METRO STATE UNIVERSITY

ICS 232 Computer Organization & Architecture
Final Exam Review

You may use up to 10 pages of single-side notes and the textbook. You may not use a computer, phone, calculator or any other electronic equipment except to compile and run exam questions. You may not communicate with any other student via in-person or via electronic or any other means.

1. Numeric conversions:
   a. binary conversions - ones complement, twos complement
   b. Hex addition / subtraction
   c. Big vs Little Endian
2. ASCII / Unicode conversion
3. Intel Assembly

   a. Example 1:

```
 if (a == b)
      c = 1;
   else
      c = 2;

   MOV      EAX, A
   CMP      EAX, B
   JNE      else
   MOV      C, 1
   JMP      endif
else:
   MOV      C, 2
endif:

A  DWORD    0
B  DWORD    0
C  DWORD    0
```

4. Intel Assembly

What is the equivalent C code and what will be stored in var1

```
    MOV EAX, 1
    MOV EBX, 100
    MOV ECX, 101
    SUB EBX, ECX
    JG  next1
    INC EAX
next1:
    MOV var1, EAX        ; what is stored in var1

    int eax, ebx, ecx, var1;

    eax = 1;
    ebx = 100;
    ecx = 101;
    ebx = ebx - ecx;
    if (ebx <= 0)
       eax += 1;
    var1 = eax;       // var1 = 2
```

5. Cache: how lines are mapped from addresses, performance
6. Address translation page 391, figure 6.23
7. Disk
    a. Access = seek time + rotational delay + transfer time
    b. RAID = RAID 0 - optimal performance, no redundancy
       RAID 1 - mirror - great performance, total redundancy, high cost
8. CPU scheduling / Context Switching / Interrupts - FIFO, SJF, Priority, Round-Robin
9. I/O types - programmed, memory mapped, DMA, interrupts
10. RISC vs CISC differences - page 534, table 9.1
11. Intel 64, ARM 64, RISC-V basic architecture
    a. Stack management - EBP and ESP registers
    b. CISC vs RISC
    c. ISA of ARM, RISC-V (number of registers, instruction width, caching)
    d. ARM-64: Data and instruction cache, can run 32- and 64-bit programs, MMU manages memory mapping
12. Networking
    a. UDP vs TCP
    b. TCP sequence numbers and Windowing

   c. IP addresses
13. Java VM
    a. Bytecode
    b. Garbage collection
14. Quantum Computing
    a. Quantum mechanics and Qbit
    b. Decoherence
15. Compilers
    a. Optimization
    b. Static vs shared libraries

Examples:

```
-99 (10) to hex

99 / 2 = 49 r 1
49 / 2 = 24 r 1
24 / 2 = 12 r 0
12 / 2 = 6 r 0
6  / 2 = 3 r 0
3  / 2 = 1 r 1
1  / 2 = 0 r 1

0 1 1 0 0 0 1 1 = 63 (16)
1 0 0 1 1 1 0 0 = one complement
              +1
1 0 0 1 1 1 0 1 = 9D (16)


 1
 9A4C
+16B2
=====
 B0FE
 3210

11 * 16^3 + 0 * 16^2 + 15 * 16^1 + 14 * 16^0 = 45310 (10)
```

ICS 232 Computer Organization & Architecture
Final Exam Review

```
  9
 9A4C
-16B2
=====
 839A
```

Big Endian

```
00 00 B0 FE
```

Little Endian

```
FE B0 00 00
```

"789 a Σ

```
ASCII = 37 38 39 20 61 20 ??
Unicode =          0037 0038 0039 0020 0061 0020 03A3
Unicode (little) = 3700 3800 3900 2000 6100 2000 A303
```

int xx[3] = {1, 2, 3};

```
1              2              3
00 00 00 01    00 00 00 02    00 00 00 03 (big endian)
01 00 00 00    02 00 00 00    03 00 00 00 (little endian)
```

Cache:
Each addresses tag + directMap + offset
Offset bit = size of each cache lines
directMap size = size of cache

cache line = 16 bytes
cache size = 256 lines

address = 12345678_{16} = 12345 67 8
offset = 8_{16}
cache entry = 67_{16}
tag = 12345_{16}

METRO STATE UNIVERSITY

ICS 232 Computer Organization & Architecture
Final Exam Review

Virtual Address:
Page number + offset
offset size = size of each page frame

page size = 4096
virtual address space = 4096 pages
physical address space = 1024 pages

```
Page table
 0 - 01
 1 - 0D
 2 - 02
 3 - N/A
 4 - 04
```

Virtual Address = XXX YYY_{16}

001 010_{16} -> 0D 010_{16}

002 $0FF_{16}$ -> 02 $0FF_{16}$

003 100_{16} -> page fault

Memory Layout:
address 0:
- code
- global data and constants
- heap (grows to higher addresses)
- ...
- stack (grows to lower addresses)

address N (highest)