CS152 Section 2

Q1. Iron Law

For each of the three terms, come up with three techniques that improve that term.

Instructions/Program:

CPI:

Cycle Time:

Q1.1 Explain how each term changes given the proposed change (increase, decrease, no change).

Quiz 1, 2011: [In a classic RISC pipeline] Modifying the ISA (and thus the microarchitecture) to use hardware interlocking instead of software interlocking for both branch delay slots and load-use delay slots

Instructions/Program:

CPI:

Cycle Time:

Quiz 1, 2013: Remove hardware floating-point instructions and instead use software subroutines for floating-point arithmetic

Instructions/Program:

CPI:

Cycle Time:

Q2 Pipelining:

Q2.1 Label the data hazards in the follow RISC-V assembly snippet:

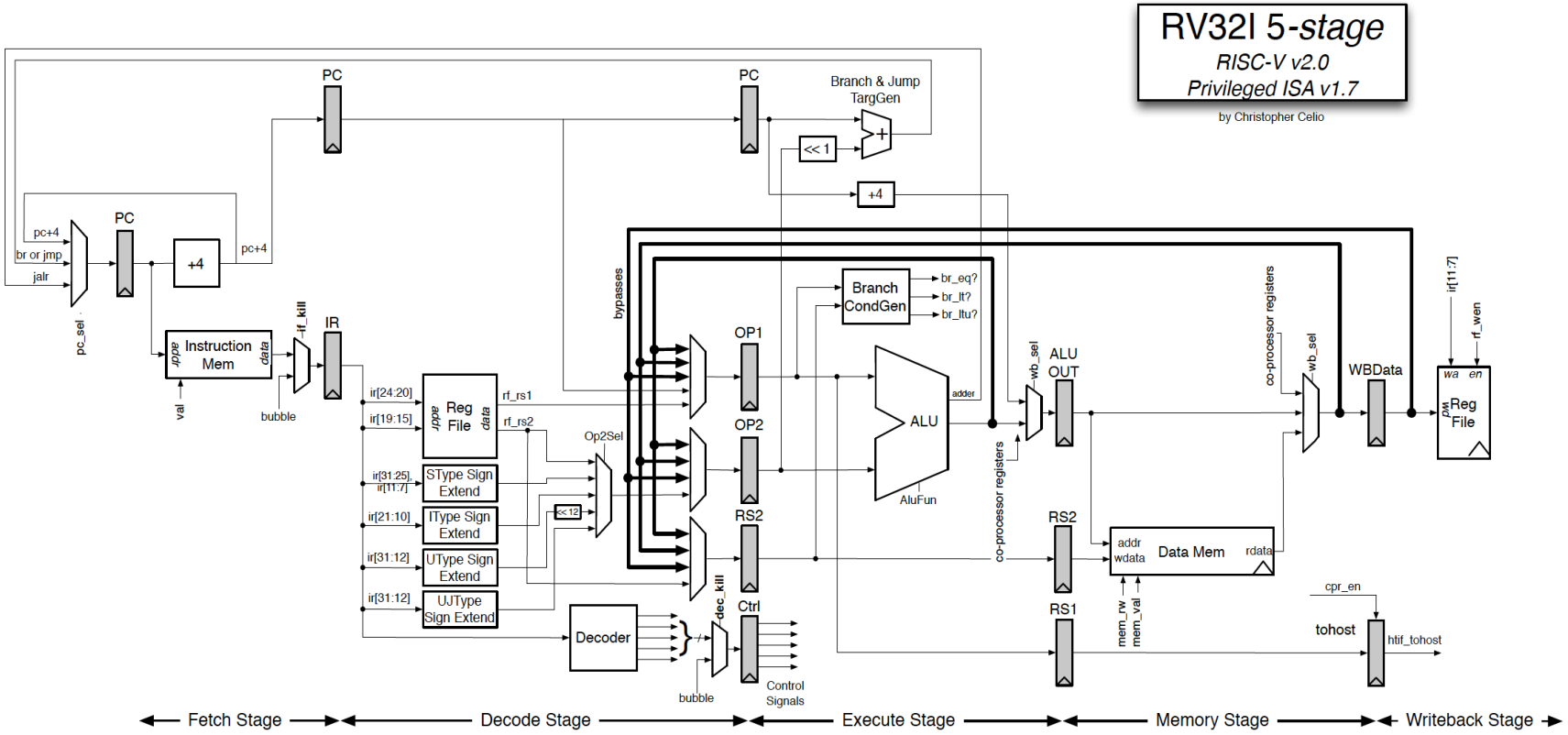
```
ADDI  x1, x0, 4
SW    x1, 8(x2)
SLLI  x3, x1, 1
ADD   x3, x2, x3
LW    x1, 0(x3)
```

Q2.2 What does the following code do? How many iterations does it run?

Memory:
0x400: 0x0
0x404: 0xD40
...
0x800: 0x9F0
0x804: 0x400
...
0x9F0: 0x400
0x9F4: 0x0
...
0xD40: 0x0
0xD44: 0x9F0

```
        ADD   x1, x0, x0
        ADDI  x2, x0, 0x800
Loop:   LW    x2, 4(x2)
        ADD   x1, x2, x1
        BNEQ  x2, x0, Loop
```

Q2.3 For the following pipeline, fill out the pipeline table for the code snippet above*:*



(Follow Q2.3) Questions:

- For the prior code and pipeline what is the CPI (branches always predicted not taken)?

- Give an expression for CPI for K iterations (branches always predicted not taken)?

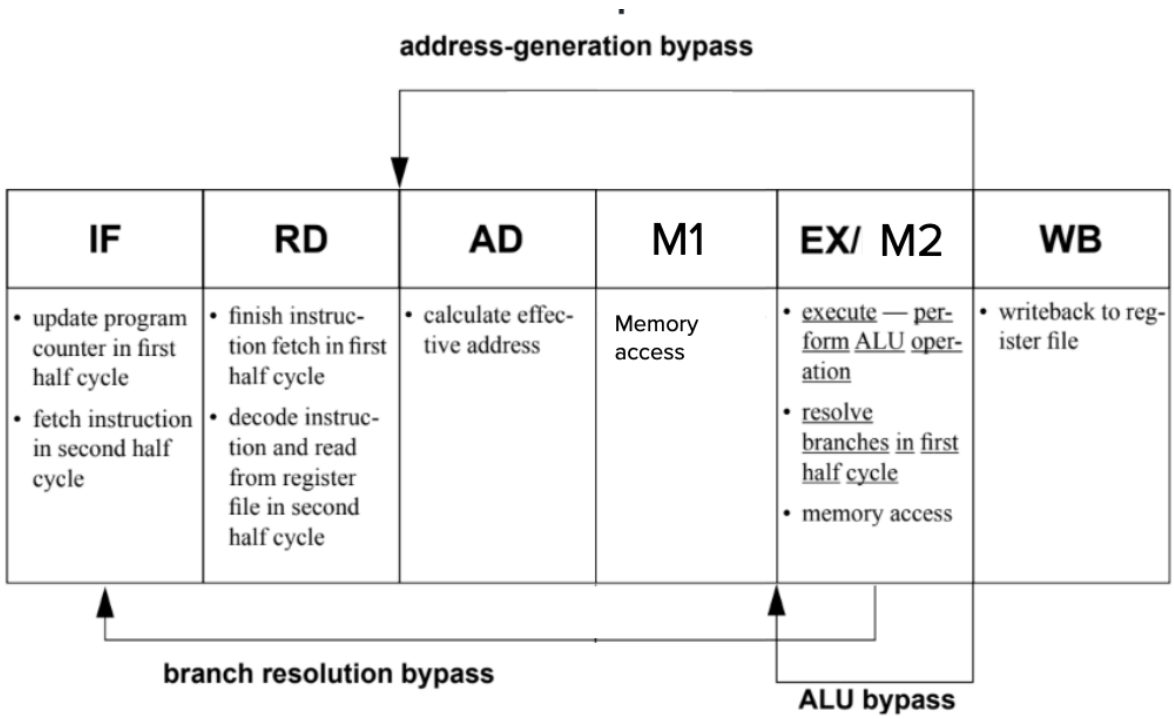
- With perfect branch prediction?

Considering splitting M into M1 and M2.

- How does CPI change?

- How does CPI change when the M stage is made to be N stages long? (Give an expression)

Q2.4 Fill out the pipeline table for the following pipeline, assuming branches are always predicted not taken [M. Golden and T. Mudge, "A comparison of two pipeline organizations," 1994]



What is the CPI for this pipeline?

How does CPI change when:

- There is only one MEM stage?

- N memory stages?

Suppose classic RISC Pipeline closes timing @ 1 GHz.

- (For application), at what frequency does the AGI-2 pipeline perform better?

- AGI-N vs LUI-N (classical RISC pipeline with N memory stages)?

Q3. Exceptions

In this question, we will use the same pipeline as Q2.3

```
        0x2000   add x1, x0, x0     # x1 <= x0 + x0
bar:    0x2004   bne x1, x0, exit
        0x2008   bge x1, x0, foo    # branch if greater or equal
        0x200c   addi x1, x1, -100
        0x2010   add x5, x5, x5
                 ……
foo:             ……
exit:            ……
```

Quiz 2016: The first time the instruction in PC 0x2008 executes (bge), it raises an exception. If the pipeline provides precise exceptions, what must the value of x1 be by the time the pipeline is flushed and the handler executes? What is the value if the pipeline has imprecise exceptions?

Precise exception:

Imprecise exceptions:

Classic RISC Pipeline:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
ADD	F	D	X	M	W																									
ADDI		F	D	X	M	W																								

AGI Pipeline:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
ADD	F	R	A	M1	X	W																								
ADDI		F	R	A	M1	X	W																							