READ AND FOLLOW THESE INSTRUCTIONS.

- Do not begin until you are told to do so.
- You have 50 minutes; budget your time. The questions are not of equal weight; do not spend too much time on a question that is not worth many points.
- Read through all of the questions before starting to work.
- This exam is **closed notes**. You may use the MPC823 data book, the "PowerPC Programming Pocket Book", and/or a printout of Appendix F from the green book as reference material. You may *not* share reference materials with other students.

Name: _____

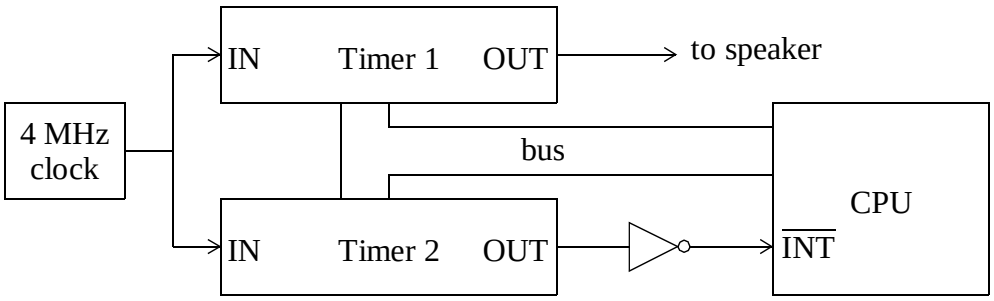
Uniqname: _____

I have neither given nor received aid on this exam, nor have I concealed any violation of the Honor Code.

Signature: _____

Question	Points	Score
1	25	
2	45	
3	30	
Total	100	

1. (25 pts.) This problem involves a very simple sound subsystem for a microprocessor-based system. The output of a 16-bit timer is connected to a speaker; thus, square waves of various frequencies can be sent to the speaker to generate tones. A second 16-bit timer generates an interrupt when it is time to change the frequency (i.e., it controls the tone's duration). (This is basically the setup that drives the built-in speaker on a PC.) Both timers are driven by a 4 MHz clock. The following diagram illustrates the hardware configuration:



Other than their connections, the two timers are identical. Each interfaces to software via three 16-bit registers: a count value register, a limit (reference) register, and a control/status register. The control/status register has the following layout:

Control Register Layout:

0 ... 4	5	6	7	8 ... 15
0 0 0 0 0	EN	OUT	TOG	PRE

- The EN bit must be set to 1 to enable the counter.
- The OUT bit of the control register is connected to the unit's OUT pin: you can read this bit to check the current state of the pin and you can write to it to change the output state.
- The TOG bit determines the behavior of the output pin. If TOG=1, the state of the OUT pin is complemented (toggled) each time the count value reaches the max count. If TOG=0, the OUT pin is set to 1 (high) when the count value reaches the max count; in this case, OUT can only be set to 0 by software.
- The PRE field provides the prescaler value (minus one). As with the MPC823 timers, the input clock frequency can be divided by any integer from 1 to 256 by setting this field to the divisor minus one (0-255). A PRE value of zero corresponds to a divisor of one, effectively turning off prescaling.

a. (5 pts.) Derive values for Timer 1's control and limit registers that will produce a 500 Hz tone on the speaker. The square wave driving the speaker must be symmetric (50% duty cycle).

b. (10 pts.) The system must be able to generate tones in the range 10 Hz to 20 kHz. Derive an appropriate control register value for Timer 1 such that tones in this range can be generated by changing only the limit register.

c. (5 pts.) Derive an appropriate control register value for Timer 2 (the timer that generates the interrupt). A limit register value of 1 should generate an interrupt after 25 microseconds.

d. (5 pts.) Given your answer to part (c), what is the longest duration that a single tone can be played before an interrupt occurs?

2. (45 pts.) In this problem, you will write an ISR to drive the sound subsystem of question 1. The tone duration timer is the only interrupting device, and is it connected to the single external interrupt line of the PowerPC CPU (i.e., the SIU and CPM interrupt controllers are not involved).

   The sequence of tones is stored in a memory array. There are two halfwords for each tone, one for the frequency and one for the duration. The values have all been pre-processed so that you can use them directly as the limit register values for Timers 1 and 2 respectively. The symbol TONEPTR is the address of a memory location that contains a pointer to the current location in the array; specifically, the address of the Timer 1 limit value of the *next* tone to be played.

a. (15 pts.) The middle part of the timer ISR is given below. Complete the ISR. Don't worry about running off the end of the array of tones (you can assume it is infinitely long). The timer registers are at the following addresses:

Register	Timer 1	Timer 2
control	0xFF000000	0xFF000010
count	0xFF000004	0xFF000014
limit	0xFF000008	0xFF000018

timer_ISR:

```
lis     r3, TONEPTR@h
ori     r3, r3, TONEPTR@l
lwz     r4, 0(r3)           ; load pointer to next tone info
lis     r5, 0xff000000@h    ; load base address of timer registers
lhz     r6, 0(r4)           ; get next tone period
sth     r6, 8(r5)           ; update Timer 1 limit reg
lhz     r6, 2(r4)           ; get next tone duration
sth     r6, 0x18(r5)        ; update Timer 2 limit reg
addi    r4, r4, 2           ; update array pointer
stw     r4, 0(r3)           ; store back in TONEPTR
```

For parts (b) to (d), assume you add another timer to your system to generate a once-per-second interrupt to track the time of day. This new timer shares the PowerPC external interrupt line with the tone-duration timer. The SIU and CPM controllers are still not used.

b. (5 pts.) When an interrupt occurs, how will your system determine which timer caused it?

c. (5 pts.) What will determine which timer has higher priority?

d. (12 pts.) If the once-per-second interrupt performs a lot of processing, you will want to handle interrupts from the tone-duration timer within the once-per-second service routine. What steps do you need to take to make this work? Be very specific about the operations you need to do and the order in which you need to do them. Don't worry about prioritization: that is, you may assume that the once-per-second interrupt will not occur again during the once-per-second ISR.

e. (8 pts.) List the primary advantage and primary disadvantage of a vectored interrupt scheme (relative to the all the other schemes discussed in class).

3. (30 pts) Consider a six-bit analog-to-digital converter with an input range of 0-10V.

a. (6 pts.) After a conversion, the ADC returns a value of 41. Assuming an ideal converter, what is the possible voltage range of the analog signal?

b. (8 pts.) If your program assumes that the analog input is in the center of this range, what is the range of possible error in volts? In LSBs? What is the name for this error?

c. (6 pts.) The ADC data sheet specifies a worst-case non-linearity of ±1/5 LSB. Now what is the possible voltage range of the analog signal?

d. (5 pts.) Given only the information from part (c), what is the differential non-linearity of the ADC?

e. (5 pts.) If this is a flash-type ADC, how many voltage comparators does it contain?