:** Image of the simulator output for your sequence, showing correct operation at maximum clock frequency. Annotate clock period on plot.
**SUBMIT:** Image of the simulator output when the clock period is 20% shorter than above. Annotate clock period on plot and highlight where the first incorrect value is observed in the waveform.
**SUBMIT:** Based on the time and location of the first incorrect value, explain whether it appears to actually be caused by your suspected critical path, or during some earlier cycle of the simulation.

**Step 6: Measure Switching Power** [15pts]

Measure the power consumption (in µW) of the extracted 4-bit accumulator netlist when applying your test vectors from step 2 at a clock frequency that is 80% of the maximum frequency from step 5. How much of the total power is static power, and how much is dynamic? You may need to perform some additional measurements to find out. Estimates are acceptable if they are justified and based on sound reasoning.

**SUBMIT:** Average power consumption and description of how you obtained the measurement.
**SUBMIT:** Describe your approach for distinguishing static vs dynamic power. Report the breakdown that you calculated.