ECE116C Midterm 2 12/04/2019

Name:

UID:

Allotted Time: 90 min. **Total Points: 40**

Instructions

1. One sheet of paper (double sided, handwritten) allowed.
2. Calculators allowed.
3. MIPS reference sheet (detachable from front of your book) is also allowed.
4. -1point/min of late submission
5. Please explain all your answers clearly. No points without explanation.
6. Cross out anything that you don't want graded.

Q1. 5+5 =10 points

(a) Assuming that a cache miss penalty is essentially time to read data from main memory, cache reads being 10X faster than the memory reads, how high should the cache miss rate be for cache to be essentially useless ?

(b) A processor has a 4 entry TLB and 4KB pages. What is the maximum memory a program should access to maximize its performance ?

Solution:

(a) Let the number of instructions be N
Let's say cache read takes c cycles, then memory read would take 10c cycles
For the cache to be useless, let the cache miss rate be m

$10c * N = c*N + m*N*10c$
$10 = 1 + 10m$
$m = 0.9$

(b) To maximize performance, it should always be a hit in the TLB. For that to happen, no more than 4 distinct pages should be used by the program since anything more than 4 would result in a TLB miss.
Therefore, maximum memory = 4*4KB = 16KB

Q2. 4+6 =10 points

Assume a virtual memory system that has a page table and 16 byte pages. The total physical memory in the system is 4096 bytes (i.e., a 12 bit physical address), however each process only has access to 256 bytes (i.e., an 8 bit virtual address). A memory dump of a portion of physical memory is given below, along with a page table.

Physical Address	*Data (bytes)*															
080	2A	1B	0C	3D	5A	02	13	12	22	13	4A	0B	10	21	21	12
090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0C0	02	1B	A1	2C	11	31	22	33	11	12	14	2B	11	2B	15	13
0D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0E0	02	01	03	04	11	01	01	0B	11	10	12	13	00	03	11	0B
0F0	00	00	01	10	00	00	00	00	00	00	00	00	00	00	00	00
100	1A	2B	3C	4D	5E	01	10	02	20	03	40	0A	11	22	01	10
110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
130	01	1A	A0	2B	10	21	12	13	01	02	04	2A	01	1B	02	03
140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
160	01	01	02	03	10	01	02	0A	10	11	02	03	00	02	10	0A
170	01	02	02	01	01	02	03	04	00	00	00	00	00	00	00	00

Page Table:

0	08
	09
	0A
	0B
	0C
	0D
	0E
	0F
	10
	11
	12
	13
	14
	15
	16
P	17

(a) Give the physical address of virtual address 70

(b) What is the decimal value of the 32 bit integer at the virtual address EC? What assumptions did you make in arriving at your answer?

Solution:

(a) Virtual address = 0x70 = 0111 0000. Here, 0111 = virtual page number and 0000 = page offset.
In the page table, we look at index (virtual page number) 0x7 = 0111. That entry in the page table holds physical page number 0x0F.
The page offset from the virtual address is concatenated with the physical page number to get the final physical address
Final physical address = physical page number + page offset = 0x0F0

(b) Virtual address EC – translates to physical address 16C
If you assume big endian representation – then 16C stores 0x0002100A = 135178
If you assume little endian representation – then 16C stores 0x0A100200 = 168821248

Q3. 5 + 4 + 6 = 15 points

You have a 2-way set associative L1 cache that is 8KB, with 4-word cache lines. Writing data to L2 takes 10 cycles. You get the following sequence of writes to the cache -- each is a 32-bit address in hexadecimal:

0x1000 – 0001 0000 0000 0000 – set 0 (miss)
0x1004 – 0001 0000 0000 0100 – set 1 (miss)
0x1010 – 0001 0000 0001 0000 – set 4 (miss)
0x11c0 – 0001 0001 1100 0000 – set 112 (miss)
0x2000 – 0010 0000 0000 0000 – set 0 (miss)
0x21c0 – 0010 0001 1100 0000 – set 112 (miss)
0x3400 – 0011 0100 0000 0000 – set 0 (miss, evict 0x1000 if LRU; evict 0x2000 if MRU)
0x3404 – 0011 0100 0000 0100 – set 1 (miss)
0x3f00 – 0011 1111 0000 0000 – set 192 (miss)
0x2004 – 0010 0000 0000 0100 – set 1 (miss, evict 0x1004 if LRU; evict 3404 if MRU)
0x1004 – 0001 0000 0000 0100 – set 1 (hit if MRU, miss if LRU – evict 0x3404)

(a) How many cache misses occur with an LRU policy?
(b) How many cache misses occur with a most-recently used policy?
(c) How long does a read-miss eviction take if the cache is write-back, write-allocate? What about a write-miss? (Assume the cache line is dirty)
How long does a read-miss eviction take if the cache is write-through, write-allocate? What about a write-miss?

Solution:

(a) 11 misses for word addressable scheme - there is only word offset (last two bits of the address in red). The index would be 8 bits (shown in blue) because:
Total cache size = 8KB
Cacheline size (size of each cache block) = 4words = 4*4B = 16B
Total number of cache blocks = 8KB/16B = 512 blocks
Number of blocks per set = 2 (2-way set associative)
Number of sets = 512/2 = 256 = 2^8
Therefore, number of index bits = 8.
(b) 10 misses
(c) 10 cycles for both read-miss eviction and write-miss eviction. This is because eviction penalty = number of cycles needed to write the old dirty data that is being replaced in L1 back into L2. Since, this is a write-back cache, the dirty line in L1 that needs to get replaced in order to bring in the new data into L1 has to be written back into L2.
(d) No eviction penalty for both read-miss and write-miss. This is because, the cache is write through. So the cacheline that is getting replaced in L1 has another copy in L2, so there is no need to write this cacheline back into L2.

Q4. 5 points

Schedule the program below on a 2 issue pipeline. The pipeline has two independent datapaths which can execute any two independent arithmetic instructions. Assume all needed operands are already in registers before the program starts. Assume no register renaming.

a = b+c – I1
e = a+d – I2
a = b-c – I3
f = a+d – I4

Solution:

P1 – I1:	IF	ID	EX	MEM	WB																				
P2 – NOP:	-	-	-	-	-																				
P1 – I2:										IF	ID	EX	MEM	WB											
P2 – I3:										IF	ID	EX	MEM	WB											
P1 – I4:																				IF	ID	EX	MEM	WB	
P2 – NOP:																				-	-	-	-	-	