VLSI Design : Assignment-3

Spring 2021, IIIT Hyderabad (Instructor: Abhishek Srivastava)
Due date : 2 Apr, 2021 (18:00 hrs)

Instructions:

1. Submit your assignment as a single file in pdf format (Name_RollNo.pdf)
2. Use the given 180 nm technology file for the NGSPICE simulations
3. Consider lengths of NMOS and PMOS to be equal ($L_n = L_p$), and $V_{DD} = 1.8V$ until stated otherwise
4. Use 'set curplottitle= Your-name-roll-question-number-part' for every plot in your report so that it is printed on the top of each plot
5. Answers should be complete and must be presented in a systematic way with explanation, plots, annotations net-lists and HDL description
6. Utilize moodle platform to discuss and clear your questions. Discussion is highly encouraged.

1. Size the pass transistor logic based multiplexor shown in Fig. 1 such that the average delay from input (A or B) to output (Y) is minimized. It is given that the electrical effort (C_{out}/C_{in}) for each path (input to output) is 2. Length of each transistor in the circuit is same (L=0.18 μm). Widths shown in the figure represents the ratio of PMOS and NMOS devices. There is a constraint for the output inverter that the total width $W_p + W_n = 6W$. Use NGSPICE simulations by including parasitic capacitances of the devices to find the sizes to minimize the delay. For your final design, show the functionality of the multiplexor with appropriate waveforms and report the delay of your circuit.

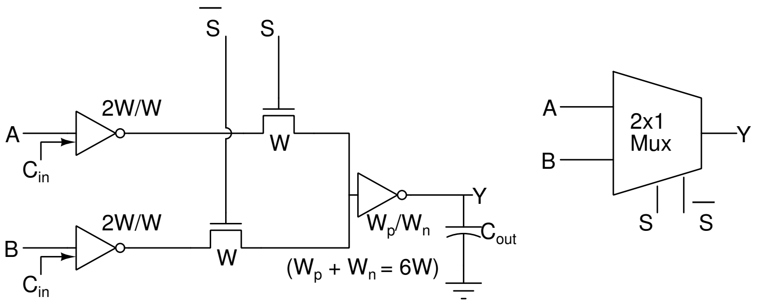


Figure 1

2. Use the 2×1 multiplexor (only) shown in Fig. 2 and implement the following logic function *(Hint: Use Shanon's expansion.)*:

$$f = x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$$

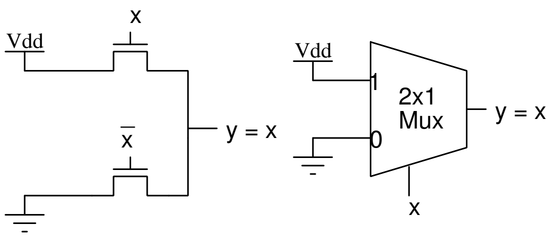


Figure 2

Each input (x_1 to x_4) should see a high input impedance (goes to gates of pass transistors) and the load capacitance at the output is equivalent to 4 times the minimum sized inverter. Use the same size (W) for pass transistor obtained in the previous problem.

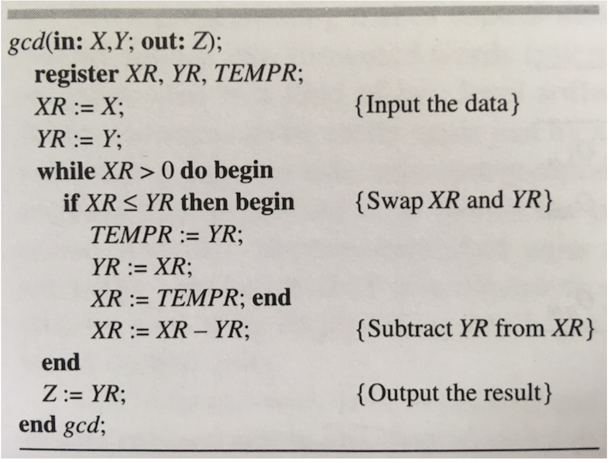
(a) Give the circuit diagram

(b) Write the spice netlist for the circuit and verify the functionality with simulations. Attach your plots.

(c) Find the minimum transition time taken by output (f) to change from - i) 0 to 1 (t_{PLH}) and ii) 1 to 0 t_{PHL}. Clearly mention the input combination and paths for charging and discharging the output in your circuit.

(d) Do you observe any difference in t_{PLH} and t_{PHL}. If yes, then can you suggest some modification in your circuit to make them nearly equal. *(Hint: Charging path (V_{DD} to out) and discharging path (out to ground) should have equal number of transistors of same size. Repeaters (inverters) can be inserted at appropriate nodes to achieve the same.)*

3. *(GCD control unit design, Reference - Computer Architecture and Organisation, third edition by John P Hayes):* You are asked to design control unit of a simple digital circuit that can find the greatest common divisor (GCD) of two given positive integers. A variant of Euclid's Algorithm proposed by Cormen, Leiserson and Rivest in 1990 is given in Fig. 3, which can be used to find the GCD of two integers X and Y. For example GCD(20,16)=4 and GCD(27,17)=1.



```
gcd(in: X,Y; out: Z);
  register XR, YR, TEMPR;
  XR := X;                        {Input the data}
  YR := Y;
  while XR > 0 do begin
    if XR ≤ YR then begin         {Swap XR and YR}
      TEMPR := YR;
      YR := XR;
      XR := TEMPR; end
      XR := XR − YR;              {Subtract YR from XR}
  end
  Z := YR;                        {Output the result}
end gcd;
```

Figure 3

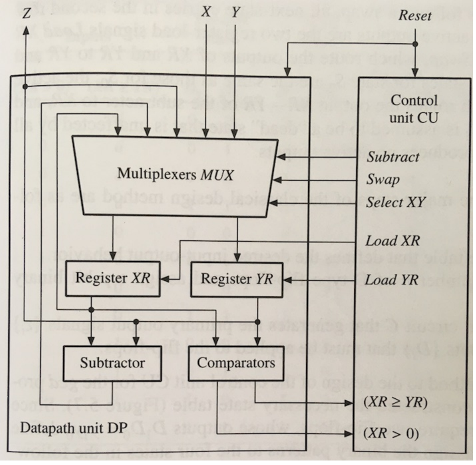
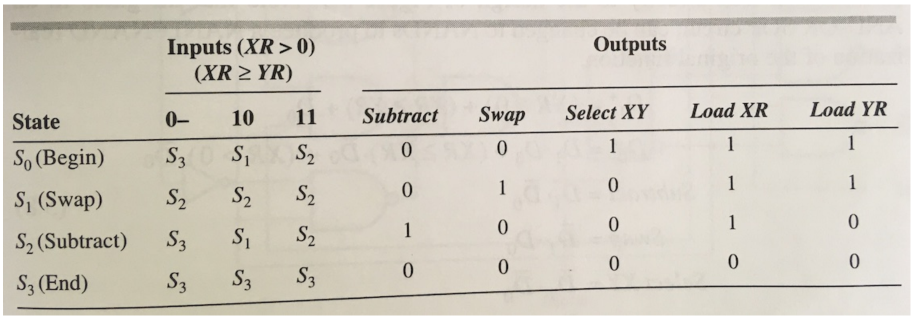


Figure 4

From the algorithm shown in Fig. 3, it is identified that following sub-circuits will be required to implement the complete circuit- 1) registers to hold X and Y data, 2) comparator unit (X$>$ 0 and X$\leq$Y), 3) subtracter unit (X = X - Y) and 4) set of multiplexers to realize the swap operation. The input to the circuit will be X, Y, a clock signal (CLK) and a reset signal. The circuit will also need control signals (instructions) such as subtract, swap, load-X and load-Y, etc.. Fig. 4 depicts the block diagram to realize the GCD algorithm by showing the data-path unit and control unit of the circuit.

In order to generate control signals or instructions, we need to evolve a finite state machine (FSM) for the control unit shown in Fig. 4. The 4 states shown in Fig. 5 can be used to build the FSM of the circuit.



	Inputs ($XR > 0$) ($XR \geq YR$)			Outputs				
State	0–	10	11	*Subtract*	*Swap*	*Select XY*	*Load XR*	*Load YR*
S_0 (Begin)	S_3	S_1	S_2	0	0	1	1	1
S_1 (Swap)	S_2	S_2	S_2	0	1	0	1	1
S_2 (Subtract)	S_3	S_1	S_2	1	0	0	1	0
S_3 (End)	S_3	S_3	S_3	0	0	0	0	0

Figure 5

With the help of above informations, do the following:

(a) Build the state diagram corresponding to the state table shown in Fig. 5.

(b) Give the circuit diagram of the control unit using digital blocks (gates/muxes/flip-flops).

(c) Implement the control unit circuit using Verilog HDL and show its functionality with simulation results. Use positive edge clock transitions in your design.

(d) Give a behavioural description of the data path unit using Verilog HDL and show the complete functionality of your circuit with simulation results

(e) Compute GCD of (27,19) and (24,16) with the help of your HDL model and show the simulation results

Acknowledgement: Figures in problem 3 are taken from the book - Computer Architecture and Organisation, third edition by John P Hayes.