Introduction to Computer Architecture

Assignment 3

Due January 09, 2024

Grading Policy:
The ultimate goal for all presentation and assignments is review and understand.
The grading process may concentrate more on completeness than correctness.

*****CHAPTER 3 - ILP*****

1. [20 = 10 + 10]

Assume the following code and latencies.

Latencies beyond single cycle	
Memory LD	+3
Memory SD	+1
Integer ADD, SUB	+0
Branches	+1
fadd.d	+2
fmul.d	+4
fdiv.d	+10

```
Loop:   fld     f2,0(Rx)
I0:     fmul.d  f2,f0,f2
I1:     fdiv.d  f8,f2,f0
I2:     fld     f4,0(Ry)
I3:     fadd.d  f4,f0,f4
I4:     fadd.d  f10,f8,f2
I5:     fsd     f4,0(Ry)
I6:     addi    Rx,Rx,8
I7:     addi    Ry,Ry,8
I8:     sub     x20,x4,Rx
I9:     bnz     x20,Loop
```

a. Reorder the instructions to improve performance of the code. How many cycles does your reordered code take?
b. Use loop unrolling to unroll two iterations of the loop in the reordered code in question c.

2. [20 = 10 + 10]

Please explain the principles of register renaming and provide an example code snippet to showcase how it can be accelerated by register renaming.

*****CHAPTER 4 - DLP*****

3. [20 = 10 + 10]

Consider the following code, which multiplies two vectors that contain single-precision complex values:

```
for (i=0;i <300;i++) {
     c_re[i] = a_re[i] * b_re[i] – a_im[i] * b_im[i];
     c_im[i] = a_re[i] * b_im[i]+a_im[i] * b_re[i];
}
```

Assume that the processor runs at 700 MHz and has a maximum vector length of 64. The load/store unit has a start-up overhead of 15 cycles; the multiply unit, 8 cycles; and the add/subtract unit, 5 cycles.

a. Assuming chaining and a single memory pipeline, how many chimes are required? How many clock cycles are required per complex result value, including start-up overhead?

b. Now assume that the processor has three memory pipelines and chaining. If there are no bank conflicts in the loop's accesses, how many clock cycles are required per result?

*****CHAPTER 5 - TLP*****

4. [40 = 10 + 10 + 10 + 10]

The simple, bus-based multiprocessor illustrated in the following figure represents a commonly implemented symmetric shared-memory architecture. Each processor has a single, private cache with coherence maintained using the snooping coherence protocol. Each cache is directed-mapped, with four blocks each holding two words. To simplify the illustration, the cache-address tag contains the full address and each word shows only two hex characters, with the least significant word on the right. The coherence states are denoted M, S, and I for Modified, Shared, and Invalid.

Treat each action below as independently applied to the initial state as given in the figure.

What is the resulting state (i.e., coherence state, tags, and data) of the caches and memory after the given action? Show only the blocks that change, for example, P0.B0: (I, 120, 00 01) indicates that CPU P0's block B0 has the final state of I, tag of 120, and data words 00 and 01.

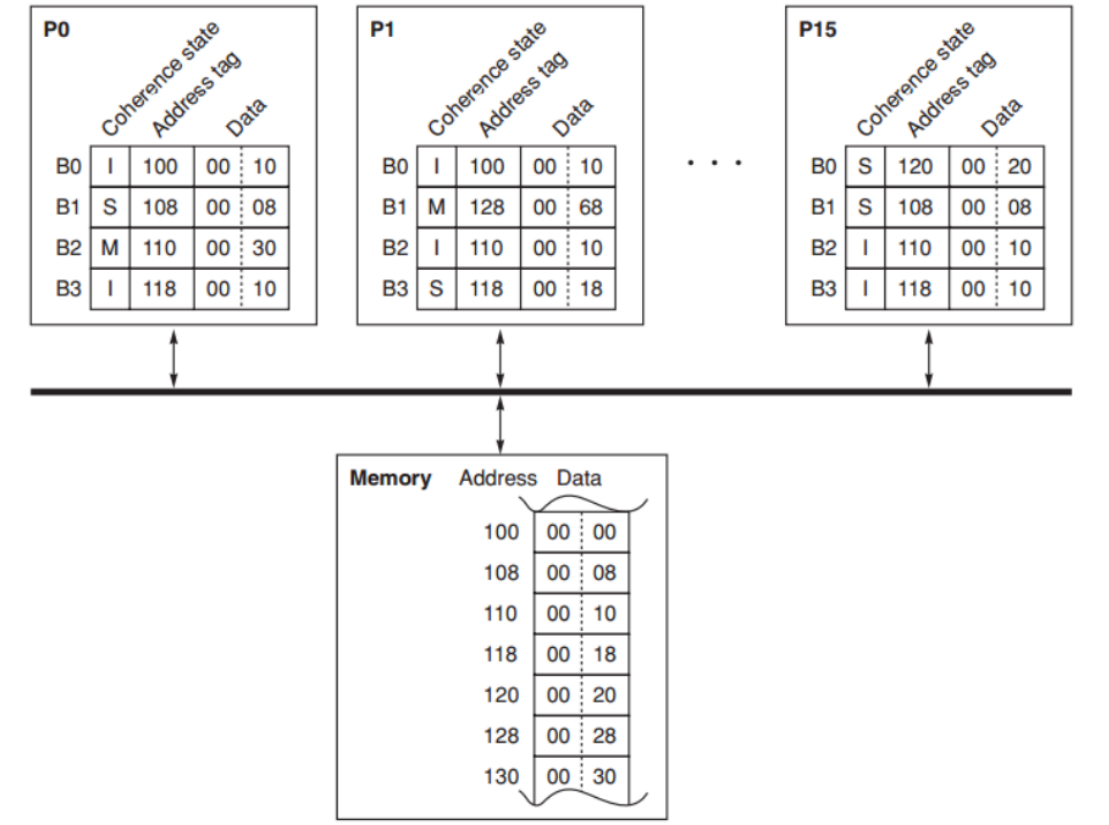
Also, what value is returned by each read operation?

a. P0: read 120

b. P0: write 120 ← 80

c. P3: write 120 ← 80

d. P1: read 110


P0
Coherence state
Address tag
Data
B0 I 100 00 10
B1 S 108 00 08
B2 M 110 00 30
B3 I 118 00 10
P1
Coherence state
Address tag
Data
B0 I 100 00 10
B1 M 128 00 68
B2 I 110 00 10
B3 S 118 00 18
P15
Coherence state
Address tag
Data
B0 S 120 00 20
B1 S 108 00 08
B2 I 110 00 10
B3 I 118 00 10
Memory Address Data
100 00 00
108 00 08
110 00 10
118 00 18
120 00 20
128 00 28
130 00 30