**计算机系统概论期中考试**

**PART A: Short answers (28 points)**

**A1 (3 points):** Add the two hexadecimal 2’s complement integers below:

x90A

+ x2018

**A2 (3 points):** The IR (Instruction Register) holds the instruction that is to be executed. Given that the instruction bits can be hold in the MDR, why is IR necessary?

**A3 (3 points):** Perform the following logical representations. Express your answers in hexadecimal notation.

x1958 AND (x1958 OR x2015) AND (x2018 OR (x1958 AND x2018))

**A4 (3 points):** About IEEE.

**A5 (4 points)**: Given two 6-bit inputs A and B, where A=x30 and B=x12. If C is a 6-bit output from an adder, where C=A+B, answer the following questions:

1. (2 points) Assuming that A, B, and C are treated as unsigned numbers, is C a valid result? If not, explain why not. If yes, write the values of C in decimal.
2. (2 points) Assuming that A, B, and C are treated as 2’s complement numbers, is C a valid result? If not, explain why not. If yes, write the value of C in decimal.

**A6 (4 points)**: Is the pair of functions f1, f2 together logically complete? Prove that your answer is correct. Hint: prove that the two functions can construct the AND, OR and NOT functions.

A B f1 f2

0 0 1 0

0 1 1 0

1 0 0 1

1 1 0 0

**A7 (4 points)**: Consider the following LC-3 instruction (x3500 is the address at which the instruction is located):

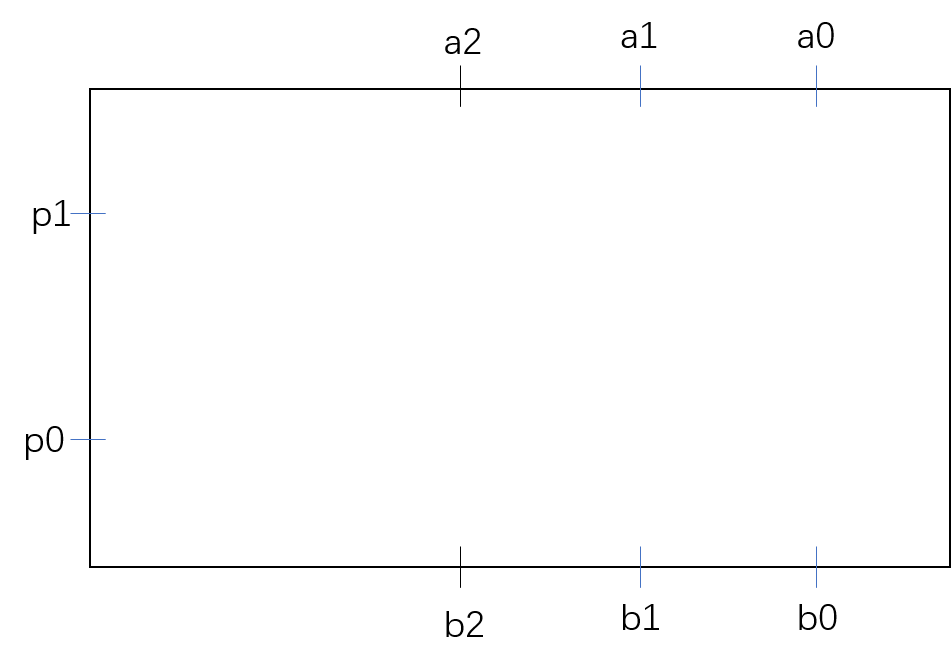
X3500 LD R5, ; we want to put the value x2333 in register R5.

Given the above instruction, what is the range of memory addresses at which the value x2BFF can be stored such that the above instruction can be executed successfully?

**A8 (4 points)**: The current LC-3 memory contains a total of 1 Megabit. If we were to extend it to be byte-addressable as well as DWORD-addressable (a DWORD is 32 bits) without changing its size, and without changing the size of the instructions (16 bits), how many bits would be necessary in the following registers?

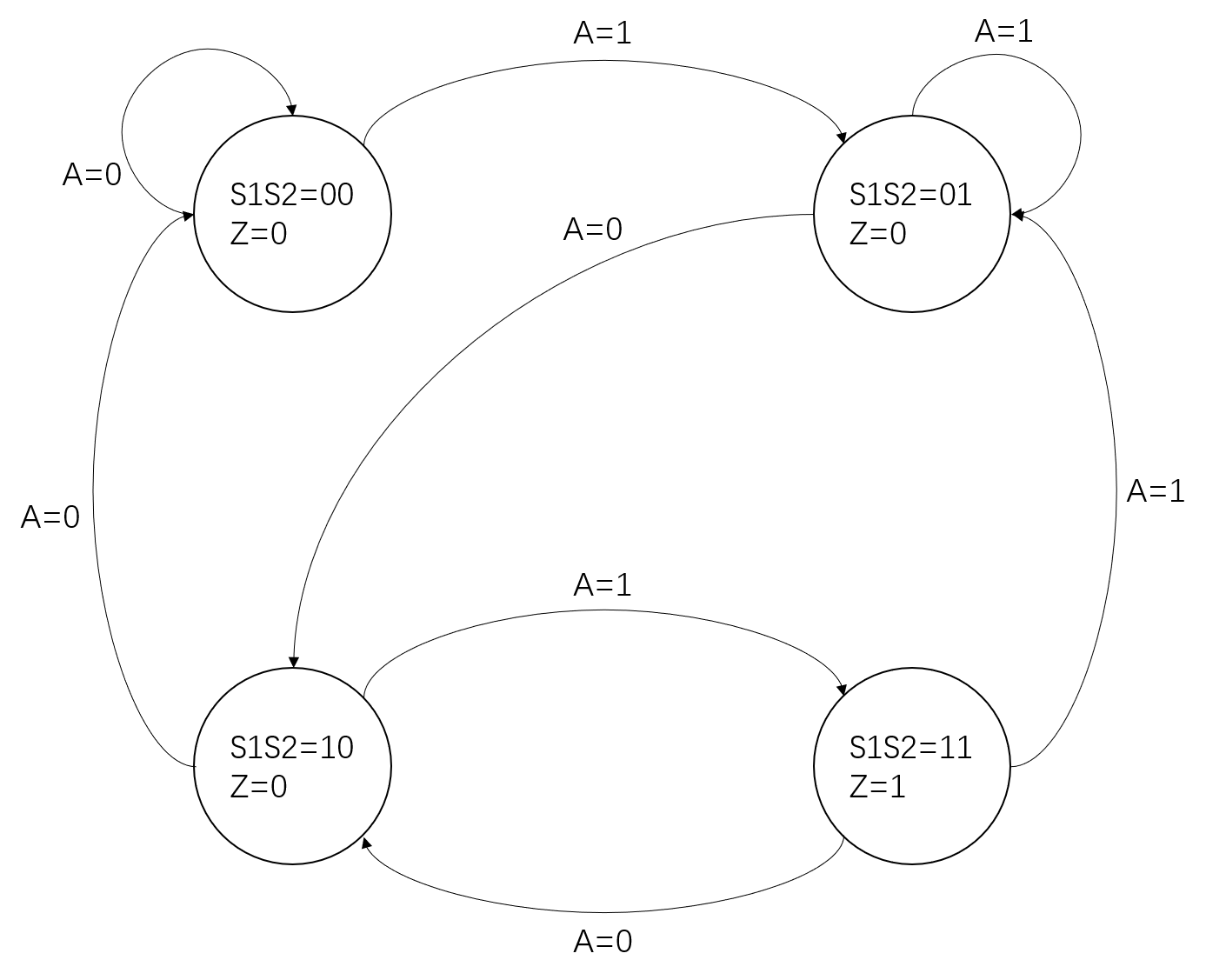
**PART B: Digital Logic Structure (35 points)**

**B1 (7 points)**: Using only one decoder, three 2-input AND and three NOT gates, implement a circuit that takes 3-bit a2a1a0 and two position bits p1p0 as input, a 3-bit b2b1b0 as output. The circuit needs to reset (sets to 0) the value of the bit at the position p1p0 in input a2a1a0. E.g., for input a2a1a0=111 and p1p0=01, output b2b1b0=101. For p1p0=11, no bits should be cleared.



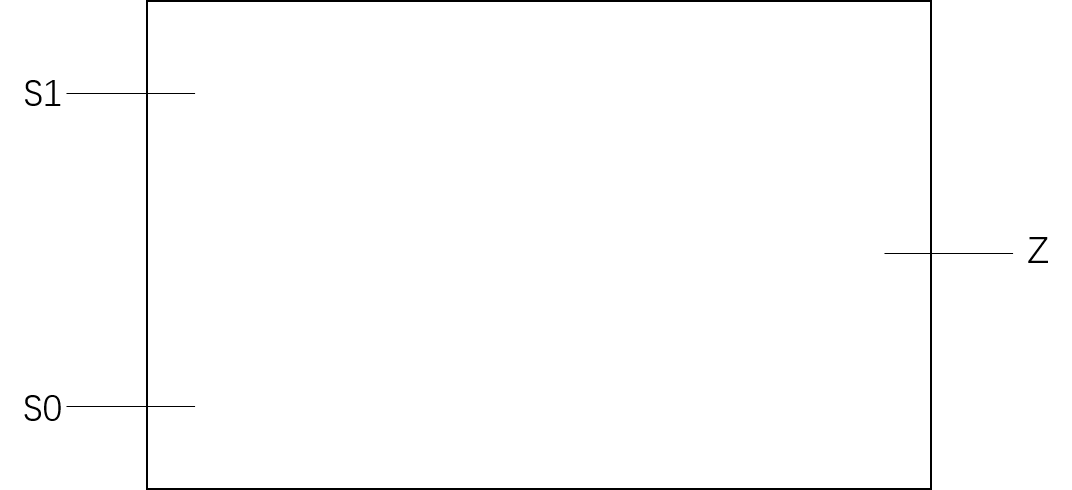
**B2 (10 points)**

1. (5 points) the input “0101110101” is fed into the FSM shown below serially, one bit at a time, starting with the most left bit. If the machine is initially in state S1S0=00, what series of state transitions does the state machine go through upon receiving the input? Note: in the table below, at time n, the state (S1S0) represents the transformation state after input at time n-1.

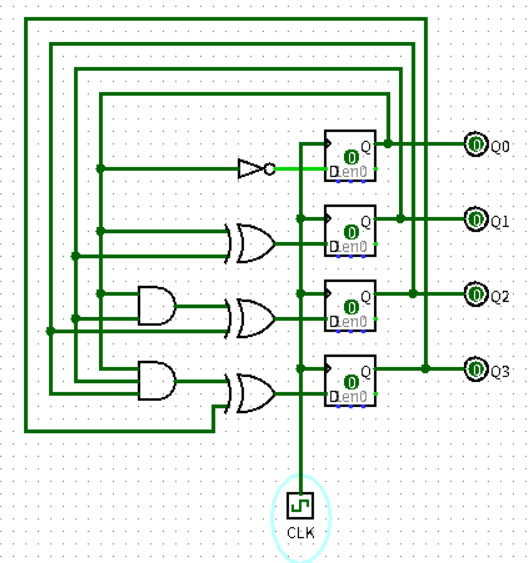


|  |  |  |
| --- | --- | --- |
| Time | State (S1S0) | Input (A) |
| 0 | 00 | 0 |
| 1 |  | 1 |
| 2 |  | 0 |
| 3 |  | 1 |
| 4 |  | 1 |
| 5 |  | 1 |
| 6 |  | 0 |
| 7 |  | 1 |
| 8 |  | 0 |
| 9 |  | 1 |

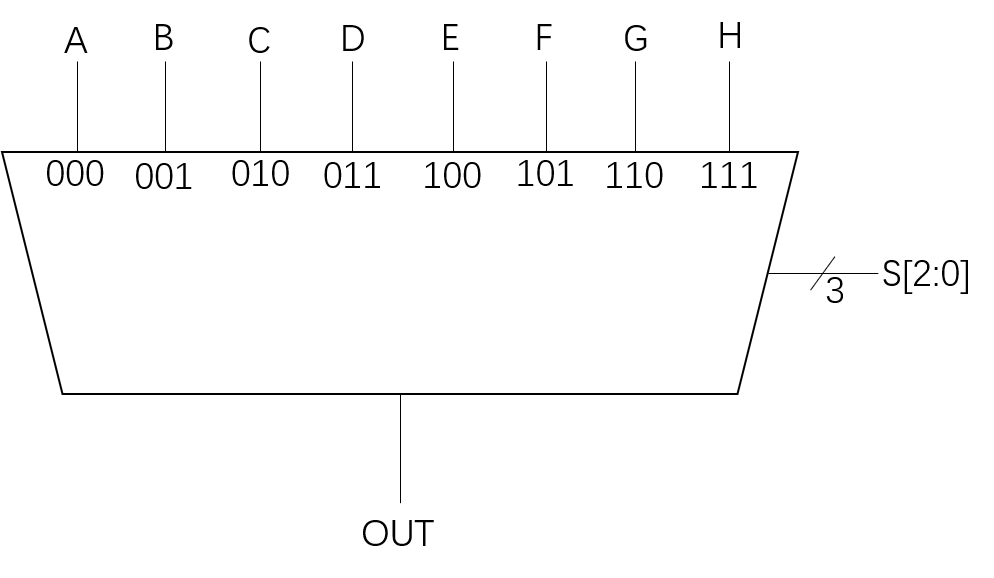
1. (5 points) The FSM from part (i) represents a sequence recognizer which outputs 1(Z=1) when it decode the sequence 101. The output Z is a function of the current state (S1S0) only. Draw the output logic for this sequence recognizer. You can use no more than 3 gates.



**B3 (8 points)** Describe what the following sequential circuit does in a sentence. Assume that the output Q3Q2Q1Q0 is initially 0000.



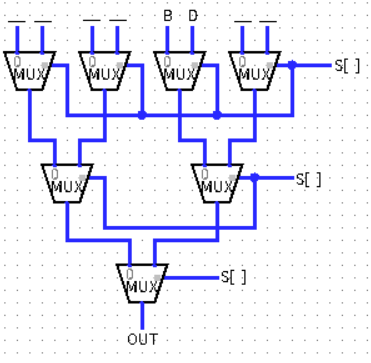
**B4 (10 points)** An 8-to-1 mux (shown below) outputs one of the eight sources, A, B, C, D, E, F, G, H, depending on S[2:0], as shown. Note the value of S[2:0] corresponding to each source is shown just below the input to the mux. For example, when S[2:0]=001, B is provided to the output.



We can implement an 8-to-1 mux with a logic circuit of 2-to-1 muxes, as shown below. In this case, the 0 and 1 below the two inputs to each mux correspond to the value of the select line that will cause that input to be provided to the output of that MUX.

Note that only two of the sources are shown. Note also that none of the select bits are labeled.

**Your task: complete the 8-to-1 mux.**



**PART C: The Von Neumann Model (20 points)**

For the following question, refer to the LC-3 Von Neumann Model below. The current PC and register file is provided, and the LC-3 computer is about to perform a FETCH.

**C1 (4 points)** The data in memory locations x3080, x3081 and x3082 are indicated in hexadecimal on the diagram. For each data word shown in memory, translate them into LC-3 assembly language.

|  |  |
| --- | --- |
| Address | LC-3 |
| x3080 |  |
| x3081 |  |
| x3082 |  |

**C2 (16 points)** In the diagram below, the LC-3 is about to start the instruction cycle for the instruction at x3080. Update the diagram to reflect the state of the machine after the execution of three full instruction cycles (before fetch the forth instruction). Clearly write the final values in the following components within the diagram below.

i) (4 points) Register File

ii) (4 points) IR and PC

iii) (4 points) MAR and MDR

iv) (4 points) Memory

x3080

x3081

x3082

x3083

x3084

x3085

x7680

x5705

x1441

xD249

x0002

x5341

x4251

xFFF1

x098B

x1842

x008E

Register File

R0

R1

R2

R3

R4

R5

R6

R7

IR

PC

MDR

MAR

Memory

x3080

**PART D: LC-3 Programming (16 points)**

**D1 (10 points)** The code below is an incomplete program to multiply two positive integers in R1 and R2 respectively. The result is written into R0. This code is missing the very first instruction in x3000 and part of the instruction in x3004. Provide these missing parts.

|  |  |  |
| --- | --- | --- |
| x3000 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_ |
| x3001 | 0001 0010 0111 1111 | ADD R1,R1,#-1 |
| x3002 | 0000 1000 0000 0010 | BRn x3005 |
| x3003 | 0001 0000 1000 0000 | ADD R0,R2,R0 |
| x3004 | 0000 \_\_\_\_ \_\_\_1 1111 | BR\_\_\_ x3001 |
| x3005 | 1111 0000 0010 0101 | HALT |

**D2 (6 points)** Assuming the code above is completed to correctly calculate R1\*R2, what function would be calculated if the instruction in x3003 were replaced with the following. Phrase your answer in terms of R1 and R2.

1. ADD R0,R0,#-1
2. ADD R0,R0,R1