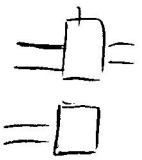
CS2102 Digital Logic Design Final Exam

10:10-11:50am, Thursday, January 14, 2010

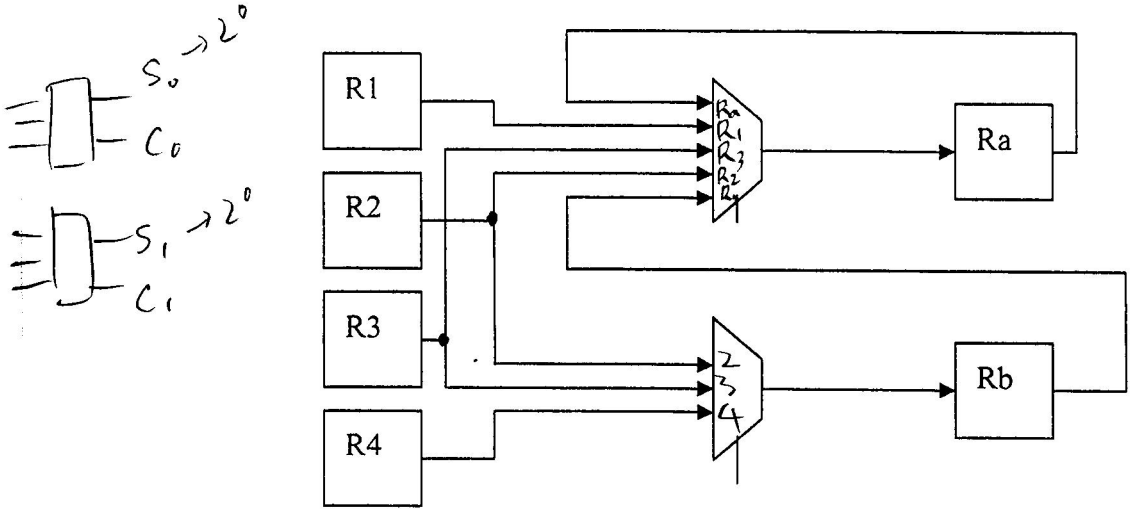
1. (20%)
   A. What is *noise margin of a gate*?
   B. What *propagation delay of a gate*?
   C. What is *set-up time of a flip-flop*?
   D. What is *design reuse*?



2. (10%) Design a 7-input (A_6, A_5, …, A_0) 3-output (X_2, X_1, X_0) combinational circuit using four, and only four, 1-bit full adders as your basic building blocks. X is a binary number indicating the number of 1s in A. For example, when A is "1010101" X should be "100" and when we have "1111111" in A, X should output "111."

3. (10%) From the datapath of the small computer of Chapter 9, you have learned how to make a Two-Read-One-Write(2R1W) register file. In many advanced CPU design, register files of more read/write ports are needed for parallel processing. Design a Three-Read-Two-Write(3R2W) register file of four 8-bit registers.

4. (15%) Suppose all registers below are positive-edge-triggered and synchronized to a common clock signal. List every set of register-transfer operations possible in a cycle. For example, it can perform two transfers, Ra ← R1 and Rb ← R3, simulatneously in a cycle.



5. (20%) Given the datapath depicted in Figure 1, what are the values the controller should set for the following signals in order to perform the following four instructions?

	MB	MD	MW	RW	AA	BA	DA
R5 ← M[R5]	X	1	0	1	101	XXX	101
M[R3] ← R7	0	X	1	0	101	111	XXX
R2 ← R7 + 2	1	0	0	1	111	XXX	010
R3 ← R0 – R3	0	0	0	1	000	011	011

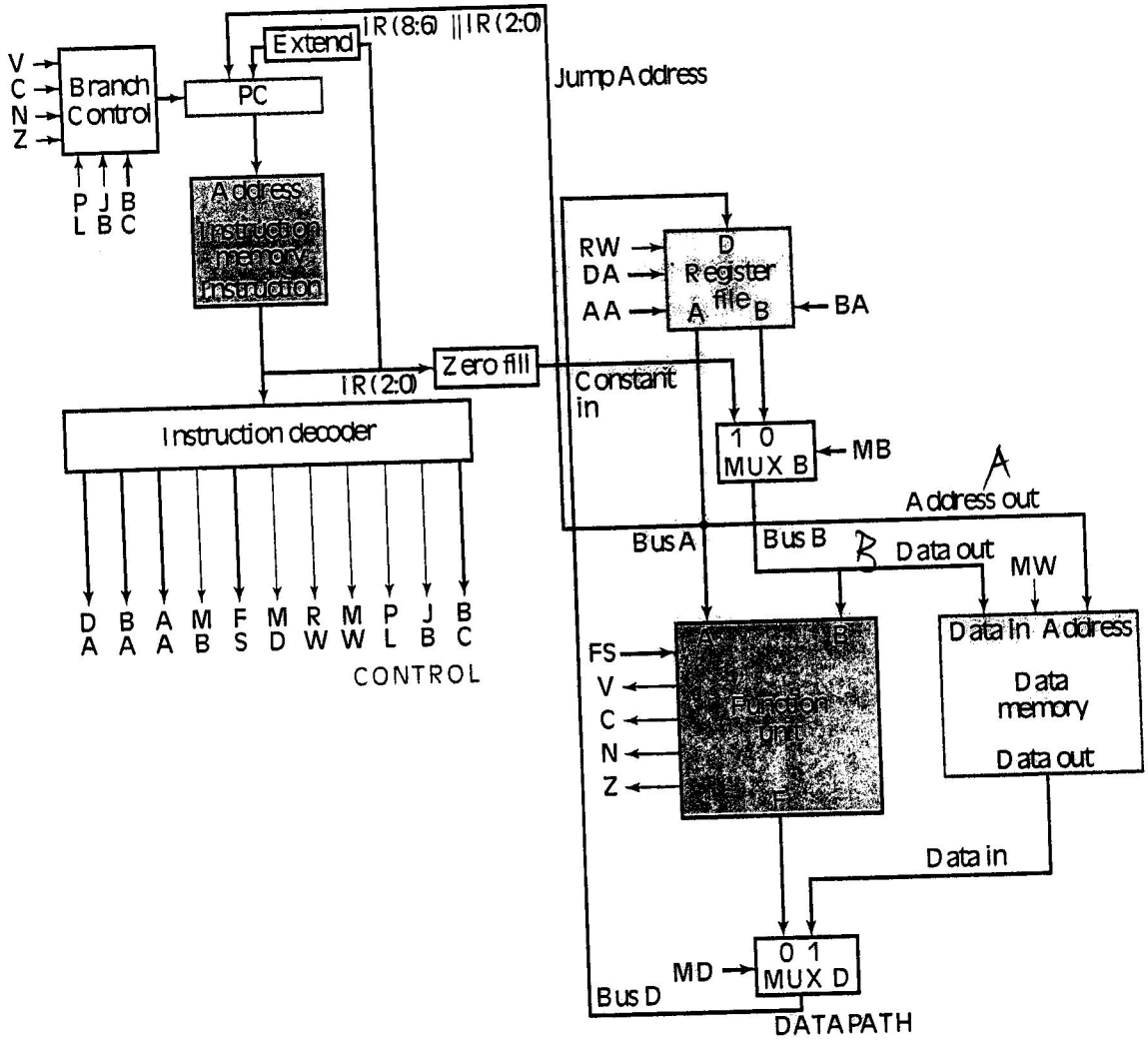


Figure 1

6. (20 pts) Use the multi-cycle computer from the text book and copied in Figure 2. Design a microprogram (in state-transition diagram) so that it can perform the new instructions

   A. Reorder_RF; //the instruction swaps the conetnts of registers R0…R7 to that of R7…R0.

   B. M[Rc] ← M[Rb] - M[Ra]; // Ra, Rb, Rc serve as pointers to data memory locations

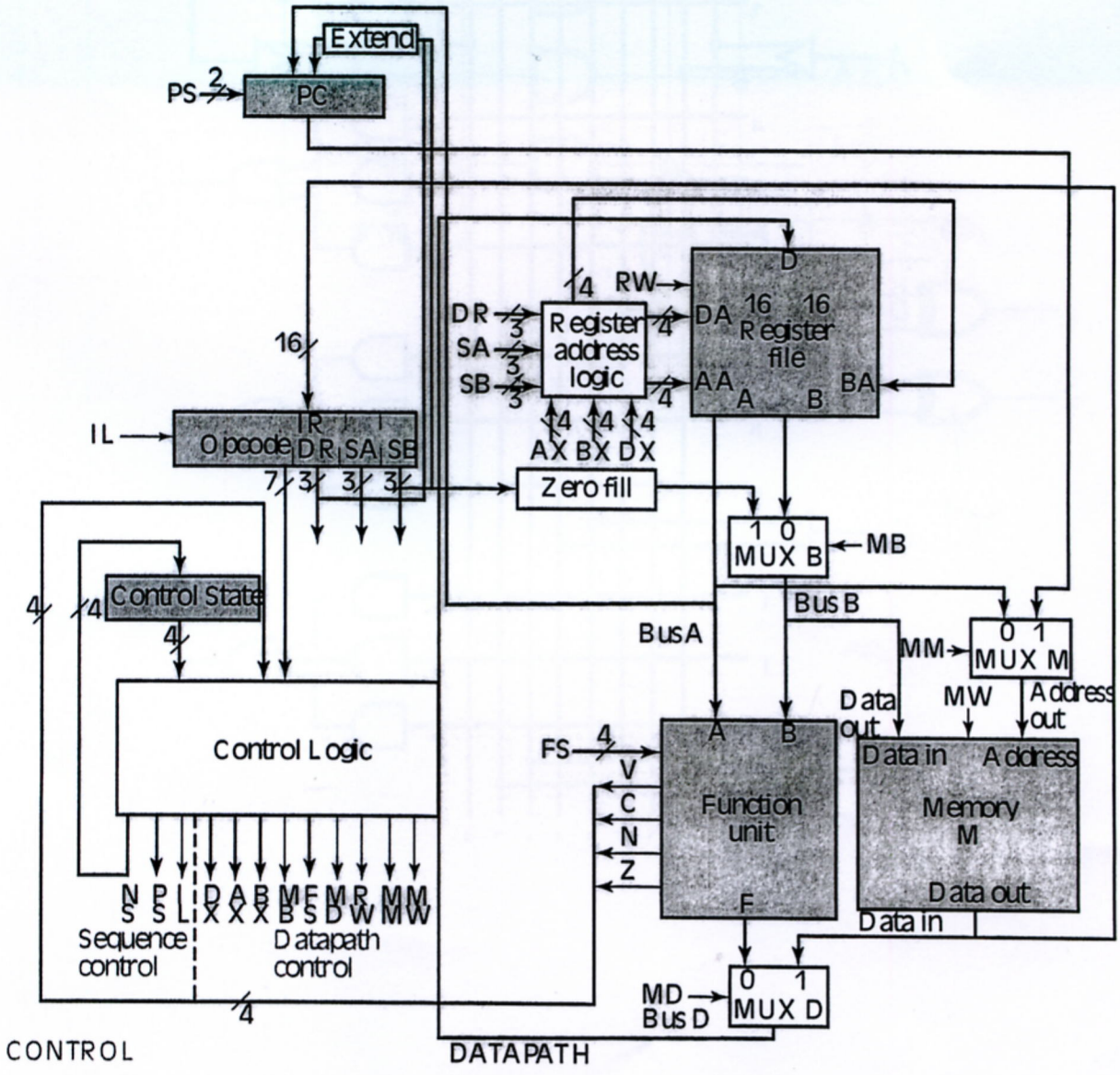
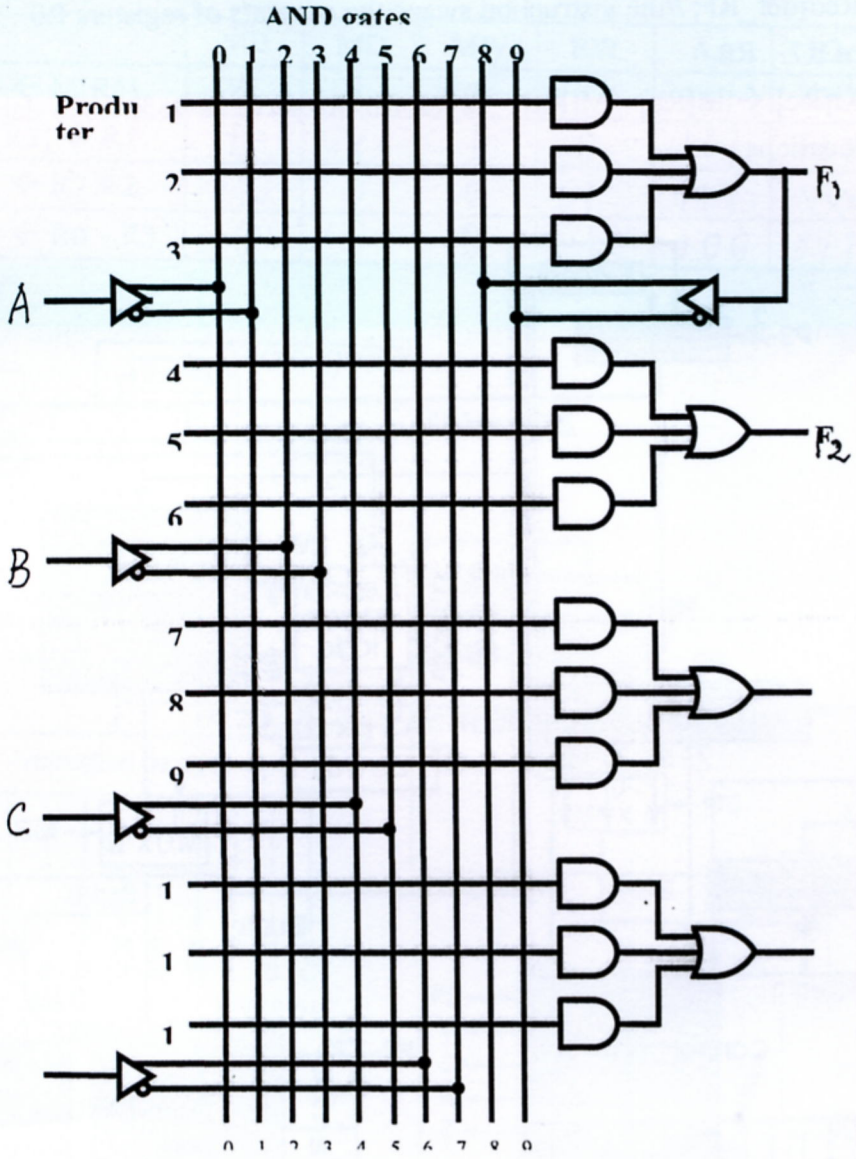
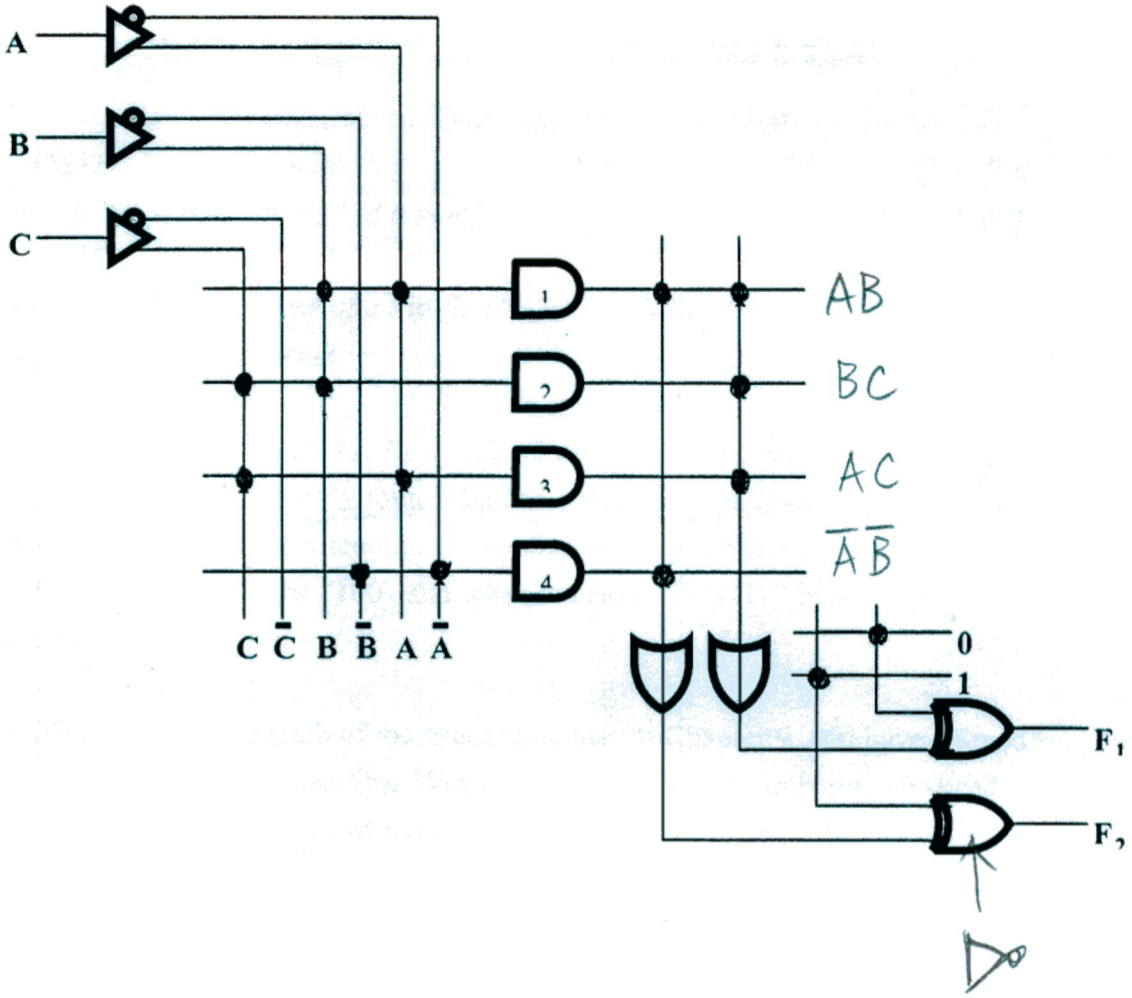


Figure 2 A multi-cycle CPU

7. (15%) Program the following Programmable Array Logic (PAL) such that it performs the same function as that of the Progammable Logic Array (PLA) in the next page.




A
B
C
1
2
3
4
AB
BC
AC
$\overline{A}\,\overline{B}$
C $\overline{C}$ B $\overline{B}$ A $\overline{A}$
0
1
F_1
F_2