**Lab 4: Written by Tyler Holland**

**Non Optimized Code: RAW Hazards are commented with //HAZARD**

**Also: if it cannot be resolved by forwarding, Unresolved is added**

.file 1 "matmul.c"

.section .mdebug.abi32

.previous

.text

.align 2

.globl main

.ent main

main:

.frame $fp,1224,$31 # vars= 1216, regs= 1/0, args= 0, extra= 0

.mask 0x40000000,-8

.fmask 0x00000000,0

subu $sp,$sp,1224

sw $fp,1216($sp) **//HAZARD with $sp**

move $fp,$sp

sw $0,1204($fp) **//HAZARD with $fp**

$L2:

lw $2,1204($fp)

slt $2,$2,10 **//HAZARD with $2, Unresolved**

bne $2,$0,$L5

j $L3

$L5:

sw $0,1208($fp)

$L6:

lw $2,1208($fp)

slt $2,$2,10 **//HAZARD with $2, Unresolved**

bne $2,$0,$L9

j $L4

$L9:

lw $3,1204($fp)

move $2,$3 **//HAZARD with $3, Unresolved**

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$3 **//HAZARD with $2**

sll $3,$2,1 **//HAZARD with $2**

lw $2,1208($fp)

addu $2,$3,$2 **//HAZARD with $2, Unresolved**

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$fp **//HAZARD with $2**

addu $2,$2,800 **//HAZARD with $2**

sw $0,0($2) **//HAZARD with $2**

lw $2,1208($fp)

addu $2,$2,1 **//HAZARD with $2, Unresolved**

sw $2,1208($fp)

j $L6

$L4:

lw $2,1204($fp)

addu $2,$2,1 **//HAZARD with $2, Unresolved**

sw $2,1204($fp)

j $L2

$L3:

sw $0,1204($fp)

$L10:

lw $2,1204($fp)

slt $2,$2,10 **//HAZARD with $2, Unresolved**

bne $2,$0,$L13

j $L11

$L13:

sw $0,1208($fp)

$L14:

lw $2,1208($fp)

slt $2,$2,10 **//HAZARD with $2, Unresolved**

bne $2,$0,$L17

j $L12

$L17:

sw $0,1212($fp)

$L18:

lw $2,1212($fp)

slt $2,$2,10 **//HAZARD with $2, Unresolved**

bne $2,$0,$L21

j $L16

$L21:

lw $3,1204($fp)

move $2,$3

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$3 **//HAZARD with $2**

sll $3,$2,1 **//HAZARD with $2**

lw $2,1208($fp)

addu $2,$3,$2 **//HAZARD with $2, Unresolved**

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$fp **//HAZARD with $2**

addu $5,$2,800 **//HAZARD with $2**

lw $3,1204($fp)

move $2,$3 **//HAZARD with $3**

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$3 **//HAZARD with $2**

sll $3,$2,1 **//HAZARD with $2**

lw $2,1208($fp)

addu $2,$3,$2 **//HAZARD with $2, Unresolved**

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$fp **//HAZARD with $2**

addu $4,$2,800 **//HAZARD with $2**

lw $3,1204($fp)

move $2,$3 **//HAZARD with $3, Unresolved**

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$3 **//HAZARD with $2**

sll $3,$2,1 **//HAZARD with $2**

lw $2,1212($fp)

addu $2,$3,$2 **//HAZARD with $2, Unresolved**

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$fp **//HAZARD with $2**

lw $3,0($4)

lw $2,0($2)

addu $4,$3,$2 **//HAZARD with $2**

lw $3,1212($fp)

move $2,$3 **//HAZARD with $3, Unresolved**

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$3 **//HAZARD with $2**

sll $3,$2,1 **//HAZARD with $2**

lw $2,1208($fp)

addu $2,$3,$2 **//HAZARD with $2, Unresolved**

sll $2,$2,2 **//HAZARD with $2**

addu $2,$2,$fp **//HAZARD with $2**

addu $2,$2,400 **//HAZARD with $2**

lw $2,0($2) **//HAZARD with $2**

addu $2,$4,$2 **//HAZARD with $2, Unresolved**

sw $2,0($5)

lw $2,1212($fp)

addu $2,$2,1 **//HAZARD with $2, Unresolved**

sw $2,1212($fp)

j $L18

$L16:

lw $2,1208($fp)

addu $2,$2,1 **//HAZARD with $2, Unresolved**

sw $2,1208($fp)

j $L14

$L12:

lw $2,1204($fp)

addu $2,$2,1 **//HAZARD with $2, Unresolved**

sw $2,1204($fp)

j $L10

$L11:

move $sp,$fp

lw $fp,1216($sp) **//HAZARD with $sp**

addu $sp,$sp,1224

j $31

.end main

**Optimized Code: RAW Hazards are commented with //HAZARD**

**Also: if it cannot be resolved by forwarding, Unresolved is added**

.file 1 "matmul.c"

.section .mdebug.abi32

.previous

.text

.align 2

.globl main

.ent main

main:

.frame $sp,1200,$31 # vars= 1200, regs= 0/0, args= 0, extra= 0

.mask 0x00000000,0

.fmask 0x00000000,0

subu $sp,$sp,1200

move $13,$0

$L11:

sll $5,$13,2

addu $3,$5,$13 **//HAZARD with $5**

sll $4,$3,3 **//HAZARD with $3**

addu $3,$4,$sp **//HAZARD with $4**

addu $2,$3,800 **//HAZARD with $3**

li $9,9 # 0x9

$L10:

addu $9,$9,-1

sw $0,0($2)

bgez $9,$L10

addu $2,$2,4

addu $13,$13,1

slt $6,$13,10 **//HAZARD with $13**

bne $6,$0,$L11

move $13,$0

$L26:

sll $9,$13,2

addu $8,$9,$13 **//HAZARD with $9**

sll $7,$8,3 **//HAZARD with $8**

move $12,$7 **//HAZARD with $7**

move $9,$0

addu $8,$7,$sp

move $10,$sp

$L25:

lw $7,800($8)

move $11,$8

addu $5,$10,400

addu $4,$12,$sp

li $6,9 # 0x9

$L24:

lw $24,0($4)

lw $14,0($5)

addu $15,$7,$24

addu $6,$6,-1

addu $7,$15,$14

addu $4,$4,4

bgez $6,$L24

addu $5,$5,40

addu $9,$9,1

slt $25,$9,10 **//HAZARD with $9**

sw $7,800($11)

addu $10,$10,4

bne $25,$0,$L25

addu $8,$8,4

addu $13,$13,1

slt $10,$13,10 **//HAZARD with $13**

bne $10,$0,$L26

j $31

addu $sp,$sp,1200

.end main

**A Hazard that can be resolved by moving statements (changes are in bold):**

**In non-optimized code:**

**Original:**

lw $2,1208($fp)

addu $2,$3,$2

sll $2,$2,2

addu $2,$2,$fp

addu $4,$2,800

**lw $3,1204($fp)**

move $2,$3 **//HAZARD with $3, Unresolved**

sll $2,$2,2

**Fixed:**

lw $2,1208($fp)

addu $2,$3,$2

**lw $3,1204($fp)**

sll $2,$2,2

addu $2,$2,$fp

addu $4,$2,800

move $2,$3

sll $2,$2,2

**Explanation:** Because $3 isn't being used in the steps before the command (mode $2, $3) that causes the hazard that can't be resolved with forwarding, I was able to move the lw that deals with $3 up a few notches to get rid of the hazard.

The Optimized code has no unresolved hazards.

**Average CPI:**

**Note: Number of instructions are calculated on the next page**

**Fill/drain cost is 4 in a MIPS pipeline**

**Non-Optimized:**

CPI = [(instructions per cycle \* cycles) + fill/drain cost] / instructions

= [(55989 \* 1) + 4] / 55989 = **1.000071443**

**CPI = 1.000071443**

**Optimized:**

CPI = [(instructions per cycle \* cycles) + fill/drain cost] / instructions

= [(8615\* 1) + 4] / 8615 = **1.000464306**

**CPI = 1.000464306**

**Run time of the code with 1GHz processor:**

**Non-Optimized:**

Instruction count:

4 - main = 4

**Inside first i for loop: multiply by 10**

4 - $L2, i < 10 =40

1 - $L5, j = 0 =10

**Inside first j for loop: multiply by 10**

4 - $L6, j < 10 =400

15 - $L9, C[i][j] = 0; =1500

**end j loop**

4 - $L4, i++ =40

**end i loop**

1 - $L3, i = 0 =1

**Inside second i for loop: multiply by 10**

4 - $L10, i < 10 =40

1 - $L13, j = 0 =10

**Inside second j loop: mult. by 10**

4 - $L14, j < 10 =400

1 - $L17, k = 0 =100

**Inside k loop: mult. by 10**

4 - $L18, k < 10 =4000

49 - $L21, C[i][j] += A[i][k] + B[k][j] = 49000

**end k loop**

4 - $L16, j++ =400

**end j loop**

4 - $L12, i++ =40

**end i loop**

4 - last statements =4

**Total: 55989 instructions**

CPI: **1.000071443**

Total Run Time on 1Ghz CPU:

1Ghz = 1 x 10^9(billion) cycles/second = 1 cycle/ 1 billionth of a second = 1 cycle/nanosecond

[1 cycle/nanosecond] \* [1.000071443 cycles/instruction] \* [55989 instructions] = **55993 nanoseconds**

**Total Run Time = 55993 nanoseconds**

**Optimized:**

Instruction count:

2 - main =2

**$L11 : the first i loop, mult. by 10**

6 - $L11 =60

**$L10 : the first j loop, mult. by 10**

3 - $L10 before bgez $9 $L10 =300

**end j loop**

4 - $L10 after bgez $9 $L10 =40

**end i loop**

1 - out of loops =1

**$L26 : the second i loop, mult. by 10**

7 - $L26 =70

**$L25 : the second j loop, mult. by 10**

5 - $L25 =500

**$L24 : the k loop, mult. by 10**

7 - $L24 before bgez $6 $L24 =7000

**end k loop**

6 - $L24 after bgez $6 $L24 and before bne $25 $0 $L25 =600

**end j loop**

4 - $L24 after bne $25 $0 $L25 =40

**end i loop**

2 - remaining parts =2

**Total: 8615**

CPI: **1.000464306**

Total Run Time on 1Ghz CPU:

1Ghz = 1 x 10^9(billion) cycles/second = 1 cycle/ 1 billionth of a second = 1 cycle/nanosecond

[1 cycle/nanosecond] \* [1.000464306cycles/instruction] \* [8615 instructions] = **8619 nanoseconds**

**Total Run Time = 8619 nanoseconds**

**Speedup of Optimized version compared to non-optimized:**

**Non-optimized speed: 55993 nanoseconds**

**Optimized speed: 8619 nanoseconds**

[Non-optimized time / optimized time] = % speed gain

55993 / 8619 = 6.4964

**Total speedup: 649.64% faster**