Additional 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. 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. Consider the IEEE single-precision floating-point number:

C1CF0000

(a) (10 points) Determine the decimal value of this number.

(b) (10 points) Determine the IEEE double-precision representation of this number, in hex format.

2. A circuit is required which will add 3 5-bit unsigned numbers using a carry-save adder (CSA) and a ripple carry adder.

(a) Draw the diagram for this circuit as an interconnection of full adders. (In other words, show each of the individual full adders and indicate how they are connected together.)

(b) Suppose that the decimal computation 11 + 21 + 31 is to be performed using the circuit of part (a). Determine the sum and carry vectors which are output from the CSA and then perform the binary addition of those two vectors to show that the correct binary result is obtained at the output of the ripple carry adder.

3. A file called **familiar.v** contains the following:

```
module SlightlyDifferent;

reg      [31:0] P, Q;
reg      [63:0] R;
reg             PS, QS, RS;
reg       [7:0] PE, QE;
reg      [22:0] PF, QF;
reg      [10:0] RE;
reg      [51:0] RF;
reg       [3:0] m1, m2;
reg       [7:0] m3, m4;
reg       [6:0] f;
integer         pval, qval, rval;

initial begin

P = 32'hbfc00000;
Q = 32'h40100000;
PS = P[31]; PE = P[30:23]; PF = P[22:0];
QS = Q[31]; QE = Q[30:23]; QF = Q[22:0];

m1 = {1'b1, PF[22:20]};
case (PE)
   8'd128:  pval = m1*2;
   8'd129:  pval = m1*4;
   8'd130:  pval = m1*8;
   default: pval = m1;
endcase
if (PS == 1) pval = (-1)*pval;

m2 = {1'b1, QF[22:20]};
case (QE)
   8'd128:  qval = m2*2;
   8'd129:  qval = m2*4;
   8'd130:  qval = m2*8;
   default: qval = m2;
endcase
if (QS == 1) qval = (-1)*qval;

m3 = m1*m2;
if (m3[7] == 1) begin
   f = m3[6:0];
   RE = 1;
end
else begin
   f = {m3[5:0], 1'b0};
   RE = 0;
end
```

```
RS = 0;
RE = RE + PE + QE + 769;
RF = {f, 45'b0};
R = {RS, RE, RF};
m4 = {1'b1, f};
case (RE)
   11'd1024: rval = m4*2;
   11'd1025: rval = m4*4;
   11'd1026: rval = m4*8;
   default:  rval = m4;
endcase

$display("%b , %b , %b , %b", m1, m2, m3, m4);
$display("%d , %d , %d", pval, qval, rval);
$display("%h", R);

end

endmodule
```

If the file **familiar.v** is simulated using Verilog, give the output that will be produced by the **$display** statements. Be sure to clearly show how you obtained your answer!

4. Let X and Y be the following two signed 10-bit numbers:

X = 1111111100 , Y = 1110101011

The Booth encoding procedure discussed in class is applied to Y and the product P = XY is calculated using the following equation:

$$P = \sum_{\substack{k=0 \\ (k\ even)}}^{8} z_k X 2^k$$

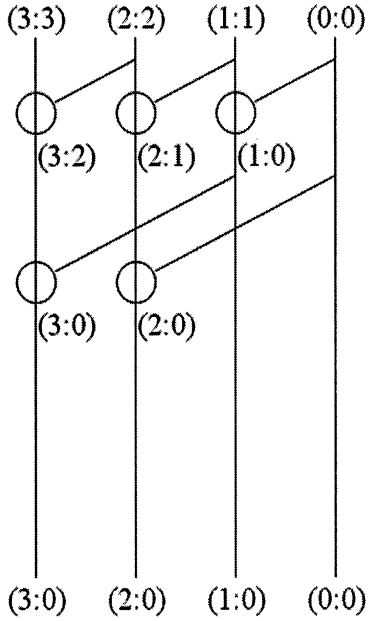
Determine the value of each of the coefficients z_k. Then, using decimal compuations, write out each of the terms in the summation and add them up to determine the product. Finally, check your answer by calculating the product directly (again working in decimal) using the equation P = XY.

5. The circuit below shows a certain 4-bit parallel prefix carry tree, where each circle is an instance of the "o" operator, which is defined as follows:

$$(P_{i:m}, G_{i:m}) \circ (P_{m-1:j}, G_{m-1:j}) = (P_{i:m}P_{m-1:j}, G_{i:m} + P_{i:m}G_{m-1:j})$$
$$= (P_{i:j}, G_{i:j})$$

The inputs at the top of the circuit are the propagate signals $P_{i:i} = P_i = A_i + B_i$ and generate signals $G_{i:i} = G_i = A_iB_i$ at each individual bit position i, for i = 0, 1, 2 and 3.

This circuit is to be used for the addition of the two 4-bit unsigned binary numbers 0110 and 1101, where the carry into the LSB is $C_0 = 0$. Determine the values of $P_{i:j}$ and $G_{i:j}$ at the output of each of the five "o" operators for this specific addition calculation and then show how those values are used to calculate the correct sum bits S_0, S_1, S_2, S_3 and the final carry out C_4.



6. Consider the following portion of an assembly language program for some processor, where the registers are indicated by `ri`, for some integer `i`, and where the meanings of the instructions are given in parentheses:

```
Instruction I1:     add r4, r5, r6 (r4 = r5 + r6)
Instruction I2:     sub r6, r7, r5 (r6 = r7 - r5)
Instruction I3:     or  r4, r6, r4 (r4 = r6 or r4)
```

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

7. A computer with a virtual memory system has the following parameter values: memory is byte-addressable, the virtual address is 40 bits, the page size is 128K bytes and each page table entry is 48 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 of the page table.

8. Here is a series of address references given as word addresses: 4, 5, 12, 10, 4, 20, 21, 13, 2, 5, 20, 18, 2, 4, 10, 21. Show the hits and misses and the final cache contents for a fully associative cache with one-word blocks and a total size of 8 words. Assume that LRU replacement is used.

9. A program has a branch statement. When the program is executed, the outcomes of the branch are as follows (T = taken, N = not taken): T, N, T, T, N, N, T, N, T, T. List the predictions and give the prediction accuracy (expressed as a percentage) for each of the following branch predictors:

(a) 1-bit predictor, initialized to predict not taken
(b) 2-bit predictor, initialized to weakly predict not taken

10. Consider a 2 GHz eight-way multiple issue processor.

(a) Determine the peak execution rate in instructions per second.

(b) Determine the best case CPI.

(c) Determine the best case IPC.

(d) Assuming that each pipeline has 5 stages, determine the maximum number of instructions in execution at any given time.