CENTRAL UNIVERSITY OF TECHNOLOGY, FREE STATE
SENTRALE UNIVERSITEIT VIR TEGNOLOGIE, VRYSTAAT
YUNIVESITHI E BOHARENG YA THEKENOLOJI, FOREISTATA

Department of Information Technology

INSTRUCTIONAL OFFERING	COMPUTER ARCHITECTURE IV
SUBJECT CODE	CRA42AB
EXAMINATION	3 HOUR WRITTEN PAPER
CREDITS	12
NQF LEVEL	7
TYPE OF LEARNING	VOCATIONAL TRAINING

UNIT	SPECIFIC OUTCOME	ASSESSMENT CRITERIA	CREDITS	NOTIONAL HOURS	WEEKS
1	Demonstrate an understanding of the major technological milestones in computer architecture. Demonstrate an understanding of technologies that made the various computer generations possible. Demonstrate an understanding of both the components in a computer system and how they work. Demonstrate an understanding of some common terminology, including common prefixes in the context of a fictitious computer advertisement.	Provide a historical overview of computing in general, pointing out the many milestones in the development of computing systems. Introduce the necessary terminology, the basic components in a computer system, the various logical levels of a computer system, and the von Neumann computer model.	1	10	1

2	Demonstrate an understanding of how computer represent values. Demonstrate an understanding of the different representation formats (different bases) used by the computer. Demonstrate an understanding of the three methods for representing signed numbers: signed magnitude, one's complement, and two's complement. Use the techniques for base conversion (Decimal to binary conversions, and binary to hexadecimal relationship) and memorize the smaller hexadecimal and binary numbers. Perform Binary arithmetic.	Describe the various ways in which computers can store and manipulate numbers and characters. The numbers bases and the typical numeric representation techniques, including one's complement, two's complement, and BCD are discussed, as well as Unicode character representations. Addition, subtraction, multiplication, and division are also addressed. Fixed and floating point representations are introduced too.	2	20	2
3	Demonstrate an understanding of the basic circuit configurations used to construct computer systems. Demonstrate an understanding of how the Boolean expressions are physically implemented. Demonstrate an understanding of how Boolean algebra relates to digital components. Demonstrate an understanding of the Combinational circuits, and sequential circuits.	Describe classic representation of digital logic and how it relates to Boolean algebra. Cover both combinational and sequential logic in detail. More complex circuits, such as buses and memory, are also included.	2	20	2

4	Demonstrate an understanding of computer architecture in general. Demonstrate an understanding of the basic architectural organization involved in program execution. Demonstrate an understanding of the syntax of language very close to the machine language (e.g. Assembly Language programming). Write simple programs using Assembly Language Programming.	Explain basic computer organization and introduces many fundamental concepts, including the fetch-decode-execute cycle, the data path, clocks and buses, register transfer notation, and of course, the CPU. A very simple architecture, MARIE, and its ISA are presented. MARIE exhibits the classical von Neumann design, and includes a program counter, an accumulator, an instruction register, 4096 bytes of memory, and two addressing modes. Assembly Language programming is introduced to reinforce the concepts of instruction format, instruction mode, data format, and control that are presented earlier. The primary objective in introducing assembly is to further the understanding of computer architecture in general.	3	30	3
5	Demonstrate an understanding of instruction formats (including big versus little endian, use of registers, and instruction length). Demonstrate an understanding of addressing modes (with considerable attention paid to various modes). Demonstrate an understanding of stack machines. Demonstrate an understanding of pipelining (the fetch-decode execute cycle)	Provide a closer look at the instruction set architectures, including instruction formats, instruction types, endianess, and addressing modes. Instruction level pipelining is introduced as well.	1	10	1

6	Demonstrate an understanding of the types of memory, the memory hierarchy, Cache memory, and Virtual memory. Demonstrate an understanding of how system memory (registers, cache, and main memory), online memory (hard disk), near line memory (optical disk), and offline memory (tapes and floppy disks) work together to provide acceptable performance at minimal cost.	Explain basic memory concepts, such as RAM and the various memory devices, and also addresses the more advanced concepts of the memory hierarchy, including cache memory and virtual memory.	1	10	1
7	Demonstrate an understanding of the importance and application of Amdahl's Law. Demonstrate an understanding of I/O architecture, and the differences among the I/O control methods. Demonstrate an understanding of different RAID levels, including the advantages and disadvantages of each.	Explain detailed overview of I/O fundamentals (including interrupt handling), bus communication and protocols, and typical external storage devices, such as magnetic and optical disks, as well as the various formats available for each. DMA, programmed I/O, and interrupts are covered as well. In addition, some various techniques for exchanging information between devices are introduced. RAID architectures are covered in detail.	1	10	1
8	Demonstrate an understanding of some important aspects of multiprocessor and multi-computer systems. Demonstrate an understanding of the motivation for RISC machines and how they differ with other traditional architecture (e.g. Von Neumann architecture). Demonstrate an understanding of the necessity of interconnection networks.	Provide an overview of alternative architectures that have merged in recent years. RISC, Flynn's Taxonomy, parallel processors, instruction-level parallelism, multiprocessors, interconnection networks, shared memory systems, cache coherence, memory models, superscalar machines, neural networks, systolic architectures, dataflow computers, and distributed architectures are covered.	1	10	1