Computer Science 605.411

Module 9 Example Set 2

1. A memory system employs dynamic RAM that has a pre-charge time of 5ns and an access time of 25ns per read operation. Each read operation transfers 32 bits of data.

a) What is the minimum cycle time for this memory system?

The minimum cycle time = 30ns (5ns for pre-charging plus a 25ns access time).

b) What is the bandwidth for this memory system?

Each read operation transfers 4 bytes (32 bits) of data. Therefore, the bandwidth is 4 bytes/30ns = 1.333 x 10^8 bytes per second.

2. The null terminated character string "South Bridge" is stored in memory beginning at address 0x100C004. The first character in the string is "S", the second is "o", etc. The null character at the end contains eight 0 bits.

a) Assuming that big endian memory storage order is used, show (in hex) the contents of each of the following locations in this byte addressable memory:

Address	Contents
0x100C004	0x53="S"
0x100C005	0x6F="o"
0x100C006	0x75="u"
0x100C00B	0x72="r"

b) Assuming that little endian memory storage order is used, show (in hex) the contents of each of the following locations in this byte addressable memory:

Address	Contents
0x100C004	0x53="S"
0x100C007	0x74="t"
0x100C00A	0x42="B"
0x100C00F	0x65="e"

The string is stored from first character to last for both orders. The address of the string is the address of the leading (i.e., first) character in the string.

3. The contents of the memory bytes at locations 0x80000 through 0x80003 are as follows:

Address	Contents
0x80000	0xA9
0x80001	0x87
0x80002	0xD5
0x80003	0x43

Show, in hex, the contents of register $t0 produced by each of the instructions listed below assuming first that the byte addressable system employs big endian storage order and then that it employs little endian storage. Register $t1 contains the value 0x80000, **lw** is load word, **lh** is load half word, **lhu** is load half word unsigned, **lb** is load byte, and **lbu** is load byte unsigned.

	Big endian	Little endian
lw $t0,0($t1)	0xA987D543	0x43D587A9
lhu $t0,2($t1)	0x0000D543	0x000043D5
lb $t0,3($t1)	0x00000043	0x00000043
lbu $t0,0($t1)	0x000000A9	0x000000A9
lh $t0,0($t1)	0xFFFFA987	0xFFFF87A9
lh $t0,2($t1)	0xFFFFD543	0x000043D5

4. A memory system contains 16 modules that are organized so as to create a low order interleaved memory. Each memory module is 8 bits wides and the total storage capacity of 2 GB is provided collectively by the 16 modules. Recall that one GB is 2^{30} bytes.

a) How many bits would be required within the address for the module number?
Four bits would be required to specify module numbers 0 through 15. With low order interleaving, the module number resides in the low or rightmost bits.

b) What is the minimum number of bits required for the offset field within the address to allow any location within a memory module to be accessed?
Since there are 16 modules, the number of bytes within each module would be $(2*2^{30})/16 = 2^{27}$ hence 27 bits are required for the offset.

c) Write an instruction sequence that would fill each location within module 3 with all 1 bits.
The module number is 3, hence the address range for module 3 would have the binary pattern 0011 in the rightmost 4 bits and the 27 bits to the left of the module number field would range from 0 to 134217727. The following instruction sequence would work:

```
               li      $t2,255           # put all 1's into the low byte of $t2
               li      $t3, 134217727  # set count to fill all bytes within module
              li      $t4,3              # address of first byte in module 3
     loop:     sb      $t2,0($t4)         # fill next byte in module with all 1's
               addiu  $t3,$t3,-1          # decrement loop counter
               bge    $t3,$0,loop        # repeat until all bytes have been written
               addiu  $t4,$t4,16          # increment offset within address
```

Note that the code takes advantage of the branch delay slot to increment the offset field by adding 16 (which has the effect of leaving the module number fixed and changing only the offset field).

5. a) An interleaved memory system contains four memory modules each of which is 32-bits wide and each module has a separate MDR (memory data register). The CPU communicates with the memory over a 32-bit bus. For each instruction fetch, the appropriate memory module takes 8 clock cycles to obtain the instruction and place it into the MDR associated with the memory module. One clock cycle is required to transfer the contents of the MDR to the CPU. What would be the minimum number of cycles required for the CPU to fetch a sequence of 8 consecutive instructions from this memory system?

Since the instructions are consecutive, they would appear in adjacent memory words each of which is in a different memory module. The interleaved memory can read 4 instructions into the 4 separate MDRs in parallel. Hence it would take 8 cycles to place the first 4 instructions into the respective MDRs. However, since the bus is only 32 bits wide, only one instruction at a time could be transferred from a MDR to the CPU. So it would take 8 cycles to read the first 4 instructions into the MDR then 4 cycles to transfer the instructions (one after the other) to the CPU. While the instructions are being transferred from the MDRs to the CPU, the read of the next 4 instructions could be started in parallel with these transfers. Thus the first 4 cycles of the next read would overlap with the 4 MDR transfers leaving only 4 cycles remaining until the next 4 instructions are placed into the MDRs. Therefore the minimum total number of cycles needed to fetch all 8 consecutive instructions would be 8 + 4 + 4 + 4 = 20 cycles (8 cycles to read the first 4 instructions, 4 cycles to transfer the MDR contents and start the second read, 4 cycles to complete the second read and 4 cycles to transfer the MDR contents to the CPU) .

b) A 32-bit wide memory module requires 8 clock cycles to read out an instruction into its MDR and 1 cycle to transfer the contents of the MDR to the CPU over a 32-bit bus. What is the minimum number of clock cycles required for the CPU to fetch a sequence of 8 consecutive instructions from this single memory module?

The single memory module can only read one instruction at a time. Hence the first instruction would require 9 cycles to reach the CPU (8 cycles to fill the MDR and 1 cycle to transfer the MDR contents to the CPU). While each instruction is on its way to the CPU, the read of the next instruction could be in progress. Hence each instruction after the first would arrive at the CPU 8 cycles following the preceding instruction. Therefore the minimum total number of cycles required to fetch all 8 consecutive instructions would be 9 + 7*8 = 65 cycles.

6. Thirty-two bit addresses are used to access a byte addressable data memory with a storage capacity of 256 MB. The memory is implemented using some number of memory modules each of which has a width of 32 and a depth of 1 M (i.e., 1048576).

a) How many memory modules are required for this data memory?

A cell within a module is 32 bits (i.e., 4 bytes) wide. Hence each module contains 4*1048576 bytes (4 MB). So a total of 256 MB / 4 MB = 64 modules are required.

b) Explain how a 32-bit address would be used to read a single byte from this data memory.

The low order 2 bits within the address would be used as the offset within the 32-bit memory cell selected by the remaining address bits. Each read would obtain the 4 bytes contained in the selected 32-bit cell and the low 2 bits would indicate which byte from the group of 4 to use.

c) What is the address of the next to last byte in module 33 (counting from 0)?

Since each module has a depth of 1048576, a 20-bit offset would be required to indicate which cell within module 33 to select. The low order 2 address bits indicate the byte offset within the selected cell. The module number would be indicated by the next 6 bits to the left of the cell offset field. Hence the address format would be:

Four 0 bits	6-bit module number	20-bit cell offset within module	2-bit byte offset within selected cell

Since the data memory is 256 MB in size, the corresponding address range is 0x00000000 to 0x0FFFFFFF. So the leftmost 4 address bits would always be 0.

The next to last byte within each 4-byte cell has offset 2. The final cell in each module would correspond to the maximum offset (0xFFFFF). So the next to last byte within a module would be byte 2 within the final cell which has the offset 0xFFFFF. Hence the address of the next to last byte in module 33=0x21 = binary 100001 is:

00001000011111111111111111111110 = 0x087FFFFE