# Project: Fixed Point Division

Mahendra Khinchi

## Solution:

**Mathematical Part:**

### 0.1 Initial Setup

$$
\begin{aligned}
N &= 8 \\
\text{scale\_factor} &= 2^8 = 256 \\
D &= 23 \quad \text{(denominator)} \\
\text{num} &= 15 \quad \text{(numerator)} \\
D_{\text{fixed}} &= \text{round}(23 \times 256) = 5888 \\
x_0 &= \frac{1}{17} \approx 0.0588235 \\
x_{\text{fixed}} &= \text{round}(0.0588235 \times 256) = 15
\end{aligned}
$$

### 0.2 Newton-Raphson Iterations

**Formula:**

$$
\begin{aligned}
x_{n+1} &= x_n \times (2 - D \times x_n) \\
x_{n+1}^{\text{fixed}} &= \left( x_n \times \left( 2 \times \text{scale\_factor} - \frac{x_n \times D_{\text{fixed}}}{2^N} \right) \right) \div 2^N
\end{aligned}
$$

**Iterations:**

$$
\begin{aligned}
x_1 &= (15 \times (512 - 345)) \div 256 = 9 \\
x_2 &= (9 \times (512 - 207)) \div 256 = 10 \\
x_3 &= (10 \times (512 - 230)) \div 256 = 11 \\
x_4 = x_5 &= (11 \times (512 - 253)) \div 256 = 11
\end{aligned}
$$

**Final Calculation:**

$$
\begin{aligned}
\text{result\_fixed} &= \text{num} \times x_{\text{fixed}} = 15 \times 11 = 165 \\
\text{result\_floating} &= \frac{165}{256} \approx 0.64453125
\end{aligned}
$$

# 1 MATLAB Implementation

```
N = 8;
scale_factor = 2^N;
D = 23; % Denominator
num = 15; % Numerator

% Convert to fixed-point
D_fixed = round(D * scale_factor);  % Convert denominator to fixed
    point
x0 = 1 / 17;
x_fixed = round(x0 * scale_factor);  % Convert initial guess 1/D to
     fixed point

% Newton-Raphson Iteration in Fixed-Point using bit shifts
for i = 1:5
    x_fixed = bitshift(x_fixed * (2 * scale_factor - bitshift(
        x_fixed * D_fixed, -N)), -N);
end

% Compute 15/23 in fixed point using bit shifts
result_fixed = num * x_fixed;

% Convert to floating point for verification
result_floating = result_fixed / scale_factor;

% Display Results
disp(['Fixed-Point Result: ', num2str(result_fixed)]);
disp(['Floating-Point Result: ', num2str(result_floating)]);
```

# 2 Verilog Implementation

```
module fixed_point_division;
    parameter N = 8;
    parameter SCALE = 1 << N;
    parameter D = 23;
    parameter NUM = 15;

    reg [16:0] D_fixed;
    reg [16:0] x_fixed;
    reg [16:0] temp;
    integer i;

    initial begin
        D_fixed = D * SCALE;  // Fixed-point representation of D
        x_fixed = (SCALE / 17); // Initial approximation of 1/D

        for (i = 0; i < 5; i = i + 1) begin
            temp = x_fixed * (2 * SCALE - ((x_fixed * D_fixed) >> N
                ));
            x_fixed = temp >> N;
        end

        // Multiply by NUM and compute the final result
```

```
        temp = NUM * x_fixed;

        // Display results
        $display("D_fixed: %d", D_fixed);
        $display("Final x_fixed: %d", x_fixed);
        $display("Result Fixed: %d", temp);
        $display("Result Floating: %f", temp * 1.0 / SCALE);

        $stop;
    end
endmodule
```