Register Number | | | | | | | | | |

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110

(An Autonomous Institution, Affiliated to Anna University, Chennai)

<Name of the Department>

Continuous Assessment Test – I / II / III

Question Paper

Degree & Branch	B.E. & Computer Science and Engineering				**Semester**	IV
Subject Code & Name	UCS1401 & Computer Organization and Architecture				**Regulation:**	**2018**
Academic Year	2021-2022	**Batch**	2020-2024	**Date**	**26-03-2022**	**FN**
Time: 90 Minutes	**Answer All Questions**				**Maximum: 50 Marks**	

Part – A (6×2 = 12 Marks)

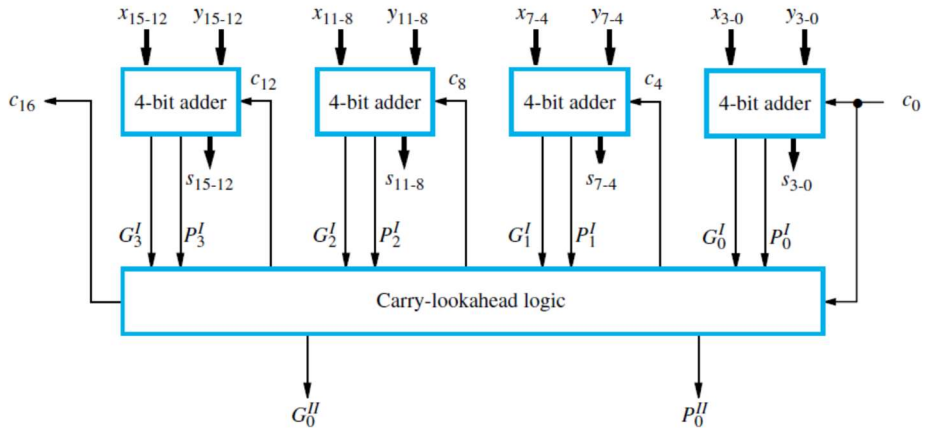
K1	1. Define Amdahl's law. $\text{Speedup}_{\text{overall}} = \frac{\text{Ex Time}_{\text{old}}}{\text{Ex Time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$	CO1	PI 1.3.1 PI 2.1.3
K2	2. Consider a hypothetical processor with an instruction of type LW R1, 20(R2), which during execution reads a 32-bit word from memory and stores it in a 32-bit register R1. The effective address of the memory location is obtained by the addition of a constant 20 and the contents of register R2. Infer the addressing mode implemented by this instruction. ANS: Base Register Indirect	CO1	PI 1.3.1
K3	3. The 32-bit hex value B0A29C47 is stored in the location starting from address 1000. Apply the concept of byte addressing in both is Big endian representation and Little endian representation. Identify the value of the byte in address 1002 if the system is Big endian? And if Little endian? ANS: LE : A2 BE: 9C	CO1	PI 1.3.1
K1	4. A computer has a processor with a 24 bit address bus, 32 bit data bus and 8 control lines. Find the maximum amount of memory that the computer can address. ANS: 16MB	CO1	PI 1.3.1 PI 2.1.3
K2	5. Demonstrate the 2's complement operation for the given operation (-2) - (-7). (use 8-bit representation) (-2) - (-7) =(-2)+7 = 5 (2) = 0000 0010	CO2	PI 1.3.1 PI 1.4.1 PI 2.1.3 PI 13.3.1

	(-2) = 2's comp = 1111 1110 (7) = 0000 0111 (-2) = 1111 1110 (7) = 0000 0111 ---------- (5) 0000 0101		
K1	6. What are the advantages of 2's complement over 1's complement?. ANS: 1. Only one representation for zero in 2's complement Design of arithmetic circuit is easy for 2's complement	CO2	PI 1.3.1

Part – B (3×6 = 18 Marks)

K3	7. A memory unit with a capacity of 65,536 words of 25 bits each. It is used in conjunction with a general-purpose computer. The instruction code is divided into four parts an indirect mode bit, opcode, two bits that specify a processor register and an address part. a. Construct the instruction word format indicating the number of bits and the function of each part. (2 Marks) Estimate the following: b. The maximum number of operation that can be in the computer if the instruction is stored in one memory word. (1 Mark) c. The number of processor registers that are there in the computer and how many bits are there in processor registers, MAR, MBR and PC. (3 Marks) ANS: Memory Size =65,536 words No. of bits to encode memory address = $2^8$6x 2^{10} = 2^{16} Number of Registers = 4 No. of bits to encode register no = 2^2 Size of the memory word =25 bits a) Address - 16 bits Register code - 2 bits Indirect - 1 bit 9 bits 25-19 = 6 bits for opcode a) 2^6 = 64 operations b) 1 6 2 16 =25 bits I | Opcode | Register | Address b) 4 processor register specified by R = 0 0 0 1 1 0 1 1 25 bits each c) MBR has 25 bits MAR & PC has 16 bits	CO1	PI 1.3.1 PI 2.1.3
K3	8. Develop MIPS assembly language program for the following high level language statement *while (save[i] == k) i += 1* Assume i in $s3, k in $s5, address of save in $s6.	CO1	PI 1.3.1

	ANS: 8.
	Loop: sll $t1,$s3,2 # Temp reg $t1 = i * 4
	To get the address of save[i], we need to add $t1 and the base of save in $s6:
	add $t1,$t1,$s6 # $t1 = address of save[i]
	Now we can use that address to load save[i] into a temporary register:
	lw $t0,0($t1) # Temp reg $t0 = save[i]
	The next instruction performs the loop test, exiting if save[i] ≠ k:
	bne $t0,$s5, Exit # go to Exit if save[i] ≠ k
	The next instruction adds 1 to i:
	addi $s3,$s3,1 # i = i + 1
	The end of the loop branches back to the *while* test at the top of the loop. We just add the Exit label after it, and we're done:
	j Loop # go to Loop
	Exit:		

K2	9. Design a 4 bit adder using carry Look ahead Logic ANS:  	CO2	PI 1.3.1 PI 1.4.1 PI 2.1.3 PI 2.2.2 PI 13.3.1 PI 13.3.2

Part – C (2×10 = 20 Marks)

K3	10. Consider a Graphics card which is executing a program that consists of 50% of its total execution time is spent in floating point operations(FP) and 20% of its total execution time is spent in floating point square root operations (FPSQR). Option 1: improve the FPSQR operation by a factor of 10. Option 2: improve all FP operations by a factor of 1.6 Choose the best design alternative. ANS: $Speedup_{overall} = \frac{Time_{org}}{Time_{enh}} = \frac{1}{(1 - Fraction_{enh}) + \frac{Fraction_{enh}}{Speedup_{enh}}}$ $Speedup_{FPSQR} = \frac{1}{(1-0.2)+(\frac{0.2}{10})} = \frac{1}{0.82} = 1.22$ $Speedup_{FP} = \frac{1}{(1-0.5)+(\frac{0.5}{1.6})} = \frac{1}{0.8125} = 1.23$ → Option 2 slightly faster	CO1	PI 1.3.1 PI 2.1.3
	(OR)		
K3	11.We want to compare the computers R1 and R2, which differ that R1 has the machine instructions for the floating-point operations, while R2 has not (FP operations are implemented in the software using several non-FP instructions). Both computers have a clock frequency of 400 MHz. In both we execute the same program, which has these following mixture of instructions. ANS:	CO1	PI 1.3.1 PI 2.1.3

Type the command	Dynamic Share of instructions in program (p_i)	Instruction duration (Number of clock periods CPI_i)	
		R1	R2
FP addition	16%	6	20
FP multiplication	10%	8	32
FP division	8%	10	66
Non - FP instructions	66%	3	3

a) Calculate the CPI for the computers R1 and R2. (5 Marks)
b) Calculate the CPU program execution time on the computers R1 and R2, if there are 12000 instructions in the program? (5 Marks)

ANS:

a) $CPI = \sum_{i=0}^{3} CPI_i * p_i$

Computer R1:

$CPI = \sum_{i=1}^{3} CPI_i * p_i = 0{,}16 * 6 + 0{,}1 * 8 + 0{,}08 * 10 + 0{,}66 * 3 = 4.54$

Computer R1 needs an average of 4.54 clock periods for one instruction

Computer R2:

$CPI = \sum_{i=1}^{3} CPI_i * p_i = 0{,}16 * 20 + 0{,}1 * 32 + 0{,}08 * 66 + 0{,}66 * 3 = 13.66$

Computer R2 needs an average of 13.66 clock periods for one instruction

$CPU_{time} = Number\_of\_instructions * CPI * t_{CPU}$

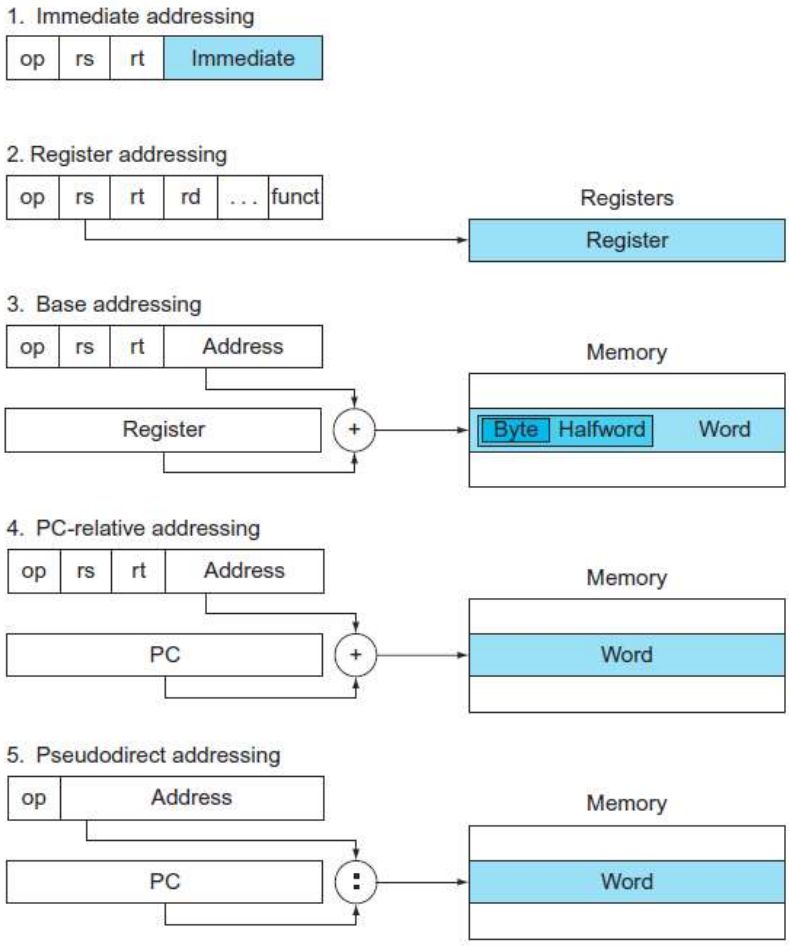
Computer R1:

$CPU_{\text{time}} = 12000 * 4.54 * 1/(400 * 10^6) = 136.2 * 10^{-6} = 136.2\ \mu s$

Computer R2:

$CPU_{\text{time}} = 12000 * 13.66 * 1/(400 * 10^6) = 410 * 10^{-6} = 410\ \mu s$

K2	12. Explain different types of addressing modes in MIPS architecture with examples. ANS:	CO1	PI 1.3.1

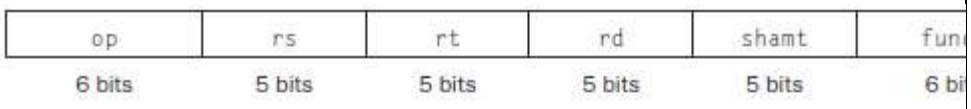


(OR)

13. Explain about different MIPS instruction formats.

ANS:

R type



Here is the meaning of each name of the fields in MIPS instructions:

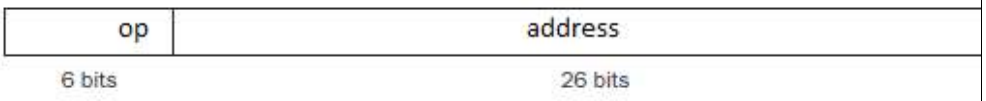
- *op:* Basic operation of the instruction, traditionally called the opcode.
- *rs:* The first register source operand.
- *rt:* The second register source operand.
- *rd:* The register destination operand. It gets the result of the operation
- *shamt:* Shift amount. (Section 2.6 explains shift instructions and this t will not be used until then, and hence the field contains zero in this se
- *funct:* Function. This field, often called the *function code,* selects the s variant of the operation in the op field.

I type:

K2

CO1

PI 1.3.1

	 J type: 		

Prepared By	Reviewed By	Approved By
Course Coordinator	PAC Team	HOD

Guidelines

1. The question paper should be set in accordance with Bloom's Taxonomy (APPENDIX – A: next page) . The questions in a desired knowledge level must contain the respective action verbs.
2. The Knowledge level (Eg. <K2>), course outcome (Eg. <CO2>), and the program indicators (Eg. <1.2.1>) should be mentioned against each question and subdivisions in the respective columns.
3. Both the questions in "either or" type must be set in the same knowledge level and must be from the same CO.
4. In the case of "either or" type questions, the keyword (OR) must be in a separate row.
5. In the case of sub-divisions in a question, it is preferable to have the same knowledge level.
6. The marks assigned to each question in the case of subdivisions should be mentioned clearly at the end of the question within brackets and with a keyword Marks. (Eg. (5 Marks)).
7. Add the keyword "Options" before the choices of an objective type question in Part A.

8. Once the question paper is set, its adherence to the guidelines in terms of knowledge levels and marks distribution has to be approved by the QP Scrutiny Team.

APPENDIX – A
Bloom's Taxonomy Action Verbs

K Level	Bloom's Definition	Action Verbs
K1 Remember	Exhibit memory of Previously learned Material by recalling facts, terms, basic concepts, and answers.	Choose, Define, Find, How, Label, List, Match, Name, Omit, Recall, Relate, Show, Spell, Tell, What, When, Where, Which, Who, Why.
K2 Understand	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions and stating main ideas.	Classify, Compare, Contrast, Demonstrate, Explain, Extend, Illustrate, Infer, Interpret, Outline, Relate, Rephrase, Show, Summarize, Translate
K3 Apply	Solve problems to new situations by applying acquired	Apply, Build, Construct, Develop, Experiment with, Identify, Interview,

	knowledge, facts, techniques, and rules in a different way.	Make use of, Model, Organize, Plan, Select, Solve, Utilize
K4 Analyse	Examine and break information into parts by identifying motives or causes. Make inferences and find evidence to support generalizations	Analyze, Assume, Categorize, Conclusion, Discover, Dissect, Distinguish, Divide, Examine, Function, Inference, Inspect, Motive, Relationships, Simplify, Survey, Take part in, Test for, Theme
K5 Evaluate	Present and defend opinions by making judgments about information, validity of ideas, or quality of work based on a set of criteria.	Agree, Appraise, Assess, Award, Choose, Compare, Conclude, Criteria, Criticize, Decide, Deduct, Defend, Determine, Disprove, Estimate, Evaluate, Explain, Importance, Influence, Interpret, Judge, Justify, Mark, Measure, Opinion, Perceive, Prioritize, Prove, Rate, Recommend, Rule on, Select, Support, Value
K6 Create	Compile information together in a different way by combining elements in a new pattern or proposing alternative solutions.	Adapt, Build, Change, Choose, Combine, Compile, Compose, Construct, Create, Delete, Design, Develop, Discuss, Elaborate, Estimate, Formulate, Happen, Imagine, Improve, Invent, Make up, Maximize, Minimize, Modify, Original, Originate, Plan, Predict, Propose, Solution, Solve, Suppose, Test, Theory

::Q15::Consider a hypothetical processor with an instruction of type LW R1, 20(R2), which during execution reads a 32-bit word from memory and stores it in a 32-bit register R1. The effective address of the memory location is obtained by the addition of a constant 20 and the contents of register R2. Choose one of the following best reflects the addressing mode implemented by this instruction for operand in memory?

{
~Immediate Addressing
~Register Addressing
~Register Indirect Scaled Addressing
=Base Indexed Addressing
}

. The 32 bit hex value 30A79847 is stored in the location starting from address 1000. What is the value of the byte in address 1002 if the system is Big endian? And if Little endian?.

5. State Amdahl's law.
6. Perform the operation (-3) - (-5) using 8bit ALU. (Use 2's complement representation)

A computer has a processor with a 24 bit address bus, 32 bit data bus and 8 control lines. Calculate the maximum amount of memory that the computer can address

An instruction is stored at the location 300. with its address field at the location 301. The address filed has the value 400. The content of 400 is 500.The processor register R_1 contains the number 200. Evaluate the EA if the addressing mode of the instruction is

(a) direct (b) indirect (c) relative (d) register indirect (e) Indexed with R_1 as index register.

Solution:

(a) direct → 400
(b) indirect → 500
(c) relative → 302+400 = 702
(d) reg.indirect → 300
(e) indexed → 400+200 = 600

PC → 300
301
302