University of Michigan **EECS 373**
Fall 2001

EXAM 1

Instructions

1. This is an open books/notes exam.
2. It comprises 25 multiple-choice questions, each worth 4 points.
3. Each question has only one correct answer.
4. For each question, indicate your answer on the answer sheet provided.
5. You have 75 minutes to complete the exam.
6. Please make sure you enter your name in the space below AND on the answer sheet.
7. By signing below, you certify that your conduct throughout the exam has been in accordance with the College of Engineering Honor Code.
8. At the end of the exam, please turn in this exam booklet AND your answer sheet.

Name **SOLUTION**

Signature _____

ISAs, alignment, and endian modes [20 points]

1. Consider the instruction `addi r2, r3, n` , where *n* is an immediate. What is the maximum value *n* can take?

a. ***$2^{15} - 1$***
b. $2^{16} - 1$
c. 2^{15}
d. 2^{16}
e. None of the above

2. Assuming a single-cycle memory, how many cycles will the bus transaction corresponding to the instruction `lhz r2, 3(r4)` take?

a. 2
b. 4
c. 6
d. Depends on the value in the register `r4`.
e. Depends on the value stored at the effective address.

3. Which of the following instructions affect(s) general-purpose registers other than `r3`?

I. `lwzu r3, 4(r4)`
II. `and. r3, r2, r3`
III. `cmpwi r3, 100`
IV. `bdnzt eq, target`

a. I
b. I and II
c. III and IV
d. IV
e. None

4. Which of the following transaction scenarios give(s) rise to incorrect/unpredictable bus operation?

I. One master, one slave
II. One master, several slaves
III. Several masters, one slave
IV. Several masters, several slaves

a. I
b. II
c. I and III
d. III and IV
e. II, III, and IV

5. Which of the following statements are always true?

I. Big-endian machines can support aligned accesses only with a 32-bit data bus.
II. Little-endian machines can support aligned accesses even with an 8-bit data bus.
III. It is possible for a big-endian machine to support unaligned accesses.
IV. It is possible for a little-endian machine to support unaligned accesses.

a. I and II
b. II
c. II and III
d. III and IV
e. II, III, and IV

Assembly code [28 Points]

Questions 6—9 refer to the following simple program.

```
          .data
          .align 2
          .equ size, 4
          .equ max, 42
stuff:    .word  9, 1, 4, 27, 42, 2, 6, 78, 11, 28, 19, 22
iter:     .skip 1
          .text
          .align 2
_start:   lis  r2, stuff@h
          ori  r2, r2, stuff@l
          li   r3, size
          li   r4, max
loop:     lwz  r5, 0(r2)           # Loop while value read
          cmp  r4, r5              # from memory <= max
          blt  exit
          add  r2, r2, r3
          b    loop
exit:     b    exit
```

6. In memory, the data section is allocated:

   a. before the text section (i.e., at a lower address).
   b. after the text section.
   c. at the same address as the text section.
   d. There is not enough information given to answer this question.

7. The first two instructions of the program have the effect of:

   a. loading the least-significant byte of the array into r2.
   b. loading the least-significant word of the array into r2.
   c. loading the address of the array into r2.
   d. loading the most-significant byte of the array into r2.
   e. loading the most-significant word of the array into r2.

8. Suppose the array `stuff` were changed so that it were declared as follows:

```
stuff:    .byte 9, -1, 4, 27, -42, 2, 6, 78, 11, 28, 19, 22
```

Indicate which of the following changes are absolutely necessary for the program to work correctly:

I. Change the first `.align 2` to `.align 0` so the array elements will be byte-aligned.
II. Change `size` from `4` to `1`.
III. Change `lwz` to `lba` to sign-extend the bytes.
IV. Change `lwz` to `lbz` and add code to sign-extend the bytes.
V. Change `cmp` to `cmpl` so that the program performs a signed comparison.

a. I, II, and IV
b. II and III
c. II and IV
d. II, III, and V
e. II, IV, and V

9. Suppose that the compare instruction were changed to read as follows:

```
cmp  4, r4, r5
```

Indicate which of the following branch instructions could replace the existing `blt` instruction and allow the program to function correctly:

```
I.  bc  12, 4, exit
II. bc  12, 16, exit
III. bc  4, 4, exit
IV. bc  4, 17, exit
```

a. I
b. II
c. III
d. II or IV
e. IV

In Questions 10—12, you will be given the specification of an action and a number of code snippets that presumably perform that action. Indicate which of the code sequences will always perform as specified.

10. Load the immediate value `0xDEADBEEF` into `r4`.

I.
```
addis  r4, r4, 0xDEADBEEF@h
ori    r4, r4, 0xDEADBEEF@l
```

II.
```
lis  r4, 0xDEADBEEF@h
ori  r4, r4, 0xDEADBEEF@l
```

III.
```
lis   r4, 0xDEADBEEF@h
addi  r4, r4, 0xDEADBEEF@l
```

a. I
b. II
c. I and II
d. III
e. All of the above

11. Branch to `label` if bit 27 of `r4` is 1.

I.
```
andi.  r5, r4, 1 << (31 – 27)
bne    label
```

II.
```
li    r5, 0x10
and   r6, r4, r5
cmpi  r6, 1
beq   label
```

III.
```
rlwinm  r6, r4, 28, 31, 31
cmpi    r6, 1
beq     label
```

IV.
```
andi.  r5, r4, 0x10
cmpi   2, r5, 0
bc     4, 10, label
```

a. I and III
b. II and III
c. II and IV
d. I, II, and III
e. I, III, and IV

12. Load an unsigned half-word into `r3` from the address in `r4`, increment it by 1, and store it back to its original address.

I.
```
lha   r3, 0(r4)
addi  r3, r3, 1
sth   r3, 0(r4)
```

II.
```
lhz   r3, 0(r4)
addi  r3, r3, 1
sth   r3, 0(r4)
```

III.
```
lhax  r3, r4, r0
addi  r3, r3, 1
sthx  r3, r4, r0
```

IV.
```
lhzx  r3, r4, r0
addi  r3, r3, 1
sthx  r3, r4, r0
```

a. I
b. II
c. I and III
d. II and IV
e. All of the above

Bus transactions [24 points]

13. What bus signals does the processor drive/assert during a bus read?

a. Address bus, TSIZE[0:1], read/write* (logic low), TS*
b. Address bus, TSIZE[0:1], read/write* (logic high), TS*
c. Address bus, data bus, read/write* (logic high), TS*
d. Address bus, TSIZE[0:1], data bus, read/write* (logic high), TS*
e. None of the above

14. What bus signals does a peripheral drive/assert during a bus write?

a. TA*
b. TA*, read/write* (logic low),
c. TA*, read/write* (logic low), data bus
d. read/write* (logic low), data bus
e. None of the above

15. When is the data latched by the peripheral during a write cycle?

a. TS* and the rising edge of the bus clock
b. TA* and the falling edge of the bus clock
c. TA* and the rising edge of the bus clock
d. TA*
e. None of the above

16. What signals are valid when TS* is low on both a read and a write transaction?

a. Address bus, TSIZE[0:1]
b. Address bus, data bus, TSIZE[0:1]
c. Address bus, data bus
d. Address bus, TA*
e. None of the above

17. What signals would you combine logically to enable an I/O device tri-state buffer? Assume a word read.

a. Address decode, TA*
b. Address decode, TA*, TSIZE[0:1]
c. Address decode, TA*, read/write*
d. Address decode, TS*, read/write*
e. None of the above

18. A read/write register is located at 0x0200 0040. You decode address lines A6, A7, A24, A25. As in lab, locations A0—A5 are not on the bus. How many shadow locations are there?

a. $2^{22}-1$
b. 2^{22}
c. 2^{21}
d. $2^{20}-1$
e. None of the above

Memory Interfacing [28 points]

For Questions 19—21, assume you are using an MPC823 hooked up to the memory hardware you designed in Lab 4. The memory will function as in the macros in the lab assignment. These macros are given in Figures 1 and 2 below.

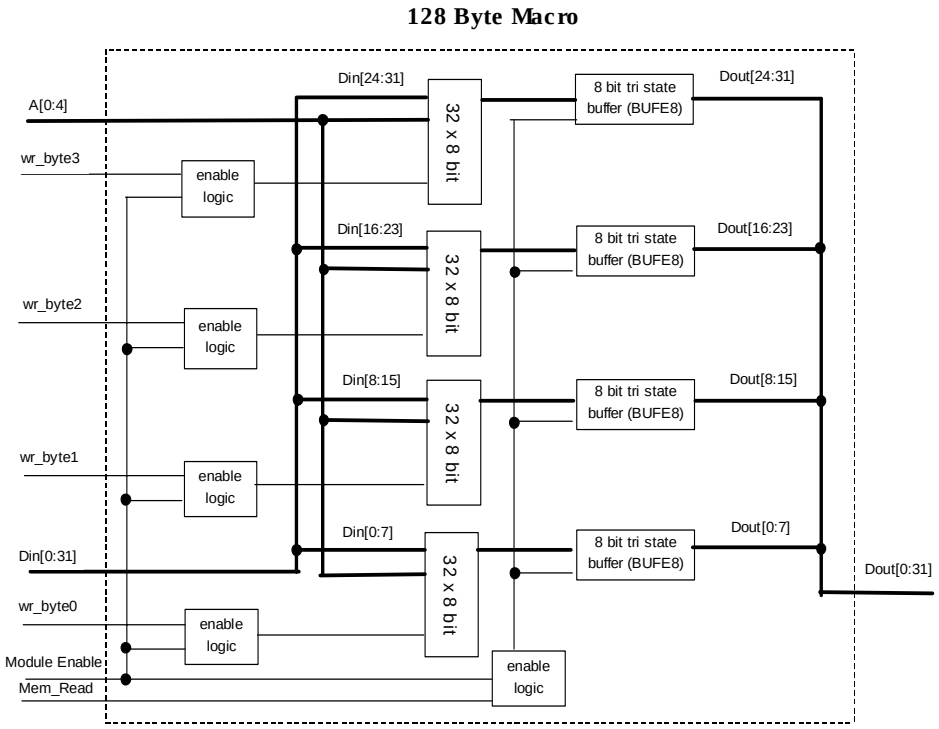

Figure 1: 128-byte macro

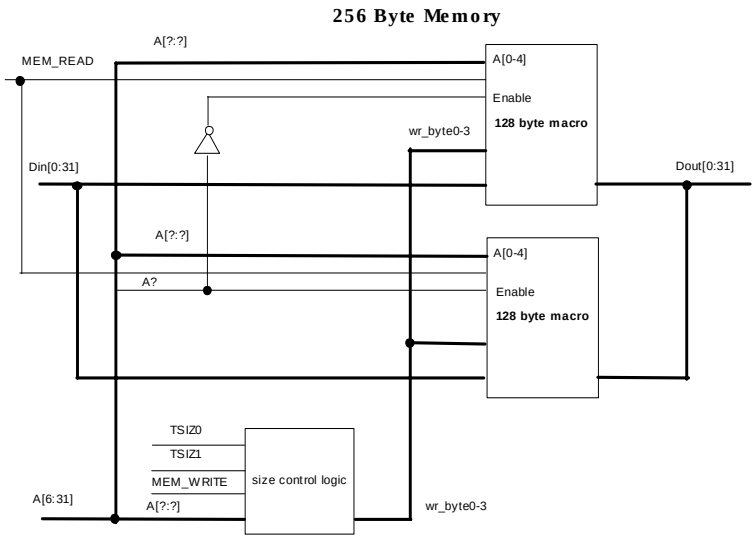

Figure 2: 256-byte macro.

19. Assume the following facts about the current state of the memory and registers.

Register	Contents	Memory location	Contents
`r3`	0x3000004	0x300000	0x12345678
`r4`	0xDEAD	0x300004	0x0059FF00
		0x300008	0x00ADBEEF

If you perform the instruction `sth r4, 3 (r3)`, which of the following will appear on the Logic Analyzer for the instruction?

	Transaction 1			Transaction 2		
	Address	Data	TSIZE	Address	Data	TSIZE
a.	0x300007	0x0000DEAD	10	No Second Transaction		
b.	0x300007	0xAD0000DE	10	No Second Transaction		
c.	0x300007	0x0000000DE	01	0x300008	0xAD000000	01
d.	***0x300007***	***0xDEDE00DE***	***01***	***0x300008***	***0xAD000000***	***01***
e.	0x300007	0x0059FFDE	01	0x300008	0xADADBEEF	01

20. Assuming the next instruction is `lhz r4, 3(r3)` and the memory has changed to reflect the `sth` command from Question 19, what will appear on the Logic Analyzer?

	Transaction 1			Transaction 2		
	Address	Data	TSIZE	Address	Data	TSIZE
a.	0x300007	0x0000DEAD	10	No Second Transaction		
b.	0x300007	0xAD0000DE	10	No Second Transaction		
c.	0x300007	0x0000000DE	01	0x300008	0xAD000000	01
d.	0x300007	0xDEDE00DE	01	0x300008	0xAD000000	01
e.	***0x300007***	***0x0059FFDE***	***01***	***0x300008***	***0xADADBEEF***	***01***

21. Consider the following statements.

I. The worst-case unaligned access will take 3 transactions.
II. Half words are only unaligned if they contain bytes in different words (they cross a word boundary).
III. Byte accesses are always aligned.
IV. Unaligned word accesses can always be broken into aligned half-word accesses.

Which of these statements are always true?

a. I, II, and IV
b. I, III, and IV
c. II and III
d. I and III
e. II and IV

For Questions 22—25, assume your are still using a hardware design like the one in Figures 1 and 2, but now with a new processor that is exactly like the MPC823 except for one modification: We have added the ability for triple-byte transactions. To do this, we modified the functionality of TSIZE. The new tables are given below:

Read Cycle Data Bus Table

Size	TSIZE [0:1]		A30	A31	D0-D7	D8-D15	D16-D23	D24-D31
Byte	0	1	0	0	OP0	-	-	-
	0	1	0	1	-	OP1	-	-
	0	1	1	0	-	-	OP2	-
	0	1	1	1	-	-	-	OP3
Half-Word	1	0	0	0	OP0	OP1	-	-
	1	0	1	0	-	-	OP2	OP3
Triple-Byte	1	1	0	0	OP0	OP1	OP2	-
	1	1	0	1	-	OP1	OP2	OP3
Word	0	0	0	0	OP0	OP1	OP2	OP3

Write Cycle Data Bus Table

Size	TSIZE [0:1]		A30	A31	D0-D7	D8-D15	D16-D23	D24-D31
Byte	0	1	0	0	OP0	-	-	-
	0	1	0	1	OP1	OP1	-	-
	0	1	1	0	OP2	-	OP2	-
	0	1	1	1	OP3	OP3	-	OP3
Half-Word	1	0	0	0	OP0	OP1	-	-
	1	0	1	0	OP2	OP3	OP2	OP3
Triple-Byte	1	1	0	0	OP0	OP1	OP2	-
	1	1	0	1	-	OP1	OP2	OP3
Word	0	0	0	0	OP0	OP1	OP2	OP3

22. What is the worst-case number of transactions for a word access?

a. 1
b. 2
c. 3
d. 4
e. None of the above

23. What is the worst-case number of transactions for a half-word access?

a. 1
b. 2
c. 3
d. 4
e. None of the above

24. How many unaligned word accesses do <u>not</u> use this new triple-byte feature?

a. 0
b. 1
c. 2
d. 3
e. None of the above

25. Referring to this new processor, which parts of your hardware would you have to modify in order to support the triple-byte feature? Assume your hardware was that given in Figures 1 and 2.

a. Only the write decode hardware.
b. Only the read decode hardware.
c. Both the write and read control hardware.
d. Nothing needs to be changed.