Q1 Memories

15 Points

Consider a processor with the following given characteristics:

- Page size: 4KB (= 4096B)
- Separate data and instruction TLBs.
- Data TLB:
  - Fully associative
  - 4 entries
  - block size of 1
  - LRU replacement policy
- Data L1 cache:
  - Physically addressed
  - 8-way set associative
  - Total size: 4KB
  - Cache line size: 64B
  - LRU replacement policy

The following simple C code and its nonoptimized compiled assembly code fill four arrays `a`, `b`, `c`, `d` according to the content of an input array `z`. In the program, `a`, `b`, `c`, `d`, and `z` are five different arrays of bytes (i.e., type `int8_t` in C) that are page-aligned. As can be seen in the assembly code, the variables `i`, `tmp`, `a`, `b`, `c`, `d`, `z` are stored in registers.

```
for (int64_t i = 0; i < 4096; i++) {

  int8_t tmp = z[i];

  a[i] = tmp;
  b[i] = tmp + 1;
  c[i] = tmp + 2;
  d[i] = tmp + 3;

}
```

```
    lui  x2, 1
    addi x3, x0, 0
loop:
    add  x4, x1, x3
```

```
lb   x5, 0(x4)
add  x4, x10, x3
sb   x5, 0(x4)
addi x6, x5, 1
add  x4, x11, x3
sb   x6, 0(x4)
addi x6, x5, 2
add  x4, x12, x3
sb   x6, 0(x4)
addi x6, x5, 3
add  x4, x13, x3
sb   x6, 0(x4)
addi x3, x3, 1
blt  x3, x2, loop
```

Q1.1

2 Points

What is the total number of memory accesses (= load & store operations) of this program? Write the number below:

20480

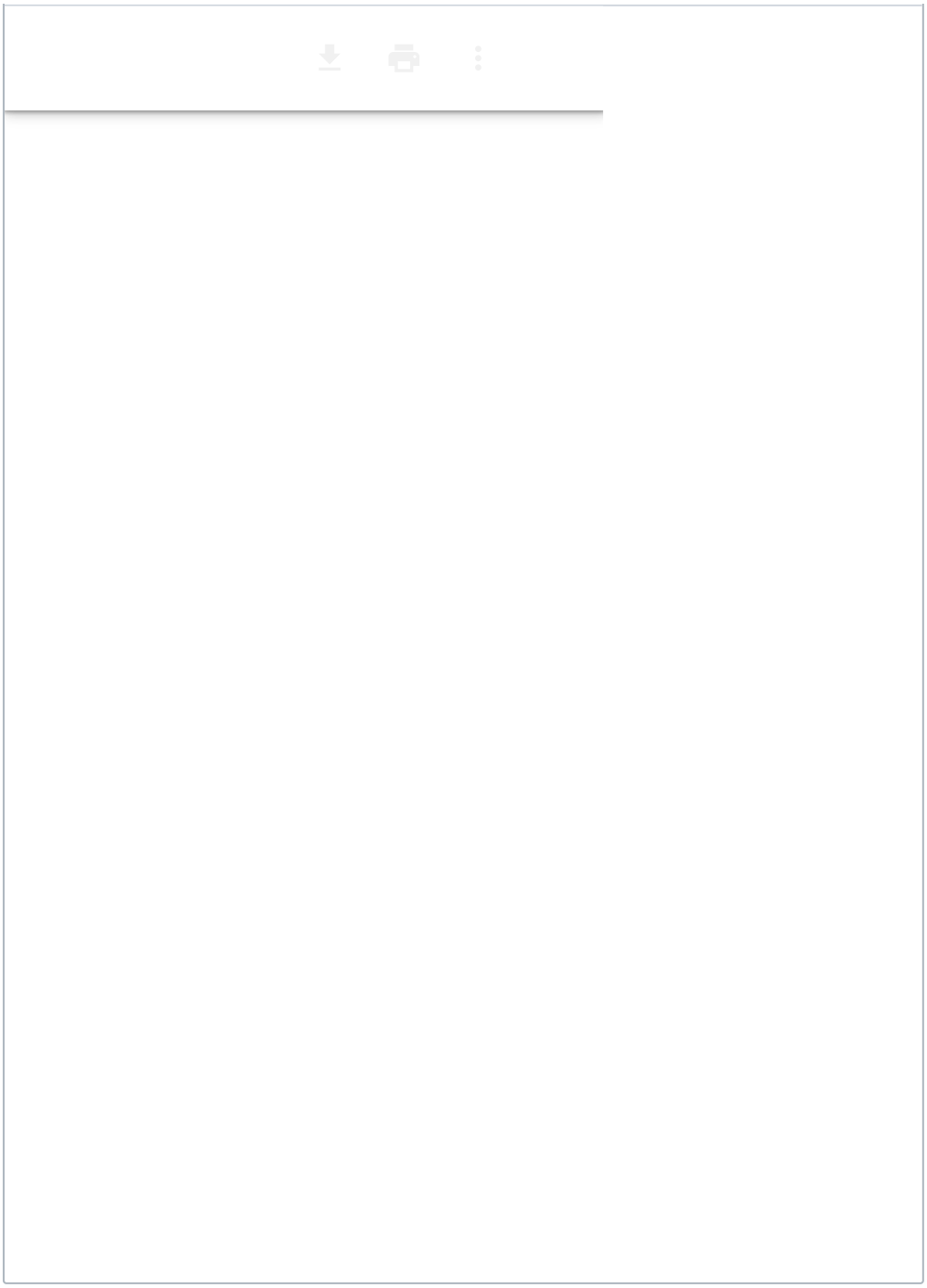
Q1.2

4 Points

Assuming that the data TLB is initially empty, count the number of memory references that result in data TLB misses. What is the corresponding TLB miss rate? Clearly explain.

Upload your solution here.

▾ EPSON030.PDF Download



Q1.3

2 Points

Find the number of sets of the L1 data cache.

8

Q1.4

3 Points

For any given iteration `i`, are the elements `z[i]`, `a[i]`, `b[i]`, `c[i]`, and `d[i]` necessarily mapped to the same set? Explain why. (Hint: remember that `z`, `a`, `b`, `c`, `d` arrays are all page aligned)

EPSON030.PDF Download

Q1.5

4 Points

Assuming that the L1 data cache is initially empty, count the number of L1 data misses of this program. What is the corresponding L1 data miss rate? Clearly explain.

EPSON030.PDF Download

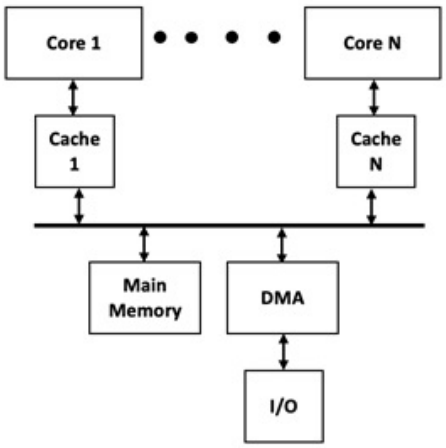
Q2

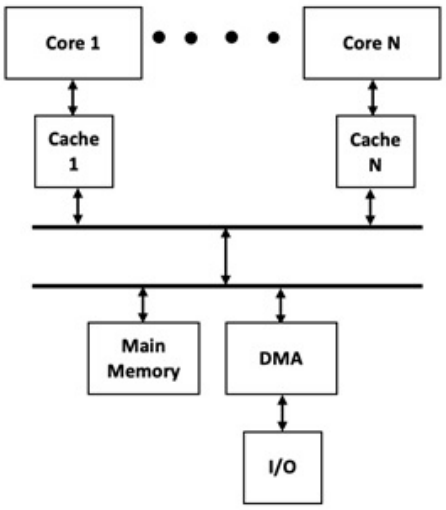
6 Points

Consider the multi-core systems shown in the figure below. Each core has a private cache attached to it.

In System 1, there is one bus which connects the caches, the Main Memory and a direct-memory-access (DMA) engine which facilitates I/O-to-Memory transactions.
In System 2, there are two buses, one is a core-side bus, and the other is a memory-side bus. Such a configuration allows two simultaneous accesses on the two buses if the sender-receiver pairs are completely contained within that bus. For example, a core-to-core communication can happen in parallel over the core bus while a DMA to memory transaction is active in the memory bus.

Assume all buses can support 1 transaction every 2 cycles. Also assume there is a DMA request once every 10 cycles which goes through the bus connected to DMA and main memory (each request is one bus transaction).





(1) (2)

Q2.1
1 Point

If there is no cache coherence implemented (ignore any correctness issues), how many processor to main memory transactions can System 1 support per 10 cycles without bus becoming a bottleneck ?

4

Q2.2
1 Point

If there is no cache coherence implemented (ignore any correctness issues), how many processor to main memory transactions can System 2 support per 10 cycles without any bus becoming a bottleneck ?

4

Q2.3
2 Points

Now assume that an invalidating snooping cache coherence protocol is implemented where everytime a shared value is written to a cache, an invalidation message is broadcast to all other caches for that address over the bus (assume this invalidation message is one bus transaction). If one of the cores executes a write to a shared location every 10 cycles, now how many processor to main memory transactions can System 1 support per 10 cycles without bus becoming a bottleneck ? Explain.

Answer: 3

Invalidation Message = 1 Bus Transaction
DMA Request = 1 Bus Transaction
10 cycles - 2cycles/transaction * (1 + 1) = 6 cycles
But, each remaining main memory transaction takes 2 cycles/trans action as well. Thus, only 3 main memory transactions are possibl e.

Q2.4
2 Points

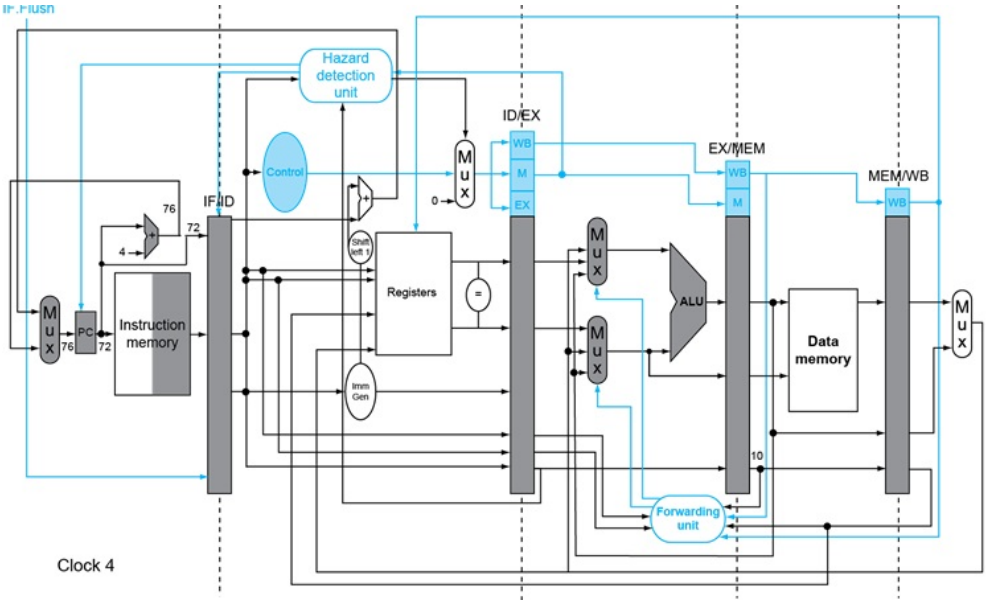
In the case above, how many processor to main memory transactions can System 2 support per 10 cycles without any bus becoming a bottleneck ? Explain.

Answer: 4

Invalidation Message = 1 Core Side Bus Transaction
DMA Request = 1 Memory Side Bus Transaction
PARALLEL!!!
10 - 2 cycles/transaction * (1 || 1) = 8 cycles
But, each remaining memory transaction takes 2 cycles/transaction as well. Thus, only 4 main memory transactions are possible with n o bottleneck.

Q3

11 Points



The RISC-5 pipeline with 1 cycle branch misprediction penalty is shown above. Assume all forwarding is enabled including register file bypassing. We will investigate the load-use hazard with this pipeline.

Q3.1

1 Point

Consider the following program:
LD x1, 100(x2)
ADD x3, x1, x2

How many NOPs/stalls would be needed for this program ?

1

Q3.2
2 Points

Now consider the following program
LD x1, 100(x2)
BEQ x1, x3, L1

How many NOPs/stalls would be needed for this program ?

1

Q3.3
3 Points

You consider getting rid of load-use hazard by swapping locations of MEM and EX stages in the pipeline. What additional hardware would be needed to make sure that load instruction executes correctly in this case ? Explain.

You would need either an Adder or an ALU in the MEM stage beca use you will have to compute the offset for load and store instructi ons before you call them in the MEM stage.

Q3.4
1 Point

Will the following program execute without NOPs/stalls now (assuming hardware of previous question) ?

LD x1, 100(x2)
ADD x3, x1, x2

○ No

◉ Yes

Q3.5
2 Points

How many NOPs/stalls would be needed for the following program under this new hardware. Assume all possible forwarding.

ADD x1, x2, x3
ADD x4, x1, x2

0

Q3.6
2 Points

How many NOPs/stalls are needed for the following program under this new hardware. Assume all possible forwarding.

ADD x1, x2, x3
LD x4, 100(x1)

0

Q4
8 Points

State true or false with a brief explanation.

Q4.1
2 Points

There are no advantages to having variable-length instructions.

False. There is an advantage to having variable-length instructions. In languages such as x86-64, they chose to use variable-length ins tructions because addressing is a lot more flexible in this case. In a ddition, common instructions typically will have shorter encoding a nd use less space in the instruction cache. Thus, x86 chips will nee d less instruction cache space for the same performance.

Q4.2

2 Points

Suppose I limit the number of branches in programs running on a processor to just 2 each, then for the 1-bit branch history table discussed in class, I only need two entries in it.

False. You only need 1 entry because a 1 bit branch history table ca n contain 2 values 2^1 = 2. With just 2 processors you only need lo g_2(2) = 1 entry.

Q4.3

2 Points

Latency and throughput always improve together.

False. Latency and throughput are related, but the higher the laten cy, the longer the time it takes, and the lower the throughput. They are actually opposite instead of improving together.

Q4.4

2 Points

Integers can only overflow while floating point numbers can overflow and underflow.

True. Integers can overflow when you add two big ones together. I ntegers will negatively overflow when you add two large negative numbers together. They will never underflow. Floating point numb

ers can overflow when you add two big numbers together. (Just try adding the max of whatever floating point type to itself). Floating p oint CAN underflow when the solution to your arithmetic is smaller than the smallest representable floating point value for that floatin g point type (32 or 64 bit).

Final

GRADED

STUDENT
LIANG, NEVIN

TOTAL POINTS
29 / 40 pts

QUESTION 1

Memories **12** / 15 pts

1.1	(no title)	**2** / 2 pts
1.2	(no title)	**4** / 4 pts
1.3	(no title)	**2** / 2 pts
1.4	(no title)	**3** / 3 pts
1.5	(no title)	**1** / 4 pts

QUESTION 2

(no title) **6** / 6 pts

2.1	(no title)	**1** / 1 pt
2.2	(no title)	**1** / 1 pt
2.3	(no title)	**2** / 2 pts
2.4	(no title)	**2** / 2 pts

QUESTION 3

(no title) **7** / 11 pts

3.1	(no title)	**1** / 1 pt
3.2	(no title)	**0** / 2 pts
3.3	(no title)	**3** / 3 pts
3.4	(no title)	**1** / 1 pt
3.5	(no title)	**2** / 2 pts
3.6	(no title)	**0** / 2 pts

QUESTION 4

(no title) **4** / 8 pts

4.1	(no title)	**2** / 2 pts
4.2	(no title)	**0** / 2 pts
4.3	(no title)	**0** / 2 pts
4.4	(no title)	**2** / 2 pts