Solutions:	**ME Spring 1999**	**Course**	**COMP 180**
Due Date	**30 March 1999**	**Sections:**	**L1, L3, L4**
Time	**19:00-21:00**	**Instructor**	**George Baciu**

NOTE: Only simple calculators are permitted. Please make sure that their is no information relevant to the course stored anywhere in the memory of the calculator. Please read the questions carefully and try to understand them before attempting to answer. Please work out the answers neatly in the space provided. When indicated, please fill the answer in the table provided.

Question	Marks	Score
1	10	
2	05	
3	10	
4	10	
5	15	
6	15	
7	15	
8	15	
9	05	
TOTAL	100	

1. [10/100] **Decimal to binary:**
   What decimal number does this two's complement binary number represent:

   Solution:

Base ten		Base two							
-500_{ten}	=	1111	1111	1111	1111	1111	1110	0000	1100
invert	=	0000	0000	0000	0000	0000	0001	1111	0011
add one	+	0000	0000	0000	0000	0000	0000	0000	0001
500_{ten}	=	0000	0000	0000	0000	0000	0001	1111	0100

2. [05/100] **Binary 16bit to 32bit extension:**
   Given the following binary number represented as a 16-bit two's complement

Base two							
				1101	1011	1100	1001

   convert the number into a 32-bit two's complement representation

   Solution:

Base two							
1111	1111	1111	1111	1101	1011	1100	1001

3. [10/100] **Decimal to IEEE 754 Fl. Point:**
Given the following decimal number: -1.25×10^{-1} Convert the number into 32-bit IEEE 754 floating point format.

Solution:

$$\begin{aligned}
-1.25 \times 10^{-1} &= -125 \times 10^{-3} && (1)\\
&= -\frac{125}{1,000} && (2)\\
&= -\frac{1}{8} && (3)\\
&= -\frac{1}{2^3} && (4)\\
&= -1 \times 2^{-3} && (5)\\
&= -0.001 && (6)\\
&= -1.00000000000000000000000 \times 2^{-3} && (7)\\
&= -1.00000000000000000000000 \times 2^{124-127} && (8)
\end{aligned}$$

Hence,

$$\begin{aligned}
S &= 1 && (9)\\
exp &= 124_{ten} && (10)\\
&= 0111\ 1100_{two} && (11)
\end{aligned}$$

Hence, the floating point representation is:

Base two									
1	0111	1100	0000	0000	0000	0000	0000	000	

4. [10/100] **Binary IEEE 754 to decimal:**
Convert the following 32-bit IEEE 754 floating point format number into a decimal number:

Base two		
0	0100 0011	1101 0000 0000 0000 0000 000

Solution:

$$\begin{aligned}
exp &= 0100\ 0011_{two} && (12)\\
&= 2^6 + 2^1 + 2^0 && (13)\\
&= 64 + 2 + 1 && (14)\\
&= 67 && (15)\\
E &= exp - bias && (16)\\
&= 131 - 127 && (17)\\
&= -60 && (18)\\
0.1101_{two} &= 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} && (19)\\
&= 1/2 + 1/4 + 1/16 && (20)\\
N_{ten} &= (1 + 13/16) \times 2^{-60} && (21)\\
&= (1 + 0.8125) \times 2^{-60} && (22)\\
&= 1.8125 \times 2^{-60} && (23)\\
&= \boxed{1.5721 \times 10^{-18}}_{ten} && (24)
\end{aligned}$$

5. [15/100] **Performance and CPI:**
Suppose we have a computer with the following instruction classes and the corresponding CPI's:

Instruction Class	**CPI**
Data transfers	2
Arithmetic operations	1
Branches	3

(a) Calculate the total number of clock cycles it takes to execute a program with 1000 data transfers, 5000 arithmetic operations, and 2000 branches.

Solution:

Instruction Class	**CPI**	**N**	**Total Clock Cycles**
Data transfers	2	1000	2000
Arithmetic operations	1	5000	5000
Branches	3	2000	6000
Total Clock Cycles			13000

(b) Suppose we run a benchmark suite using a CPU with the instruction classes from above. Given 70% of those instructions were arithmetic operations, 10% branches, and the remaining 20% data transfers, compute the average CPI for the computer running this benchmark suite

Solution:

Instruction Class	**CPI**	$F_i\%$	$C_i \times F_i$
Data transfers	2	20	0.40
Arithmetic operations	1	70	0.70
Branches	3	10	0.30
Average CPI			1.40

6. [15/100] **MIPS Instruction types:**
Given the following MIPS 32-bit instruction format: Imagine a modified integer MIPS

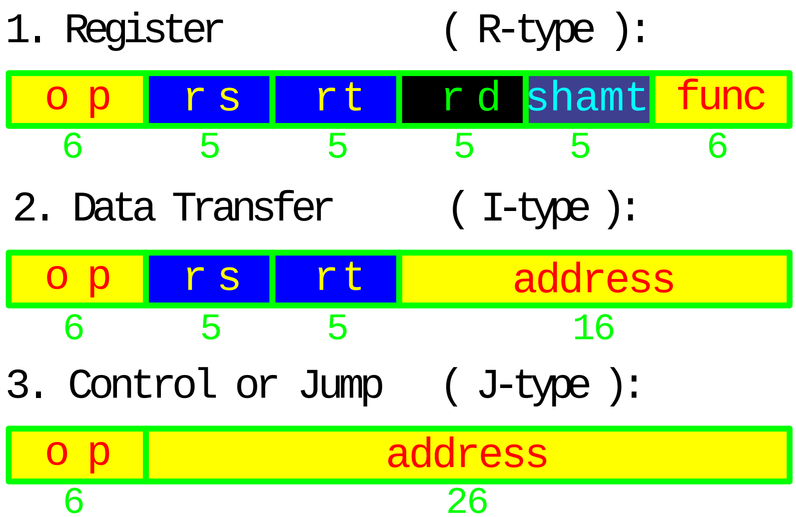

Figure 1: MIPS instruction types and their formats

instruction set with a register file containing 64 general purpose registers rather than the usual 32. Assume that we still want to use a fixed instruction length of four bytes and that the total number of operations must remain unchanged. Also, assume that you can expand and contract fields in an instruction, but you cannot omit them.

(a) How would the format of the R-type instruction change? Label all the fields with their name and bit length.

OPcode	Rs	Rt	Rd	shamt	func
6bits	6bits	6bits	6bits	2bits	6bits

(b) How would the format of the I-type instruction change? Label all the fields with their name and bit length.

OPcode	Rs	Rt	Branch Address
6bits	6bits	6bits	14bits

(c) How would the format of the J-type instruction change? Label all the fields with their name and bit length.

OPcode	Branch Address
6bits	26bits

7. [15/100] **MIPS Assembly programming:**
Given the following assembly program,

```
addi    $19,$0,20
lw      $17,4($19)
add     $20,$19,$16
sw      $20,8($19)
```

trace it and show the final contents of the following registers and memory locations. Please fill in your answer in the table below.

Solution:

Register/memory	**Before**	**After**
Mem[24]	64	64
Mem[28]	16	24
\$16	4	4
\$17	0	64
\$19	0	20
\$20	0	24

8. [15/100] **One-bit ALU functions:**
Given the following 1-bit ALU:

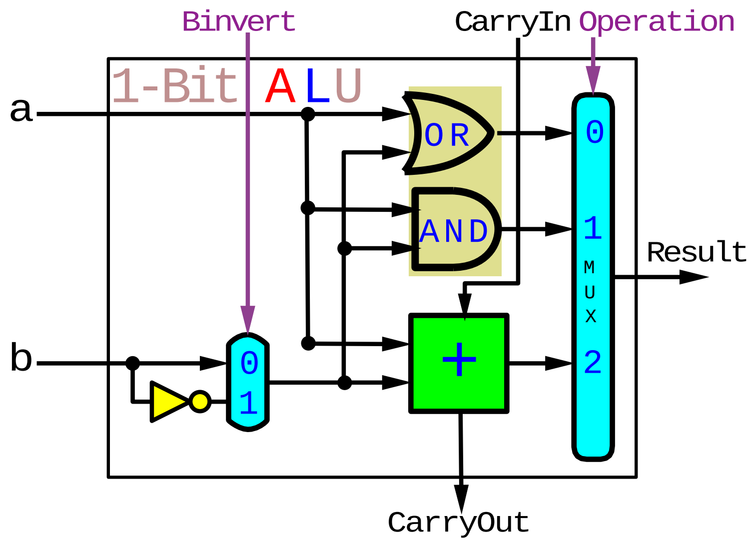

Figure 2: MIPS instruction types and their formats

Show the values of the control lines for the corresponding ALU operations:

Solution:

Binvert	**Operation**	**Instruction**
0	01	AND
0	00	OR
0	10	ADD
1	10	SUB

9. [05/100] **64-bit Set-on-Less Than (SLT)**
   Assembly Language.
   Write a series of instructions to perform a 64-bit SLT (set on less than). The operands are passed in the 32-bit registers as follows:

   - operand 1 in registers \$5 and \$4
   - operand 2 in registers \$7 and \$6.

   The most significant word is in the higher number register, i.e. 5 and 7. The result is to be stored in register \$8.

   Solution:

```
SLT:
     slt   $8, $4, $6
     beq   $5, $7, chk
     slt   $8, $5, $7
chk: ...
```