# Design of IEEE-754 Single-Precision Floating-Point Fused Multiply-Add (FMA)

Handout: 2024/10/15

Due: 2024/11/05

1. (Fused Multiply Add, FMA)
   Given three IEEE 754 single-precision floating-point numbers A, B, C. the fused multiply-add (FMA) performs the computation of D= A*B+C. Design an arithmetic unit that computes the FMA. Note that your design should perform the alignment/normalization only once for the combined floating-point FMA. Assume the tuples of (sign, exponent, and mantissa) for the five numbers: A, B, C, P=A*B, and D are (sa, ea, ma), (sb, eb, mb), (sc, ec, mc), (sp, ep, mp), and (sd, ed, md) respectively. Note that ep=ea+eb.
2. (Alignment for C Only)
   After extracting mantissa and exponents from A, B, C, perform the alignment shift for mc at the same time when computing the product of ma*mb, so that the total latency of FMA could be reduced. Note that alignment for mc might be right-shift (ep-ec>0), or left-shift (ep-ec<0). It You only need to consider -25 <= ep-ec <= 47. Why? Let mc' denote the shifted mc after alignment shift. mc' is Q28.48 format. Why? Hint: If ep – ec = +w, mc is alignment-right-shift uy w bits. If ep – ec = -w, mc is alignment-left-shift by w bits.
3. (Range/Precision Analysis)
   The format of the magnitude of the mantissa for A, B, C is Q1.23 with the hidden integer bit of 1. But after converting the sign-magnitude representation to 2's complement format, the new signed mantissa format for A, B, C become Q2.23 with two integer bits because the range of signed mantissa (ma, mb, mc) is either between 1 and 2, or between -2 and -1. Thus, the range of the mantissa mp for P=A*B is either between 1 and 4, or between -4 and -1. The precision of the mantissa mp has ulp=$2^{-46}$. The format of the subsequent signed fixed-point addition is Q27.47. Why ?
4. (Special Cases)
   Add a control unit to consider the special numbers (zero, infinity, NaN) and sub-normal format for inputs A, B, C. For example, the output D=A in case A=0 or B=0. Also you need to detect overflow/under flow if the exponent of the output is > 127 or < -149 (sub-normal numbers).
5. (Non-Pipelined Implementation)
   First, design a non-pipelined (i.e., single-cycle) version for the FMA. That is, all the hardware is pure combinational logic without use of any flip-flop. Write a testbench to verify your register transfer level (RTL) design. Name the non-pipelined design as FLP_FMA.
6. (Logic Synthesis of FLP_FMA)
   Use Synopsys Design Compiler to synthesize the non-pipelined RTL design to gate-level netlist, and verify the gate-level design again. Note that you should synthesize your design with at least three different constraints: *delay-optimization*, *area-optimization*, and *in-between* (with reasonable delay constraint). In general, the curve of area versus delay looks look like a reciprocal curve.
7. (Area/Timing Reports of FLP_FMA)
   What is the total area of the non-pipelined FMA for each synthesis (with a particular constraint)? What is the critical path delay of the non-pipelined FMA for each synthesis (with a particular constraint) ? Note that you have at least three results, one for each synthesis with the aforementioned constraints.
8. (Pipelined Implementation)
   Then, design a pipelined FMA by partitioning it into several pipelined stages. The following lists the suggested pipelined stages:
   1st stage: unpack, alignment, ma*mb
   2nd stage: fixed-point signed addition

3rd stage: normalization
4th stage: rounding, pack

Note that each pipelined stage can be modelled as a separate Verilog module so that you can easily identify the critical path delay of each pipeline stage. Name the 4-stage pipelined design as FLP_FMA_4.

9. (Logic Synthesis of FLP_FMA_4)
   Use Synopsys Design Compiler to synthesize the pipelined RTL design to gate-level netlist, and verify the gate-level design again. Note that you should synthesize your design with at least three different constraints: delay-optimization, area-optimization, and in-between (with reasonable delay constraint).
10. (Area/Timing Reports of FLP_FMA_4)
    What is the total area of the pipelined FMA design? What is the critical path delays of each pipelined stage in the pipelined FMA? What is the critical path delay of the entire pipelined FMA? Which pipelined stage is the critical stage, i.e., with the longest delay? Note that you should have at least three results, one for each different synthesis constraint.
11. (Comparison with FXP_FMA)
    Compare the area and delay of the floating-point FMA design with the fixed-point multiply-add design. You might refer to the following RTL code segment for fixed-point multiply-add. Name the fixed-point multiply-add design as FXP_FMA.

```
wire signed [31:0] a, b, c, d, p;
wire signed [63:0] dp;

assign dp = a * b + {32{c[31]}, c}; // sign-extend c before addition with a*b
assign d = dp[63:32];   // take the 32 most significant bits (MSB) as the final output
```

12. (Summary Table)
13. The following table summarizes the synthesis results for the aforementioned designs. Fill in the table with the synthesis results.
    ??? add the table from slide 4 of the ALU_HW2 ppt file

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| FXP_FMA | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| FLP_FMA | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| FLP_FMA_4 | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

14. (P&R of FLP_FMA_4)
    Perform placement-and-routing (P&R) for the FLP_FMA_4 design. Mark the four pipelined stages in the physical layout view.
15. (References)
    Some related papers are listed below for your reference.
16. You have several weeks to do this entire homework. But you need to submit intermediate reports every week. The weekly reports show your progress in each week. The complete report dues as shown above. The suggested progress report for each week is as follows.
    Week 1: Draw a block diagram of overall architecture and identify the major components in the architecture such as unpack, multiply, alignment, add, normalize, round, unpack. Give detailed description for each of the major components. Design and verify some of the components.

Week 2: Design and verify the rest of the components. Then, synthesize the non-pipelined FMA and verify the gate-level netlists with various area/delay constraints

Week 3: Design and Verify the pipelined FMA

17. Upload your report to NSYSU course website with proper file names showing your student ID and homework number. Example: ARITH HW2_1 M1130400xx 王小明. The attached compressed file should include all HDL codes, results of simulation, results of synthesis, final report, and all the supporting documents.
18. The grading criterion are as follows:

    **(40%)** RTL code, testbench, simulation results, and synthesis reports for the non-pipelined FMA**.**

    **(40%)** RTL code, testbench, simulation results, and synthesis reports of the pipelined FMA

    **(20%)** Final report which contains the figures for the overall architectures, explanations of your designs and verification strategy, and your comments.

References:

[1] T. Lang and J.D. Bruguera , "Floating-Point Multiply-Add Fused with Reduced Latency", IEEE Trans. Computers, vol. 53, no. 8, Aug. 2004.

[2] "Bridge Floating-Point Fused Multiply-Add Design", IEEE Trans. VLSI Systems, vol. 16, no. 12, Sept. 2008.