Answers to ENGN1630 Second Optional Problem Set – Fall 2017

1.)
```
module p1h2ay11 (input [10:00] A, input clk, output reg SIGOUT);

wire [10:0] sum;
wire [11:0] cy;

reg [10:0] accum;
genvar i;

generate
for (i = 0; i < 11; i = i + 1) begin: AddTree
      full_add(A[i], accum[i], cy[i], sum[i], cy[i + 1]);
      end
endgenerate

assign cy[0] = 1'b0;

always @ (posedge clk)
begin
accum <= sum[10:0];
SIGOUT <= SIGOUT^cy[11];
end

endmodule

module full_add (input wire a, b, cin, output wire s, cout);

assign s = a^b^cin;
assign cout = (a && b) || (a && cin) || (b && cin);

endmodule
```

This implementation uses structural components to avoid having to write both the sum and carry expressions repeatedly. A tedious but perfectly reasonable alternate to the generate loop would be to write twelve structural lines of the form:

```
   full_add U1 (A[0], accum[0], 1’b0, sum[0], cy[1]);
   full_add U2 (A[1], accum[1], cy[1], sum[1], cy[2]);
```

Each bit of the adder, whether the least significant half adder or one of the 10 full adders, has two outputs, CARRY and SUM so it can be built in one CLB of this product family. There are sufficient flip flops in the CLB that the register does not occupy additional blocks. The output square wave generator, however, does require and additional half of a CLB. So the total number of CLBs is 11.5.

All three bits, Cin(), A(), and B() are required in calculating both the sum bit to be registered and the carry out bit. Here is a CLB wired to do the third bit. The drawing assumes that the output of the G lookup table goes to the D input of the YQ flip-flop. The X combinational output is fed by appropriate multiplexor selection from the F-LUT and is the carry out signal to the next adder or the square wave generator block. (Note: there is a typo on the original block diagram and the output of the F-LUT is labeled G instead of F.)

The G-LUT implements $G4 \oplus G3 \oplus G2$ and has 16 non-zero entries; the F-LUT implements $F2 \bullet \left(F3 \oplus F4\right) + F3 \bullet F4$

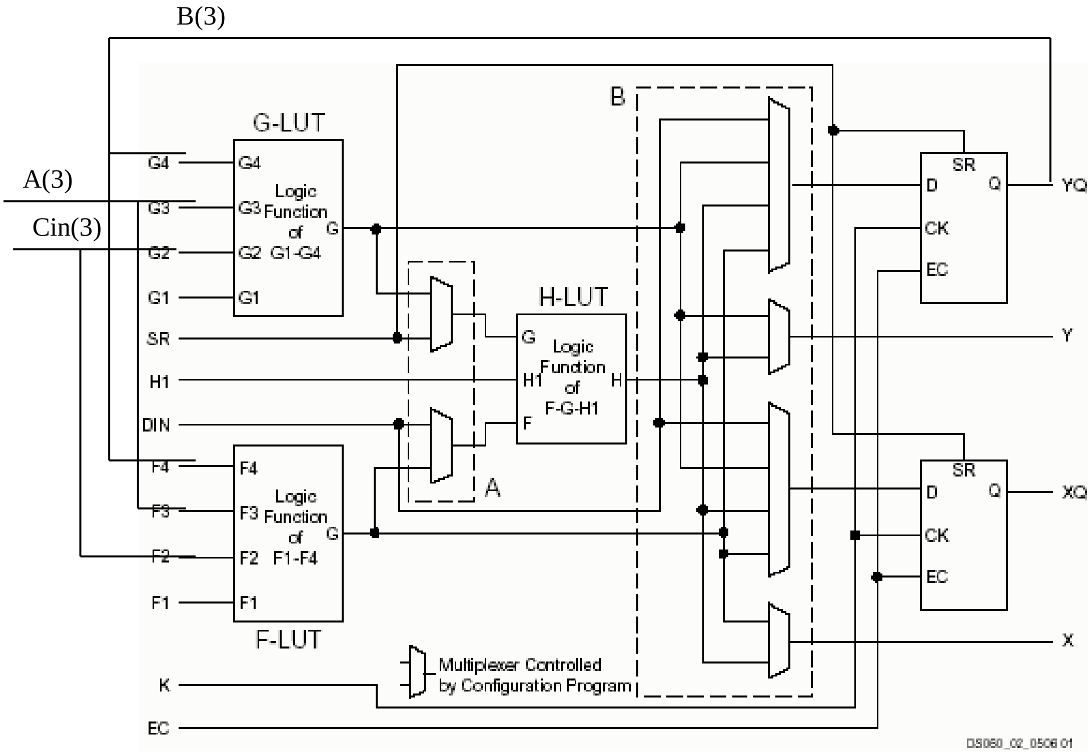
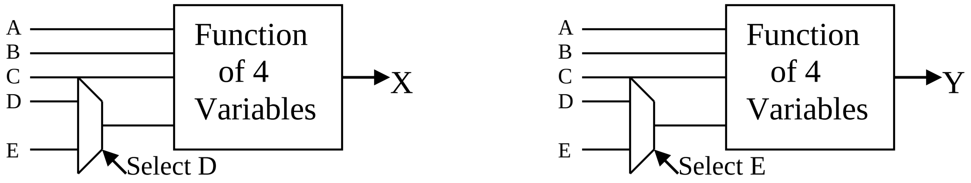


Figure 2: **Spartan/XL Simplified CLB Logic Diagram (some features not shown)**

2.) The first constraint is that each CLB can have only 5 inputs and 2 outputs. One way to get this is:

CLB #	GATE #
1	1,6
2	5,4
3	2,3

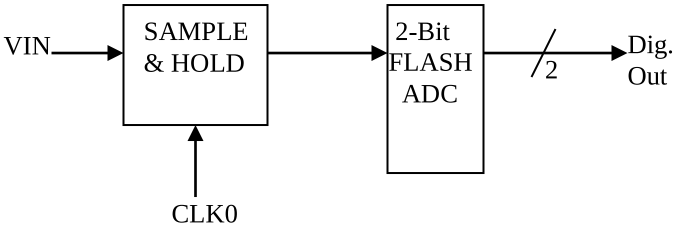


The division is forced because each gate of the original design has 3 inputs. Only pairs of gates sharing an input can fit within the limit of 5 inputs. For gate # 3, first consult fig. 5 shown in the problem. The combinational form for CLB #3 has to be "FG Mode" because there are to be two combinational outputs. The block is purely combinational, so the outputs are set as X = F and Y = G. These set up a configuration with the inputs split as in the figure with two LUTs above.

Suppose one connects A as the $\overline{R}$ of the flip-flop, B as the output of gate 4, C as unused, D as the D-input of the flip-flop, E from the Q output of the flip-flop, X as the gate3 output, and Y as the $\overline{Q}$ output of the flip-flop. While there are other possibilities from rearranging the inputs, this arrangement requires the CLB to be programmed for $F = \overline{A \cdot B \cdot D}$ and $G = \overline{A \cdot B \cdot E}$. The lookup table itself is written as the truth table of the NAND function with the table filled in such a was as to make the C input have no effect on the output.

3.1) $D_0 = B_0 \cdot \overline{B_1} + B_2$ and $D_1 = B_1$. The output of the small flash converter is a thermometer code, that is 000 is for Vin below the B0 negative input, 001 is above the bottom divider potential but below the mid-point, and so on. No input with a zero to the right of a 1 is possible so those don't cares simplify the expressions substantially.

3.2) Contents of the last subranging block, BLOCK0:



There is no DAC or analog amplifier needed because this is the end of the chain.

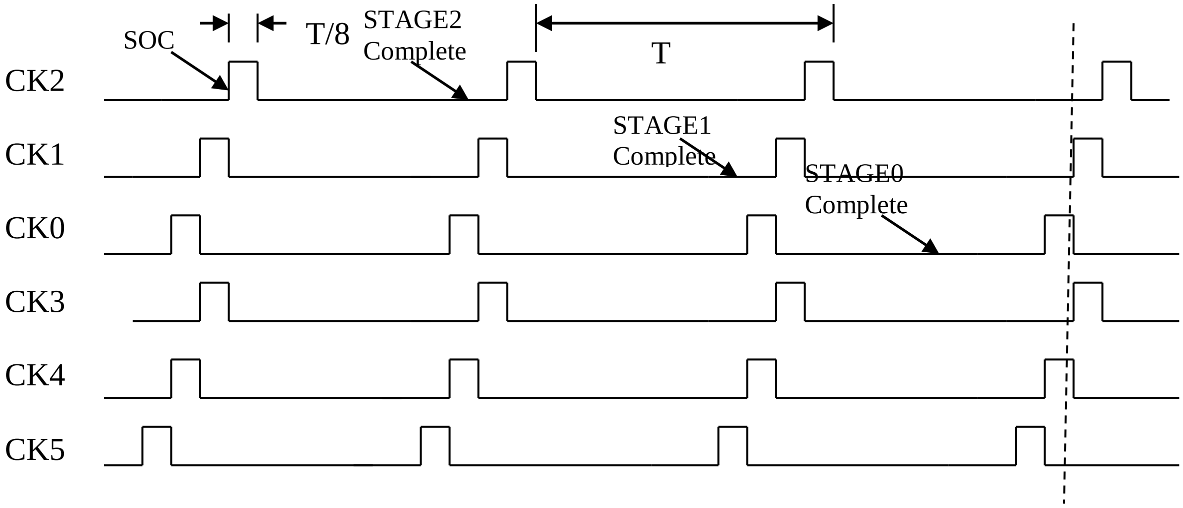
3.3) Each DAC subtraction section reduces the range of the input signal by a factor of 4 which the amplifier must make up to give the next section a full input signal. Thus the required gain is X4 (2-Bits or $2^2 = 4$ or 12 DB).

3.4) The pulse width of these signals sets the sampling time of the first sample and hold circuit. Its width is determined by the need for that sample and hold to settle to full LSB accuracy. The time between pulses is long enough for the ADC/DAC and analog difference amplifier to settle before the second stage sampling begins. That sampling must be finished before a new sample can start at VIN. Therefore $\boxed{T_{TOTAL} = 2T_{S\&H} + T_{ADC/DAC-SETTLE}}$

3.5) The number of delay stages in the shift registers are:

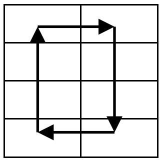
In SR2: 2 stages
In SR1: 1 stage

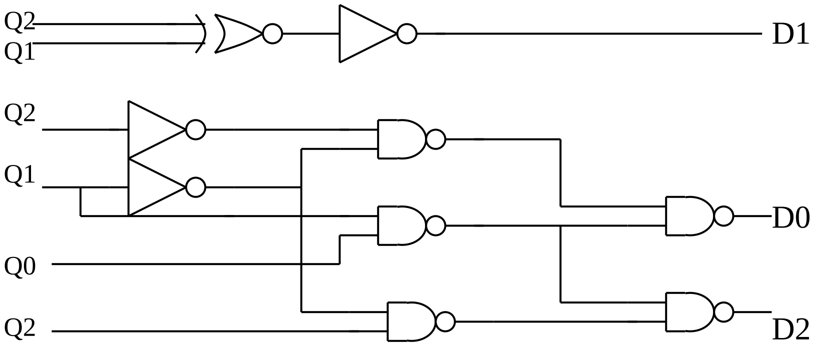
3.6)



4.) Transistors M5 and M6 are a transmission gate which is closed when A is LOW (and $\overline{A}$ from the inverter on the left is HIGH). This TG connects B to the output. If A is HIGH, transistors M3 and M4 function as an inverter between B and the output. (When A is LOW, transistors M3 and M4 cannot be turned on by B because the gate-source voltages are incorrect. Transistor 3 has its nominal source at VDD so the terminal connected to B by the transmission gate is its real source. Similarly with the nominal source of M4 at ground, that transistor's actual source is connected to the same node as the source of M3 and through the transmission gate to B. Both devices have B on their gates so they are turned off because of zero gate-source voltage.) Thus: $OUT = \overline{A} \cdot B + A \cdot \overline{B} = A \oplus B$. This is a very poor implementation of the XOR gate because the output is not fully restored to VDD and GND levels. The output signal is always taken from one of the two input signals and not from the power supply. The speed will be slow and logic levels will degrade in successive stages of this type.

5.) This is a straightforward counter design. One way to find the sequence is to traverse a 3 variable K-map clockwise through the 8 boxes. This leads to:





6.) The current drawn by the NAND gate when the ENABLE signal is HIGH is I = P/V = .005/5 = 1 ma. The portion attributable to C_L is

$$I = \frac{P_L}{V_{DD}} = \frac{C_L f_C V_{DD}^2}{V_{DD}} = C_L f_C V_{DD} = 4 \cdot 10^{-11} 2.5 \cdot 10^6 5 = 0.5 mA.$$

The NAND gate alone, therefore, would draw 1ma – 0.5 ma = 0.5 mA.