Problem Set 12

Justin Ely

605.204.81.FA15 Computer Organization

23 November, 2015

1

1. Input
2. Memory
3. Datapath
4. Control
5. Output (given)

2

1. Data input
2. Data output
3. Data processing
4. Data storage

3

The correct answer is **D) 32 bits**. MIPS machines have 32 32-bit registers, and this is the size of a word.

4

The correct answer is **C) 2's complement**.

5

This statement is **false**. Though there is a good deal of ambiguity to this question, compilers these days are written and optimized by very experienced developers to produce very efficient compiled code. Though a specific program written directly in assembly by a highly skilled developer could be more efficient than it's high-level, compiled, equivalent, in general this is not the case.

6

The correct answer is **B) Terabyte**. 1 Terabtye is 10^{12} bytes. This means $1,099,511,627,776$ bytes $*\frac{1TB}{10^{12}bytes}$ $= 1.099$ TB $\sim$ 1 TB.

7

The correct answer is **A) less than zero**. The binary representation of **D49B** is 1101 0100 1001 1011. The leading 1 in two's compliment arithmetic means that this value is negative.

8

- Alan Turing: C) - computing theory
- Joseph Jacquard: D) - control program on punch cards
- J. Presper Eckert: A) - ENIAC
- Vinton G. Cerf: B) -TCP/IP

9

- .text: signifies to the assembler that the following section is assembly code.
- .asciiz: stores the following in memory as a null-terminated string.
- .word: stores the following values as words in memory.
- .globl: tells the assembler that the following label is considered global and is able to be referenced from other files.

10

The closest this value can come in 8 bits is 0.1101 0111, which evaluates to 0.83984.

11

The **PROLOG** code exists to save the state of the registers before a subroutine call so that data isn't lost. The end code to restore registers back to their initial state is called the **EPILOG**.

12

Having the HI register be -1, thus the last 16 bits are -1, means the result of the calculation was NaN.

13

The correct answer is **C)**, assign machine memory addresses to symbol labels. Though A) and B) are beneficial for many reasons, they are not required by all compilers.

14

- A) **Addition of two positive numbers will never overflow.** This statement is false. An easy example would be to add the largest possible integer to the largest possible integer of a given precision. This result is too large, and overflow will occur.
- B) **Addition of a positive and a negative number will never overflow.** This statement is true. The largest positive integer added to the largest negative number can be, at most, 0. Thus, this situation can never overflow.
- C) **Multiplication of a positive and negative number will never overflow.** This statement is false. The resulting exponent can be incredibly large, regardless of the sign, and outside the allowable number range on either the positive or negative side.

15

The main reason a two-pass assembler is necessary is because, **C)**, symbols may be used as operands before they are defined. If this happens, a one-pass assembler will not be able to resolve the symbol when it's first used.

16

Overflow is when the result of a computation is larger than the largest representable value.

17

strcpy:	ADD	$t0	$zero	$zero	
0000 4020					
loop:	ADD	$t1	$t0	$a1	
0105 4820					
	LB	$t2	0	$t1	
812A 0000					
	ADD	$t3	$t0	$a0	
0104 5820					
	SB	$t2	0	$t3	
A16A 0000					
	BEQ	$t2	$zero	finish	
1140 000C					
	ADDI	$t0	$t0	1	
2108 0000					
	J	loop			
0800 001C					
finish:	JR	$ra			
03E0 0008					

18

The BGE pseudo instruction requires two individual MIPS instructions:

```
slt $t0, $t6, $s3
beq $t0, $zero, FINISH
```

19

The MIPS language has to subtract immediate because the add immediate instruction can be used with a negative value to achieve the same result.

20

- **Register:** jr

- **Base:** lw
- **Program Counter:** beq
- **Immediate:** addi
- **Pseudo Direct:** j

21

1. Normalization is the standard representation of exponential numbers.
2. It allows the simplification of the algorithms and hardware.
3. It allows for higher accuracy in the represented values.

22

The first ”killer app” was email. In it’s first year it accounted for 75% of the ARPANET’s traffic.

23

0’s in every bit represents exactly 0 in IEEE 754 format, specifically positive 0 (+0), as the sign bit indicates a positive value.

24

The first bit encodes the sign, which a 1 means the value is negative. The next 8 bits represent the exponent. Here, 127 is subtracted from the exponent value of 138, $138 - 127 = 11$, to encode to 2^{11}. The last 23 bits represent the mantissa, which has the value 1.968017578125. Thus, sign * exponent * mantissa yields $-1 * 2^{11} * 1.968017578125 = -\mathbf{4030.5}$.

25

- Translate mnemonic operation codes to machine codes is performed in pass **two**.
- Save addresses assigned to labels is performed in pass **one**.
- Scan for label definitions is performed in pass **one**.
- Write object file is performed in pass **two**.

26

- a) Object instruction code. **Text Segment**
- b) Relative address of global symbol. **Relocation**
- c) Modification information. **Symbol**
- d) Program name. **Header**
- e) Data values. **Data Segment**

27

Yes, MIPS does allow arithmetic with 1-byte and 2-byte operands, albeit indirectly. The load byte and load halfword instructions can be used to load 1-byte and 2-byte operands into registers, upon which arithmetic operations can be performed.

28

This statement is **false**. Natural languages are ambiguous and verbose, which make them very unsuitable to use as programming languages.

29

This statement is **true**. While a mouse with a track ball doesn't require an embedded processor, an optical mouse does require an embedded processor to perform real-time computation of the optical image.

30

- Guard Bit: is an extra bit used in arithmetic calculations. It is used to reduce round-off error after the operands have been aligned.
- Round Bit: is another extra bit used in arithmetic calculations. It is used to reduce round-off error after the operands have been aligned.
- Sticky Bit: is a bit used in arithmetic calculations to indicate the presence of less-significant digits that have been truncated.
- NaN: is a reserved binary representation that means Not a Number. This value is used as the result of invalid operations such as the square root of a negative number.

31

RAID stands for Redundant Array of Inexpensive Disks and is an umbrella term for a selection of strategies that provide increased data security and performance through redundancy. RAID 1 is a specific strategy that writes data to two or more hard drives simultaneously. This way, read requests can be handled by any available drive, and no data is lost through the failure of a single disk (or multiple, given the total number of copies).

32

The correct answer is **C)** Supports the operation of a specific computer. System Software includes thinks like the OS and hardware drivers that interface with specific hardware. Though it can be translated to many machines of different architectures, it is not innately machine independent.

33

Booth's algorithm is a technique for multiplying two's compliment integers that increases efficiency in most cases, though it is not universally more efficient. The algorithm takes advantage of rows of repeated 1's or 0's where only a bit shift is needed instead of multiple additions.

34

This statement is **true**. An example of a blocked program is one that is waiting for a memory resource that another program is actively using. Once the other program releases the memory, then the blocked program can resume.

35

In this situation, **program 2** should be given priority. By the time of dispatch, all I/O operations have been performed. Therefore, the program with a smaller CPU requirement should have priority.

36

- A) One priority scheme is First-Come-First-Served (FCFS). In this scheme, each process is immediately added in to the queue. The queue is processed sequentially, where the process that has waited the longest is always the next to execute. This scheme doesn't take any other criteria into account, such as program execution time or memory usage.

- B) Another priority scheme is Round-Robin. This scheme gives each running process a set amount of time before a context switch occurs. This scheme also doesn't take into account any other criteria such as individual program execution time or memory usage. This strategy favors short-running programs.

37

The Grammar describes the **(B) SYNTAX** of the programming language. The Semantics, on the other hand, is the higher-level meaning of the individual statements.

38

This statement is **false**. The I/O channel would cause an interrupt when the I/O process is requested and started, and would return from the interrupt when the process in completed.

39

The correct answer is **C)**. The main purpose of Interrupts is to allow more efficient processing of multiple concurrent tasks. A simple example would be trying to type in the terminal while a computation algorithm is executing. Without interrupts to process the keystrokes, and input would need to wait until the, potentially long, running programs finishes executing.

40

op code	source register	destination	constant or address
100011	01001	01000	0000010010110000
35	9	8	1200
lw	$t1	$t0	1200

The MIPS instruction is for this binary pattern is: **lw, $t0 1200($t1)**.

41

The other main task is to provide the user-interface to the computer. This makes the computer much more friendly and easy to use by providing an abstraction to the actual hardware.

42

LC	Operation	Op1	Op2	Result
(1)	:=	#1		Indx
(2)	*	#2	KLM	t3
(3)	-	t3	#1	t4
(4)	JGT	Indx	#25	(16)
(5)	-	Indx	#1	t1
(6)	*	t1	#10	t2
(7)	-	t4	#1	t5
(8)	+	t2	t5	t6
(9)	*	t6	#4	t7
(10)	+	t2	t4	t8
(11)	*	t8	#4	t9
(12)	:=	XYZ[t9]		ABC[t7]
(13)	+	#1	Indx	t10
(14)	:=	t10		Indx
(15)	JMP			(4)
(16)				

To optimize the given code, lines 10, 11, 12, and 13 were removed as duplicate operations. The values calculated were already available from operations performed in previous lines. Additionally, lines 5 and 6 do not require the Indx variable to calculate, and could thus be moved outside the loop to avoid repeated calculation.

The optimization methods used here were eliminating common subexpressions and moving loop invariants.

43

- **SCANNER:** The scanner parses through the source program and identifies the tokens.
- **PARSER:** The parser builds the parse tree for the program. This can use the bottom-up or top-down approach.
- **CODE GENERATOR:** The code generator takes the parse tree and generates quadruples of the program code. These quadrules help later optimization and generation of machine/assembly code.

44

The major flaw with BGP is that it automatically trusts users. Data packets can be sent and received by anyone, with no encryption and no user verification. This makes the information sent on the internet very insecure, where the only real protection from malicious eavesdropping is the sheer volume of information to process.

45

Two additional optimization methods are:

1. Folding, where computations involving multiple constants are combined into a single expression.
2. Loop jamming, where multiple loops that go over the same index are combined into a single loop.

46

1. Channel Program: **D)** Set of instructions for special i/o.
2. Context Switching: **G)** Saving/restoring registers by Interrupt Processor.
3. Dead-Lock: **A)** Set of programs in a Circular Wait.
4. Dispatching: **E)** Select next program to process by CPU.
5. Storage Protection Key: **H)** Half byte used for memory access control.
6. Inhibit Interrupt: **B)** Set mask to prevent Interrupt Processing.
7. Page table map: **F)** Data area used by the Memory Manager.
8. Program status word: **C)** Register containing interrupt mask.

47

Yes, although a Virtual memory management system is in-place, some level of the machine still needs to access physical memory. At the level of physical-memory interaction, memory-protection hardware will be needed.

48

The Timer Interrupt often has a lower priority than the Operating System Service Request (OSSR) because a direct request from the OS often cannot wait for a program to finish. The Timer Interrupt is part of program scheduling, where programs are only allowed to run for a certain time before a context switch happens and another program is loaded. If the OSSR had a lower priority than the Timer, then an OS request could only execute in-between running programs.

49

- A) The OS can abort all deadlock processes. This will resolve the deadlock situation, and when the programs are restarted there should be enough of a difference in start time to avoid the same deadlock on the next run.
- B) The OS can successively abort deadlocked processes until the deadlock is resolved. This terminates some programs and leaves others running, but eliminates fewer running jobs that shutting down all-offending jobs.

50

- First, the OS must store all register values so that the currently running program can proceed once the interrupt has been processed.
- Second, the OS must transfer control to the interrupt handler.

51

No, it is never safe for a MIPS user program to use the **$k0 or $k1** registers. These two registers are reserved, by convention, for the OS to use in the event of an exception. All other registers are saved and restored when control is given back to the program, but the data in these two registers could be lost or changed at any time.

52

Assembly Languages should be avoided for general application development because they don't contain sufficient abstraction for most problems. Assembly languages are very close to the machines and require explicit handling of every register to perform any operation. This process requires a great deal of effort and is prone to human error. Additionally, the built-in instruction set is incredibly limited when compared to a high-level programming language. This makes the computation of most common tasks very time-consuming and costly.

53

To support exponentiation, an additional class needs to be added (exponent), and the exponent class needs to be another type of term. This looks like:

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

54

LC	Operation	Op1	Op2	Result
(1)	*	#3	EX	i1
(2)	*	#5	WHY	i2
(3)	*	EX	WHY	i3
(4)	+	i1	i2	41
(5)	+	i3	i4	i5
(6)	:=	i5		ZEE

55

1. file access control
2. passwords required for user accounts
3. memory access protection
4. supervisor modes
5. privileged accounts
6. privileged I/O channels

56

Yes, I did complete the online Course Evaluation Survey.