#### COURSE CODE: COMP 2601

**COURSE TITLE: Computer Architecture**

**LEVEL: 2**

**NO. OF CREDITS: 3**

**SEMESTER: 1**

**PRE-REQUISITE(S): COMP 1600**

**COURSE DESCRIPTION**

This course describes how computers work with the goal of gaining a better understanding of how computers work from a programmer’s point of view rather than the hardware designer’s view. Topics include: Digital Logic and Digital Systems, Machine Level Representation of Data, Assembly Level Machine Organization, Memory System Organization and Architecture, Interfacing and Communication, Multiprocessing and Alternative Architectures, and Performance Enhancements. The overarching theme of the course is the hardware-software interface; in particular, focusing on what the programmer needs to know about the underlying hardware to achieve high performance for his or her code.

**COURSE RATIONALE**

Computing professionals should not regard the computer as a black box that executes programs by magic. This course serves students in two ways. First, for those who want to continue studying computer architecture, embedded systems, and other low-level aspects of computer systems, it lays the foundation experience needed to make the quantitative tradeoffs in more advanced courses meaningful. Second, for those students interested in other areas of computer science, it solidifies an intuition about why hardware is as it is and how software interacts with hardware.

**COURSE CONTENT**

1. Digital Logic and Digital Systems
   1. Combinational vs. sequential logic
2. Machine Level Representation of Data
   1. Signed and twos-complement representations
   2. floating-point systems
3. Assembly Level Machine Organization
   1. Instruction sets and types (data manipulation, control, I/O)
   2. Instruction formats, Addressing modes
   3. Assembly/machine language programming
   4. Subroutine call and return mechanisms
   5. I/O and interrupts
4. Processor Structure & function
   1. Control unit: instruction cycle stages
   2. Instruction pipelining
   3. Introduction to instruction-level parallelism (ILP)
   4. Control unit: hardwired realization vs. microprogrammed realization
5. Memory System Organization and Architecture
   1. Memory hierarchy: importance of temporal and spatial locality
   2. Main memory organization and operations
   3. Latency, cycle time, bandwidth, and interleaving
   4. Cache memories (address mapping, block size, replacement, and store policy)
   5. Virtual memory (page table, TLB)
6. Interfacing and Communication
   1. Buses: bus protocols, arbitration, direct-memory access (DMA)
   2. I/O fundamentals: handshaking, buffering, programmed I/O, interrupt driven I/O
   3. Interrupt structures: vectored and prioritized, interrupt acknowledgment
   4. External storage, physical organization, and drives
   5. RAID architectures
7. Multiprocessing and Alternative Architectures
   1. Introduction to SIMD vs. MIMD and the Flynn Taxonomy
   2. Shared memory multiprocessors/multicore organization
   3. Multiprocessor cache coherence
8. Performance Issues

**COURSE ASSESSMENT**

|  |  |
| --- | --- |
| **Method of Evaluation** | **Percentage of Grade** |
| **Coursework**  1 Assignment: 10%  1 Group Project: 10%  7 Mini-quizzes: 10%  Course work examination 2: 20% | 50 % |
| **Final Exam** | 50 % |

**TEACHING STRATEGIES**

The course will be taught via online lectures, tutorials, and virtual class group work. Students will be expected to work in groups as well as individually on course assignments and projects. Additionally, class presentations (via video) will be used to help students internalize the content.

**RESOURCES**

* Lecture notes - PowerPoint containing notes in slides will be made available.
* Stallings, William. Computer Organization & Architecture, Designing for Performance. Pearson, 11th edition.
* J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 5th Edition, Morgan Kaufmann Publishing Co., Menlo Park, CA. 2012

**COURSE CALENDAR: 39 hours** (3 hours per week)

|  |  |  |
| --- | --- | --- |
| Week | Topic | Objectives |
| 1 | Introduction  [\*Chapter 1]  \*Note that chapters refer to the Stallings text, however the topics can also be found in the second text and external links provided each week. | * Organization and Architecture * Structure and Function * History of Computers * Embedded Systems * ARM Architecture |
| 1 | Digital Logic – logic gates, combinational circuits (adder, decoder, multiplexer), sequential circuits (SR, D and JK flip-flops, parallel registers, binary counter), Karnaugh maps  [Chapter 12 – Digital Logic] | * Draw circuit diagrams given a Boolean function * Use a Truth table to derive the sum of products * Use Karnaugh Maps in minimizing a Boolean function * Discuss the operation of an adder, decoder, multiplexer, and comparator * Discuss the operations of a flip flop (SR, D, JK) * Discuss the operation of a parallel register, shift register, synchronous binary counter |
| 2,3 | Data Representation – integer representation, integer arithmetic, floating-point representation, floating point arithmetic  [Chapter 11 – Computer Arithmetic] | * Discuss the representations of integers: sign-magnitude, two’s complement * Perform two’s complement arithmetic * Use Booth’s Algorithm in the multiplication of integers in the two’s complement representation * Discuss the IEEE754 standard for representing floating point numbers * Discuss floating-point arithmetic and issues * Discuss the issues that arise in representations of integers and floating-point numbers |
| 4,5 | Instruction Set Architectures – design issues, types of operations, addressing modes,  a light introduction to MIPS  [Chapter 13 & 14 – Instruction Sets: Addressing Modes and Formats] | * Describe the machine instruction characteristics * Discuss the categories of machine operations * Discuss issues in the design of an instruction set * Trace a program written in a set of generic machine instructions * Define the various addressing modes * Determine the address of an operand given an addressing mode |
| 6,7 | Processor Structure and Function – processor organization, register organization, instruction cycle, instruction pipelining, RISC and CISC computers  [Chapter 16 – Processor Structure and Function  Chapter 17 – Reduced Instruction Set Computers] | * Describe the organization of a processor * Discuss the role of the major registers in the stages of an instruction cycle * Discuss the role of instruction pipelining * Determine the performance of a processor supporting instruction pipelining * Discuss the methods used to deal with the problems that arise during instruction pipelining * Discuss instruction pipelining on a RISC computer |
| 8 | Control unit – control unit operation, hardwired control unit, microprogrammed control unit, microinstructions  [Chapter 19 – Control Unit Operation and Microprogrammed Control] | * Discuss the operations of a control unit * Define the microoperations for each stage of the instruction cycle * Differentiate between horizontal and vertical microinstructions * Describe how a horizontal microinstruction is interpreted * Differentiate between a hardwired and a microprogrammed control unit * Describe the operations of a microprogrammed control unit |
| 9 | Cache Memory – cache memory principles, cache design, direct, associative, and set-associative mapping  [Chapter 5 - Cache Memory] | * Discuss cache memory principles * Discuss the issues in cache design * Differentiate between direct, associative, and set-associative mapping * Perform calculations involving cache mapping functions |
| 10 | I/O architecture – external devices, I/O modules, programmed I/O, interrupt I/O, Direct Memory Access, I/O Interrupt processing, the external I/O interface  [Chapter 8 – Input/Output  Chapter 7 – External Memory] | * *Discuss the organization and performance characteristics of external memory devices: magnetic disk, optical disk, RAID (student reading)* * Discuss need for an interface between the processor and the external devices * Discuss the major functions of an I/O Module * Differentiate between programmed I/O, interrupt driven I/O, and direct memory access * I/O interrupt processing |
| 11 | Multicore and parallel architectures  [Chapter 20 – Parallel Processing  Chapter 21 – Multicore Computers] | * Describe the trend of modern computer architectures towards multi-core and that parallelism is inherent in all hardware system * Discuss the concept of parallel processing beyond the classical Von Neumann model. * Describe alternative parallel architectures such as SIMD and MIMD |
| 12 | Performance Issues | Project |
| 13 | Review |  |