**CPSC 250 Computer Systems and**

**Assembly Language Programming**

**Final Study Guide**

**Chapter 01 – (10% of total questions)**

1. Understand classes of computing applications and their characteristics (Section 1.1)
2. Be able to identify the “Eight Great Ideas in Computer Architecture” (Section 1.2)
3. Understand the idea of abstraction (Section 1.3)
   1. Review Figure 1.3 in the text
   2. Review Figure 1.4 in the text to understand the power of abstraction.
   3. Be able to identify the 5 classic components of computer organization – see Figure 1.5 in the text.
4. Understand and able to measure performance (Section 1.6)
   1. Throughput and Response Time (see example pg. 30 of text)
   2. Improving performance (see example pg. 34 of text)
   3. Understand the “Classic CPU Performance Equation” (see example on pg. 37 of text)
5. Understand what is meant by the term “Power Wall” (pg. 40)
6. Understand Parallelism (pg. 43 of text)

Questions is this area will most likely consist of short answers and simple calculations to determine/measure performance.

**Chapter 02 – (20% of total questions)**

1. Understand Operations of the Computer Hardware (Section 2.2)
   1. Be able to identify/understand most of the MIPS assembly language (see Figure 2.1 of the text). You **DO NOT** have to memorize all MIPS instructions but understand what the basic instructions do.
      1. Memory Operands (Section 2.3)
      2. Signed/Unsigned Numbers (section 2.4)
      3. Logical Operations (Section 2.6)
      4. Instructions for Making Decisions (Section 2.7)

Questions is this area will most likely consist of code snippets where you may have to convert from a higher-level language (i.e. C) to assembly, write a small code snippet to perform a specific task/function or determine the results of a particular assembly code snippet.

**Chapter 03 – (10% of total questions)**

1. Integer Addition/Subtraction (Section 3.2)
   1. Be able to add/subtract binary numbers
   2. Understand/Identify overflow/underflow
2. Integer Multiplication/Division (section 3.3/3.4)
   1. Be able to multiply binary numbers (see example on pg. 183 of text)
   2. Be able to divide binary numbers (see example on pg. 189 of text)
3. Floating Point (Section 3.5)
   1. Understand floating point representations
   2. Understand the difference between single precision vs. double precision.

Questions in this area will most likely consist of converting a binary or decimal number to a floating-point number or converting a floating-point number back to a binary/decimal number. To keep the calculations small only 8-bit numbers will be used.

**Chapter 04 – (30% of total questions)**

1. Understand the Basic MIPS Implementation (Section 4.1)
   1. Review the implementation overview (pg. 245/246 of text, slides 3/4 of text lectures)
   2. Understand multiplexing (slides 5/6 of text lectures)
2. Understand Logic Design Conventions (Section 4.2)
   1. Understand combinational elements (slide 8 of lectures)
   2. Understand sequential elements (slide 9 of lectures)
   3. Understand clocking methodology (See Figure 4.3 of text and slide 11 of lectures)
3. Building a Datapath (Section 4.3)
   1. Identify processor datapath elements (See Figure 4.5, 4.6, 4.7 of text and slide 13-20 of text lectures)
4. Overview of Pipelining (Section 4.5)
   1. Identify/describe the 5-stages of a MIPS pipeline (Slide 32 of text lectures)
   2. Understand the different pipelining hazards (pg. 277 – 281 of text, slides 37-43 of text lectures)
   3. Understand/Identify Data Hazards (Section 4.7, 4.8)
      1. Forwarding (see Figure 45.2 of text, slides 39-41)
      2. Review code scheduling to avoid pipeline stalls (slide 42 of text lectures)
   4. Understand/Identify Control Hazards (Section 4.8)
      1. Understand Branch Prediction (See Figure 4.61 of text, slides 44, 45 of text lectures)
   5. Understand Data Hazards in ALU Instructions (slide 67 of text notes)
      1. Identify dependencies for forwarding (slide 68 of text notes)
      2. Be able to detect the need to forward an instruction (slides 68,69 of text lectures)
   6. Understand Branch Hazards (slides 83, 84 of text lectures)
      1. Understand data hazards for branches (slides 87-89 of text lectures)

Questions for this section will most likely consist of short answers as well as pipeline diagrams depicting pipelines and pipeline hazards. You could be requested to identify various potential data/control hazards and propose forwarding or prediction to reduce/eliminate the hazards.

**Chapter 05 – (30% of total questions)**

1. Understand the memory hierarchy (slides 2-4 of text slides)
   1. Understand the principle of locality
   2. Understand the difference between spatial and temporal locality
2. Understand the different Memory Technologies and their uses (Section 5.2)
   1. DRAM, Flash, Disk Storage (slides 5-7, 11-14)
3. The Basics of Cache (Section 5.3)
   1. Understand Cache Memory, Direct Mapped Cache (slides 17- 25 of text lectures)
4. Measuring/Improving Cache Performance (Section 5.4)
   1. Understand Cache Misses using block size considerations (slides 27-29 of text lectures)
   2. Understand Cache Write-Though and Write-Back (Slides 30-31 of text lectures)
   3. Be able to calculate cache performance (slides 36-36 of text slides)
5. Understand Associate Caches
   1. Understand fully associative and n-way set associative caches (slides 40-45 of text lectures)

Questions for this section will most likely consist of short answers as well as calculations to determine the memory address of various cache architectures (i.e. direct mapped, n-way set associative or fully associative).