Digital Wavelet Decomposition Group

Final Report, 6/21/01

Group Members

Bob Henz	henz.2@osu.edu (contact)
Eric Balster	ericbalster@excite.com (secondary contact)
Dan Shen	shen.100@osu.edu
Gang Xiong	xiong.14@osu.edu
Xiaoli Yang	yang.346@osu.edu

Tape-out technology:	AMI 0.5 process
Tape-out Deadline:	6/25/01

Project Description

For our project we decided to implement an 8-bit Haar Discrete Wavelet Transform digitally. Although we will be implementing only a one-dimensional wavelet transform we plan for this to be a first step towards implementing different wavelet transforms in two dimensions for image processing. For now we will keep it simple due to time constraints and the inexperience of our group with Cadence and the fabrication process.

Time Schedule

The clock divide-by-2 circuit can be easily implemented with a D-flip/flop, and the shift right operation can be implemented by ignoring the LSB of the 8-bit value. Therefore, these components will not be considered in our time schedule but are included in the system block diagram for clarity and completeness. As can be seen in Table 1 we were able to meet our deadline.

Task Description	Who	Expected Completion	Actual Completion
Schematic for 8-bit Adder	Bob & Xiaoli	5/18/01	5/18/01
Schematic for 8-bit Register	Eric	5/18/01	5/18/01
Schematic for 8-bit Negation	Dan & Gang	5/18/01	5/18/01
Simulation of entire system (schematic level)	Entire Group	5/25/01	6/01/01
Layout of Register	Eric	6/01/01	6/08/01
Layout of Adder	Xiaoli, Dan, & Gang		
Layout of misc. components and help with layout questions etc.	Bob		
Run LVS and verify functionality	Entire Group	6/08/01	6/15/01
Design Freeze (no more layout changes!)	Entire Group	6/08/01	6/15/01
Combine Layout	Bob	6/15/01	6/20/01
Tape-out Deadline	Entire Group	***6/25/01***	***6/25/01***

Table 1: Task assignment schedule.

System Overview

A conceptial system diagram is show below. It's based on the calculations which are necessary to implement a Haar Wavelet transform. The system block diagram below divides the system into several units which need to be designed: An 8-bit register, a 2's complement inverter, and an 8-bit adder.

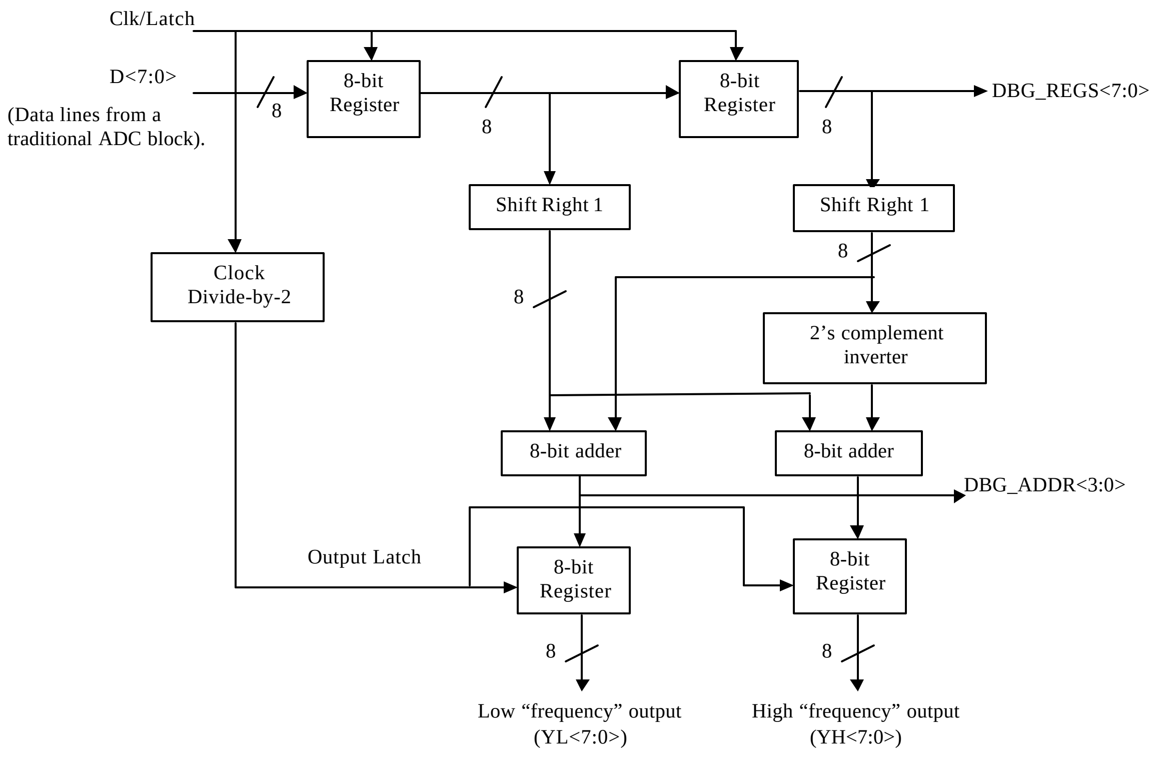

Figure 1: System block diagram for a Haar DWT decomposition.

The actual system schematic is shown if Figure 2. Notice that several short-cuts we implemented. For example there is no Shift Right unit since implementing such an operation simply requires dropping bit zero and shifting all the other lines over 1 pin. Also the 2's complement inverter is simply implemented by a bank of inverters and applying a '1' for Cin on the appropriate Adder unit.

Also notice the buffers on the clock line which are used to reduce the affects of fanout of the CLK signal.

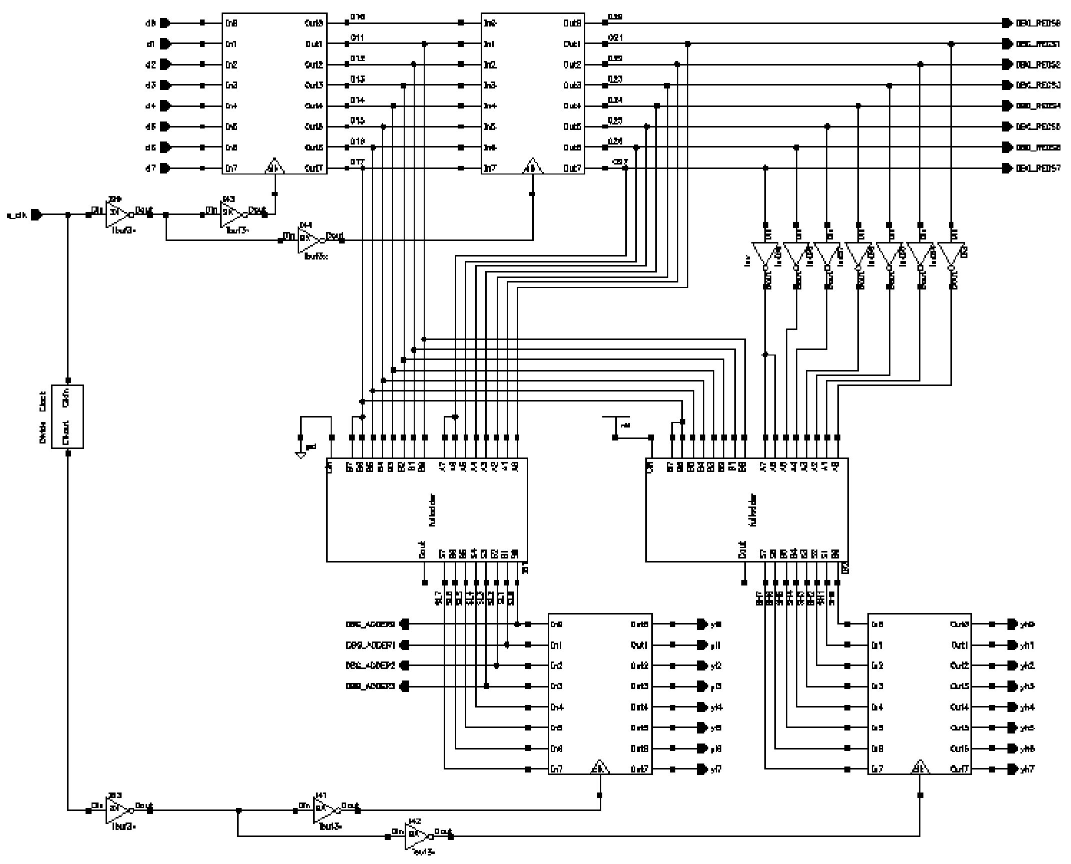
Figure 2: Schematic of the entire system.

System Simulation Performance

The system has no reset. The registers can be reset by clocking in two zero-valued bytes at start up. The system is rising-edge triggered with an eight-bit input being clocked in every clock cycle but a sixteen-bit output being clocked out only *every other* clock edge. Only the registers are clocked so the maximum clock speed is determined by the propagation delay of the registers and the Adders which is discussed later in this report. Figure 3 shows the delay between the rising edge of CLK and the change in the output lines. The delay is ~1.80ns.

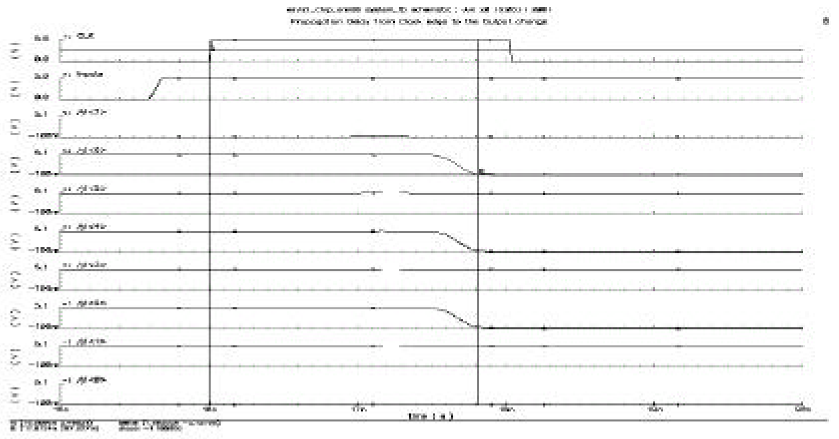
Figure 3: Delay from CLK rising-edge to change-in-output.

Pins
For this project we used the 40-pin package. Twelve of the 40 pins we were able to use for debugging and analysis purposes so that we can "probe" different intermediate signals on the chip once we get it back from fabrication.

Pin Name	#	I/O	Pin Description
D<7:0>	8	I	Sampled input. (Value between –128 and +127.) 2's signed compliment number.
YL<7:0>	8	O	Low frequency filter output. 2's signed compliment number.
YH<7:0>	8	O	High frequency filter output. 2's signed compliment number.
CLK	1	I	Input clock used to latch inputs and outputs.
DBG_REGS<7:0>	8	O	Debug signal to make sure our registers are latching the appropriate values.
DBG_ADDR<3:0>	4	O	Debug signal to make sure our adder is working. (Only the lowest 4 bits are looked.)
VDD	1	I	Digital power pin.
GND	1	I	Digital ground.
Unused	1	X	Reserved for future use.

Table 2: Pin-out table for 40-pin package.

Once the system layout was assembled and the signal routing between components completed, the entire system was dropped into the padframe provided by the NCSU kit.

To reiterate, the system really has only two major components, the adders and the registers. The other components are trivial and therefore will not be discussed in any more detail. The delay between rising clock edge and change in output was found to be ~1.8ns; however, it should be noted that the propagation delay through the register + the propagation delay through the adder will ultimately limit our maximum clocking speed.

Adder Unit

After considering several different designs for the 1-bit adder units we chose the design presented in Chapter 8 (page 526) of *Principles of CMOS VLSI Design, 2nd ed.* (Neil H. E. Weste, Kamran Eshraghian). This design is a transmission-gate adder which takes in 2 bits and a carry and generates a sum bit and a carry-out. We entered this schematic into Cadence and simulated. The circuit can be seen in Figure 4.

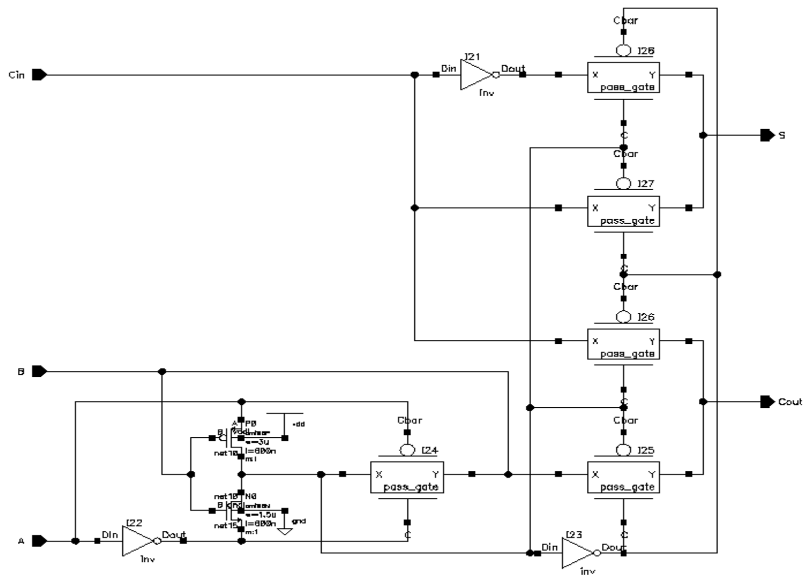

Figure 4: Schematic of one-bit adder

Once we had the 1-bit adder unit entered into Cadence we simulated it with Analog Artist and verified that it worked correctly. Next we chained 8 of these units together to create an 8-bit adder as is shown in the following Figure 5.

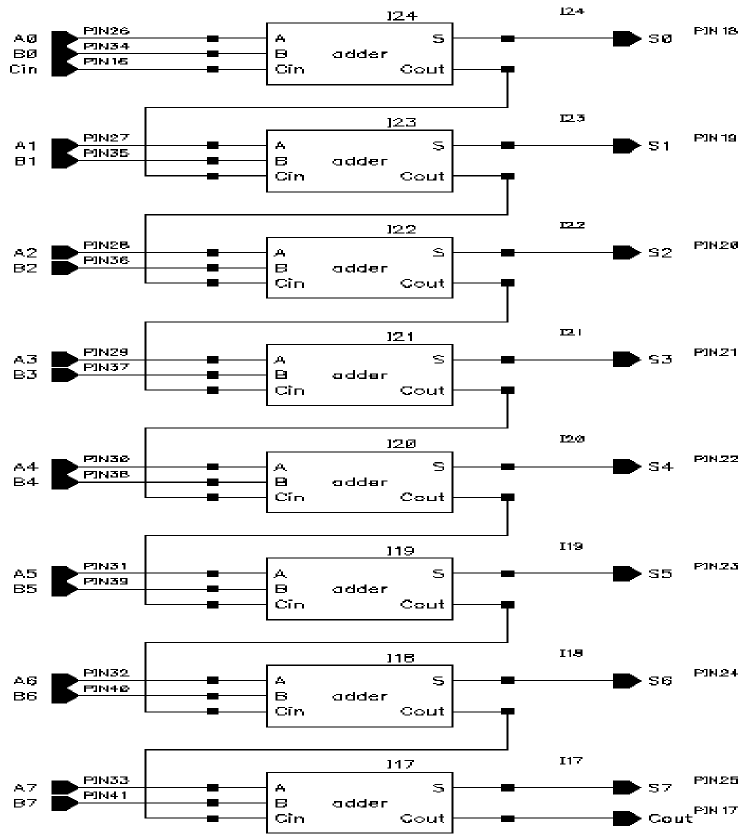

Figure 5: Full 8-bit ripple-carry adder constructed of 8 1-bit adders.

Finally we were able to simulate the adder with several different inputs and verify that it worked correctly. In our simulation example (see Figure 6), the period is 30ns, and the simulation interval is 80ns. The net1 is B0 and net17 is A0, the net 3 is B1 and the net19 is A1. S0=A0+B0+Cin(here we set Cin0 0), S1=A1+B1+Cin1(here Cin1=Cout0). It is shown that the delay is ~0.35ns. For the entire 8-bit ripple carry adder (worst-case scenario) the delay is ~2.4ns.

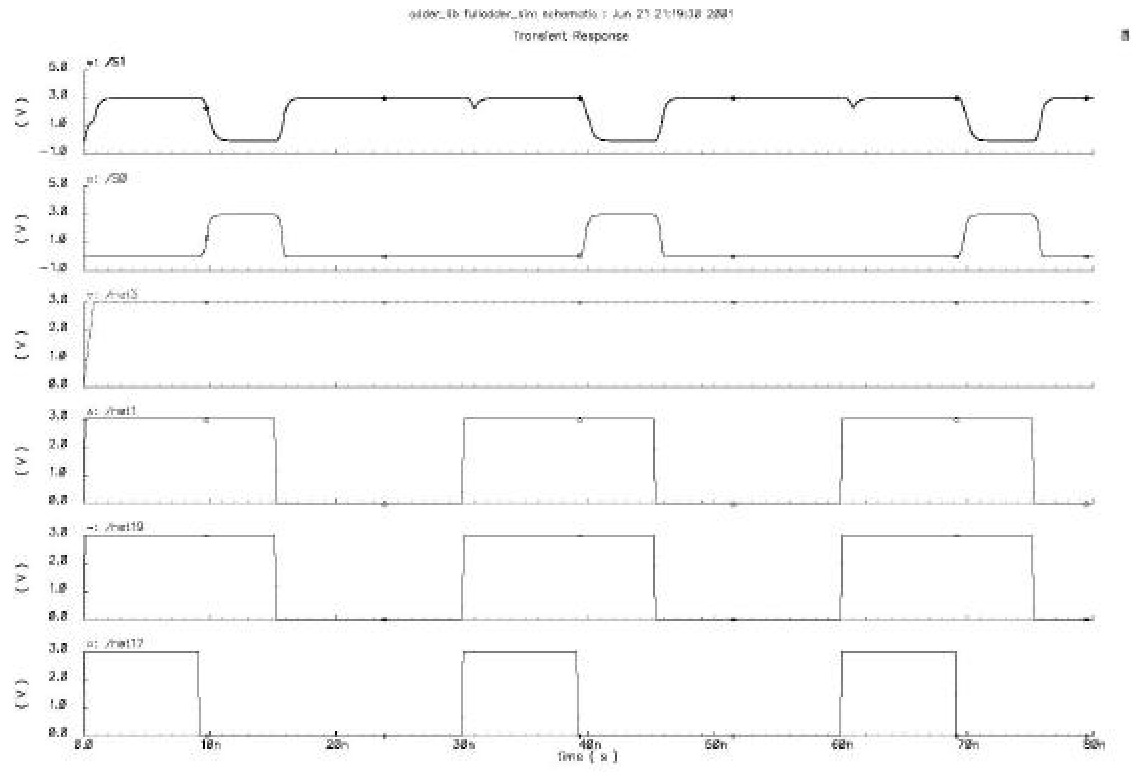

Figure 6: Post-Layout Simulation results

8-bit Register

An 8-bit register is needed in several areas of the final design of the Wavelet decomposition circuitry. Each tap of the low-pass and high-pass filters must include a register to hold the delayed signal values. Also, at the output, it is beneficial to register values for a smooth output signal, see Figure 1.

Our register is based on a D-type flip-flop design.   A D-type latch is first designed and is given in Figure 7.

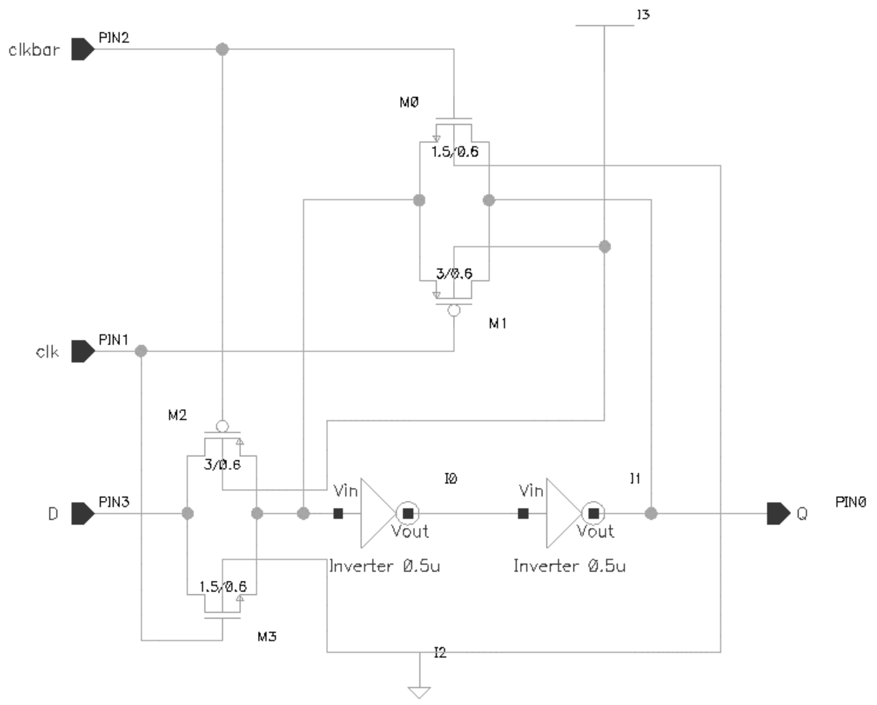

Figure 7: D-type latch

Two of the D-latches are tied together in series to form the master and slave of a D-type flip-flop. The Schematic for the D-type flip-flop is give in Figure 8.

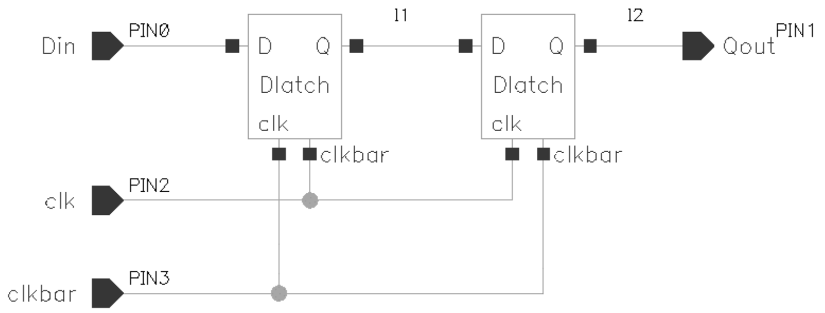

Figure 8: D-type Flip-Flop

The 8-bit register is simply 8 D-type flip-flops set in parallel with an inverter tying the signals *clk* and *clkbar* together. The schematic of the 8-bit register is given in Figure 9.

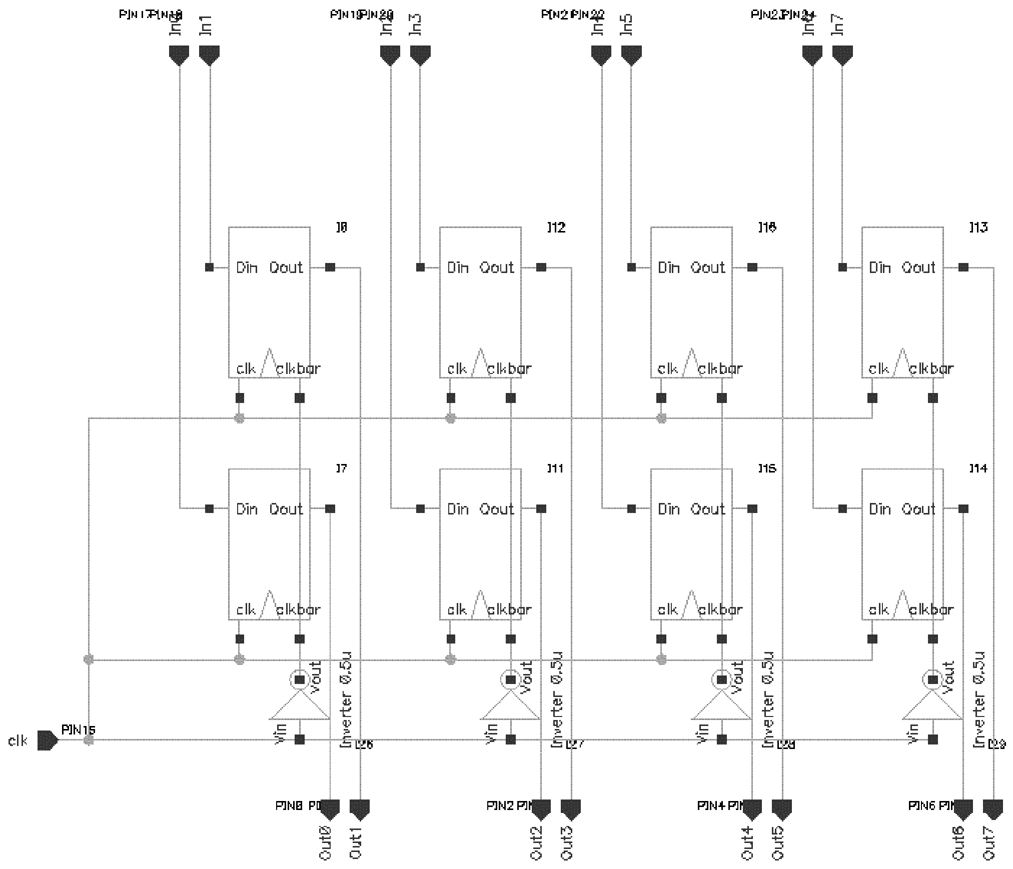

Figure 9: 8-bit register

Note that there are four inverters between the signals *clk* and *clkbar*. The four inverters are used instead of one to fan out the *clockbar* signal. The four inverter are able to drive the 8 flip-flops more quickly and thus improve the register's performance.

A simulation is run with the 8-bit register to ensure proper functionality using the Analog Artist tool in Cadence. Each of the eight register inputs are driven by the same input signal, and the outputs are verified. The results of the simulation are given in Figure 10.

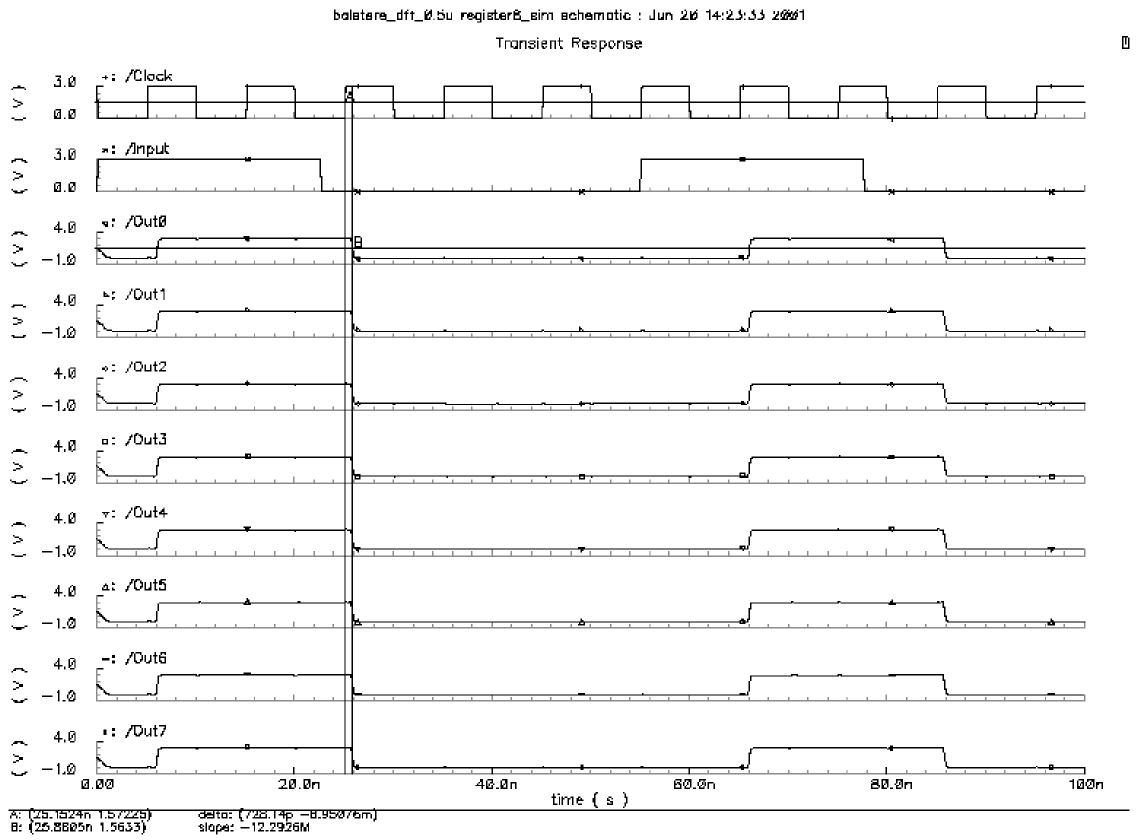

Figure 10: 8-bit Register Simulation Results

As seen from Figure 10, the register operates properly. Also, the delay of the registers is measured. As seen in Figure 10 the gate delay of the register is

$$delay = 728.14\,ps\,. \qquad (1)$$

The layout of the 8-bit register is complete and it is LVS clean. The layout of one D-type flip-flop is given in Figure 11.

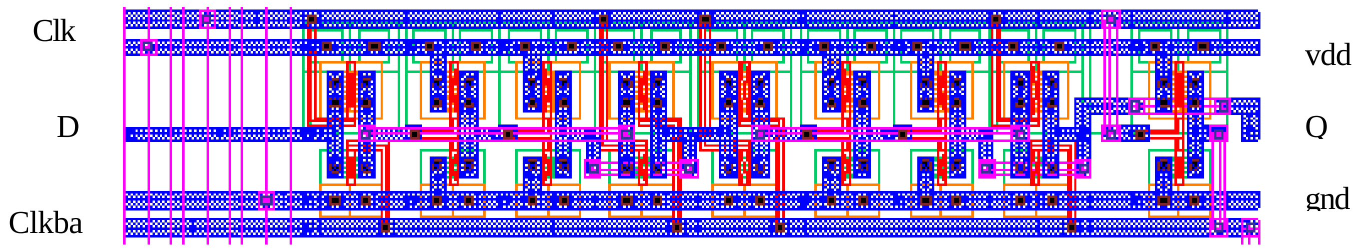

Figure 11: D-type flip-flop Layout

Although it may be difficult to see the various gates associated with the flip-flop layout, it is designed to minimize resource area. In the final layout design, 8 of these layout designs are stacked vertically, thus achieving an 8-bit register. Note, the signals *Clk* and *Clkbar* are tied together with an inverter seen in the right-hand-side of the layout. As shown in Figure 9, this inverter is used only on four of the eight flip-flops. However, to ensure a more robust design, the metal runs are established in all of the flip-flops, and extra "fan-out" inverters may be added easily if needed in future designs. Also, note that the *D* and *Q* signals are at the exact same vertical dimensions, and the *Clk*, *vdd!*, *gnd!*, and *Clkbar* signals run completely across the flip-flop layout. This works well for any design that may cascade registers. The register may easily be copied in the horizontal direction to form a cascaded set.

In summary, the register layout, seen partially in Figure 11, is designed to minimize resource area and facilitate ease in future designs. The layout is completely DRC clean and passes LVS with no errors when compared to the schematic given in Figure 9. The simulation gives correct and promising results.

Conclusion

Since the maximum clocking speed is limited by the propagation delay through the register followed by the propagation delay through the adder, the maximum clocking speed will be ~1/(2.4ns + 0.8ns) = ~300 MHz. This could easily be improved by implementing a carry look-ahead (CLA) adder instead of a ripple-carry.

Our chip is ready for fabrication. Bob took care of combining the layouts and doing the routing between the individual components. Because we are implementing digital circuitry (and also due in part to our inexperience), routing of the signals took up about 50% of the die space available to us inside of the pad-frame. When the entire layout was complete there was very little room remaining.

The combination of parts would have gone more smoothly if we'd agreed on a specific format for layout before plunging ahead. For example the layout would have been simpler if the adder and register's pins were ordered the same way from top to bottom (i.e. 0 to 7 vs 7 to 0). However because all group members were inexperienced, we did not think of such issues ahead of time.

In summary, all components were laid out in schematics, simulated, laid-out, DRCed and LVSed. This provides our group with a high degree of confidence that our chip will work when it returns from fabrication.