Pipelined processors
Example 1

E. Sanchez

Example 1

Considering the following pseudo-code:

```
for (i = 0; i < 100; i++) {
        $Y[i] = X[i]^2 + X[i] / Z[i]$
}
```

Suppose that vectors X[i] and Z[i] contain 100 FP numbers, were previously saved in memory, and $Z[i] \neq 0$.

Assume the RISC-V processor RV32G with single-precision FP operations presented below:

- Integer ALU: 1 clock cycle
- Data memory: 1 clock cycle
- FP multiplier unit: pipelined 4 stages
- FP arithmetic unit: pipelined 2 stages
- FP divider unit: not pipelined unit, requiring 4 clock cycles
- branch delay slot: 2 clock cycle (disabled)
- forwarding is enabled.

Example 1 – [cont]

1. Write an assembly program for the architecture RISC-V RV32G with FP extensions able to perform the previously presented pseudo-code
2. show the timing of the developed loop-based program and compute how many cycles does this program take to execute
3. using all the static optimization techniques, re-write the developed code in order to eliminate the most data hazards
4. show the timing development of the new optimized program and compute how many clock cycles does this program take to execute.

1. Write an assembly program for the architecture RISC-V RV32G with FP extensions able to perform the previously presented pseudo-code

```
#******** RISCV RV32G PROGRAM ****************
#--------------------------------------------
# STATIC OPTIMIZATION EXAMPLE
# The code implements the following algorithm
# for (i = 0; i < 100; i++) {
#   Y[i] = x[i]^2 + x[i] /z[i]
#   Z[i] ≠ 0 }
#--------------------------------------------
          .data
vect1: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10…
vect2: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10…
vect3: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10…
```

1)

	.text	CC
MAIN:	addi x1, x0, 100	
	la x2, vect1	
	la x3, vect2	
	la x4, vect3	
loop:	flw f1, 0(x2)	
	flw f2, 0(x3)	
	fmul.s f3, f1, f1	
	fdiv.s f4, f1, f2	
	fadd f5, f3, f4	
	fsw f5, 0(x4)	
	addi x2, x2, 4	
	addi x3, x3, 4	
	addi x4, x4, 4	
	addi x1, x1, -1	
	bnez x1,loop	
	nop	

2)

	.text	CC
MAIN:	addi x1, x0, 100	5
	la x2, vect1	2
	la x3, vect2	2
	la x4, vect3	2

	.text	CC
MAIN:	addi x1, x0, 100	5
	auipc X2, 0x10000	1
	addi x2, x2, -4	1
	auipc x3, 0x10000	1
	addi x3, x3, 338	1
	auipc x4, 0x10000	1
	addi x4, x4, 780	1

2)

																							cc
loop:	flw	f1, 0(x2)	F	D	E	M	W																1
	flw	f2, 0(x3)		F	D	E	M	W															1
	fmul.s	f3, f1, f1			F	D	X	X	X	X	M	W											4
	fdiv.s	f4, f1, f2				F	D	d	d	d	d	M	W										1
	fadd.s	f5, f3, f4					F	D	s	s	s	A	A	M	W								2
	fsw	f5, 0(x4)						F	s	s	s	D	E	s	M	W							1
	addi	x2, x2, 4										F	D	s	E	M	W						1
	addi	x3, x3, 4											F	s	D	E	M	W					1
	addi	x4, x4, 4													F	D	E	M	W				1
	addi	x1, x1, -1														F	D	E	M	W			1
	bnez	x1,loop															F	s	D	E	M	W	2
	brn+1																	F	D	-	-	-	1

2)

	.text		CC
MAIN:	addi	x1, x0, 100	5
	la	x2, vect1	2
	la	x3, vect2	2
	la	x4, vect3	2
loop:	flw	f1, 0(x2)	1
	flw	f2, 0(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	fadd.s	f5, f3, f4	2
	fsw	f5, 0(x4)	1
	addi	x2, x2, 4	1
	addi	x3, x3, 4	1
	addi	x4, x4, 4	1
	addi	x1, x1, -1	1
	bnez	x1,loop	2+1
	nop		1

2)

	.text		CC
MAIN:	addi	x1, x0, 100	5
	la	x2, vect1	2
	la	x3, vect2	2
	la	x4, vect3	2
loop:	flw	f1, 0(x2)	1
	flw	f2, 0(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	fadd.s	f5, f3, f4	2
	fsw	f5, 0(x4)	1
	addi	x2, x2, 4	1
	addi	x3, x3, 4	1
	addi	x4, x4, 4	1
	addi	x1, x1, -1	1
	bnez	x1,loop	2+1
	nop		

11

17

2)

11

+

17 x 99

2)

	.text		CC
MAIN:	addi	x1, x0, 100	5
	la	x2, vect1	2
	la	x3, vect2	2
	la	x4, vect3	2
loop:	flw	f1, 0(x2)	1
	flw	f2, 0(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	fadd.s	f5, f3, f4	2
	fsw	f5, 0(x4)	1
	addi	x2, x2, 4	1
	addi	x3, x3, 4	1
	addi	x4, x4, 4	1
	addi	x1, x1, -1	1
	bnez	x1,loop	2
	nop		1

11

16

1

2)

11

+

17 x 99

+

16

+

1

1710

3)

	.text	
MAIN:	addi	x1, x0, 100
	la	x2, vect1
	la	x3, vect2
	la	x4, vect3
loop:	flw	f1, 0(x2)
	flw	f2, 0(x3)
	fmul.s	f3, f1, f1
	fdiv.s	f4, f1, f2
	fadd.s	f5, f3, f4
	fsw	f5, 0(x4)
	addi	x2, x2, 4
	addi	x3, x3, 4
	addi	x4, x4, 4
	[redacted]	[redacted]
	[redacted]	[redacted]
	[redacted]	[redacted]

	.text	
MAIN:	addi	x1, x0, 100
	la	x2, vect1
	la	x3, vect2
	la	x4, vect3
loop:	flw	f1, 0(x2)
	flw	f2, 0(x3)
	fmul.s	f3, f1, f1
	fdiv.s	f4, f1, f2
	addi	x2, x2, 4
	fadd.s	f5, f3, f4
	fsw	f5, 0(x4)
	addi	x3, x3, 4
	addi	x4, x4, 4

3)

																					cc
loop:	flw	f1, 0(x2)	F	D	E	M	W														1
	flw	f2, 0(x3)		F	D	E	M	W													1
	fmul.s	f3, f1, f1			F	D	X	X	X	X	M	W									4
	fdiv.s	f4, f1, f2				F	D	d	d	d	d	M	W								1
	addi	x2, x2, 4					F	D	E	M	W										0
	fadd.s	f5, f3, f4						F	D	s	s	A	A	M	W						2
	fsw	f5, 0(x4)							F	s	s	D	E	s	M	W					1
	addi	x3, x3, 4										F	D	s	E	M	W				1
	addi	x4, x4, 4											F	s	D	E	M	W			1

12

4)

	.text		CC	
MAIN:	addi	x1, x0, 25	5	
	la	x2, vect1	2	11
	la	x3, vect2	2	
	la	x4, vect3	2	
loop:	flw	f1, 0(x2)	1	
	flw	f2, 0(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	addi	x2, x2, 4	0	12
	fadd.s	f5, f3, f4	2	
	fsw	f5, 0(x4)	1	
	addi	x3, x3, 4	1	
	addi	x4, x4, 4	1	
	flw	f1, 0(x2)	1	
	flw	f2, 0(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	12
	addi	x2, x2, 4	0	
	fadd.s	f5, f3, f4	2	
	fsw	f5, 0(x4)	1	
	addi	x3, x3, 4	1	
	addi	x4, x4, 4	1	

	flw	f1, 0(x2)	1	
	flw	f2, 0(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	addi	x2, x2, 4	0	12
	fadd.s	f5, f3, f4	2	
	fsw	f5, 0(x4)	1	
	addi	x3, x3, 4	1	
	addi	x4, x4, 4	1	
	flw	f1, 0(x2)	1	
	flw	f2, 0(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	addi	x2, x2, 4	0	
	fadd.s	f5, f3, f4	2	15
	fsw	f5, 0(x4)	1	
	addi	x3, x3, 4	1	
	addi	x1, x1, -1	1	
	addi	x4, x4, 4	1	
	bnez	x1,loop	1+1	
	nop		1	1

4)

11

+

51 x 24

+

51

1286

4+)

	.text		CC
MAIN:	addi	x1, x0, 25	5
	la	x2, vect1	2
	la	x3, vect2	2
	la	x4, vect3	2
loop:	flw	f1, 0(x2)	1
	flw	f2, 0(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	🟥	🟥	
	fadd.s	f5, f3, f4	2
	fsw	f5, 0(x4)	1
	🟥	🟥	
	🟥	🟥	
	flw	f1, 0(x2)	1
	flw	f2, 0(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	🟥	🟥	
	fadd.s	f5, f3, f4	2
	fsw	f5, 0(x4)	1
	🟥	🟥	
	🟥	🟥	

	.text		CC	
MAIN:	addi	x1, x0, 25	5	11
	la	x2, vect1	2	
	la	x3, vect2	2	
	la	x4, vect3	2	
loop:	flw	f1, 0(x2)	1	10
	flw	f2, 0(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	fadd.s	f5, f3, f4	2	
	fsw	f5, 0(x4)	1	
	flw	f1, 4(x2)	1	10
	flw	f2, 4(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	fadd.s	f5, f3, f4	2	
	fsw	f5, 4(x4)	1	
	flw	f1, 8(x2)	1	10
	flw	f2, 8(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	fadd.s	f5, f3, f4	2	
	fsw	f5, 8(x4)	1	

	flw	f1, 12(x2)	1	15
	flw	f2, 12(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	addi	x2, x2, 16	0	
	fadd.s	f5, f3, f4	2	
	fsw	f5, 12(x4)	1	
	addi	x3, x3, 16	1	
	addi	X1, x1, -1	1	
	addi	x4, x4, 16	1	
	bnez	x1,loop	1+1	
	nop		1	1

11 + (45 x 24) + 45

1136

4+) Enabling Branch delay slot

	.text		CC
MAIN:	addi	x1, x0, 25	5
	la	x2, vect1	2
	la	x3, vect2	2
	la	x4, vect3	2
loop:	flw	f1, 0(x2)	1
	flw	f2, 0(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	fadd.s	f5, f3, f4	2
	fsw	f5, 0(x4)	1
	flw	f1, 4(x2)	1
	flw	f2, 4(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	fadd.s	f5, f3, f4	2
	fsw	f5, 4(x4)	1
	flw	f1, 8(x2)	1
	flw	f2, 8(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	fadd.s	f5, f3, f4	2
	fsw	f5, 8(x4)	1

	flw	f1, 12(x2)	1
	flw	f2, 12(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	addi	x2, x2, 16	0
	fadd.s	f5, f3, f4	2
	fsw	f5, 12(x4)	1
	addi	x3, x3, 16	1
	addi	x1, x1, -1	1
	addi	x4, x4, 16	1
	bnez	x1,loop	1+1
	nop		1

	flw	f1, 12(x2)	1
	flw	f2, 12(x3)	1
	fmul.s	f3, f1, f1	4
	fdiv.s	f4, f1, f2	1
	addi	x2, x2, 16	0
	fadd.s	f5, f3, f4	2
	fsw	f5, 12(x4)	1
	addi	x1, x1, -1	1
	addi	x3, x3, 16	1
	bnez	x1,loop	1
	addi	x4, x4, 16	1
	nop		1

4+) Enabling Branch delay slot

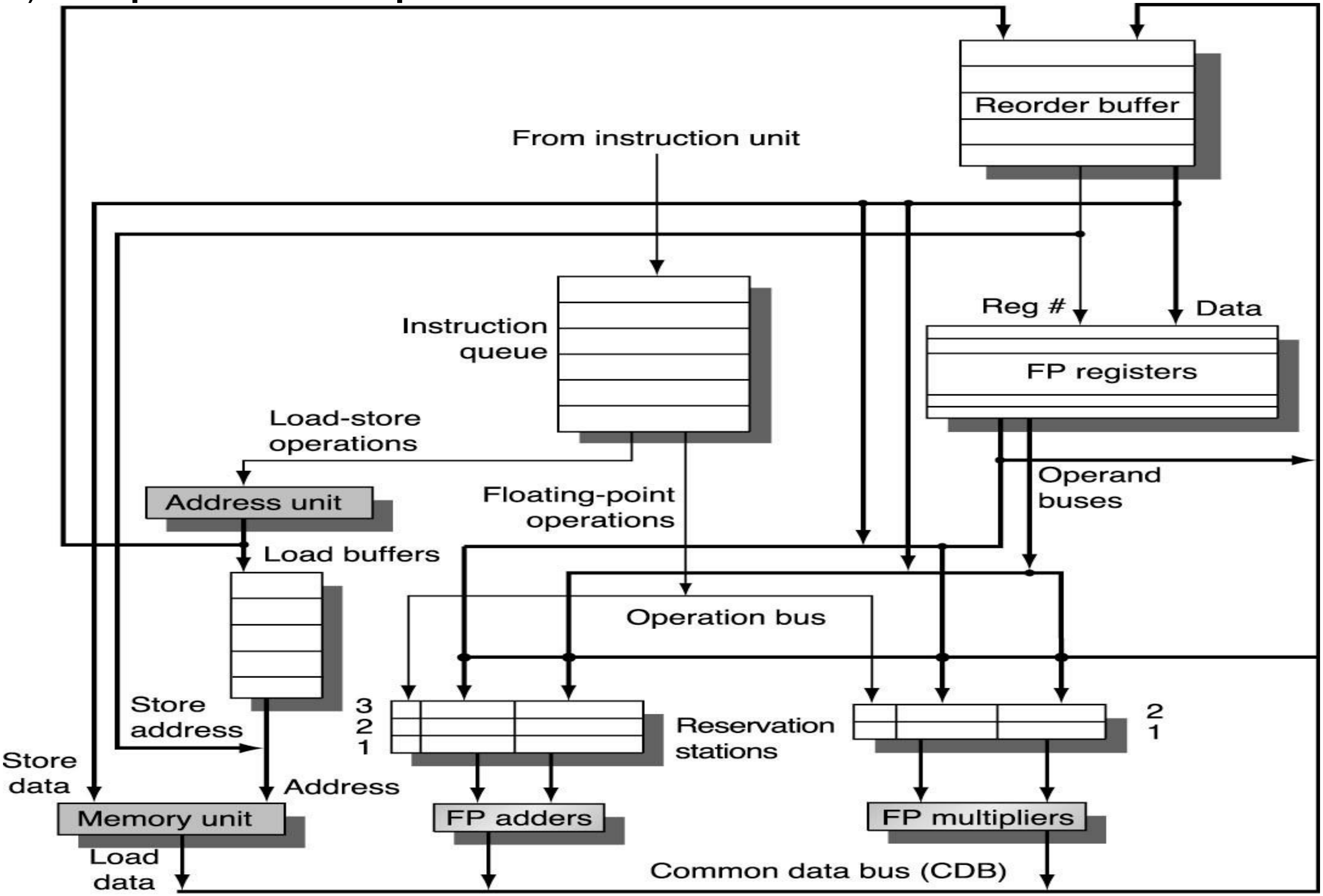
	.text		CC	
MAIN:	addi	x1, x0, 25	5	**11**
	la	x2, vect1	2	
	la	x3, vect2	2	
	la	x4, vect3	2	
loop:	flw	f1, 0(x2)	1	**10**
	flw	f2, 0(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	fadd.s	f5, f3, f4	2	
	fsw	f5, 0(x4)	1	
	flw	f1, 4(x2)	1	**10**
	flw	f2, 4(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	fadd.s	f5, f3, f4	2	
	fsw	f5, 4(x4)	1	
	flw	f1, 8(x2)	1	**10**
	flw	f2, 8(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	fadd.s	f5, f3, f4	2	
	fsw	f5, 8(x4)	1	

	flw	f1, 12(x2)	1	**14**
	flw	f2, 12(x3)	1	
	fmul.s	f3, f1, f1	4	
	fdiv.s	f4, f1, f2	1	
	addi	x2, x2, 16	0	
	fadd.s	f5, f3, f4	2	
	fsw	f5, 12(x4)	1	
	addi	x1, x1, -1	1	
	addi	x3, x3, 16	1	
	bnez	x1,loop	1	
	addi	x4, x4, 16	1	
	nop		1	**1**

11 + (44 x 25) + 1

1112

5) **Multiple-issue with speculation**



5)

	Instruction	Issue	EXE	READ MEM	CDBx2	COMMIT
1	flw f1, 0(x2)	1	2m	3	4	5
	flw f2, 0(x3)	1	3m	4	5	6
	fmul.s f3, f1, f1					
	fdiv.s f4, f1, f2					
	fadd f5, f3, f4					
	fsw f5, 0(x4)					
	addi x2, x2, 4					
	addi x3, x3, 4					
	addi x4, x4, 4					
	addi x1, x1, -1					
	bnez x1,loop					
2	flw f1, 0(x2)					
	flw f2, 0(x3)					
	fmul.s f3, f1, f1					
	fdiv.s f4, f1, f2					
	fadd f5, f3, f4					
	fsw f5, 0(x4)					
	addi x2, x2, 4					
	addi x3, x3, 4					
	addi x4, x4, 4					
	addi x1, x1, -1					
	bnez x1,loop					

5)

	Instruction	Issue	EXE	READ MEM	CDBx2	COMMIT
1	flw f1, 0(x2)	1	2m	3	4	5
	flw f2, 0(x3)	1	3m	4	5	6
	fmul.s f3, f1, f1	2	5x		9	10
	fdiv.s f4, f1, f2	2	6d		10	11
	fadd f5, f3, f4	3	11a		13	14
	fsw f5, 0(x4)					
	addi x2, x2, 4					
	addi x3, x3, 4					
	addi x4, x4, 4					
	addi x1, x1, -1					
	bnez x1,loop					
2	flw f1, 0(x2)					
	flw f2, 0(x3)					
	fmul.s f3, f1, f1					
	fdiv.s f4, f1, f2					
	fadd f5, f3, f4					
	fsw f5, 0(x4)					
	addi x2, x2, 4					
	addi x3, x3, 4					
	addi x4, x4, 4					
	addi x1, x1, -1					
	bnez x1,loop					

5)

	Instruction	Issue	EXE	READ MEM	CDBx2	COMMIT
1	flw f1, 0(x2)	1	2m	3	4	5
	flw f2, 0(x3)	1	3m	4	5	6
	fmul.s f3, f1, f1	2	5x		9	10
	fdiv.s f4, f1, f2	2	6d		10	11
	fadd f5, f3, f4	3	11a		13	14
	fsw f5, 0(x4)	3	4m			14
	addi x2, x2, 4	4	5i		6	15
	addi x3, x3, 4	4	6i		7	15
	addi x4, x4, 4	5	7i		8	16
	addi x1, x1, -1	5	8i		9	16
	bnez x1,loop	6	10j			17
2	flw f1, 0(x2)					
	flw f2, 0(x3)					
	fmul.s f3, f1, f1					
	fdiv.s f4, f1, f2					
	fadd f5, f3, f4					
	fsw f5, 0(x4)					
	addi x2, x2, 4					
	addi x3, x3, 4					
	addi x4, x4, 4					
	addi x1, x1, -1					
	bnez x1,loop					

5)

	Instruction	Issue	EXE	READ MEM	CDBx2	COMMIT
1	flw f1, 0(x2)	1	2m	3	4	5
	flw f2, 0(x3)	1	3m	4	5	6
	fmul.s f3, f1, f1	2	5x		9	10
	fdiv.s f4, f1, f2	2	6d		10	11
	fadd f5, f3, f4	3	11a		13	14
	fsw f5, 0(x4)	3	4m			14
	addi x2, x2, 4	4	5i		6	15
	addi x3, x3, 4	4	6i		7	15
	addi x4, x4, 4	5	7i		8	16
	addi x1, x1, -1	5	8i		9	16
	bnez x1,loop	6	10j			17
2	flw f1, 0(x2)	7	8m	9	10	17
	flw f2, 0(x3)	7	9m	10	11	18
	fmul.s f3, f1, f1	8	11x		15	18
	fdiv.s f4, f1, f2	8	12d		16	19
	fadd f5, f3, f4	9	17a		19	20
	fsw f5, 0(x4)	9	10m			20
	addi x2, x2, 4	10	11i		12	21
	addi x3, x3, 4	10	12i		13	21
	addi x4, x4, 4	11	13i		14	22
	addi x1, x1, -1	11	14i		15	22
	bnez x1,loop	12	16j			23

5+)

3

Instruction	Issue	EXE	READ MEM	CDBx2	COMMIT
flw f1, 0(x2)	13	14m	15	16	23
flw f2, 0(x3)	13	15m	16	17	24
fmul.s f3, f1, f1	14	17x		21	24
fdiv.s f4, f1, f2	14	18d		22	25
fadd f5, f3, f4	15	23a		25	26
fsw f5, 0(x4)	15	16m			26
addi x2, x2, 4	16	17i		18	27
addi x3, x3, 4	16	18i		19	27
addi x4, x4, 4	17	19i		20	28
addi x1, x1, -1	17	20i		21	28
bnez x1,loop	18	22j			29