Parametric, Floating Point Arithmetic Logic Unit

by Nathan Sketch, Saher Al-Khrayyef, Raymon Benjamin

Objectives

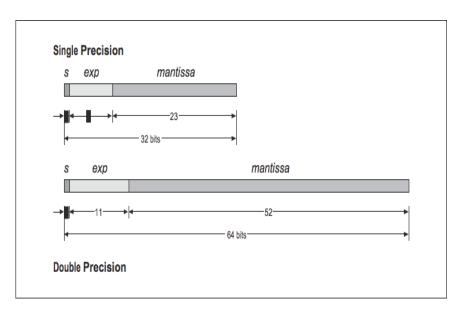
- Design Parametric Floating Point Arithmetic Logic Unit
- Addition
- Subtraction
- Multiplication
- Interface the FPU to a ARM processor via an AXI4 lite bus.

IEEE single and double precision floating point representation

This figure shows the IEEE single and double precision

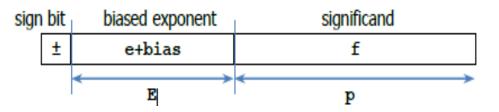
floating point format.

- Where s is the sign bit.
- Exp is the exponent, it depends on the binary number size.
- Mantissa, also it depends on the binary number size.
- Bias single is 127 and double is 1023



IEEE-754 Standard Representation

- The representation is as follows:
 - $X = \pm 1. f \times 2^{e}$



Significand: f is the mantissa. We should add 1 to the beginning of the mantissa before start addition/subtraction.

Significand range should be $[1,2-2^{-}p] = [1,2)$

Biased Exponent:

- \succ E is the number of the bits.
- ➢ Bias = 127 or 1023
- ≻ e= exp bias

Floating Point Addition/Subtraction Equations

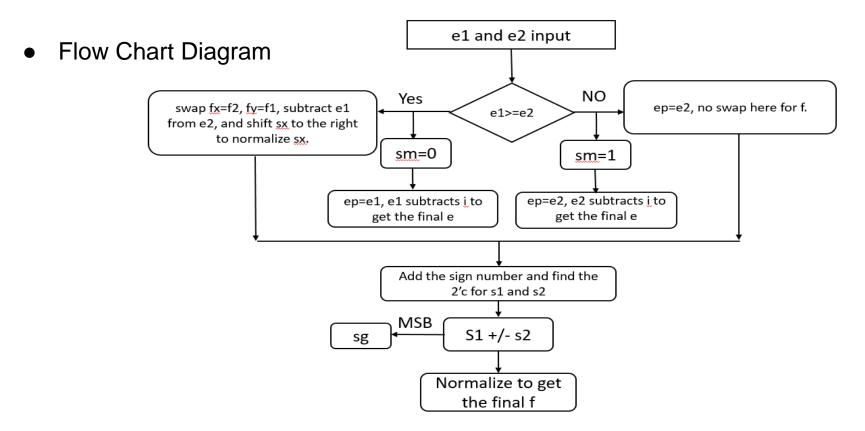
- *b*1=±*s*1*2^*e*1, *s*1=1.*f*1
- *b*2=±*s*2*2^*e*2, *s*2=1.*f*2
- $b1+b2=\pm s1^{*}2^{e1}\pm s2^{*}2^{e2}$

Floating Point Adder/Subtractor Sign

- An add-subtract operation has three sign inputs, a_sign, b_sign, and op.
- We can transform the normal equation into one that better serves the output floating-point format.

-a+(-b)	a-(+b)
-a-b	a-b
-(a+b)	a-b

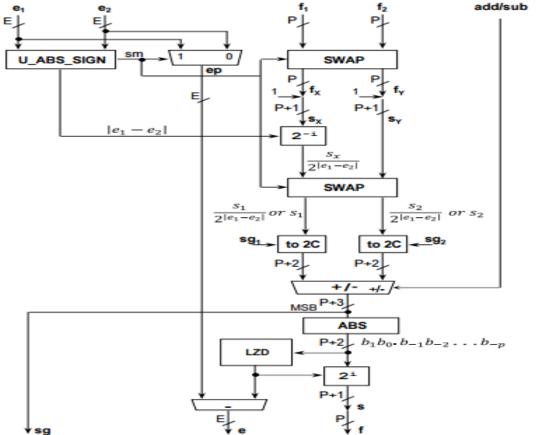
DESIGN OF FLOATING POINT ADDITION/SUBTRACTION



Normalized left shift

- The normalized left shift in the post-normalization step removes leading zeros.

DESIGN OF FLOATING POINT ADDITION/SUBTRACTION Cont.



Addition Simulation Part 1

SixtyFourBitFPAddSub_t	o.vhd 🗙 🔚 Thii	rtyTwoBitFloPAddSub_	_tb_behav.wcfg >	<						
		0.000000000 ms	•							
Name	Value	0.00000 ms	0.00002 ms	0.00004 ms	0.00006 ms	0.00008 ms	0.00010 ms	0.00012 ms	0.00014 ms	0.000
t 🗄 📲 🕷 num 1[63:0]	0.0	0.0	9.25	0.0	1.0	9.25	10.5	5013.4563	186247.256	53
- 🖬 📲 num2[63:0]	0.0	0	0	9.25	1.0	10.5	9.25	186247.256	5013.4563	2
🖪 🖬 📲 expected[63:0]	0.0	0.0	9.	25	2.0	19	75	191260	. 7123	53
	0.0	0.0	9.	25	2.0	19	75	191260	.7123	53
🗤 testAddSub	0									
🕼 expWidth	11					11				
🕼 fracWidth	52					52				
-										
I Contraction of the second										
•										
-										
-										
1										
1										
	<	> <								

Addition Simulation Part 2

SixtyFourBitFPAddSul	b_tb.vhd 🗙 🔚 Thir	tyTwoBitFloPAddS	ub_tb_behav.wcf	g* X						00
						0.000240000 ms				
Name	Value	0.00020 ms	0.00021 ms	0.00022 ms	0.00023 ms	0.00024 ms	0.00025 ms	0.00026 ms	0.00027 ms	0.00028
t ⊡¶i num1[63:0]	3.0			33333333				.0		-10.5
🖥 📲 num2[63:0]	6.75539944105574	2.966765144	67627e-307	3.74519403	96316e+307	6.75539944	105574e+15	1.35107988	821115e+16	-9.25
🖬 📲 expected[63:0	0] 6.75539944105575	53.33333	3333333	3.74519403	96316e+307	6.75539944	105575e+15	1.35107988	821115e+16	-19.75
	6.75539944105575	53.33333	33333333	3.74519403	96316e+307	6.75539944	105575e+15	1.35107988	821115e+16	-19.75
🕼 testAddSub	0									
1. expWidth 1. fracWidth	11					11				
🕼 fracWidth	52					52				
	< >	<								

Subtraction Simulation Part 1

	(h) 5	SixtyFourBitFPAddSub_tb.vhd 🗙 🔚 Thirty	/TwoBitFloPAddSub_tb	_behav.wcfg ×							×
	`							0.000080000 ms			^
		Name	Value	l0.00000 ms	10.00002 ms	10.00004 ms	10.00006 ms	0.00008 ms	10.00010 ms	0.00012 ms 0.0	
) (2 +	🗉 📲 num 1[63:0]	2.5	0.0	9.25	(0.0)	3.5		5	× 1.5 ×	
		num2[63:0]	3.5		10	9.25	2.5	3.5	1.5	2.5	
	0	🛨 📲 expected[63:0]	-1.0	0.0	9.25	-9.25	1.0	-1.0	1.0	-1.0	
	<u>,</u>	±- a testSum[63:0]	-1.0	0.0	9.25	-9.25	1.0	-1.0	1.0	-1.0 (
	<u>•</u>	🕼 testAddSub	1								
		🕼 expWidth	11				11				
		🕼 fracWidth	52	<			52				
	<u>(</u>										
	≥r ⊮										
	Ē.										
	۰Ľ										
	⊷∥										
	μ										
											\sim
			< >	<							>

Subtraction Simulation Part 2

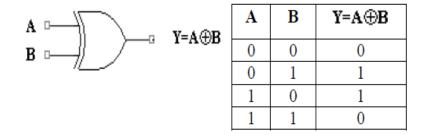
(Wh	🕲 SixtyFourBitFPAddSub_tb.vhd 🗙 🖀 ThirtyTwoBitFloPAddSub_tb_behav.wcfg 🗙										
₩				0.000141830	ms						<u>^</u>
	Name	Value	10	0.00014 ms	0.00016 ms	0.00018 ms	10.00020 ms	10.00022 ms	0.00024 ms	0.00026 ms	10.00028 ms
0+	🖪 📲 num 1[63:0]	1018.952	1.5	1018.952	143.841	3e+17	5.3893	x 5.5	-1018.952	1018.952	x-1018.952
0-	🖬 📲 num2[63:0]	143.841	2.5	143.841	1018.952	5.3893	3e+17	\$ 5.5	143.841	-143.	841
0	🖭 🔣 expected[63:0]	875.111		875.111	-875.111	3e+17	-3e+17	0.0	-1162.793	1162.793	-875.111
кя <mark>8</mark>	E testSum[63:0]	875.111		875.111	-875.111	3e+17	-3e+17	0.0	-1162.793	1162.793	-875.111
-	U testAddSub	1									
M	🕼 expWidth	11					11				
	🕼 fracWidth	52					52				
1											
P		urBitFPAddSub_tb/f	racWidth								
÷											
G.											
-۲											
א											
100											
_											
		<	> <								~ ~
											/

DESIGN OF FLOATING POINT MULTIPLICATION EQUATIONS

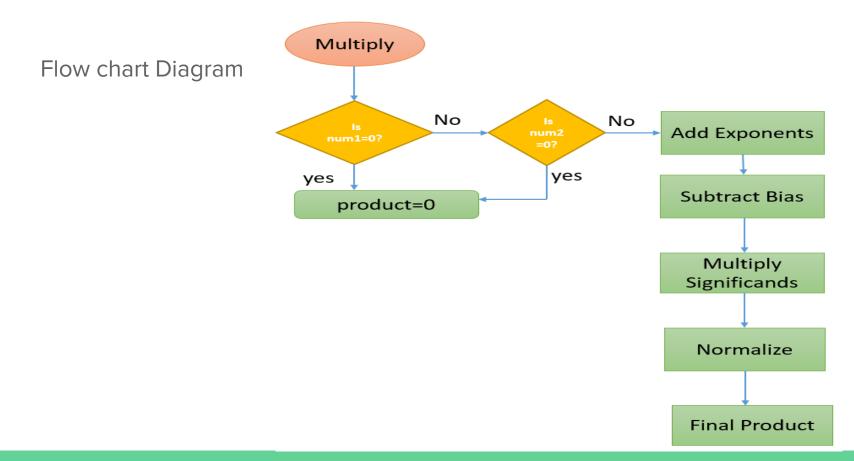
- $b1 = \pm s1^{2} e1$ $b2 = \pm s2^{2} e2$
- $b1 \times b2 = (\pm s1^{*}2^{e1}) \times (\pm s2^{*}2^{e2}) = \pm (s1 \times s2)2^{e1+e2}$
- $s = (s1 \times s2) \in [1,4).$

DESIGN OF FLOATING POINT MULTIPLICATION

- Sign Bit Calculation
- Multiplying two number's result is a negative sign if one of the multiplied numbers is of a negative value. By the aid of a truth table we find that this can be obtained by XORing the sign of two inputs.



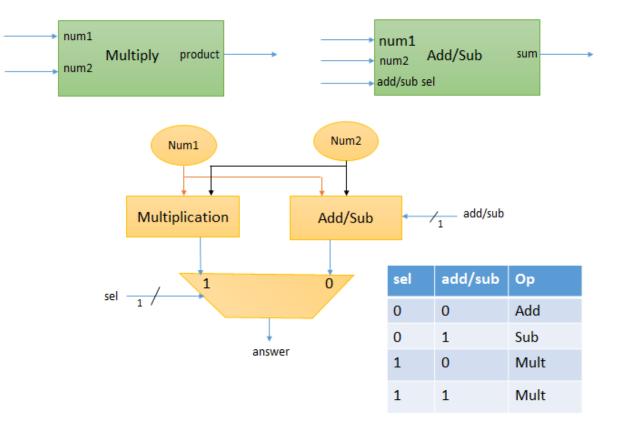
DESIGN OF FLOATING POINT MULTIPLICATION Cont.



Multiplication Simulation

SixtyFourBitFPMult_tb_	behav.wcfg* ×								02
								0.000	136170 ms
Name	Value		0.00006 ms 0.00007	ms 0.00008 ms	0.00009 ms	0.00010 ms	0.00011 ms	0.00012 ms	0.00013
🖽 📲 num 1[63:0]	-8.6489	0.0	1.0		25		4563	-8.648	
🖽 📲 num2[63:0]	-96632.223	-9.25	1.0	10	. 5	18624	7.256	-96632.	223
🖬 📲 expected[63:0]	835762.4335047	0.0	1.0	97.	125	93374247	8.950913	835762.43	35047
testproduct[63:0]	835762.4335047	0.0	1.0	97.	125	93374247	8.950913	835762.43	35047
1e expWidth	11				11				
🕼 fracWidth	52				52				
	< >	<							

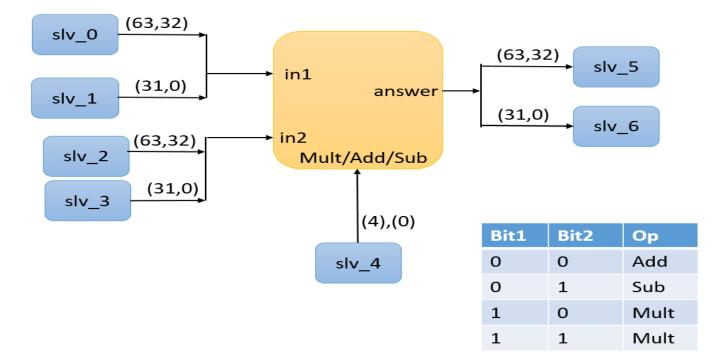
Floating Point Top Circuit



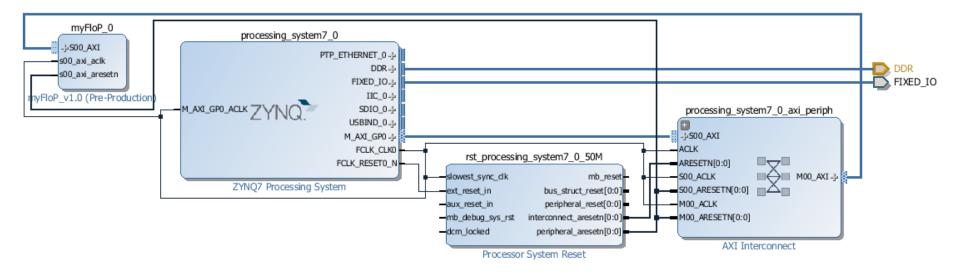
Top Floating Point Arithmetic Circuit Simulation

€					0.000020000 ms				
	Name	Value	0.00000 ms	0.00001 ms	0.00002 ms	0.00003 ms	0.00004 ms	0.00005 ms	0.00006 ms 0
0	🖬 📲 num 1[63:0]	4061fae978d4fdf4		d013a92a	4061fae9		\langle	40b39574d013a	
Q	🖪 📲 num2[63:0]	408fd79db22d0e56	4106bc3a	Oc49ba5e	408fd79d	b22d0e56)	(4106bc3a0c49b	a5e
Q	🖪 📲 answer[63:0]	c08b58e353f7ced9	410758e5	b2ca57a7	c08b58e3	53f7ced9	(41cbd3e3c779b	782
8	expected[63:0]	c08b58e353f7ced9	410758e5	b2ca57a8	c08b58e3	53f7ced9		41cbd3e3c779b	783
		01	0	0	0	1)		10	
	us exprindui	11				11			
		52				52			
Ľ									
2									
-4									
Ģ									
槽	_								
3									

Addition/Subtraction/Multiplication AXI4 lite Interface



IP Block Diagram



Challenges/Improvements

- Challenges VHDL
- Improvements Testbench and Divider

Any Questions?

Thank You