

Pipeline architecture: example 2



E. Sanchez, M. Sonza Reorda

Politecnico di Torino

Dipartimento di Automatica e Informatica (DAUIN)

Torino - Italy

This work is licensed under the Creative Commons (CC BY-SA) License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>



Example 2

Let consider 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 each 100 FP numbers
- were previously saved in memory
- Z[i] ≠ 0.

Assume the following MIPS64 architecture:

- 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: 1 clock cycle
- forwarding is enabled.

```
                .data
vetX:   .double 2, 4, 6, 8...
vetZ:   .double 1, 2, 3, 4...
vetY:   .double 0, 0, 0, 0...
```

```
                .text
MAIN:  daddui R1,R0,100
        daddui R2,R0,vetX
        daddui R3,R0,vetZ
        daddui R4,R0,vetY
loop:  l.d     F1,0(R2)
        l.d     F2,0(R3)
        mul.d   F3,F1,F1
        div.d   F4,F1,F2
        add.d   F5,F3,F4
        s.d     F5,0(R4)
        daddi   R2,R2,8
        daddi   R3,R3,8
        daddi   R4,R4,8
        daddi   R1,R1,-1
        bnez    R1,loop
        halt
```

Let consider the assembly
code on the left,
implementing the
considered C code.

Example 2 – [cont]

- 1. show the timing of the proposed loop-based program and compute the number of clock cycles this program requires to execute**
- 2. using static optimization techniques, re-write the developed code in order to reduce the impact of data hazards**
- 3. compute how many clock cycles the new program takes to execute.**

1)

	.text	CC
MAIN:	daddui R1,R0,100	
	daddui R2,R0,vetX	
	daddui R3,R0,vetZ	
	daddui R4,R0,vetY	
loop:	l.d F1,0(R2)	
	l.d F2,0(R3)	
	mul.d F3,F1,F1	
	div.d F4,F1,F2	
	add.d F5,F3,F4	
	s.d F5,0(R4)	
	daddi R2,R2,8	
	daddi R3,R3,8	
	daddi R4,R4,8	
	daddi R1,R1,-1	
	bnez R1,loop	
	halt	

1)

[illegible]

1)

	.text	CC
MAIN:	daddui R1,R0,100	5
	daddui R2,R0,vetX	1
	daddui R3,R0,vetZ	1
	daddui R4,R0,vetY	1
loop:	l.d F1,0(R2)	1
	l.d F2,0(R3)	1
	mul.d F3,F1,F1	4
	div.d F4,F1,F2	1
	add.d F5,F3,F4	2
	s.d F5,0(R4)	1
	daddi R2,R2,8	1
	daddi R3,R3,8	1
	daddi R4,R4,8	1
	daddi R1,R1,-1	1
	bnez R1,loop	2
		1

8

8

+

17 x 100

1708 ccs

17

2)

	.text	
MAIN:	daddui	R1,R0,100
	daddui	R2,R0,vetX
	daddui	R3,R0,vetZ
	daddui	R4,R0,vetY
loop:	l.d	F1,0(R2)
	l.d	F2,0(R3)
	mul.d	F3,F1,F1
	div.d	F4,F1,F2
	add.d	F5,F3,F4
	s.d	F5,0(R4)
	daddi	R2,R2,8
	daddi	R3,R3,8
	daddi	R4,R4,8
	daddi	R1,R1,-1
	bnez	R1,loop
	halt	

			cc
MAIN:	daddui	R1,R0,25	
	daddui	R2,R0,vetX	
	daddui	R3,R0,vetZ	
	daddui	R4,R0,vetY	
loop:	l.d	F1,0(R2)	
	l.d	F2,0(R3)	
	mul.d	F3,F1,F1	
	div.d	F4,F1,F2	
	daddi	R2,R2,8	
	add.d	F5,F3,F4	
	s.d	F5,0(R4)	
	daddi	R3,R3,8	
	daddi	R4,R4,8	

3) anticipating daddi

[illegible]

3) anticipating daddi

[illegible]

3+) loop unrolling

			CC	
MAIN:	daddui	R1,R0,25	5	8
	daddui	R2,R0,vetX	1	
	daddui	R3,R0,vetZ	1	
	daddui	R4,R0,vetY	1	
loop:	l.d	F1,0(R2)	1	12
	l.d	F2,0(R3)	1	
	mul.d	F3,F1,F1	4	
	div.d	F4,F1,F2	1	
	daddi	R2,R2,8	0	
	add.d	F5,F3,F4	2	
	s.d	F5,0(R4)	1	
	daddi	R3,R3,8	1	
	daddi	R4,R4,8	1	12
	l.d	F1,0(R2)	1	
	l.d	F2,0(R3)	1	
	mul.d	F3,F1,F1	4	
	div.d	F4,F1,F2	1	
	daddi	R2,R2,8	0	
	add.d	F5,F3,F4	2	
	s.d	F5,0(R4)	1	
	daddi	R3,R3,8	1	
	daddi	R4,R4,8	1	

	l.d	F1,0(R2)	1	12
	l.d	F2,0(R3)	1	
	mul.d	F3,F1,F1	4	
	div.d	F4,F1,F2	1	
	daddi	R2,R2,8	0	
	add.d	F5,F3,F4	2	
	s.d	F5,0(R4)	1	
	daddi	R3,R3,8	1	
	daddi	R4,R4,8	1	15
	l.d	F1,0(R2)	1	
	l.d	F2,0(R3)	1	
	mul.d	F3,F1,F1	4	
	div.d	F4,F1,F2	1	
	daddi	R2,R2,8	0	
	add.d	F5,F3,F4	2	
	s.d	F5,0(R4)	1	
	daddi	R3,R3,8	1	
	daddi	R1,R1,-1	1	
	daddi	R4,R4,8	1	
	bnez	R1,loop	1	
	halt		1	

$$\begin{array}{r}
 8 \\
 + \\
 51 \times 25 \\
 \hline
 1283
 \end{array}$$

3++) loop unrolling II

			CC	
MAIN:	daddui	R1,R0,25	5	8
	daddui	R2,R0,vetX	1	
	daddui	R3,R0,vetZ	1	
	daddui	R4,R0,vetY	1	
loop:	l.d	F1,0(R2)	1	10
	l.d	F2,0(R3)	1	
	mul.d	F3,F1,F1	4	
	div.d	F4,F1,F2	1	
	add.d	F5,F3,F4	2	
	s.d	F5,0(R4)	1	
	l.d	F1,8(R2)	1	10
	l.d	F2,8(R3)	1	
	mul.d	F3,F1,F1	4	
	div.d	F4,F1,F2	1	
	add.d	F5,F3,F4	2	
	s.d	F5,8(R4)	1	

	l.d	F1,16(R2)	1	10	4+)
	l.d	F2,16(R3)	1		
	mul.d	F3,F1,F1	4		
	div.d	F4,F1,F2	1		
	add.d	F5,F3,F4	2		
	s.d	F5,16(R4)	1	14	8 + 44 x 25 + 1 ----- 1109
	l.d	F1,24(R2)	1		
	l.d	F2,24(R3)	1		
	mul.d	F3,F1,F1	4		
	div.d	F4,F1,F2	1		
	daddi	R2,R2,32	0		
	add.d	F5,F3,F4	2		
	s.d	F5,24(R4)	1		
	daddi	R1,R1,-1	1		
	daddi	R3,R3,32	1		
	bnez	R1,loop	1		
	daddi	R4,R4,32	1		
	halt		1		
				1	