EEC 116 Fall 2011 Lab #5: Pipelined 32b Adder

Dept. of Electrical and Computer Engineering
University of California, Davis

Issued: November 2, 2011
Due: November 16, 2011, 4PM

Reading: Rabaey Sections 5.5, 7.1-7.3, and 11 [1].
Reference: Brunvand Chapter 7.6 [2].

OBJECTIVE

The objective of this lab is to design, lay out, and verify a pipelined 32 bit adder based on circuits you designed in Lab 4 and Prelab 5. You will measure the average power consumption of the adder over a number of digital test vectors at different clock frequencies and determine the maximum clock frequency at which your design functions correctly.

TOOL SETUP

No additional setup should be required for this lab.

D FLIP-FLOP DESIGN AND SIMULATION

Pipelining an arithmetic circuit first requires creating edge-triggered flip-flops and registers to sample input data and intermediate results like carry outputs.

Positive Edge-Triggered D Flip-Flop Create a new schematic cell view using the Library Manager for new cell called `pDFF` (for positive edge-triggered D flip-flop). Draw the schematic for the flip-flop as shown in Figure 1 of Prelab 5 with the following modifications:

1. Configure the flip-flop to be positive edge-triggered with inputs `D` and `clk` and output `Q`.

2. Generate $\overline{clk}$ from `clk` inside the flip-flop using an additional inverter.

3. Choose transistor sizes consistent with previous labs, i.e. use minimum-size inverters and choose transmission gate sizes based on scaling your solution to Prelab 5 to the typical transistor sizes you've used in previous labs.

Part 1 Flip-Flop Layout Create symbol and layout cell views for the flip-flop. It is recommended to match the height of the flip-flop cell to the full adder cells you designed in Lab 3, but you may make the flip-flop cell taller if necessary. Verify your layout is free of DRC and LVS errors. Be prepared to show your schematic, layout, and demonstrate that they are free of errors to the TA for checkoff.

Part 2 Flip-Flop Analog Simulation Create a new schematic cell view using the Library Manager for a new cell called `lab5_pDFF_tb`. In this schematic, you will instantiate your DUT, as well as additional components for testing it. Instantiate your `pDFF` cell as the DUT. Add two sets of back-to-back stimulus inverters (use `invCC_1x`) to drive the flip-flop clock and data inputs. Add a fanout-of-4 (FO4) load of `invCC_1x` cells to the flip-flop `Q` output. Be sure to instantiate the two supply voltage sources and any other voltage sources you need to supply the data and clock test waveforms. Use analog simulation (i.e., Spectre) to determine the setup time t_{su}, clock-to-Q propagation delay t_{CQ}, and hold time t_{hd} for your flip-flop. One way to do this for setup and hold time is to fix the clock waveform as a periodic square wave and create a piecewise linear (PWL) voltage source for the data waveform. Adjust the relative delay between transitions on the data waveform and the positive edge of the clock until the flip-flop output becomes logically incorrect. The smallest delay between clock and data which results in correct operation with data transitioning before the clock and after the clock transition correspond to setup and hold time, respectively. Be sure to check both low-to-high and high-to-low transitions for the data in case there are any asymmetric delays. Show your simulation waveforms to the TA for checkoff and record your simulated values in the summary table.

PIPELINED ADDER DESIGN AND SIMULATION

In this section, you will design, lay out, and simulate a pipelined 32 bit adder. The adder will be pipelined in 8b segments and follow a ripple-carry architecture, thus the maximum throughput (i.e., maximum clock frequency) will be determined the ripple-carry delay of an 8b adder stage rather than the full 32b carry chain. You would therefore expect the pipelined adder to be able to run at approximately 4 times the frequency of the adder you designed in Lab 4. Figure 1 shows a block diagram of the pipelined adder showing where the pipelining flip-flops and registers need to be inserted to guarantee that all the input and output bits arrive at the correct times to produce a valid result in the final set of registers. To build this adder requires building some subcircuits first.

8b Adder and Register Using the mirror adder cells you created in Lab 3, create a new cell called `adder8b`. Create both symbol and schematic views. Similarly, using the flip-flop cell you designed above, create symbol and schematic views for an eight bit register cell named `pReg8`. Show your schematics and symbols to the TA for checkoff.

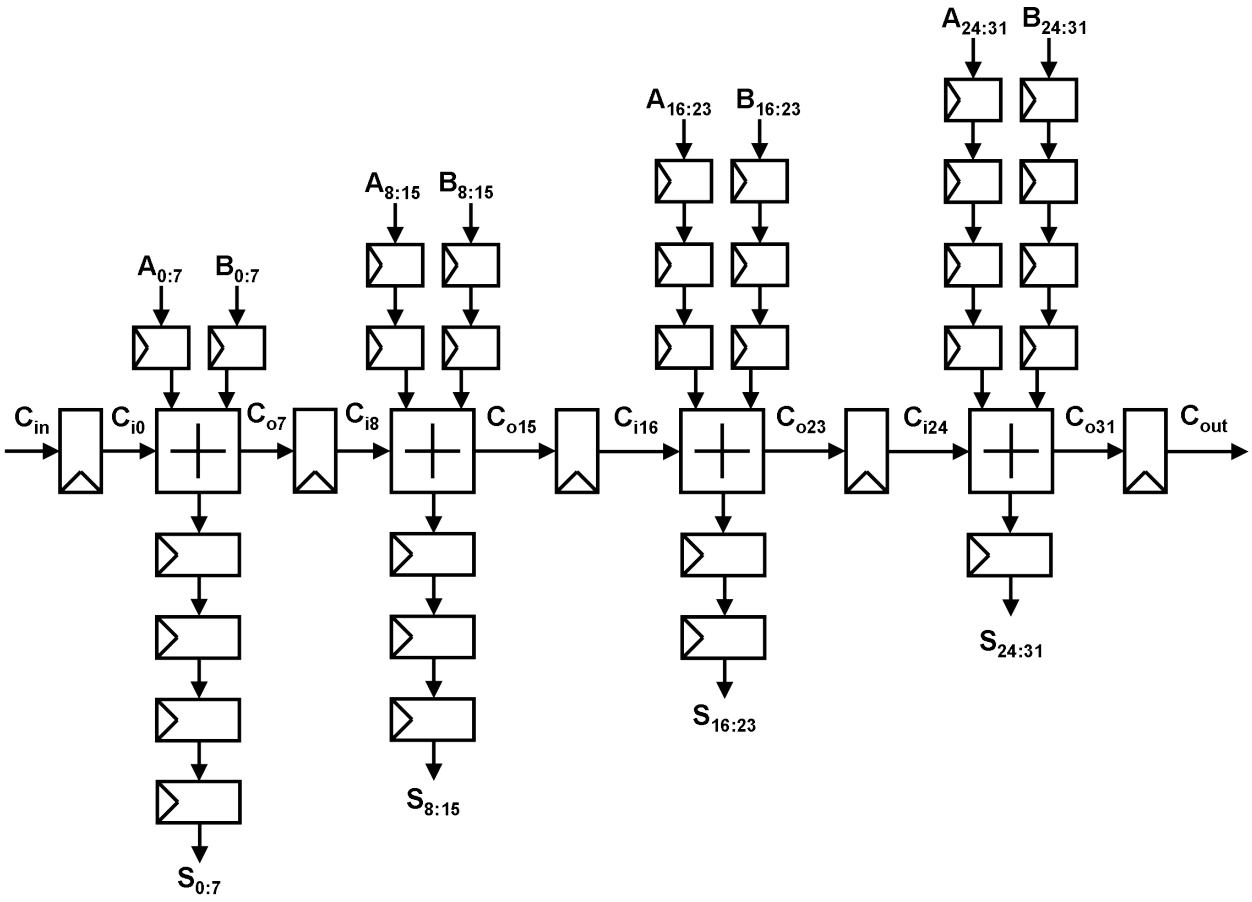

Figure 1: Pipelined 32b adder block diagram.

Part 3 Pipelined 32b Adder Design Create a new schematic view for a new cell called `adder32bPipe` and implement the schematic shown in Figure 1. Label the inputs and outputs the same as used for the `adder32` cell from Lab 4. This allows you to reuse most of the testbench schematic and test vectors you created for Lab 4 with a minimal amount of extra work. Create a symbol view for the pipelined adder cell by copying the `adder32b` symbol and adding the clock input `clk`. **It is highly recommended that you finish the adder schematic by the end of the first week to allow one full week for layout and simulation. DO NOT WAIT UNTIL THE LAST MINUTE TO START!**

You are now ready to lay out the pipelined adder. The goal is to have the layout approximate a rectangular shape rather than the parallelogram implied by the block diagram of Figure 1. This can be done by routing the carry signal diagonally from the carry output of one 8b adder to the carry input of the next 8b adder. A suggested floorplan is shown in Figure 2. The suggested floorplan implies that you may need to make multiple layout views of blocks such as `pReg8`, `pDFF`, and `adder8b` in order to make all the rows of cells the same height. Do not spend a lot of time optimizing for minimum area - if you need to stretch cells and waste some area to create a rectangular/square design, do so. In this case, the ease of integrating the rectangular design with other circuits is more important than minimizing

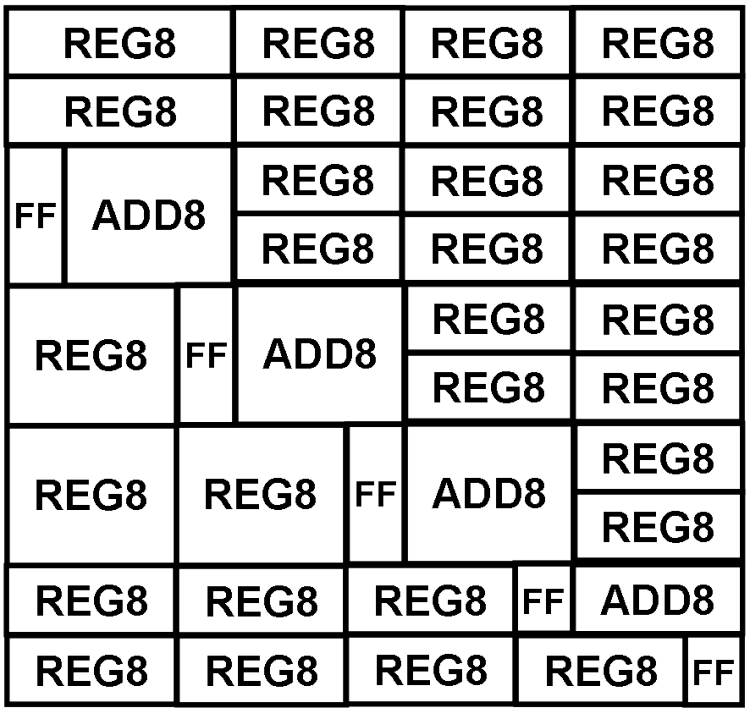

Figure 2: Pipelined 32b adder floorplan.

the total area of the adder. Verify that the final design passes DRC and LVS checks and be able to show the verification results to the TA for checkoff.

Part 4 Mixed-Mode Simulation Copy your 32b adder testbench cell from Lab 4 to a new cell named `lab5_add32pipe_tb`. Replace the `adder32b` DUT with the new pipelined adder DUT `adder32bPipe`. Modify your testbench by including the clock signal and some inverters to buffer it. Use the same configuration of inverters as the addend and carry inputs because we will want to measure the power (including the clock network power) later. Verify that your pipelined adder works correctly using mixed-mode simulation and a modified version of your Verilog stimulation file that includes the clock stimulus.

Clock Issues You may find that your circuit does not work properly the first time. One issue is the clock: you may need to buffer it by using larger inverter cells than `inv_1x` and `invCC_1x`. Go ahead and create bigger inverter cells naming them according to our convention. For example, if you need a 64× minimum size inverter running from the V_{DD} supply, name the cell `inv_64x`. You can also create bigger inverters running from the V_{CC} supply as well. Try to only use two inverter stages total for the clock buffer to simplify your testbench schematic.

Timing Issues One potential problem with connecting registers back-to-back without any logic gates in between is hold time violations. Although this is unlikely with the flip-flops you designed above, if you find such violations they can be fixed by inserting back-to-back

inverters between the register stages for the A and B inputs and the sum outputs. If you have to change your schematic in this way, you will need to modify your layout as well so that the final simulated working design can pass the LVS check.

Maximum Frequency Create a stimulus vector file which exercises the critical path delay of the adder and decrease the clock period (increase the frequency) until the adder fails in mixed-mode simulation. Record your minimum clock period and maximum clock frequency in the table. Be prepared to show your maximum clock frequency simulation waveforms to the TA for checkoff.

Power Characterization Simulate your pipelined adder design at **five** different frequencies between 1MHz and 1GHz using the power characterization stimulus vectors you developed for Lab 4. Record the average power consumption reported by your stimulus vector simulation at each frequency in Table and plot the power vs. frequency. How does the simulated power compare to the model of CMOS dynamic power dissipation from lecture?

Checkoff

Show your completed schematics, layouts, stimulus file listings, and all waveform plots to the TA for checkoff.

Report

You must hand in a **typewritten** report to receive credit for this lab. Your report can be brief, but must include the following sections in addition to the completed summary sheet attached at the end of this lab. The summary sheet will be the cover page of your lab.

1. Overview: Describe in one paragraph the objectives of the lab. State what you were testing and what data you expected to gather as a result of your experiments.

2. Procedure: Briefly document your methodology for acquiring the data you describe in the Overview. Describe how you measured the flip-flop timing parameters, verified the critical path delay of the adder, simulated power supply current and computed the average power. If you needed to buffer the clock signal in the testbench and insert extra delays to deal with timing violations, describe how you did so and how you determined the clock buffer sizes and number of additional inverter delays. Someone reading this section should be able to easily duplicate your results by following the methodology described in this section.

3. Results and Discussion: Describe succinctly the delay and power results captured in the completed summary sheet tables and any accompanying waveform plots and graphs. Do the results make intuitive sense? If not, explain why they might contradict your intuition.

Acknowledgments

Parts of this lab were inspired by lab exercises developed by Prof. David Money Harris and others at Harvey Mudd College for the class E158: Introduction to CMOS VLSI Design.

References

[1] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc., 2003.

[2] E. Brunvand, *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools*, 1st ed. San Francisco: Addison-Wesley, Inc., 2010.

EEC 116 Fall 2011 Lab #5 Summary

Name:

Grading:

Part	Checkoff	TA Initials	Date
1 `pDFF` Symbol			
1 `pDFF` Schematic			
1 `pDFF` Layout			
1 `pDFF` DRC			
1 `pDFF` LVS			
2 Flip-Flop Sim Waveform Plot			
`adder8b` Symbol			
`adder8b` Schematic			
`pReg8b` Symbol			
`pReg8b` Schematic			
3 `adder32bPipe` Symbol			
3 `adder32bPipe` Schematic			
3 `adder32bPipe` Layout			
3 `adder32bPipe` DRC			
3 `adder32bPipe` LVS			
4 Max. Frequency Sim Waveform Plot			
	Value		
4 Min. Clock Period			
4 Max. Clock Frequency			

Flip-Flop Delay Characteristics:

Parameter	Value
t_{su}(ps)	
t_{CQ}(ps)	
t_{hd}(ps)	

Adder Power Dissipation:

Parameter	Frequency	Power
f_0		
f_1		
f_2		
f_3		
f_4		