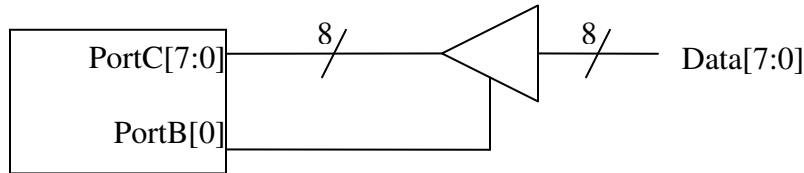


This test is open-book, open-notes, but during the test no sharing of books or notes is allowed. You have 60 minutes to finish this test.

An 8-bit tri-state buffer is wired up so that the 8-bit data bus is wired to Port C and the output enable of the buffer is wired to Port B bit 0. (The other 7 bits of Port B are “don’t care” and can be set to any value.)



When an IRQ interrupt occurs, the 68HC11 should do the following:

- Enable the tristate buffer by setting Port B bit 0 to 1.
- Read the tristate buffer by reading Port C.
- Write the contents of Port C to memory location \$0047.
- Disable the tristate buffer by setting Port B bit 0 to 0.

Write a complete Buffalo initialization routine and interrupt handler. In the initialization, consider how to set up the interrupt vector for the IRQ interrupt, how to enable maskable interrupts, how to make sure Port C is an input, and how to disable the tristate buffer. In the interrupt handler, consider how to enable the buffer and read the data.

Part of the program has already been started for you. The back of the page has the Buffalo jump-table locations.

```

PORTB EQU    $1004
PORTC EQU    $1003
DDRC  EQU    $1007

                ORG    $00EE
                JMP    IRQH            ; populates IRQ jump table entry

                ORG    $0100
MAIN          CLR    DDRC            ; Port C is all input
                CLR    PORTB         ; Make tristate disabled
                CLI    ; enable IRQ interrupt
X1           BRA    X1            ; infinite loop - nothing else to do here

IRQH         LDAA   #$01
                STAA  PORTB         ; enables tristate
                LDAA  PORTC         ; read data
                STAA  $0047         ; stores data
                CLR   PORTB         ; disables tristate
                RTI    ; done - go back

```

Table 3-1. Interrupt Vector Jump Table

INTERRUPT VECTOR	FIELD
Serial Communications Interface (SCI)	\$00C4-\$00C6
Serial Peripheral Interface (SPI)	\$00C7-\$00C9
Pulse Accumulator Input Edge	\$00CA-\$00CC
Pulse Accumulator Overflow	\$00CD-\$00CF
Timer Overflow	\$00D0-\$00D2
Timer Output Compare 5	\$00D3-\$00D5
Timer Output Compare 4	\$00D6-\$00D8
Timer Output Compare 3	\$00D9-\$00DB
Timer Output Compare 2	\$00DC-\$00DE
Timer Output Compare 1	\$00DF-\$00E1
Timer Input Capture 3	\$00E2-\$00E4
Timer Input Capture 2	\$00E5-\$00E7
Timer Input Capture 1	\$00E8-\$00EA
Real Time Interrupt	\$00EB-\$00ED
IRQ	\$00EE-\$00F0
XIRQ	\$00F1-\$00F3
Software Interrupt (SWI)	\$00F4-\$00F6
Illegal Opcode	\$00F7-\$00F9
Computer Operating Properly (COP)	\$00FA-\$00FC
Clock Monitor	\$00FD-\$00FF