Exam 2 - 1 EECE 344 Fall 2010

Name: _____

Problem	Possible	Score
1	15	13
2	15	15
3	15	13
4	20	19
5	20	10
6	15	12
Total	100	82

General information that may be useful sometime during test:

Table of Powers of Two

N	2^N		N	2^N		N	2^N		N	2^N
0	1		8	256		16	65,536		24	16,777,216
1	2		9	512		17	131,072		25	33,554,432
2	4		10	1,024		18	262,144		26	67,108,864
3	8		11	2,048		19	524,288		27	134,217,728
4	16		12	4,096		20	1,048,576		28	268,435,456
5	32		13	8,192		21	2,097,152		29	536,870,912
6	64		14	16,384		22	4,194,304		30	1,073,741,824
7	128		15	32,768		23	8,388,608		31	2,147,483,648

Please write legibly!

X.XXXXXXXXXXX

1. Information representation. We gotta have some question about number representation. Consider number system(s) that contain 12 bits, with the radix point just to the right of the MSB. For that arrangement of bits, fill in the missing elements of the following table. (Remember that maximum is right-most on the number line; minimum is left-most on the number line.)

Value	Unsigned binary pattern	Twos-complement pattern
Maximum	1.11111111111 ✓	0.11111111111 ✓
Minimum	0.00000000000 ✓	1.00000000000 ✓
$^{37}/_{64}$	0.10010100000 ✓	0.10010100000 ✓
$^{-23}/_{32}$	N/A	1.01001000000 ✓
$1^{11}/_{16}$	1.10110000000 ✓	N/A
$1^{5}/_{32}$ (NE)	N/A	1.00101000000

$\frac{32}{64} + \frac{4}{64} + \frac{1}{64}$; $\frac{16}{32} + \frac{4}{32} + \frac{2}{32} + \frac{1}{32}$; $\frac{8}{16} + \frac{2}{16} + \frac{1}{16}$; $1 + \frac{1}{8} + \frac{1}{32} = 1\frac{5}{32}$

2. General information question:

a) Where is the return address stored on a go-to-subroutine instruction?

LR ✓

b) Give a sequence of instructions (only 2 needed) that will set up the system to expect the interrupt table to be found at the third legal location for the table. That is, what is the third legal location for the interrupt table, and how do you set it up?

0
1
2

```
lis rx, 2 —
mtevpr rx
```

c) We have programmed the system recognizing that the Interrupt Controller directs the hardware to utilize the instructions found at 0x0500. If we had enough interrupts to utilize the full capabilities of the module, how many different interrupt sources could be handled by the Interrupt Controller? (Hint: it is more than the four modules we have dealt with this semester.)

trivia :-

32 ✓

d) Assume R14 contains the value 0x00100000. Consider the instruction `stw r13,<offset>(r14)`. What is the address of the word which is located as far away as possible – to higher addresses – that can be accessed by this instruction? SIMM → max = 0x7FFF

0x00107FFF (HM) ✓

e) How many different periods are available via the Fixed Interval Timer?

4 ✓ defined by 2 bits

3. Interrupt Controller Question: In the table below – which shows registers in the interrupt controller - identify the registers that need to be initialized, and give values for each. In this interrupt system, there are four interrupt sources, starting in the least significant bit position, and all are to be enabled. Also, the software activation of interrupts is not to be utilized. Assume that you want to assert the appropriate bits to reset any flags that may have remained from an earlier program.

Addr Offset	Register	Bit Pattern
0x00	ISR	0000
0x04	IPR	N/A – writing to ISR takes care of this
0x08	IER	1111
0x0C	IAR	1111
0x1C	MER	11

Now, in the space provided below, give instructions that will establish the bit patterns given above as well as to a)set the EVPR to zero and b)set up any enabling activity needed to allow interrupts to occur. Assume that the interrupt controller has been located at address `0x82340000`.

```
lis r31, 0x8234
li r30, 0
mtevpr r30
stw r30, 0(r31)      NO
li r30, 0b1111
stw r30, 8(r31)
stw r30, 0xC(r31)
li r30, 0b11
stw r30, 0x1C(r31)
wrteei 1
```

4. ISR question: For system of problem three (Interrupt Controller, 4 bits …) create an Interrupt Service Routine for the following situation. Timer Module is hooked to the most significant bit of the four identified in the question. When the timer service is requested, reset the appropriate flags, in the appropriate order, increment the value in R23 and send to the LEDs. The Interrupt controller address is identified in Problem 3. The Timer Module is located at `0x82440000`, and the LED interface GPIO is located at `0x82560000`. Do not worry about register volatility.

ENALL	PWMAO	TOINT	ENTO	ENITO	LOADO	ARHTO	CAPTO	GENTO	UDTO	MOTO
0	0	1	1	1	0	1	0	1	1	0

magic number 0x1D6

```
.org 0x500
lis r31, 0x8234  # INTC
lwz r30, 4(r31)
andi. r30, r30, 0b1000  # poll
beq  nextpoll
li  r30, 0b1000
stw  r30, 0xC(r31) # acknowledge INTC

lis r31, 0x8244  #TMR
li  r30, 0x106
stw r30, 0(r31)  # acknowledge TMR

lis r31, 0x8256 # LEDS
addi r23, r23, 1
stw  r23, 0(r31)

nextpoll:  # other polling here if appropriate
...
rfi
```

5. Data structure question. The recursive definition of the Fibonacci sequence is: F(n) = F(n-1) + F(n-2), with F(0) = 0 and F(1) = 1. Give code for a subroutine that implements this sequence. Use a stack mechanism to implement the recursion.

in gas, "sp" is aliased to "r1"

He graded this wrong, the code is correct

```
fib:  # returns r30th fib. num. in r28
      mflr  r31
      subi  sp, sp, 4
      stw   r31, 0(sp)
      subi  sp, sp, 12
      stmw  r29, 0(sp)

      li    r28, 0
      cmpw  r28, r30
      beq   endfib
      li    r28, 1
      cmpw  r28, r30
      beq   endfib

      subi  r30, r30, 1
      bl    fib
      mr    r29, r28
      subi  r30, r30, 1
      bl    fib
      add   r28, r28, r29

endfib:
      lmw   r29, 0(sp)
      addi  sp, sp, 12

      lwz   r31, 0(sp)
      addi  sp, sp, 4
      mtlr  r31

      blr
```

why 12 steps?

does this work?

F(3) = 2

r30 ← 3

r30 ← 2 ?

See Edy set

```
push
r30 ← 1
push
r28 ← 1
pop (r30←1)

r29 ← 1
r30 ← 0
push
r28 ← 0
pop (r30←0, r29←1)
r28 ← 0+1
pop (r30←2)

r29 ← 1
r30 ← 1
push
r28 ← 1
pop (r30←1, r29←1)
r28 ← 1+1
```

(2)

where is recursion?

6. Instruction coding question:

a) What instruction is represented in hexadecimal as: **0x7FD9E114**?

ØXXX XXYX XIØX XØØX XXYØ ØØ01 0001 0100

left 011111 → 31

11110 → r30
11001 → r25
11100 → r28
0 → no OE
0100 0101 0 → 266 → adder r30, r25, r28

b) Give the coding for the instruction '**andis. r2,r3,0x78AB**'. Remember that on logical instructions, register order is reversed in bit pattern.

| 29 | rS | rA | UIMM |

78AB

r2 → 00010
r3 → 00011
29 → 011101
0111 0100 0100 0011

Ox 7 4 4 3 78AB
6 2

c) Wat instruction, located at address 0x4444, is represented by **0x4811EDCC**?

ØIØØ IØØØ 0001 0001 ...

010010 → 18 → b

last digit C: 1100
no AA, no LK

LI: Ox 1 1 E D C C
+ 4 4 4 4
123210

b Ox 123210

Table B-1 lists the PPC405 instruction set in alphabetical order by mnemonic.

Table B-1: **Instructions Sorted by Mnemonic**

	0	6	9	11	12	14	16	17	20	21	22	26	30	31
add	31	rD		rA			rB			OE	266			Rc
addc	31	rD		rA			rB			OE	10			Rc
adde	31	rD		rA			rB			OE	138			Rc
addi	14	rD		rA			SIMM							
addic	12	rD		rA			SIMM							
addic.	13	rD		rA			SIMM							
addis	15	rD		rA			SIMM							
addme	31	rD		rA			00000			OE	234			Rc
addze	31	rD		rA			00000			OE	202			Rc
and	31	rS		rA			rB				28			Rc
andc	31	rS		rA			rB				60			Rc
andi.	28	rS		rA			UIMM							
andis.	29	rS		rA			UIMM							
b	18	LI											AA	LK
bc	16	BO		BI			BD						AA	LK
bcctr	19	BO		BI			00000			528				LK
bclr	19	BO		BI			00000			16				LK
cmp	31	crfD	00	rA			rB			0				0
cmpi	11	crfD	00	rA			SIMM							
cmpl	31	crfD	00	rA			rB			32				0

Table 4: **XPS INTC Registers and Base Address Offsets**

Register Name	Base Address + Offset (Hex)	Access Type	Abbreviation	Reset Value
Interrupt Status Register	C_BASEADDR + 0x0	Read / Write	ISR	All Zeros
Interrupt Pending Register	C_BASEADDR + 0x4	Read only	IPR	All Zeros
Interrupt Enable Register	C_BASEADDR + 0x8	Read / Write	IER	All Zeros
Interrupt Acknowledge Register	C_BASEADDR + 0xC	Write only	IAR	All Zeros
Set Interrupt Enable Bits	C_BASEADDR + 0x10	Write only	SIE	All Zeros
Clear Interrupt Enable Bits	C_BASEADDR + 0x14	Write only	CIE	All Zeros
Interrupt Vector Register	C_BASEADDR + 0x18	Read only	IVR	All Ones
Master Enable Register	C_BASEADDR + 0x1C	Read / Write	MER	All Zeros

Notes:

Table 4: **XPS Timer/Counter Register Address Map**

Register	Address (Hex)	Size	Type	Description
TCSR0	C_BASEADDR + 0x00	Word	Read/Write	Control/Status Register 0
TLR0	C_BASEADDR + 0x04	Word	Read/Write	Load Register 0
TCR0	C_BASEADDR + 0x08	Word	Read	Timer/Counter Register 0
TCSR1	C_BASEADDR + 0x10	Word	Read/Write	Control/Status Register 1
TLR1	C_BASEADDR + 0x14	Word	Read/Write	Load Register 1
TCR1	C_BASEADDR + 0x18	Word	Read	Timer/Counter Register 1

Control/Status Register 0 (TCSR0)

The Figure 6 and Table 7 shows the Control/Status register 0. Control/Status Register 0 contains the control and status bits for timer module 0.

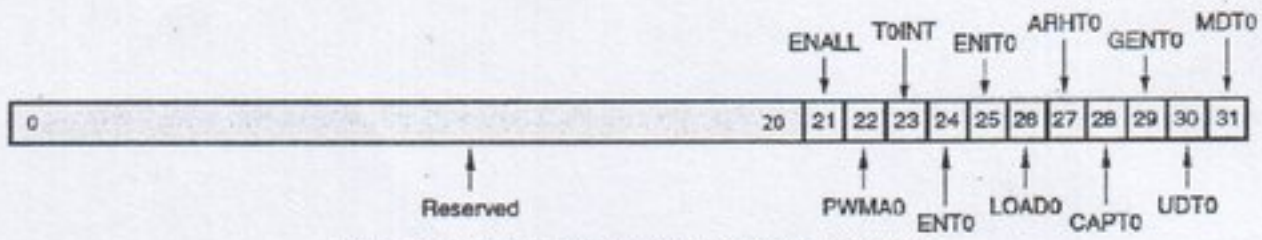

Figure 6: **Timer Control/Status Register 0 (TCSR0)**

ENALL – enable all timers

PWMA0 Enable PWM for timer 0

T0INT – Timer 0 interrupt

ENT0 – Enable timer 0

LOAD0 – Load timer 0

ARHT0 – Auto reload/hold timer 0

CAPT0 – enable capture trigger timer 0

GENT0 – Enable External Generate

UDT0 - Up Down Timer0

MDT0 – Timer 0 mode