

CS 4341. Fall 2008 Section 501. Midterm 2 Solutions

Question 1; 25 points

Design a sequential machine with one input variable x and two state variables y_1 and y_2 . and one output variable z . If $x = 0$, the count sequence is 00, 01, 11, 10. If $x = 1$, the count sequence is 00, 10, 01, 11.

1. Develop the state table In your table, the meaning of each column must be clear (10 points).
2. Develop the Boolean expressions needed for the logic circuit. You do not need to minimize any function (8 points).
3. Draw the logic diagram (7 points).

Boolean function
 z has not been
 defined!
 No need to design
 that part

x	$y_1(t)$	$y_2(t)$	$y_1(t+1)$	$y_2(t+1)$
0	0	0	0	1
0	0	1	1	1
0	1	0	0	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	0	1
1	1	1	0	0

$$y_1(t+1) = \bar{x}y_2 + x\bar{y}_1$$

$$y_2(t+1) = \bar{x}\bar{y}_1 + \bar{y}_1y_2 + xy_1\bar{y}_2$$

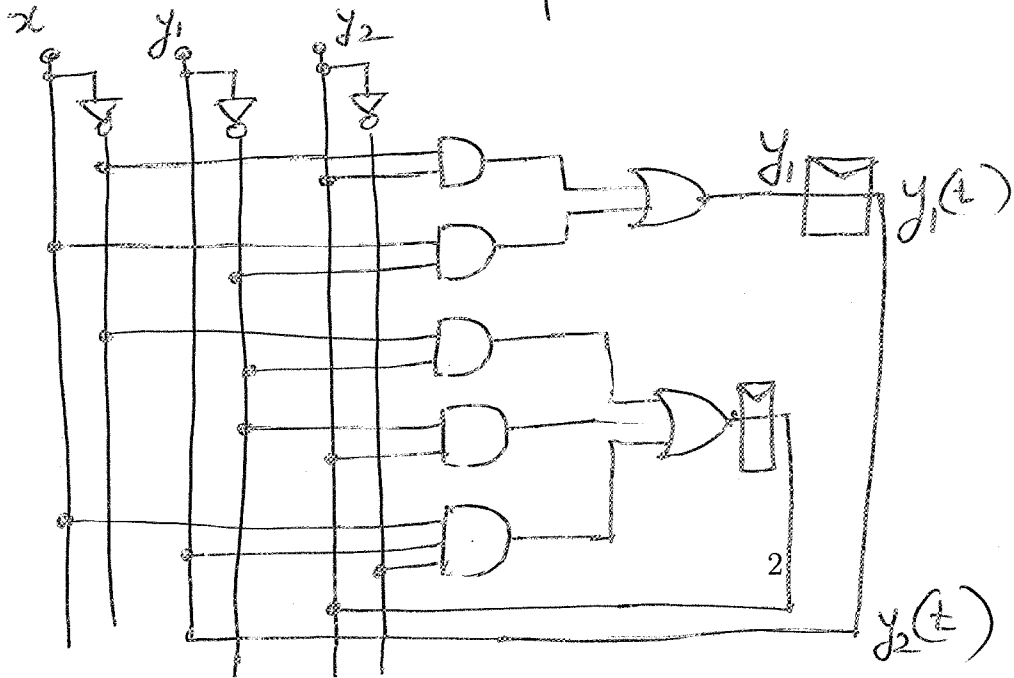


Table 1: ALU specifications

x_1x_0	Function	g_i	h_i	C_0
00	A	a_i	0	0
01	$-B$	0	$\overline{b_i}$	1
10	$2 * A$	a_i	a_i	0
11	$A + B$	a_i	b_i	0

Question 2; 15 points

In a small ALU, the data operands are A and B with bits a_i and b_i respectively. The arithmetic unit portion needs the following functionality, controlled by x_1x_0 . The bits of the operands to be fed to the RCA (ripple carry adder) are g_i and h_i . Clearly, these are functions of x_1, x_0, a_i, b_i . In addition, the overall carry input into the LSB position of the RCA is C_0 . Fill in the adjoining Table 1, the required values or functions of the data inputs.

Question 3; 15 Points

A logic unit takes in two operands $A = a_n \dots a_0$ and $B = b_n \dots b_0$ and produces output $F = f_n \dots f_0$. The operands can be chosen from any of many registers such as $R1, R2, \dots, Rk$, etc. The output can be loaded back to any of the registers, except $R1$. The only logic operations available in the unit are the bit-by-bit Exclusive-OR and bit-by-bit-AND. The register $R1$ contains constant data; all its bits are always Boolean 1. Write a sequence of microoperations to obtain the bit by bit OR operation of the contents in $R2$ and $R3$ and load the result into $R4$.

In order for the ~~AND~~^{OR} of A and B to be 1, exactly one of A, B should be 1, or both should be 1. These possibilities are mutually exclusive. Therefore

$$a + b = a \oplus b \oplus (a \cdot b).$$

The ^{better} sequence of microoperations:

~~$R5 \leftarrow R2 \oplus R3$~~
 ~~$R5 \leftarrow R2 \cdot R3$~~
 ~~$R5 \leftarrow R5 \oplus R2$~~
 ~~$R4 \leftarrow R5 \oplus R3$~~

$$\begin{aligned} R4 &\leftarrow R2 \cdot R3 \\ R4 &\leftarrow R4 \oplus R2 \\ R4 &\leftarrow R4 \oplus R3 \end{aligned}$$

Note on answers by several students:

Getting operands into the inputs at the ALU is not a microop. Inputting to ALU from Registers, ALU function & storing output of ALU in a register constitute ONE single microoperation.

Question 4 ; 10 Points

A RAM system has 1G words of memory, including 128M words of I/O ports. Note that $1K = 2^{10} = 1024$, $1M = (1K)^2$ and $1G = (1K)^3$. Each word has 32 bits. Both RAM chips and I/O port chips are available in a standard size of 128M bytes (that is, 128M rows, each row being a byte of 8 bits). How many RAM chips and how many I/O port chips are required to put together the required RAM system? What is the size of the decoder used to interconnect the chips?

$$\text{Total number of rows of RAM \& I/O chips} = \frac{1G}{128M} = \frac{1M \times 1024}{128M} = 8$$

$$\text{Number of rows of I/O chips} = 1$$

$$\text{Number of I/O chips} = \frac{32 \text{ bits}}{8 \text{ bits}} = \underline{\underline{4}}$$

$$\text{Number of rows of RAM chips} = 8 - 1 = 7$$

@

$$\text{Number of RAM chips} = 7 \times \frac{32}{8} = \underline{\underline{28}}$$

To decode between 8 rows of chips, we need a 3 by 8 decoder.

Question 5; 10 Points

Can we have a single cycle MIPS machine with a single RAM unit that contains both instructions and data? Briefly justify your answer.

Answer: NO!

The single cycle MIPS machine can have only one memory access ~~to~~ in each RAM. ~~Since~~ We need the PC to access memory for the instruction. We also may need to access the memory for data, ~~The~~ during the execution of the instruction.

The only way we can accomplish both is to have two RAM units, one for instructions and one for data.