Homework 3

- 5.3
    1. $2^{4-0+1-2} = 8$.
    2. $2^{9-5+1} = 32$.
    3. $1 + 22/(8 \times 32) = 1.086$.
    4. 4 blocks.
    5. $4/12 = 0.33$.
    6. Steps:

Address	Index	Tag	Replace	Hit
0	00000	00	0	0
4	00000	00	0	1
16	00000	00	0	1
132	00100	00	0	0
232	00111	00	0	0
160	00101	00	0	0
1024	00000	01	1	0
30	00000	00	1	0
140	00100	00	0	1
3100	00000	11	1	0
180	00101	00	0	1
2180	00100	10	1	0

$\langle$ 00000, 11, mem[3072:3104] $\rangle$
$\langle$ 00100, 10, mem[2176:2208] $\rangle$
$\langle$ 00101, 00, mem[160:192] $\rangle$
$\langle$ 00111, 00, mem[224:256] $\rangle$

- 5.6
    1. P1: $1/(0.66 \times 10^{-9}) = 1.52 \times 10^9$.
       P2: $1/(0.90 \times 10^{-9}) = 1.11 \times 10^9$.
    2. P1: $0.66 \times (1.0 + 0.08 \times \lceil \frac{70}{0.66} \rceil) = 6.31$.
       P2: $0.90 \times (1.0 + 0.06 \times \lceil \frac{70}{0.90} \rceil) = 5.11$.
    3. P1: $1.0 + 1.36 \times 0.08 \times \lceil \frac{70}{0.66} \rceil = 12.64$.
       P2: $1.0 + 1.36 \times 0.06 \times \lceil \frac{70}{0.90} \rceil = 7.36$.
       P2 is faster. $12.64 \times 0.66 = 8.34 \geq 7.36 \times 0.90 = 6.63$.
    4. Worse. $0.66 \times (1.0 + 0.08 \times (\lceil \frac{5.62}{0.66} \rceil + 0.95 \times \lceil \frac{70}{0.66} \rceil)) = 6.50 \geq 6.31$.
    5. $1.0 + 1.36 \times 0.08 \times (\lceil \frac{5.62}{0.66} \rceil + 0.95 \times \lceil \frac{70}{0.66} \rceil) = 13.04$.
    6. P2 is still faster. $13.04 \times 0.66 = 8.61 \geq 7.36 \times 0.90 = 6.63$.
       For P1 to match P2's performance:

$0.66 \times (1.0 + 1.36 \times \text{MR} \times (\lceil \frac{5.62}{0.66} \rceil + 0.95 \times \lceil \frac{70}{0.66} \rceil)) = 6.63$
$\Rightarrow \text{MR} = 6\%$.

- 5.11
  1. Steps and the final contents of TLB.
     - Steps:

Address	Virtual Page	TLB Hit	PT Hit	PF
4669	1	0	1	1
2227	0	0	1	0
13916	3	1	-	-
34587	8	0	1	1
48870	11	0	1	0
12608	3	1	-	-
49225	12	0	0	-

     - The final contents of TLB:

Valid	Tag	Physical Page
1	12	15
1	8	14
1	3	6
1	11	12

  2. Steps and the final contents of TLB.
     - Steps:

Address	Virtual Page	TLB Hit	PT Hit	PF
4669	0	0	1	0
2227	0	1	-	-
13916	0	1	-	-
34587	2	0	1	1
48870	2	1	-	-
12608	0	1	-	-
49225	3	1	-	-

     - The final contents of TLB:

Valid	Tag	Physical Page
1	2	13
1	7	4
1	3	6
1	0	5

     - Advantages:
       1. Lower TLB miss rate because each TLB entry can keep track of larger amount of memory.
       2. Smaller page table size because the number of entries for the page table is reduced.

- Disadvantages:
    1. Increase internal fragmentation because larger amount of memory within a page is never used and, thus, is wasted.
    2. Each read/write between memory and disk is more costly because of large page size.

3. - 2-way set associative:

Valid	Tag	Physical Page	Index
1	6	15	0
1	1	6	1
1	4	14	0
1	5	12	1

   - Direct mapped:

Valid	Tag	Physical Page
1	3	15
1	0	13
1	3	6
1	0	6

   TLB maps a virtual page to a physical page directly without examining the page table residing in the memory/cache.
   If there were no TLB, the time needed for accessing physical page numbers would increase significantly.
4. $5 \times 2^{32-13} \times 4/2$ = 5242880 bytes = 5 MB.
5. A virtual address is divided into 3 segments: 8 bits (1st level) + 11 bits (2nd level), 13 bits (page offset).
   Minimum of 1st level: $5 \times 256 \times 6/2$ = 3840 bytes.
   Minimum of 2nd level: $5 \times 2^{11} \times 2^8 \times 4/2$ = 5242880 bytes = 5 MB.
   Maximum of 1st level: $5 \times 256 \times 6$ = 7680 bytes.
   Maximum of 2nd level: $5 \times 2^{11} \times 2^8 \times 4$ = 10485760 bytes = 10 MB.
6. A virtual address is divided into 2 segments: 19 bits (tag) + 13 bits (page offset).
   A physical address is divided into 4 segments: 18 bits (tag) + 11 bits (index) + 1 bit (block offset) + 2 bits (byte offset).
   It is impossible to make such cache because 14 bits ≥ 13 bits.
   A 2-way associative cache is needed to increase the data size from 4 KiB to 16 KiB.

- 5.13
    1. 0 hit.
    2. 1 hit.
    3. 0 or 1 hit.
    4. MRU policy maximizes the number of hits. 1 hit.
    5. The optimal solution is to remove the address that will cause the fewest misses in the future. However, it is impossible to implement such policy since a cache controller can only predict future from cache replacement history.
    6. If caching an address with limited temporal locality causes the eviction of another block, that

address should not be cached. However, a cache controller can never know whether the pattern of address reference will exhibit temporal locality. Hence, whether to cache an address or not can have unpredictable effect on miss rate.

- 5.15
  1. $1.5 + 120/10000 \times (15 + 175) = 3.78$.
     Double: $1.5 + 120/10000 \times (15 + 175 \times 2) = 5.88$.
     Half: $1.5 + 120/10000 \times (15 + 175/2) = 2.73$.
     10% degradation: $1.5 + 120/10000 \times (15 + x) = (1.5 + 120/10000 \times 15)/0.9 \Rightarrow x = 15.6$ (cycles).
  2. Non-virtualized: $1.5 + 30/10000 \times 1100 = 4.80$.
     Virtualized: $1.5 + 120/10000 \times (15 + 175) + 30/10000 \times (1100 + 175) = 7.61$.
     Virtualized & half: $1.5 + 120/10000 \times (15 + 175) + 15/10000 \times (1100 + 175) = 5.69$.
     Compared to the I/O access time (1100 cycles), performance impact to trap to VMM cost (175 cycles) is much smaller.
  3. Virtual memory gives the programmer an illusion that the memory is infinite. Virtual machines give the guest OS an illusion that it has entire control over native hardware.
     Pros of virtual memory: higher security, more programs run at the same time.
     Cons of virtual memory: more costly to switch between programs.
     Pros of virtual machines: higher security, multiple OSes run on the same machine.
     Cons of virtual machines: decreased performance due to trap to the VMM.
     Virtual memory is desired when multiple programs run at the same time.
     Virtual machines are desired when programs needs running on a different ISA, or when security is the most-concerned issue, e.g., cloud computing.
  4. Native hardware and OS should handle and satisfy special instructions requested by guest OSes. An API may be required to instruct how each non-native instruction should be executed by the native hardware/OS.
     It is possible. The emulated system can be faster if the the native system is able to optimize and generate more efficient instructions than the emulated system. Also, the benefit of the optimized instructions has to compensate for the cost to trap to VMM.