Lab 1 Report – ECE552

I. **Performance Drop** (go.pisa-big):

```
sim: ** simulation statistics **
sim_num_insn              548130711 # total number of instructions executed
sim_num_refs              157194899 # total number of loads and stores executed
sim_elapsed_time                 12 # total simulation time in seconds
sim_inst_rate          45677559.2500 # simulation speed (in insts/sec)
sim_num_RAW_hazard_q1     278022058 # total number of RAW hazards (q1)
sim_num_RAW_hazard_q1_1cycle     28799116 # total number of RAW hazards with 1 cycle (q1)
sim_num_RAW_hazard_q1_2cycle    249222942 # total number of RAW hazards with 2 cycle (q1)
sim_num_RAW_hazard_q2     258988915 # total number of RAW hazards (q2)
sim_num_RAW_hazard_q2_1cycle    218653489 # total number of RAW hazards with 1 cycle (q2)
sim_num_RAW_hazard_q2_2cycle     40335426 # total number of RAW hazards with 2 cycle (q2)
CPI_from_RAW_hazard_q1       1.9619 # CPI from RAW hazard (q1)
CPI_from_RAW_hazard_q2       1.5461 # CPI from RAW hazard (q2)
```

1. $Slowdown = 100\left(1 - \frac{old\ time}{new\ time}\right) = 100\left(1 - \frac{old\ CPI}{new\ CPI}\right) = 100\left(1 - \frac{1}{1.9619}\right) = 49.03\%$
2. $Slowdown = 100\left(1 - \frac{old\ time}{new\ time}\right) = 100\left(1 - \frac{old\ CPI}{new\ CPI}\right) = 100\left(1 - \frac{1}{1.5461}\right) = 35.32\%$

$$\text{New CPI (non} - \text{ideal)} = 1 + \sum_{k=0}^{n} \text{stall\_freq}_k * \text{stall\_cycles}_k$$
$$= 1 + 1\_\text{cycle\_stall\_freq} * 1 + 2\_\text{cycle\_stall\_freq} * 2$$

II. Microbenchmarking

Compiled with **-O2 flag** (avoid codes getting optimized away)

/cad2/ece552f/compiler/bin/ssbig-na-sstrix-gcc mbq1.c -O2 -o mbq1

Result:

```
sim: ** simulation statistics **
sim_num_insn              23509273 # total number of instructions executed
sim_num_refs                 5163 # total number of loads and stores executed
sim_elapsed_time                1 # total simulation time in seconds
sim_inst_rate          23509273.0000 # simulation speed (in insts/sec)
sim_num_RAW_hazard_q1       9903005 # total number of RAW hazards (q1)
sim_num_RAW_hazard_q1_1cycle      1000981 # total number of RAW hazards with 1 cycle (q1)
sim_num_RAW_hazard_q1_2cycle      8902024 # total number of RAW hazards with 2 cycle (q1)
```

Observe: **#2 cycle RAW hazard = 8*#1 cycle RAW hazard**

These are possible dependencies from the main loop of microbenchmark:

red = 2 cycle stalls*; *blue = 1 cycle stalls

(for 2 cycle stalls, not all dependencies are taken every loop)

```c
int main() {
    int a = 1;        // $5
    int b = 2;        // $6
    int c = 0;        // $7
    int count = 4114;     // $3
    int i;                // $4

    for (i = 0; i < 1000000; i++) {        // $11 stores 55061
        a = b + c;          // addu $5,$6,$7
        b = a + c;          // addu $6,$5,$7
        c = a + b;          // addu $7,$5,$6
        count++;
        b = a + c;          // addu $6,$5,$7

        // // Prevent overflow
        if (a > 857808) a = a % count;
        if (b > 326498) b = b % count;
        if (c > 728974) c = c % count;
    }
    // Prevent optimized away
    printf("%lld %lld %lld", a, b, c);
    return 0;
}
```

```
400268:    42 00 00 00        addu $5,$6,$7
40026c:    00 05 07 06
400270:    42 00 00 00        addu $6,$5,$7
400274:    00 06 07 05
400278:    42 00 00 00        addu $7,$5,$6
40027c:    00 07 06 05
400280:    43 00 00 00        addiu $3,$3,1
400284:    01 00 03 03
400288:    42 00 00 00        addu $6,$5,$7
```

```
400298:   48 00 00 00      div $0,$5,$3
40029c:   00 00 03 05
4002a0:   06 00 00 00      bne $3,$0,4002b0 <main+0xc0>
```

```
4002c0:   a2 00 00 00      lui $1,32768
4002c4:   00 80 01 00
4002c8:   06 00 00 00      bne $5,$1,4002d8 <main+0xe8>
```

```
4002f0:   48 00 00 00      div $0,$6,$3
4002f4:   00 00 03 06
4002f8:   06 00 00 00      bne $3,$0,400308 <main+0x118>
```

```
400308:   43 00 00 00      addiu $1,$0,-1
40030c:   ff ff 01 00
400310:   06 00 00 00      bne $3,$1,400330 <main+0x140>
```

```
400318:   a2 00 00 00      lui $1,32768
40031c:   00 80 01 00
400320:   06 00 00 00      bne $6,$1,400330 <main+0x140>
```

```
400360:   43 00 00 00      addiu $1,$0,-1
400364:   ff ff 01 00
400368:   06 00 00 00      bne $3,$1,400388 <main+0x198>
40036c:   06 00 01 03
400370:   a2 00 00 00      lui $1,32768
400374:   00 80 01 00
400378:   06 00 00 00      bne $7,$1,400388 <main+0x198>
```

	Cycles															
Instr.	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	Comments
1	F	D	X	M	W											
2		F	d*	d*	D	X	M	W								
3			p*	p*	F	d*	d*	D	X	M	W					
4								F	D	X	M	W				
5									F	d*	D	X	M	W		

As a result, it is correct that each loop, we should expect 7->10 2 cycle stalls, which is correct, because:

1,000,000 (N) * [7->10] ~= 8,902,024

It is also correct that each loop, we should expect 1 1 cycle stall, because:

1,000,000 (N) * 1 ~= 1,000,981

➔ Confirmed!!