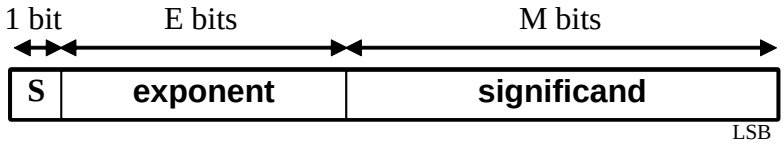
SIMPLE RISC PROCESSOR
FLOATING-POINT UNIT
v.3.0
05.01.2022

1. Requirements

Floating-point execution unit processes all floating-point instructions from the Simple RISC Processor ISA, namely ADDF and SUBF instructions.
The operands and result floating-point format comply with the floating-point data format defined in the Simple RISC Processor ISA.
The data width, the width of the significand and the width of the exponent are parameterized.

2. Simple RISC Processor Floating-Point Format

The Simple RISC Processor floating-point format is similar to the IEEE 754 floating-point standard.



The significand has an implicit leading 1 (except for the denormalized numbers).
The exponent is coded in excess-EXC, where EXC = 2^{E-1}-1. (for example, if E = 8 bits, as in the IEEE 754 standard for 32-bit numbers, the exponent is coded in excess-127, where 127=2^{8-1}-1).
The sign bit (S) is 0 for positive numbers, and 1 for negative numbers.
The table below lists all combinations and their meanings for Simple RISC Processor floating-point data:

S	exponent	significand	value
0	0 00	0 00	+0 (positive) zero
0	0 00	0 01	denormalized positive number
.	...	...	
0	0 00	1 11	
0	0 01	0 00	normalized positive number
.	...	...	
0	1 10	1 11	
0	1 11	0 00	+∞ positive infinite
0	1 11	nonzero	NaN (not a number)
1	0 00	0 00	-0 (negative) zero
1	0 00	0 01	denormalized negative number
.	...	...	
1	0 00	1 11	
1	0 01	0 00	normalized negative number
.	...	...	
1	1 10	1 11	
1	1 11	0 00	-∞ negative infinite
1	1 11	nonzero	NaN (not a number)

When NaN must be generated, the Floating-Point Unit generates an all 1 value.

3. Microarchitecture

The microarchitecture of the floating-point execution unit for the Simple RISC Processor is sketched in Figure 1. It reflects the main stages of the floating-point addition algorithm:

1. operand alignment (the operand with the smaller exponent is right shifted with a number of positions equal to the difference between exponents)
2. operand complementation, if needed, according to the operands' signs and the operation type (addition or subtraction). At most one operand needs to be complemented.
3. addition (the result may be one bit larger)
4. result complementation, if the result is negative
5. result normalization. The significand is left shifted so that its most significant nonzero bit (the implicit bit) appears next to the left of the significand field's MSB, and the exponent is increased or decreased accordingly. If the significand cannot be normalized, it is left shifted with a number of positions equal to the max exponent, and the result exponent is set to zero.

Prior to stage 2, the implicit bit for both operands must be explicitly inserted into its proper place (This bit is 0, if the operand is either zero or a denormalized number, 1 otherwise).

In addition to the above operations, that apply to actual numbers, a special logic block generates infinite or NaN result if any of the input operands is not a number, or if the addition stage produces overflow.

4. Timing

The microarchitecture is split into four pipeline stages to sustain the desired high clock frequency of the Simple RISC Processor.

Note: The block diagram below is only a sketch of the algorithm. The actual implementation might require a rearrangement of some blocks and connections, and additional blocks and signals.

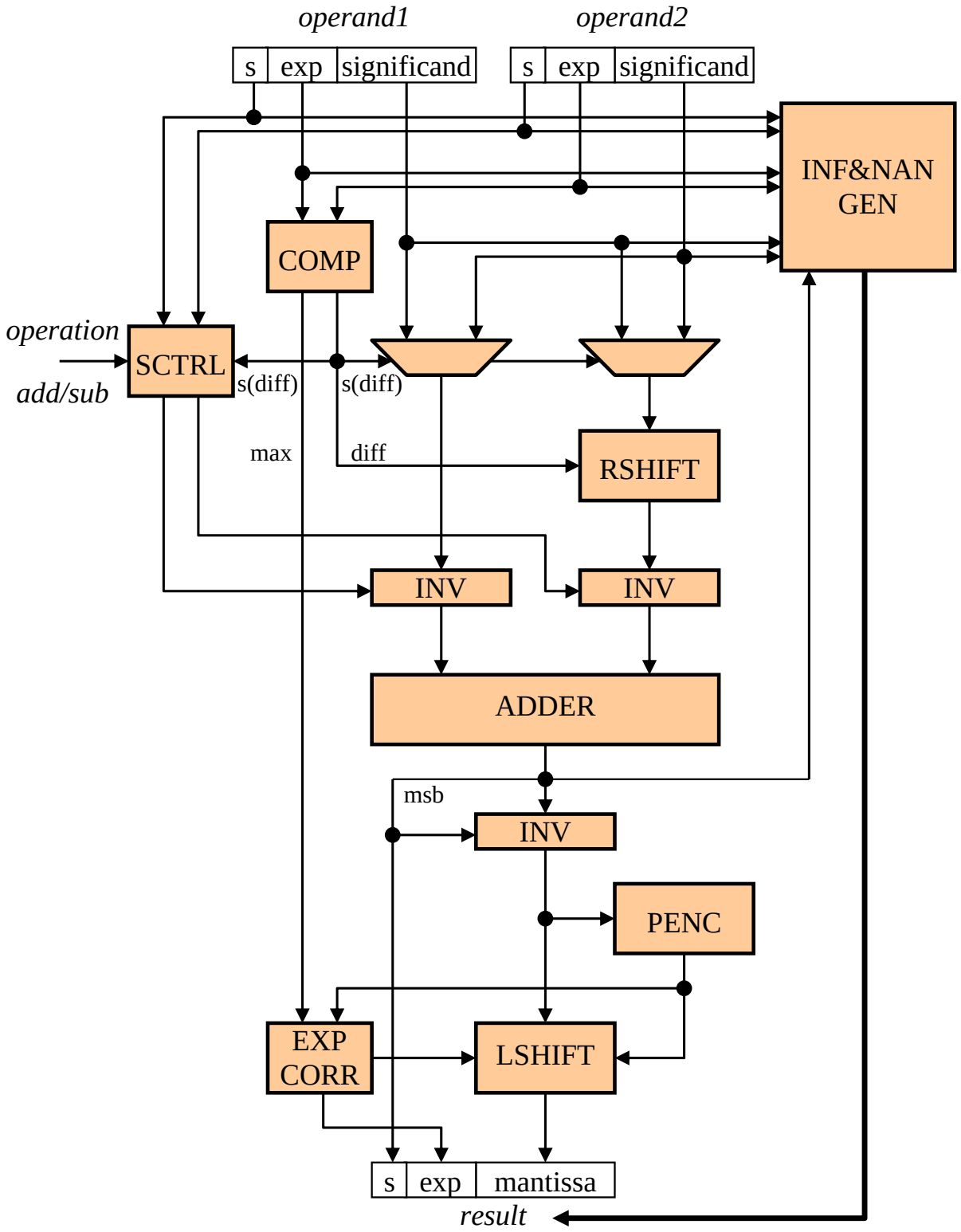


Figure 1
Simple RISC Processor floating-point execution unit block diagram.
diff - exponent difference, s(diff) - the sign of exponent difference (indicates the greatest operand), max - the greatest exponent, msb - the MSB of the adder output (also the sign bit).