

You have fifty minutes to complete this test.

Problem 1: Consider the following assembly-language program.

Starting values: [A] = \$02, [\$C300] = \$01, [\$C301] = \$03
 (Hint: [\$C300] means “contents of memory location \$C300”.)

		A	C300	C301
ADDA	\$C300	3		
SUBA	\$C301	0		
BEQ	3			
STAA	\$C301			
WAI				

What values are in A, memory location \$C300, and memory location \$C301 when the program finishes?

[A] = 0 [\$C300] = 01 [\$C301] = 03 *5 each*

Problem 2: Convert the program above into machine language. You may write the bytes of machine language next to each instruction.

ADDA	\$C300	BB	C3	00	<i>-2 per byte max 20</i>
SUBA	\$C301	B0	C3	01	
BEQ	3	27	03		
STAA	\$C301	B7	C3	01	
WAI		3E			

Problem 3: Answer the following questions about the 68HC11. *5 each*

a. What is the purpose of the 8-bit latch on the low 8 bits of the address bus of the 68HC11?

To multiplex address + data on Port C

b. Which 68HC11 output controls the load-enable of the 8-bit address latch?

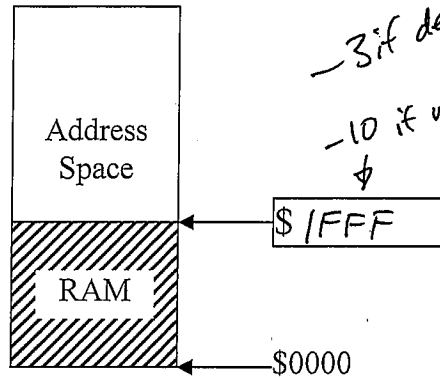
A5

c. How many unique memory locations can the 68HC11 address?

16 bits = 65,536

Problem 4: Create a memory map for 8K of RAM (8 kilobytes or 8,192 bytes) starting at address 0000. What is the last address of the RAM, in hex?

$8K = \$2000$

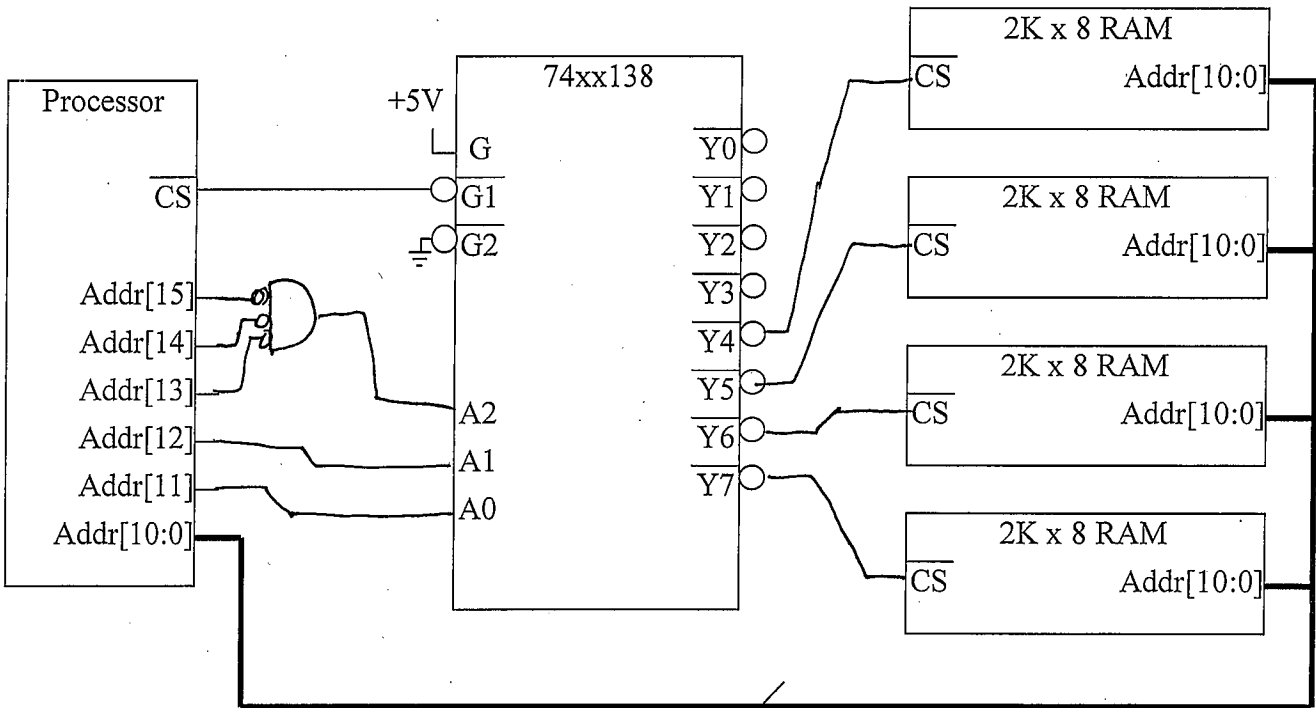


*-3 if decimal
-10 if wrong*

Problem 5: Create an address decoder arrangement for the following:

- 8K of RAM starting at address \$0000 to be built from four 2K x 8 RAM chips.
- The processor and the RAM chips have an active-low chip select.
- Part of the decoder and address bus have already been drawn.
- Do not worry about the data bus, OE, or WE signals (only CS and Address bits)
- The solution should not have any shadowing (should be fully decoded).

*2 pts per mistake
max 2.0*



11

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1