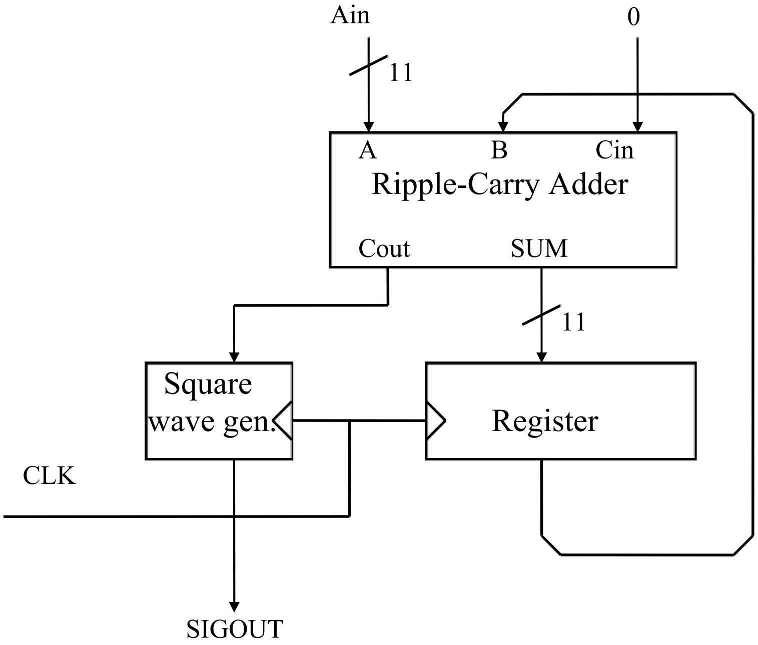
# Second Optional Homework Problem Set for Engineering 1630, Fall 2017

1. Lab B is the music box lab. If you do it, one of the choices you have for a method of synthesizing an arbitrary frequency uses an adder and register combination like the block diagram shown below. Write a Verilog description of this circuit using a ripple-carry adder for the adder block. Write out enough equations that the synthesis tool is guaranteed to make a ripple-carry adder rather than leave the problem of choosing an adder circuit to the synthesis tool. The $C_{in}$ bit is permanently zero. The $C_{out}$ signal from the adder will be asserted HIGH for only one clock cycle each time. The SIGOUT is to be a square wave at half the frequency of the $C_{out}$ signal. Observe proper coding practice for separating blocking and non-blocking statements into separate always blocks.

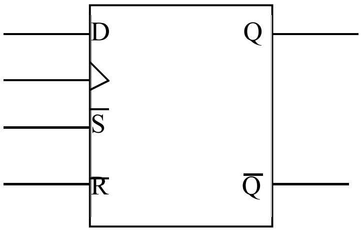
How many CLBs of the type shown below in problem 2 does it take to implement this circuit in an old Spartan/XL FPGA? Sketch how one stage from the middle of the ripple carry adder would be implemented, say for Ain(3) and B(3). In my answer, I will reproduce the CLB block diagram and I would suggest sketching on a printed copy of the CLB page. What would go in each LUT.

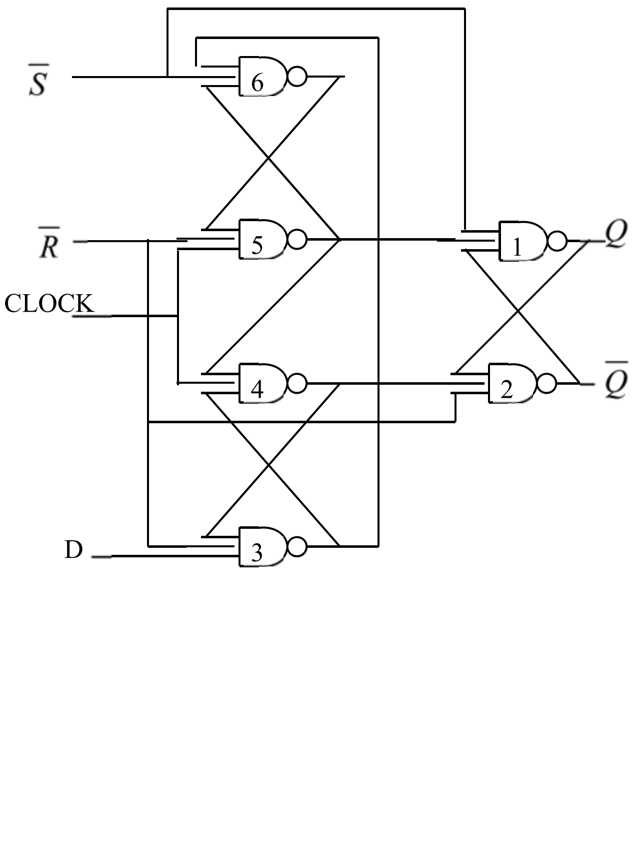


2. The gate level circuit below is an edge-triggered, D flip-flop as they were actually built in TTL and early CMOS processes. It is not a master-slave design but instead relies on propagation delays to assure reliable operation of an asynchronous state machine. While its operation is less obvious than the versions we have discussed, it has advantages in minimizing setup and propagation times. One feature is the existence of both asynchronous set and reset lines. Xilinx chip flip-flops, as well as several other

companies', FPGAs have only a single set or reset line per flip-flop and not both. If both are needed, then more than one logic block has to be allocated for the function, and the function may be realized from the gate equivalent circuit. (If you do this, timing may become problematic. This problem is really a mapping exercise not a practical procedure.) Xilinx CLBs can realize two Boolean functions of up to four inputs or one output of five inputs. Until relatively recently, there was also a restriction that a CLB could have only a total of 5 inputs. (The Spartan$^{TM}$ devices you used this year do not have that restriction. Still for timing considerations in this problem, continue to observe that limitation.) Assume you want to map the gates in the circuit below to generic CLBs with the five input restriction. Work out how the gates would have to be allocated to minimize the number of CLBs required. Probably one uses all CLBs as having just F and G outputs, so you can just work out the LUT usage. The Xilinx notation for CLB inputs is F1, F2,….G1,.. (See the appropriate figure in the Xilinx architecture datasheet or the same figure in my class handout.) For the CLB that contains gate 3, write the Boolean expressions for its F and G signals and list the assignment of CLB inputs to signals from the circuit. Which output, X or Y, will be the gate 3 output? Call the gate outputs Un where n is the gate number in the circuit. (The point of this problem is to get you to think fleetingly of the steps needed to turn a netlist into a bit file. Admittedly, one would probably not actually try to use this design because of the variable delays associated with the separate CLBs and their wiring.)

| $\overline{S}$ | $\overline{R}$ | $D$ | $C1$ | $Q^{n+1}$ |
|---|---|---|---|---|
| 0 | 1 | X | X | 1 |
| 1 | 0 | X | X | 0 |
| 1 | 1 | X | 0 | $Q^n$ |
| 1 | 1 | 0 | ↑ | 0 |
| 1 | 1 | 1 | ↑ | 1 |
| 0 | 0 | X | X | Und. |





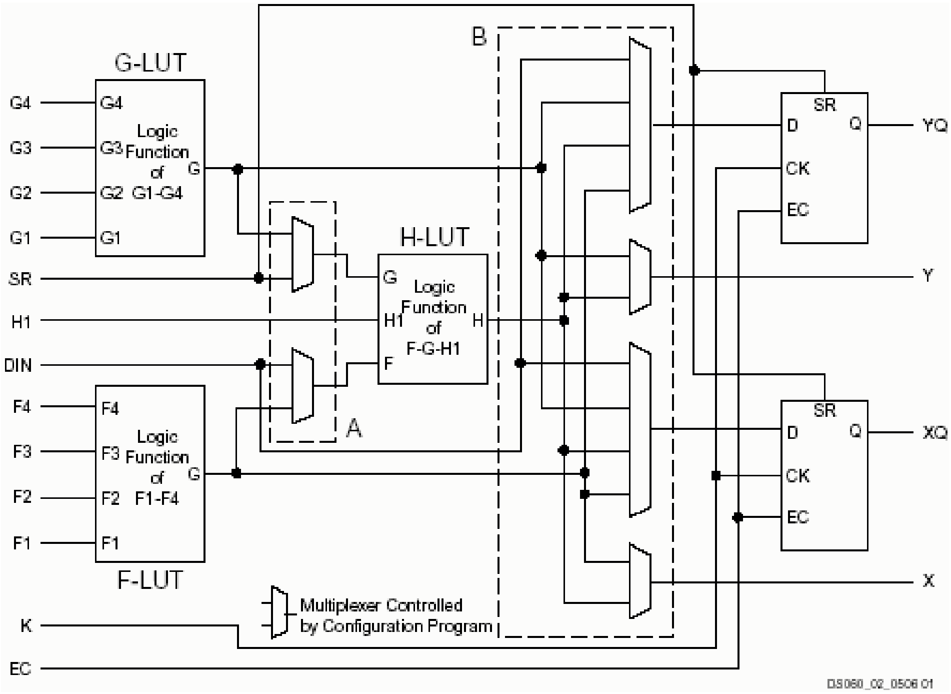
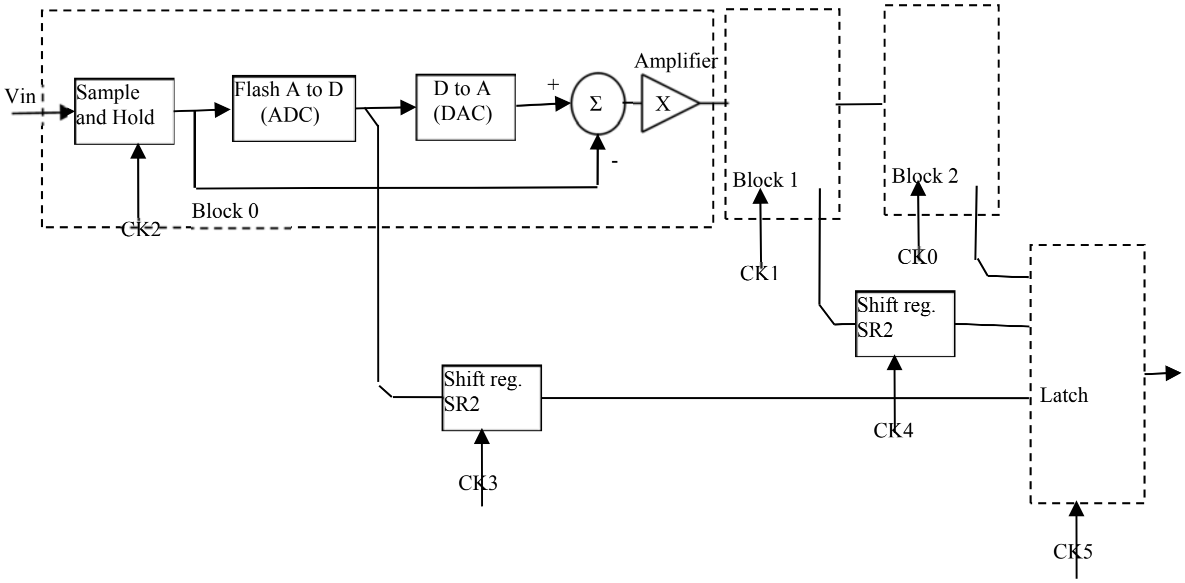


*Figure 2:* **Spartan/XL Simplified CLB Logic Diagram (some features not shown)**

3. Analog to Digital converters continue to be the subject of extensive work to improve their performance for particular purposes. This will probably continue to be the case for some years to come as the goals of low cost, high speed, and more resolution continue to be compelling and changes in semiconductor process technology open up new implementation possibilities. There are a large number of schemes that have been proposed for the task in addition to those we studied in class.

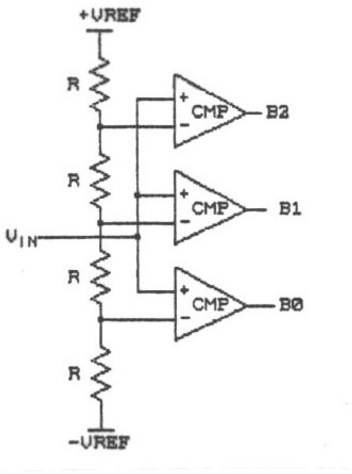
One such scheme is called a pipelined, subranging A/D converter, which is shown below in block form for a three stage system. In this problem we will consider some of the simple, primarily digital aspects of such a design. The way the system works is as follows: there are three stages. Two identical stages each consist of a sample and hold circuit, a fast but crude A/D converter, a good D/A converter, an analog subtractor, and an analog amplifier.



The input signal is first sampled, and a fast estimate with a small number of bits is made of its amplitude. These bits are the most significant bits of the final result. Simultaneously, the digital result is converted back to analog by a D/A converter and subtracted from the original analog signal. This difference is amplified back up to the same full scale voltage as the original signal, so that a crude estimate of its amplitude would give the next most significant bits of the result. This difference signal, called the residual signal, is passed on to a second stage to calculate the next bits. This process is repeated however many times is necessary to get the desired number of bits.

The pipelined subranging A/D converter draws its primary advantage from the fact the at any given time, the first stage operates on the most current analog input sample, while the second stage operates on the residual signal from the previous input sample, and so on. The simultaneous operations result in an A/D converter with a conversion speed that is only limited by the time it takes to process the analog information in one stage. One consequence of this, however, is that the result bits of the first two conversions have to be delayed until the third stage has completed its conversion. This is the function of the shift registers.

3.1) Suppose that the flash A/D converters are to be two bit, flash converters using the comparators drawn below. The full scale voltages are to be plus and minus $V_{REF}$ and that reference voltages of these values are available as shown. Draw the rest of the A/D converter, showing how the comparator inputs are connected, how many signals are used, and the logic for the priority encoding by which the outputs of the comparators ($B_0$, $B_1$,...) generate the converter outputs D0 and D1. Use Karnaugh maps, if necessary, to find the minimum logic for the priority encoding; make use of any “Don't Care” conditions in the K-maps for states which will not occur because of the way the comparators are connected.



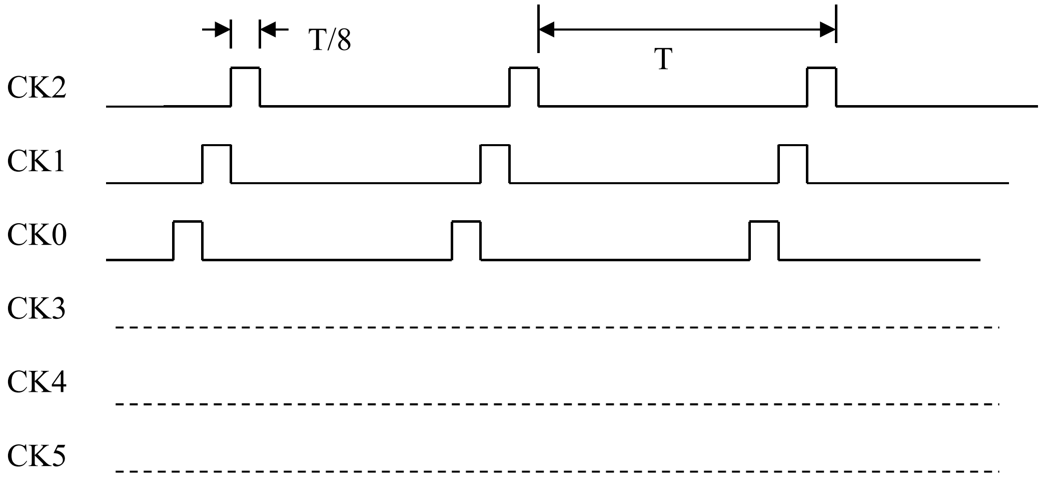
3.2) Draw or describe what is likely to be in the last subranging block BLOCK0.

3.3) How big a gain does the “AMPLIFIER” section have to have? (An amplifier multiplies its analog input signal by a constant factor called the gain. How big a factor is required in this system.)

3.4) Describe in words what sets the average conversion time of a sample, i.e.in the timing diagram of section 6 below, what factors set the time T?

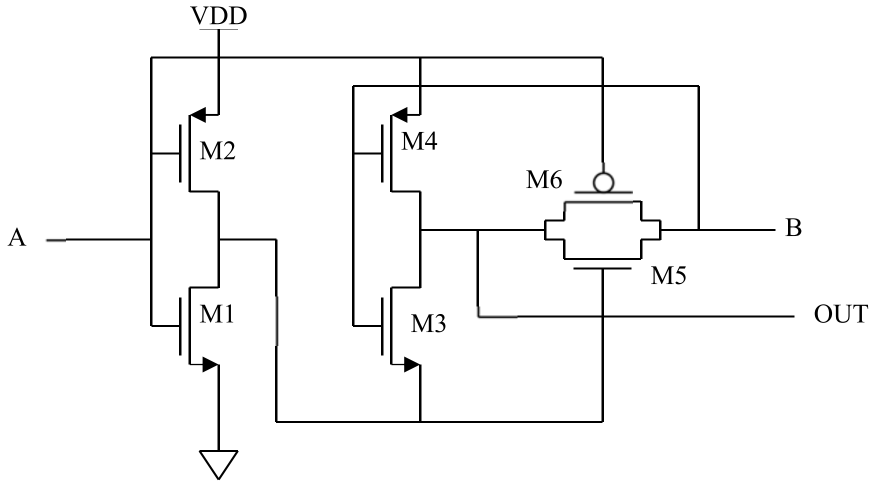
3.5) How many delay stages are required in SR2 and SR1?

3.6) On the timing diagram below, mark on the lines for CK2, CK1, and CK0 when each block is done with the conversion of a single input sample. Then draw signals for CK3, CK4, and CK5 assuming that the shift registers and output latch are positive edge triggered.



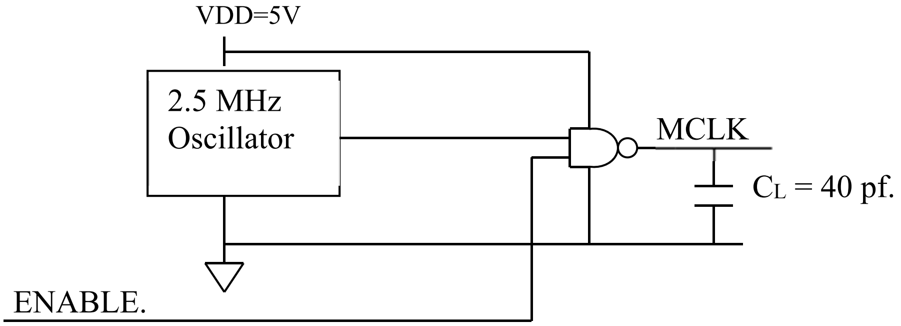
3.7) Draw a circuit of gates, counters, flip-flops, etc. of your choice which derives the clocks CK0 to CK5. Also show what SR1 would actually be in terms of these simple building blocks.

4. The gate below uses some of its transistors as a transmission gate. What is the function produced on the output of the circuit and why? This is not really a good or even usable gate. Why?



5. Suppose you wish to make a simple 3-bit Gray code counter (sequence length 8) from D flip-flops that only have a Q output (no $\overline{Q}$ ). How much logic do you need?

6. We wish to study the power dissipation in the circuit below. When the signal marked “ENABLE” goes high, there is an increase of 5 mW in the power from the power supply. The capacitor represents the load external to the gate. If that load were removed, how much current would the NAND gate draw when the circuit is enabled?


VDD=5V
2.5 MHz
Oscillator
MCLK
CL = 40 pf.
ENABLE.