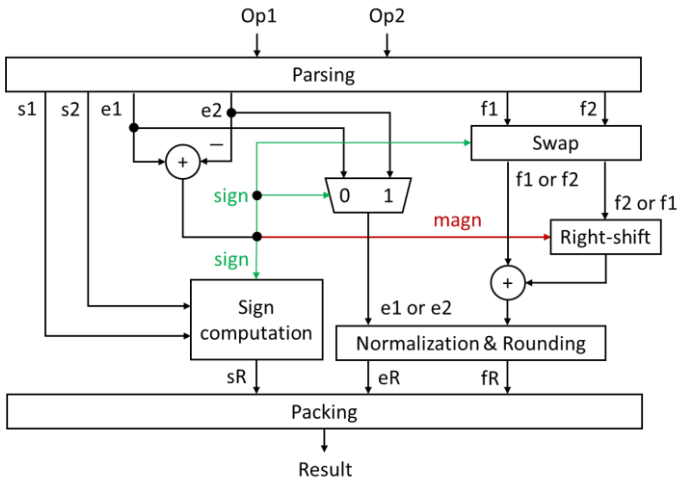
B EE 427 PA #4
Due: 03/15/2023 3:30pm

Note: You should submit your **Verilog source code** and **test bench code** to Canvas. No report and no demo for this PA (I'll run your hardware design on my side).

The following block diagram shows the architecture of a floating-point adder. Op1, Op2, and Result are floating-point numbers, and s, e, and f represent the sign, exponent, and fraction components, respectively.



Based on this architecture, implement a **combinational circuit of a single-precision floating-point adder** on the DE1-SoC board. To make this PA simple, assume that Op1 and Op2 have the same sign. Perform the following steps:

1. Create a new Quartus project using the system builder program. You'll use switches, buttons, LEDs, and 7-segment displays.
2. Generate 32-bit Op1 and Op2 signals using SW8-0 and KEY3-0 as follows:
   a. Op1[31:24] ← SW7-0 w/ SW8=0 & KEY3
   b. Op1[23:16] ← SW7-0 w/ SW8=0 & KEY2
   c. Op1[15:8] ← SW7-0 w/ SW8=0 & KEY1
   d. Op1[7:0] ← SW7-0 w/ SW8=0 & KEY0
   e. Op2[31:24] ← SW7-0 w/ SW8=1 & KEY3
   f. Op2[23:16] ← SW7-0 w/ SW8=1 & KEY2
   g. Op2[15:8] ← SW7-0 w/ SW8=1 & KEY1
   h. Op2[7:0] ← SW7-0 w/ SW8=1 & KEY0
3. Connect Result to LEDR8-0 and HEX5-0 as follows:
   a. LEDR8 ← Result[31] // sign
   b. LEDR7-0 ← Result[30:23] // exponent
   c. HEX5 (in hex) ← {hidden bit 1, Result[22:20]}
   d. HEX4 (in hex) ← Result[19:16]
   e. HEX3 (in hex) ← Result[15:12]
   f. HEX2 (in hex) ← Result[11:8]
   g. HEX1 (in hex) ← Result[7:4]
   h. HEX0 (in hex) ← Result[3:0]
4. Include the Verilog code in your project and compile it.
5. Simulate the hardware design using your test bench to verify the functionality.
6. Verify that your hardware design works properly on the board by observing the lights.

Supplement: Example of generating an operand using SW and KEY

```
// The code below is synthesized into latches (memory elements but not sensitive to a clock signal)
// KEY of DE1-SoC is an active-low signal (when pressed, its value is set to 0)
always @(*) begin
    if (KEY[3] == 0) op_a[15:8] = SW[7:0];    // There is no definition for the case where if KEY[3] == 1
                                              // → latches are created
                                              // The op_a[15:8] value is maintained when KEY[3] == 1
end
```

Reference

[1] IEEE-754 Floating Point Converter, https://www.h-schmidt.net/FloatConverter/IEEE754.html