Assignment 3 – MECHTRON 3TB4

Q1

```
module Q1(
    input clk,
    input in,
    input reset,
    output reg out
    );

reg [3:0] state;   // init state is 4'b101

always @(posedge clk or posedge reset)
begin
    if(reset == 1'b1)
    begin
        out <= 1'b0;
        state <= 4'b101;
    end
    else
    begin
        case(state)
            4'b1110:    begin                   // 1 zero
                out <= 1'b0;
                if(in == 1'b0) state <= 4'b1100;
                else state <= 4'b1;
            end

            4'b1100:    begin                   // 2 zeros
                out <= 1'b0;
                if(in == 1'b0) state <= 4'b1000;
                else state <= 4'b1;
            end

            4'b1000:    begin                   // 3 zeros
                out <= 1'b0;
                if(in == 1'b0) state <= 4'b0;
                else state <= 4'b1;
            end

            4'b0:   begin                       // 4 zeros
                out <= 1'b1;
                if(in == 1'b0) state <= 4'b0;
                else state <= 4'b1;
            end

            4'b1:   begin                       // 1 one
                out <= 1'b0;
                if(in == 1'b0) state <= 4'b1110;
                else state <= 4'b11;
            end

            4'b11:  begin                       // 2 ones
                out <= 1'b0;
```

```
                if(in == 1'b0) state <= 4'b1110;
                else state <= 4'b111;

            end

            4'b111: begin                       // 3 ones
                out <= 1'b0;
                if(in == 1'b0) state <= 4'b1110;
                else state <= 4'b1111;

            end

            4'b1111:    begin                   // 4 ones
                out <= 1'b1;
                if(in == 1'b0) state <= 4'b1110;
                else state <= 4'b1111;

            end

            default:    begin                   // init
                out <= 1'b0;
                if(in == 1'b0) state <= 4'b1110;
                else state <= 4'b1;

            end
        endcase
    end
end
endmodule
```

Q2

```
% ------- FIXED POINT Q1.7 ----------
% product is Q1.7
F = fimath('ProductMode','SpecifyPrecision','ProductWordLength',8,'ProductFractionLength',7);

y_fix = sfi(0,8,7); % initial state of 0
gain_fix = sfi(-0.982,8,7);

res_fix = zeros(100,2);
res_fix(:,1) = 1:100;

for i=1:100
    res_fix(i,2) = y_fix;
    y_fix = mpy(F,gain_fix,y_fix)+sfi(1,8,7);
end

figure(1);
plot(res_fix(:,1),res_fix(:,2));
title('fixed point step response');

% -------- FLOATING POINT ---------
y_float = 0.0;
gain_float = -0.982;

res_float = zeros(100,2);
res_float(:,1) = 1:100;

for i=1:100
    res_float(i,2) = y_float;
    y_float = gain_float*y_float+1.0;
end
figure(2);
plot(res_float(:,1),res_float(:,2));
title('floating point step response');


% -------- DIFFERENCE ------------

res_diff = zeros(100,2);
res_diff(:,1) = 1:100;

res_diff(:,2) = res_float(:,2) - res_fix(:,2);

figure(3);
plot(res_diff(:,1),res_diff(:,2));
title('difference');
```

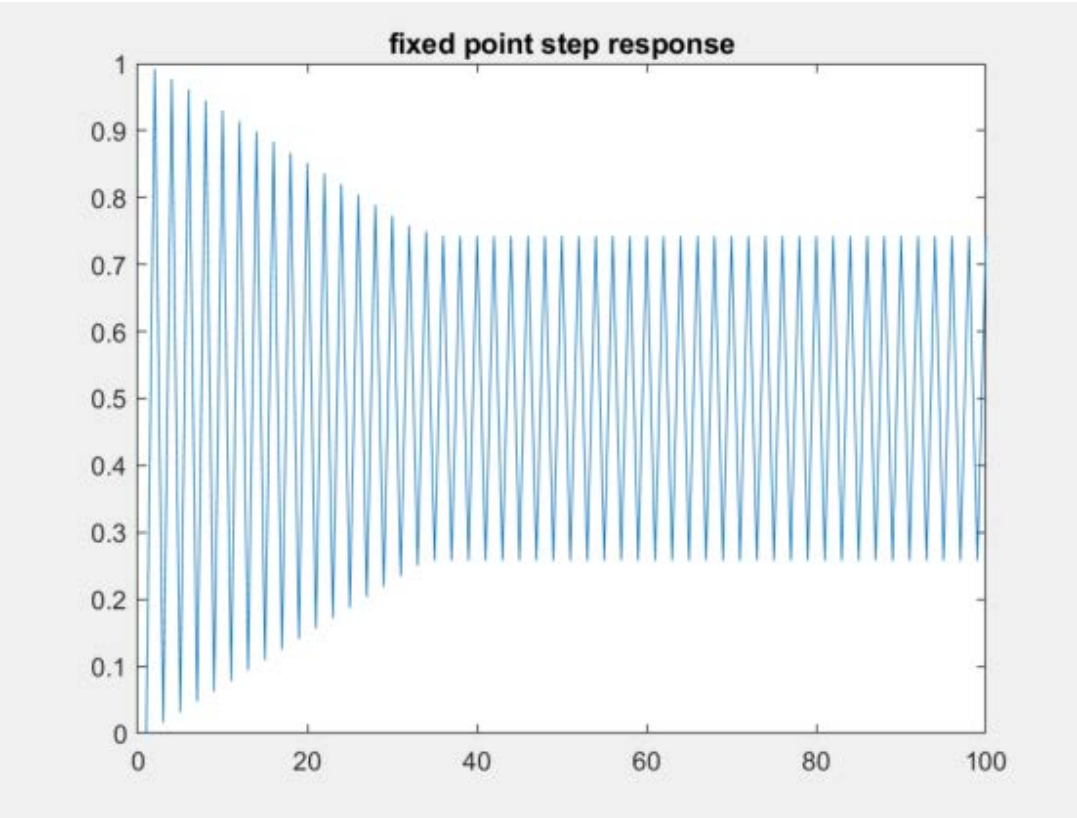

Figure 1: step response of fixed point system

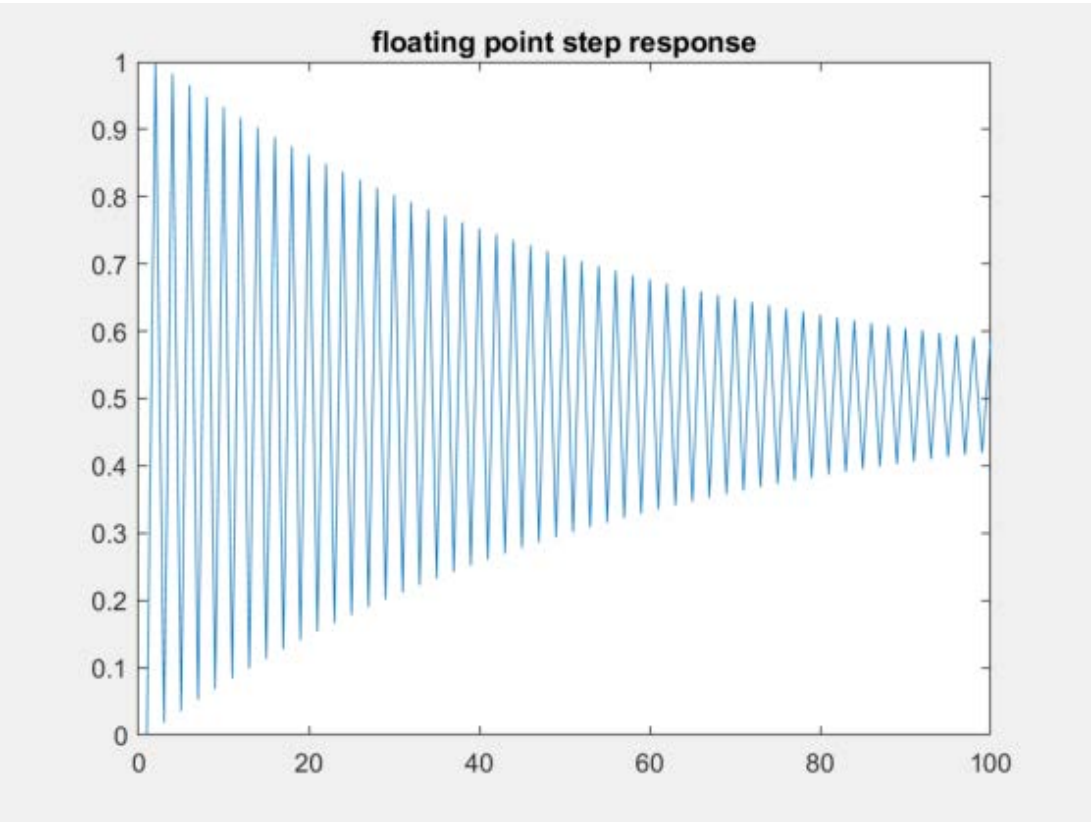

Figure 2: step response of floating point system

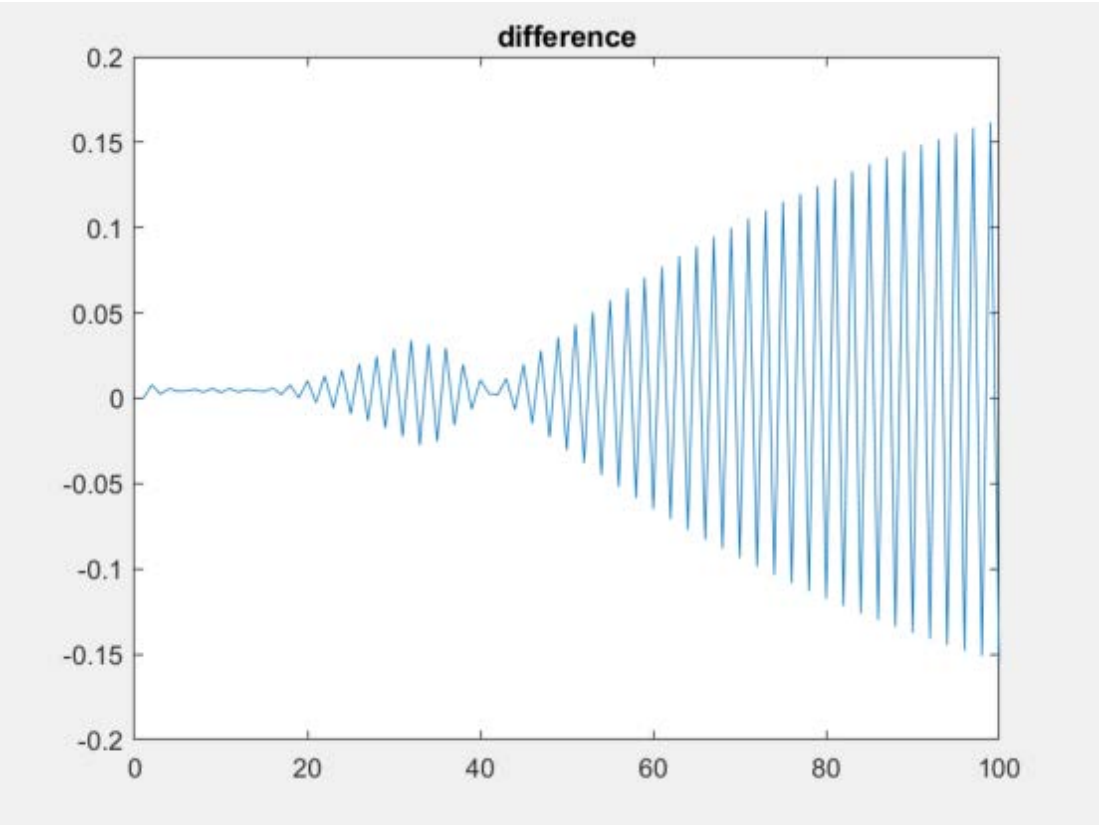

Figure 3: difference between two responses