

You have fifty minutes to complete this test.

Problem 1: Design an address space for a 68HC11 to match the following constraints:

- 8 kBytes of RAM starting at address \$0000
- 16 kBytes of ROM ending at address \$FFFF

a. What are the beginning and ending addresses of RAM and ROM?

$4k = \$1000$ $8k = \$2000$ $16k = \$4000$

RAM: \$0000 - \$1FFF

ROM: \$C000 - \$FFFF

b. Procurement got a really good deal on 4 kByte RAM chips. How many will be needed in the design?

$8k / 4k = 2$ RAM chips

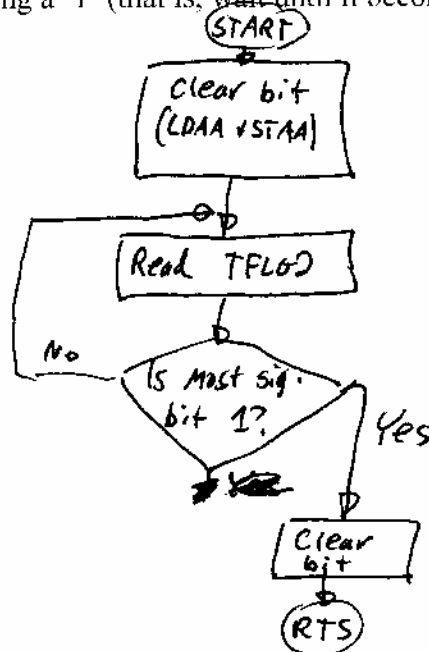
c. In the 68HC11, why is ROM typically located at the upper end of the address space (ending at \$FFFF) and RAM at the low end (starting at \$0000)?

ROM: So interrupt vectors (which go up to \$FFFF) can be in ROM

RAM: To take advantage of special addressing mode in \$0000 - \$00FF range.

Problem 2: Construct a flowchart to implement the following algorithm.

The TOF flag is bit 7 (the most significant bit) of the TFLG2 register (address \$1025). You clear the bit by writing a 1 to it. (For example, LDAA #\$80 and then STAA \$1025.) The algorithm should clear the flag, poll for the TOF flag becoming a '1' (that is, wait until it becomes 1), clear the flag again, and then return.



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

4096 = \$1000

Problem 3: Consider the following I/O design.

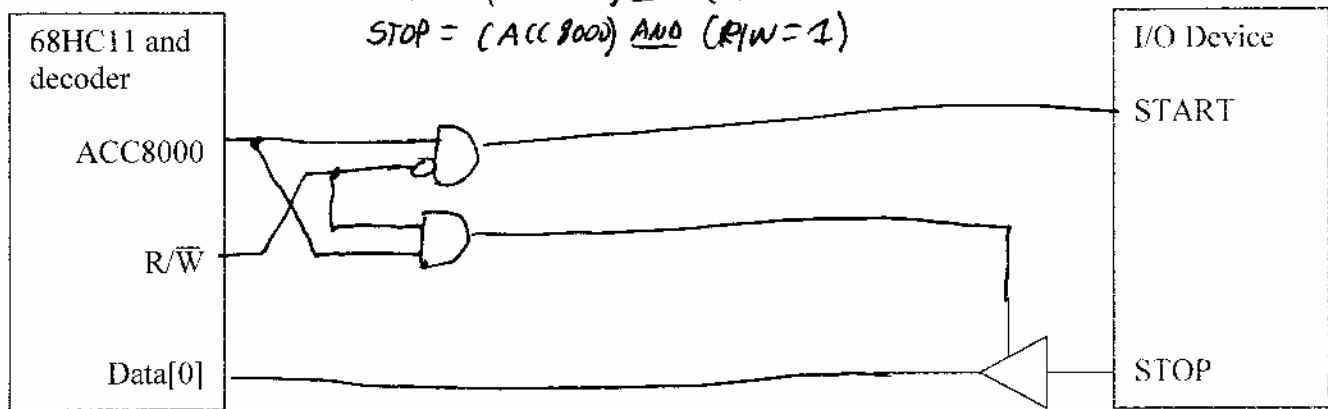
First, the I/O device has a START bit and a STOP bit. When the START bit is asserted (set to 1), the device starts some process. When the process is finished, the device makes its STOP output go high.

Second, a decoder has already been wired to the 68HC11 to produce an output that is *high* when address \$8000 is being accessed (read or written). This signal is called "ACC8000".

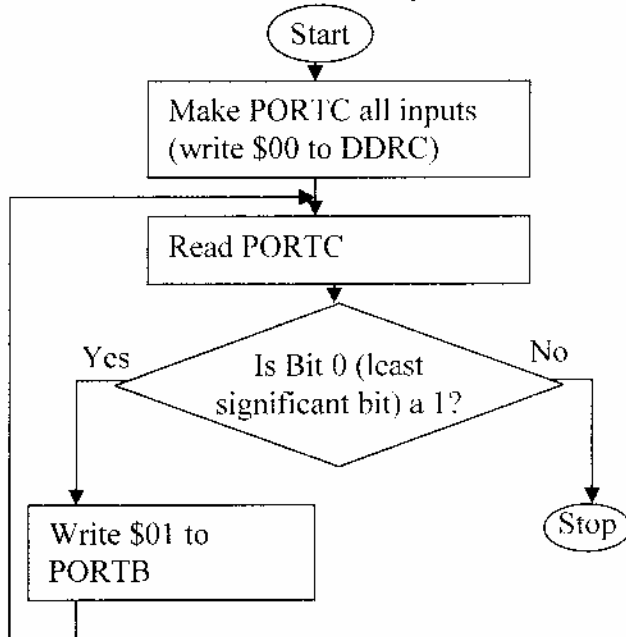
Show how to connect ACC8000, $\overline{R/\overline{W}}$, and data bus bit 0 to the START and STOP bit of the I/O device. That is, the START bit should go high when \$8000 is being written and the tristate buffer on the STOP bit should be enabled when \$8000 is being read. You may add whatever additional logic you need. Note that a tristate buffer has already been added. The tristate buffer is enabled when its enable line is high.

$$\text{START} = (\text{ACC8000}) \text{ AND } (\overline{R/\overline{W}} = 0)$$

$$\text{STOP} = (\text{ACC8000}) \text{ AND } (\overline{R/\overline{W}} = 1)$$



Problem 4: Write the code to implement the following flowchart.



```

CLR $1007 ; write 00 to DDRC
DO-AGAIN LDA $1003 ; read Port c
BMI ENDIT ; branches if MSB=1
LDA $01
STAA $1004 ; write $01 to PORTB
BRA DO-AGAIN ; go back
ENDIT WAI ; get here? time to stop
    
```

Note:

PORTB: \$1004
 PORTC: \$1003
 DDRC: \$1007

You have fifty minutes to complete this test.

25 Problem 1: Design an address space for a 68HC11 to match the following constraints:

16 kBytes of RAM starting at address \$0000
 8 kBytes of ROM ending at address \$FFFF

4K = \$1000

10 a. What are the beginning and ending addresses of RAM and ROM?

16k RAM = \$4000

8k ROM = \$2000

RAM 0000 - 3FFF
 ROM E000 - FFFF

off-by-1! -1

2 pts ea.

10 b. Procurement got a really good deal on 4 kByte RAM chips. How many will be needed in the design?

16 k bytes / 4 k bytes = 4 chips

Did 'RAM + ROM' - 4

10 c. In the 68HC11, why is ROM typically located at the upper end of the address space (ending at \$FFFF) and RAM at the low end (starting at \$0000)?

ROM: interrupt vectors

RAM: ~~is~~ direct addressing

-3 ea.

25 Problem 2: Consider the following 68HC11 timer design.

An input capture pin has been wired to the shaft of a motor. The registers have been programmed to create an interrupt every time the shaft makes one revolution. Consider the chart on page 359 and assume that the 68HC11 has a 2^{23} Hz crystal.

8 a. What is the smallest time interval that can be measured?

477 ns

8 b. What is the maximum timer-overflow interval?

0.5 s

8 c. If the timer overflows twice while the shaft makes one revolution, the measurement will not be accurate. So the shaft must make at least one revolution during the timer-overflow interval. Assuming the timer has been set up for the maximum timer overflow interval, what is the *minimum* RPM of the motor? Stated differently, what is the RPM of the motor if it makes one revolution per timer-overflow interval?

1 Rev / 0.5 s * 60 / 1 min = ~~300~~ 120 RPM

Math err: -4
 Gibberish: -5

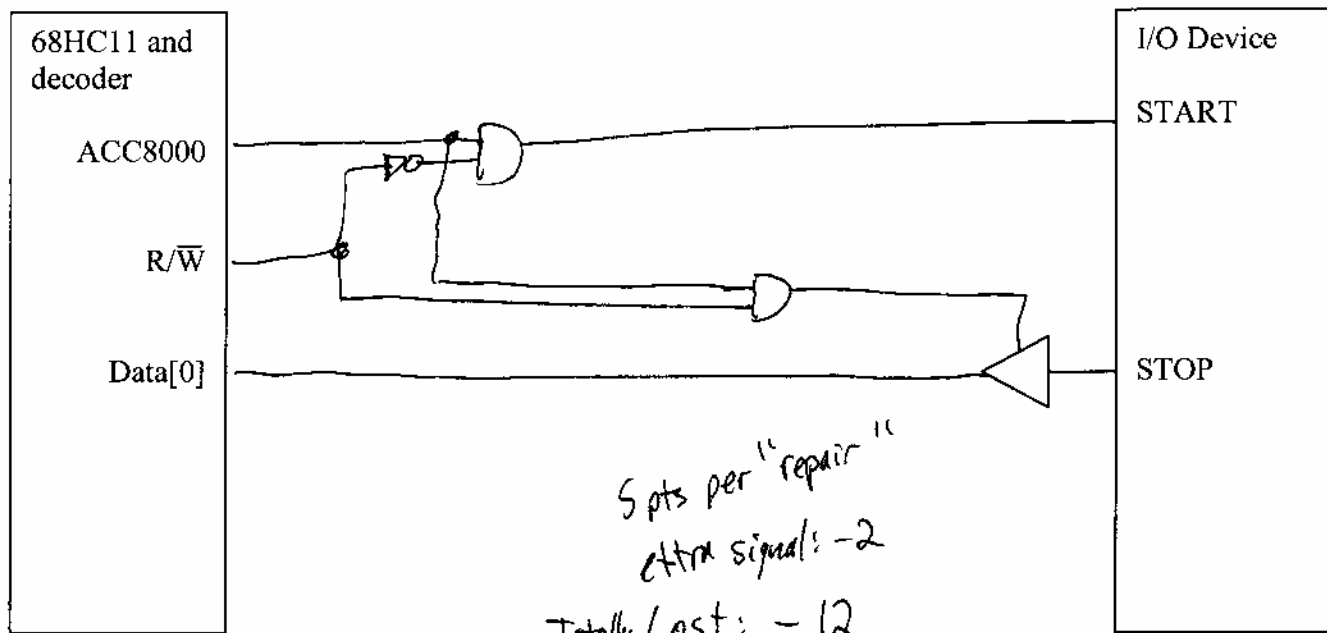
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

25 Problem 3: Consider the following I/O design.

First, the I/O device has a START bit and a STOP bit. When the START bit is asserted (set to 1), the device starts some process. When the process is finished, the device makes the STOP output go high.

Second, a decoder has already been wired to the 68HC11 to produce an output that is *high* when address \$8000 is being accessed (read or written). This signal is called "ACC8000".

Show how to connect ACC8000, $\overline{R/W}$, and data bus bit 0 to the START and STOP bit of the I/O device. That is, the START bit should go high when \$8000 is being written and the tristate buffer on the STOP bit should be enabled when \$8000 is being read. You may add whatever additional logic you need. Note that a tristate buffer has already been added. The buffer is enabled when its enable line is high.



25 Problem 4: The previous engineer set up a polled system for an I/O device, and you want to switch it over to interrupt-driven.

15 a. What steps will you need to take to process the interrupts from the I/O device? (Hint: What do you need to do to set up interrupts and get them to run correctly?)

- Write an ISR
 - Set up interrupt vector
 - Turn on maskable intr's
 - Turn on intr. source
 - Initialize the stack
- only one entry for these two: -2

10 b. What are the advantages of interrupt-driven I/O over polled I/O?

- Less CPU load