Exam Review Problems

1. The IEEE single-precision floating-point format is composed of a 1-bit sign, an 8-bit biased exponent with a bias of 127 and a 23-bit fraction.

(a) Determine the values of the IEEE single-precision floating-point format numbers X and Y, which are specified in hex as follows:

```
X = 00700000
Y = 00480000
```

Express each value as a sum of powers of 2.

(b) Determine the IEEE single-precision floating-point representation (specified in hex) of the following quantity:

```
-(8X + 16Y)
```

2. Here is a series of address references given as word addresses: 7, 8, 10, 14, 22, 5, 8, 14, 7, 9, 10, 22, 5, 2, 1. Show the hits and misses and the final cache contents for a two-way set-associative cache with one-word blocks and a total size of 8 words. Assume that the cache is initially empty and that LRU replacement is used.

3. On a certain computer, instructions in Class A have a CPI of 1, instructions in Class B have a CPI of 3 and instructions in Class C have a CPI of 4. Consider the following two code sequences: Code sequence 1 contains 3 instructions in Class A, 1 instruction in Class B and 1 instruction in Class C. Code sequence 2 contains 2 instructions in Class A, 3 instructions in Class B and 5 instructions in Class C.

(a) Determine the number of instructions executed in each sequence.

(b) Determine the number of CPU clock cycles required for each sequence.

(c) Determine the CPI for each sequence.

4. Consider the following parameter values: memory is byte-addressable, the virtual address is 44 bits, the page size is 128K bytes and each page table entry is 24 bits. Determine the number of bits in the virtual page number field, the number of bits in the page offset field, the number of page table entries and the total size (in Mbytes) of the page table.

5. Consider the following portion of a MIPS64 assembly language program:

```
XOR  R24, R11, R12  ;    instruction 1
DADD R25, R12, R24  ;    instruction 2
DSUB R19, R18, R17  ;    instruction 3
AND  R25, R16, R19  ;    instruction 4
OR   R12, R16, R17  ;    instruction 5
```

(a) Identify any flow dependences, anti-dependences and output dependences that exist amongst these instructions. (In each case, specify the register that is involved.)

(b) Suppose that instruction 1 is in the instruction fetch (IF) stage of a 5-stage pipelined MIPS64 implementation (with forwarding) during cycle 51. Determine which cycles instructions 1, 2, 3, 4 and 5 will be in the write back (WB) stage of the pipeline. (Include a table that shows the instructions moving through the pipeline as part of your answer.)

6. The IEEE double-precision floating-point format is composed of a 1-bit sign, an 11-bit biased exponent with a bias of 1023 and a 52-bit fraction.

(a) Determine the decimal value of the following IEEE double-precision floating-point format number, specified in hex:

`409ffc0000000000`

(b) Determine the IEEE double-precision floating-point representation (specified in hex) of the following decimal value:

-1023

7. A file called `easy.v` contains the following:

```
module easy;

reg  [7:0] a, b, r, s, t, u, v, w;

initial repeat (5) begin
    a = $random;
    b = $random;
    r = {a[6:3], b[4:1]};
    s = ~r;
    t = r + s;
    u = s^t;
    v = u - r;
    w = t*(v + 1);

    #10 $display($time, " %d", w);
end

endmodule
```

If the command:

```
verilog easy.v
```

is executed, give the output that will be produced by the `$display` statement. Be sure to clearly show how you obtained your answer by including all of the intermediate calculations.

8. A file called `newshuffle.v` contains the following:

```
module newshuffle;

integer i;
reg  t;
reg  [5:0] a, b, c, d, r;

initial begin
    for (i = 0; i < 4; i = i + 1) begin
        a = mix0(i);
        b = mix1(i);
        c = a << 2;
        d = b >> 3;
        t = ~(|d[5:0]);
        case (t)
            1'd0 : r = c ^ d;
            1'd1 : r = ~(c ^ d);
        endcase

        #10 $display($time, " %d %d %d %d %d", a, b, c, d, r);
    end
end

function [5:0] mix0;
  input [1:0] left;

  mix0[5:0] = { left, {2{left[0]}}, ~left };

endfunction

function [5:0] mix1;
  input [1:0] left;

  mix1[5:0] = { left, {2{left[1]}}, ~left };

endfunction


endmodule
```

If the command:

```
verilog newshuffle.v
```

is executed, give the output that will be produced by the `$display` statement. Be sure to clearly show how you obtained your answer!