

Introduction to Computer Organization

Final Exam 605.204 Summer 2017

Name : Caleb Bowers

- 1) As described by Hennessy and Patterson, the modern computer is composed of five classic hardware components. Given one, list the other four:
(4)
- INPUT, CONTROL, and OUTPUT,
DATAPATH, MEMORY.
- 2) The hardware above provides the four **basic functions** that every computer performs.
(4) What are these functions ?
- data input, data output,
data processing, and storing data.
- 3) The **integer** word length and register size of the MIPS machines is
(1)
- A) 8 bits. B) 1 byte. C) 3 bytes. D) **32 bits**.
- 4) How are **negative integer numbers** represented on the MIPS machines ?
(1)
- A) ASCII. B) 1's complement. C) **2's complement**. D) negative integers are not allowed.
- 5) Today, assembly language programs are more efficient than compiled high-level language programs, such as C++, for the same large accounting applications.
(1)
- (**True**)
- 6) 1,099,511,627,776 bytes is known as: A) Gigabyte B) **Terabyte** C) Petabyte
(1)
- 7) Given this hexadecimal representation of a 16-bit Two's Complement number:
(1) **753B**
- Is the value A) **less than zero** B) equal to zero C) greater than zero D) invalid

8) Match these people with their noted accomplishments :
(4)

Alan Turing	<u>C</u>	A) - ENIAC
L0pht	<u>D</u>	B) - TCP / IP
J. Presper Eckert	<u>A</u>	C) - computing theory
Vinton G. Cerf	<u>B</u>	D) - Internet Security

9) What is the purpose of each of the following MIPS assembler directives:

(4)

.globl declares following label follows as global allowing other files to reference it

.text Items following are put in user text segment.

.word Stores the quantities that follow as 32-bit in successive memory words.

.asciiz Store the following string in memory and null terminates it

10) Convert **0.435** to a binary fraction (total of 8 bits) For example $0.75 = 0.1100\ 0000$

(3)

$$0.435_{10} * 2 = 0.87$$

$$0.87_{10} * 2 = 1.74$$

$$0.74_{10} * 2 = 1.48$$

$$0.48_{10} * 2 = 0.96$$

$$0.96_{10} * 2 = 1.92$$

$$0.92_{10} * 2 = 1.84$$

$$0.84_{10} * 2 = 1.68$$

$$0.68_{10} * 2 = 1.56$$

In 8-bit notation $0.435_{10} = 0110\ 1111_2$

11) The MIPS subroutines generally have **PROLOG** code. What is the purpose of this code?

(2)

This code moves the stack pointer to the correct location for the proper register values.

12) The MIPS multiply hardware stores the 64 bit product in two registers, HI and LO.

(1) What does it mean when the value in the HI register is minus one (-1) ?

It means the value stored in LO is a signed value that has a negative sign (sign stored in HI)

13) Which of the following **must be** performed by **every** assembler :

(1)

A) Provide relocatable object code.

B) Process a program in two passes.

C) **Assign machine memory addresses to symbol labels.**

14) Floating point arithmetic has problems different from integer arithmetic.

(3) Which of these floating point arithmetic statements are **True** and which are **False**?

A) Addition of two positive numbers will never overflow. **FALSE**

- B) Addition of a positive and a negative number will never overflow. **TRUE**
- C) Multiplication of a positive and a negative number will never overflow. **FALSE**

15) What is the **primary** reason a two-pass assembler is necessary rather than a one-pass assembler ?
(1)

- A) Instructions may vary in length.
- B) Symbols need to be stored in the symbol table.
- C) Symbols may be defined before they are used as operands.
- D) Symbols may be used as operands before they are defined.**

16) Computer Arithmetic has a problem called '**Overflow**'. Simply what is this problem ?
(1)

When a result from an arithmetic operation exceeds the computer's number range for its given number of bits.

17) Generate the hexadecimal **object machine code** for each line in the following **MIPS** source
(20) program. (Use the Green reference card) Values are decimal numbers.

(For example: sub \$s4, \$s2, \$s1 = 0251 A022)

	<u>Memory Address</u>
.text 2044 = 0000 07FC	0000 07FC
.globl strcpy = 0000 0000	0000 0800
strcpy: add \$t0, \$zero, \$zero = 0000 4020	0000 0804
loop: add \$t1, \$t0, \$a0 = 0104 4820	0000 0808
lb \$t2, 0(\$t1) = 812A 0000	0000 080C
add \$t3, \$t0, \$a1 = 0105 5820	0000 0810
sb \$t2, 0(\$t3) = A16A 0000	0000 0814
beq \$t2, \$zero, finish = 1140 0002	0000 0818
addi \$t0, \$t0, 1 = 2108 0001	0000 081C
j loop = 0810 0808	0000 0820
finish: jr \$ra = 03E0 0008	0000 0824

18) Write the MIPS Assembler pair of hardware instructions for this pseudoinstruction:

(3)

BGT \$s4, \$t1, AGAIN

slt \$at, \$t1, \$s4

bne \$at, \$0, end

end:

19) The MIPS hardware does not have a **subtract immediate** instruction. Why?

(1)

Subtract Immediate is the same as adding the negative (MIPS supports negative constants), so this can be achieved by addi.

20) The MIPS architecture has five **Addressing Modes**.

(5) For each of the modes, give an example of an **instruction** that uses each.

Base lw \$t1, 4(\$t2)

Register jr \$ra

Immediate addi \$t1, \$t3, 12

Pseudo Direct j main

Program Counter bne \$t2, \$t4, end

21) Floating point numbers are stored in **Normalized** format. Give three reasons :

(3)

A) It simplifies exchange of data that includes floating-point numbers

B) It simplifies the floating point arithmetic algorithms to know that numbers are always in this form

C) Increases the accuracy of the numbers that can be stored in a word, since unnecessary leading zeroes are replaced by real digits to the right of the binary point

22) From the discussion readings: What was the first Internet 'killer app'?

(1)

ARPANET's e-mail

23) What decimal value does this bit pattern :

(1)

0000 0000 0000 0000 0000 0000 0000 0000

represent (exactly) in the IEEE 754 format ? +0 (since sign bit is 0)

Sign	exponent								fraction																							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

- 24) Convert the following decimal value to its MIPS floating point binary value:
(4) (IEEE 754 32-bit format) Show result as hex digits.

8765.75

The sign bit equals 0 since 8765.75 is positive.

The exponent is $2^{13} = 8192$, $127 + 13 = 140$, so exponent equals $140_{10} = 10001100_2$

The decimal is then $\frac{8765.75}{8192} = 1.070037841796875$ with the 1 being implicit in IEEE 754,

so the decimal is .070037841796875, which in binary is represented as:

$$.070037841796875 * 2 = 0.14007568359375$$

this operation is repeated iteratively for all 23 bits of the fraction to determine the binary representation (the 1's column of each answer, and then multiply the decimal by 2) of

.070037841796875. Doing this, we arrive at (at bit 7 we get 0, so the remaining bits are all 0):

000 1000 1111 0111 0000 0000

Putting all of this together we have:

Sign	exponent								fraction																					
0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0

Or, 0100 0110 0000 1000 1111 0111 0000 0000

- 25) Identify which of these Assembly process functions would **most likely** occur in
(4)

1) Pass one or 2) Pass two of a two-pass assembler

2 a) Translate mnemonic operation codes to machine codes

1 b) Save addresses assigned to labels

1 c) Scan for label definitions

2 d) Write object code

- 26) Match the statements (a - e) with the most appropriate **Object Program** record type :
(5)

D) Data Segment

S) Symbol

H) Header

T) Text Segment

R) Relocation

T a) Object instruction code.

S b) Relative address of global symbol.

R c) Modification information.

H d) Program name.

 D e) Data values.

- 27)** Does the MIPS assemble language allow for arithmetic with 1-byte and 2-byte operands?
(2) Yes / No ? Why ?

No, MIPS does not allow for arithmetic with 1-byte and 2-byte operands because arithmetic operands must be registers. Registers are 32 bits long, which is the equivalent of 4 bytes, so we cannot use 1-byte or 2-byte operands.

28) Natural languages are clear and concise, and therefore suitable for use as programming languages.
(1) (False)

29) Register \$t5 contains a negative 32-bit value in two's complement form. After performing a 1-bit **sll** instruction, followed by a 1-bit **srl** instruction. Register \$t5 would contain the absolute value of the original value.
(1) (False)

30) The IEEE 754 floating point standard includes these objects.
(4) What they are, and how they are used ??

Guard Bit : Extra bit kept on the right that holds the least significant bit of the solution. This is done to improve rounding accuracy.

Round Bit : An extra bit that holds the value to the right of the least significant bit. This is used to determine which way to round the answer of a floating point calculation.

Sticky Bit : A bit used in addition rounding that is set whenever there are nonzero bits to the right of the round bit. It is used in intermediate calculations and allows for the computer to see the difference between .0500 and .0501 for example.

NaN : The result for invalid operations. Stands for Not a Number and allows programmers to postpone some tests and decisions to later in the program

31) What is **RAID 1** ? A scheme to mitigate loss in the event of disk failure. It creates a mirroring structure of one back up disk for every accessible disk. So, there are twice as many disks needed for RAID 1 as there are for RAID 0 (no redundancies)
(2)

32) Which of the following is a primary characteristic of **System Software** ?
(1)

- A) It is totally machine independent.
- B) Will run on any PC or Unix machine.
- C) Supports the operation of a specific computer.
- D) Calculates employee paychecks.

33) What is the primary purpose of Pass One of the Two-pass Linker/Loader ? Creates the external symbol table that is a list of the external symbols and their assigned addresses

(2)

34) A process that is BLOCKED must wait for an event to occur before it can resume processing.
(1) (True)

- 35) Given two programs that are multiprogrammed in a single processor computer:
 (2) Program ONE uses a large amount of CPU time and little I/O.
 Program TWO uses a small amount of CPU time but performs a large number of I/O operations.
 Which program should get the higher priority for **dispatching** the single CPU? Why?

Program TWO should have the highest priority. Since it isn't CPU intensive it can perform its CPU tasks and then while it waits for the I/O operations to finish Program ONE can be performing its tasks that require greater use of the CPU. This can continue until both Programs are complete. Program ONE gets the CPU while Program TWO waits on I/O and then Program ONE waits while Program TWO gets the CPU for its minimally CPU intensive tasks.

- 36) The purpose of the Main Frame Operating System **Job Scheduler** is to assign resources to waiting program processes. The Scheduler has a priority scheme.
 (2) List the names of two of these schemes.

A) First-Come-First-Serve (FCFS)

B) Round-Robin

- 37) The Grammar describes the **SYNTAX** of the
 (1) programming language.

- 38) The I/O Channel causes an I/O Interrupt when the I/O process completes.
 (1) (True)

- 39) The purpose of Interrupts and the Interrupt Processing is to
 (1)
 A) Stop the computer when a fault occurs.
 B) Slow down the processing of programs.
 C) Enable the operating system to efficiently process several concurrent programs.

- 40) Given this binary bit pattern, what is the MIPS instruction as Assembly Language:
 (3)

1000 1101 0010 1000 0000 0100 1011 0000
 100011 01001 01000 0000010010110000

The first 6 bits of a MIPS instructions are the opcode and since this has a nonzero opcode, it is an I type instruction:

op						rs					rt					Constant or address															
1	0	0	0	1	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0

This translates to the MIPS instruction:

lw \$t1, 1200(\$t0)

- 41) One of the two basic functions of any Operating System is to provide the programming
(1) run-time environment. What is the other basic function ?

Provide a User Interface to make the hardware appear more friendly to users.

42) Given this Intermediate File of a Compiler :
(8)

(1)	:=	#1		Indx	Modified. Changed to := #0 Indx
(2)	JGT	Indx	#25	(20)	
(3)	-	Indx	#1	t1	Removed.
(4)	*	t1	#10	t2	
(5)	*	#2	MLK	t3	Moved outside of loop
(6)	-	t3	#1	t4	Moved outside of loop
(7)	-	t4	#1	t5	Unchanged
(8)	+	t2	t5	t6	Unchanged
(9)	*	t6	#4	t7	Unchanged
(10)	-	Indx	#1	t8	Removed. Reuses t1.
(11)	*	t8	#10	t9	Removed. Reuses t2
(12)	*	#2	MLK	t10	Removed. Reuses t3
(13)	-	t10	#1	t11	Removed. Reuses t4
(14)	+	t9	t11	t12	Unchanged
(15)	*	t12	#4	t13	Unchanged
(16)	:=	ZYX[t13]		CBA[t7]	Unchanged
(17)	+	#1	Indx	t14	Unchanged
(18)	:=	t14		Indx	Unchanged
(19)	JMP			(2)	Unchanged
(20)					

Optimize the code.

Mark which statements would be moved, modified, or removed.

Machine independent code optimization uses many techniques.

List the optimization methods you used.

The primary optimization methods used were moving of Loop Invariant statements (lines 5 and 6), removed duplicate code (lines 10-13), and removed common subexpressions (lines 3 and 10).

The new optimized code looks like:

(1)	: =	#0		Indx
(2)	*	#2	MLK	t3
(3)	-	t3	#1	t4
(4)	JGT	Indx	#25	(20)
(5)	*	t1	#10	t2
(6)	-	t4	#1	t5
(7)	+	t2	t5	t6
(8)	*	t6	#4	t7
(9)	+	t2	t4	t12
(10)	*	t12	#4	t13
(11)	: =	ZYX[t13]		CBA[t7]
(12)	+	#1	Indx	t14
(13)	: =	t14		Indx
(14)	JMP			(2)
(15)				

43) In a few phrases, describe these components of a compiler.

(3)

SCANNER -

Lexical analysis. Reads input text and finds the token objects.

PARSER -

Syntactical analysis. From tokens, it discovers the structure of the program.

CODE GENERATOR -

Semantic analysis. Using the structure it creates the machine language.

44) From the discussion readings: What is Ubuntu?

(1)

Ubuntu is a an operating system that uses the Linux kernel as its foundation.

45) Machine independent code optimization uses many techniques. List two of the methods discussed in the presentation that you **did not** use in **problem #42** (on previous page).

(2)

Reduction in Strength

Folding

46) Match the following terms with the **most correct** definition.

(8)

 D 1) CHANNEL PROGRAM

A) Set of programs in a Circular Wait.

 G 2) CONTEXT SWITCHING

B) Set mask to prevent Interrupt Processing

 A 3) DEAD - LOCK

C) Register containing interrupt mask

 E 4) DISPATCHING

D) Set of instructions for special i/o

 H 5) STORAGE PROTECTION KEY

E) Select next program to process by CPU

 B 6) INHIBIT INTERRUPT

F) Data area used by the Memory Manager

 F 7) PAGE TABLE MAP

G) Saving / restoring registers by Interrupt Processor.

 C 8) PROGRAM STATUS WORD

H) Half byte used for memory access control.

- 47) Is **memory-protection hardware** necessary on a machine that uses a Virtual (demand-paged) Memory Management system ? Yes / No ? Why ?
(2)

No. Every program has its own page table, which contains all of the pages it can access, so it doesn't have any information regarding the location of other data associated with other programs.

- 48) In many operating systems, the **Timer** Interrupt (the assigned time interval is used up) is assigned a lower priority than the **Operating System Service Request** Interrupt. Why is this done?
(2)

The Timer Interrupt only pertains to a given process, but the Operating System Service Request is an interrupt that affects the OS's entire service ability and the devices on which it is running. Since it has a larger impact on the all processes running rather than just one, it receives a higher priority. A timer interrupt just signals that another process can be context switched to.

- 49) Once the operating system detects that a **Deadlock** has occurred, how does it resolve the conflict between the processes ? List two methods.
(2)

A) The OS can abort all deadlocked processes, removing all deadlocks.

B) Successively remove deadlocked processes until the deadlock is resolved and no longer exists.

- 50) What actions **must** the Operating System perform when an interrupt is recognized ?
(2)

A) Save the program status and register contents.

B) Transfer control to the interrupt processing routine

- 51) Is it ever safe (you do not lose data values) for a MIPS user program to use registers **\$k0** or **\$k1** ?
(2) Yes / No ? Why ?

No, it is not safe since these two registers are reserved for the OS kernel to be used by the assembler and operating system, so there is no certainty that any values the user placed in those registers is maintained through program execution.

- 52) The Textbook, page 36, discusses the Classic CPU Performance Equation. Fill in the blanks:
(2)

$$\text{CPU Time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock Cycle}}$$

53) Given the following Simplified Pascal grammar :

(3)

```

< stmt-list > ::= < stmt > { ; < stmt > }
< stmt > ::= < assign > | < read > | < write > | < for >
< assign > ::= id := < exp >
< exp > ::= < term > { + < term > | - < term > }
< term > ::= < factor > { * < factor > | DIV < factor > }
< factor > ::= id | int | ( < exp > )

```

Modify the above Grammar to include the exponentiation operation. Make exponentiation the highest priority arithmetic operation.

< stmt-list >	::=	< stmt > { ; < stmt > }
< stmt >	::=	< assign > < read > < write > < for >
< assign >	::=	id := < exp >
< exp >	::=	< term > { + < term > - < term > }
< term >	::=	< factor > { * < factor > DIV < factor > }
< factor >	::=	< base > { ^ < pwr > }
< base >	::=	id int (< exp >)
< pwr >	::=	id int (< exp >)

54) Create quadruples of the form we have used in class for the following <assign> statement:

(6)

ZEE := 3 * EX - 5 * WHY + EX / WHY;

*	Ex	#3	t1
*	#5	Why	t2
/	Ex	Why	t3
-	t1	t2	t4
+	t4	t3	t5
:=	t5		ZEE

55) The Optical Mouse includes an embedded processor.

(1)

(True)

- 56)** Data Security consists of a number of methods and techniques. Name two.
(2)

(Extra credit for more than two. Max : 4 extra points.)

<u>File Access Control</u>	<u>User Account Passwords</u>
<u>User device access control</u>	<u>Information Flow Control</u>
<u>Secure Building</u>	<u>Encryption (certificates, PKI)</u>

- 57)** Don't forget to complete the online Course Evaluation Survey.
It is the only method to provide the feedback for improving the course.