**ASSUMPTION UNIVERSITY**

**Vincent Mary School of Science & Technology**

**DEPARTMENT OF COMPUTER SCIENCE**

**COURSE OUTLINE**

|  |  |  |
| --- | --- | --- |
| **COURSE ORGANIZATION** | | |
| **Course Title:** | Computer Architecture CS2202 (section 541) | |
| **Course Status:** | Major Required | |
| **Pre-requisite:** | None | |
| **Credits:** | 3 Credit Points | |
| **Semester:** | 2/2019 | |
| **Classroom** | VMS0201 (Friday 09:00 - 12:00) | |
| **Description:** | This course will cover the following areas of computer architecture: computer system components, number system, logic gates, combinatory circuits, sequential circuits, micro-operations using flip-flop, registers, arithmetic circuits and counter, computational operations and control of micro-operations, design of memory unit, design of set of instructions, design of parallel processing as well as SISD, SIMD, MISD, and MIMD architectures. | |
| **Objectives:** | * The major objective of the course is to present, as clearly and completely as possible, the nature of a modern computer architecture with various computer component levels such as, internal bus system, interrupts, internal memory system, I/O units, and CPU. * The emphasis will be on various types Computer Architectures and Computer Design covered topics such as Instruction-Set Architecture, Bus architecture, memory architecture, Pipelining and Parallel Processors. | |
| **Marks Allocation:** | Term Project (individual)\*  Class work (Quizzes & Assignments)  Midterm Examination  Final Examination (Comprehensive) | 15 %  20 %  25 %  40 % |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COURSE INSTRUCTOR** | | | | |
| **Instructor:** | Asst. Prof. Dr. Anilkumar K Gopalakrishnan |  |  |  |
| **Office:** | VMS0508 | | | |
| **Email:** | [anil@scitech.au.edu](mailto:anil@scitech.au.edu) |  |  |  |
| **Course materials:** | portal.scitech.au.edu/anilkumar |  |  |  |
| **Mobile Phone:** | 0891351711 |  |  |  |

|  |  |
| --- | --- |
| **COURSE RESOURCES** | |
| **Textbooks:** | * Digital Design and Computer Architecture, 2nd Edition, David M.H and Sarah L.H, Morgan Kaufmann, Elsevier, 2013 (ISBN: 978-0-12-394424-5) |
| **Reference(s):** | * Computer Architecture and Implementation, Harvey G. Cragon, Cambridge University Press, 2000 (ISBN: 0-52-165168-9) * Computer Architecture: A Quantitative Approach 4th Edition, David A. Patterson and John L. Hennessy, Morgan Kaufmann Publishers, 2005. (ISBN: 0-12-370490-1) |

|  |
| --- |
| **COURSE POLICIES:** |
| 1. Students are required to have 80% of class attendance to be eligible for the final examination. 2. If a student absent more than 3 classes without any relevant reason, then he/she should withdraw from the course (is mandatory!). 3. The student who misses the final exam without any relevant reason will receive a grade based on his/her total marks (means the grade will not be a ‘**W**’). 4. The student will receive a ‘**W**’ only when he/she officially withdraw from the course (that is, *the grade ‘***W’** *is given only by the system and is not by the lecturer*). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COURSE EXAMINATIONS** | | | | |
| **Midterm:** | Date: | 02-03-2020 (D-M-Y) | Time: | 12:00-14:00 (2 hrs) |
|  |  | | |
| **Final:** | Date: | 08-05-2020 (D-M-Y) | Time: | 13:00-16:00 (3 hrs) |
|  |  | | |

|  |  |  |
| --- | --- | --- |
| **COURSE CONTENTS AND TENTATIVE SCHEDULE** | | |
| **Week** | **Topic** | **Remarks** |
| Week 1,2 | Introduction-Number System & Logic Gates | Chapter 1 |
| Week 3, 4 | Combinational Logic Design | Chapter 2 |
| Week 5 | Sequential Logic Design | Chapter 3 |
| Week 6,7 | Arithmetic Circuits, Sequential Building Blocks, **Quiz1 and Revision** | Chapter 5 |
|  | **Midterm Exam** |  |
| Week 8 | Sequential Building blocks | Chapter 5 |
| Week 9 | Instruction Set design | Chapter 6 |
| Week 10 | Instruction Set design, Performance evaluation | Chapter 6 |
| Week 11 | MIPS Pipeline Architecture | Chapter 7 |
| Week 12 | MIPS Pipeline Architecture | Chapter 7 |
| Week 13-14 | Memory system | Chapter 8 |
| Week 14 | **Quiz II and revision** |  |
| Week 15 | **Project Presentation** |  |

|  |
| --- |
| **TERM PROJECT DESCRIPTION:** |
| Write a simple CPU Instruction set simulation program for a selected CPU size (either 16-bit, 24-bit or 32-bit) using any programming language (even though the pipeline simulation of the instruction set is an optional one, whoever with such an additional feature will be rewarded greatly). The ISA should include *user defined instruction sets*, *control registers*, *main memory indication*, CPI (clock per instruction), etc.  **Requirements:**   * The simulation should support 16-bit integer number or more. * The ISA should consist of at least 8 general purpose registers. * The program should be able to handle (at least) op-codes such as *mov* (move), *add* (addition), *sub* (subtraction), *mul* (multiplication), *div* (division), etc.   **Example ISA**   1. User will be allowed to input the instructions in the following way:   Assume that a 24-bit ISA with 16-bit integer data representation and 16-bit 8 GPRs (say r0,…..r7). Select any two registers from the given GPR list as a *multiplication register* for multiplication and a *remainder register* for division. An example instruction set with their decoded and encoded forms are shown below (where **r0** is a multiplication register and **r7** is a remainder register):    Decoded form Instruction meaning Encoded form  mov r1 3 r1 🡨 3 [0001001 0000 0000 0000 0011]  add r1 3 r1 🡨 r1 + 3 ------   mov r2 r1 r2 🡨 r1 ------   mul r2 -1 r0: r2 🡨 r2 \* -1 (total 32-bit result) -------  mov r3 r2 r3 🡨 r2 -------  div r3 2 r7:r3 🡨 r3 / 2 ------   end 0 0  **Values of registers after the execution of the instruction set**  During the execution of the above code sequence, the values of registers would be varied in the following way:  r1 = 3 [0000 0000 0000 0011]  r1 = 6 [0000 0000 0000 0110]  r2 = 6 [0000 0000 0000 0110]  r0: r2 = -6 [1111 1111 1111 1111 1111 1111 1111 1010]  r3 = -3 [1111 1111 1111 1101] r7: 0 [0000 0000 0000 0000]  **CPI (Clocks Per Instruction)**  CPI of the program depends on the number of clock cycles used by each instruction  **Pipelined version**  Pipelined version of instruction set simulation is an optional part of the project. But, whoever with such  an additional feature will be rewarded greatly.  **Scoring Criteria:**  The following key quality will be considered:   * ISA design & coding 10% * Presentation 2% * Report with code 3%   (*If code is not printed along with the report, then the project is not valid*). |