RESEARCH AND READING ASSIGNMENT

A) [10] [Max 1 page] Read the paper labeled R3 and comment on it. Make sure to specifically address these points: What is Virtualization? What are some of the main implementation issues and challenges? What techniques can be used to address them? What role do you think VMs will play in the future?

B) [10] Search the web for the following four memory technologies, DDR4 SDRAM, GDDR5 SDRAM, MRAM, and ZRAM and present your findings such as their speed, size, access time, and availability and cost in a table.

EXERCISES

1) [35] Consider a 2-way set associative cache that has 32 blocks and 16 bytes per block. Assume a 32 bit address.
   a) How many bits are needed to store the tag in the cache?
   b) For the above cache, assume LRU is used for cache replacement. Given the following address access sequence (data are shown in hexadecimal). For each memory access, identify its block offset, set index, and tag, decide whether it is a cache hit or cache miss. If it is a cache miss, also mark whether it is compulsory miss, conflict miss or capacity.

Address	Tag	Index	Offset	Hit/Miss	Type of miss
0x100					
0x104					
0x108					
0x200					
0x204					
0x410					
0x100					
0x108					
0x40C					
0x408					
0x300					
0x284					
0x280					
0x304					

   c) Calculate the miss rate for this memory access sequence.
   d) Assume that the cache is initially empty. After the above memory access sequence, how many cache blocks are occupied?
   e) If the same cache is direct mapped, how wide is the tag field?

2) [15] Consider a machine with a byte addressable main memory of 216 bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

   a) How is a 16-bit memory address divided into tag, index, and offset?
   b) Into what line would bytes with each of the following addresses be stored?

      0001 0001 0001 1011
      1100 0011 0011 0100
      1101 0000 0001 1101
      1010 1010 1010 1010

   c) Suppose the byte with address 0001 1010 0001 1010 is stored in the cache. What are the addresses of the other bytes stored along with it? d. How many total bytes of memory can be stored in the cache?

3) [10] Assume you have a system with the followings:

   First-level instruction and data cache (separate)

      Instruction cache: Direct-mapped, 4KB size with 8B blocks, 2% miss rate

      Data cache: direct-mapped, 8KB size with 8B blocks, 15% miss rate

   Second-level cache (unified)

      2-way set associative, 2MB size with 32B blocks, 10% miss rate

   40% of all instructions are data memory accesses

   First-level cache hits cause no stalls: 1 cycle

   Second-level hit time: 10 cycles

   Main memory access time: 100 cycles

   a) How many bits are used to index each of the caches?
   b) What is the total CPI?

4) [10] Assume you have: 32-bit addresses, 2KB Page size, 8MB Physical Memory Space, 4KB Cache with 4-way set associative and LRU replacement, 32 Byte Cache block size, 4-entry fully associative TLB.
   A program to be run on this machine begins as follows:

```
double A[1024];
int i, j;
double sum = 0;
for( i = 0; i < 1024; i++ ) // first loop
    A[i] = i;
for( j = 0; j < 1024; j += 16 ) // second loop
    sum += A[j];
```

The size of double is 8 bytes. Array A is located in memory starting at 0x1000 and stored in row major order. The cache and TLB are initially empty and NO pre-fetching is done. The only data memory references made by the program are those to array A.

a) How many bits are needed to specify the page offset? How many bits are needed to specify the physical page (frame) number? How many bits are needed to specify the virtual page number?
b) What is the hit rate for the cache in the first loop?
c) What is the hit rate for the cache in the second loop?
d) What is the TLB hit rate in the first loop?

5) [10] Assume the following system properties: 8KB Page size, 4-entry fully associative TLB with LRU replacement, 8 Physical frames with LRU replacement, 16-bit addresses
Initial TLB and page table states are provided as follows:

TLB

Valid	Dirty	Ref	Tag	Physical Page Number
1	0	1	2	6
1	1	1	5	1
0	0	0		
0	0	0		

Page Table

	Valid	Dirty	Ref	Physical page # or in Disk
0	1	0	1	5
1	1	1	0	3

2	1	0	1	6
3	0	0	0	Disk
4	1	0	1	7
5	1	1	1	1
6	0	0	0	Disk
7	1	0	1	2
8	1	0	0	0
9	1	1	1	4
10	0	0	0	Disk
11	0	0	0	Disk

Show the state of the system after each given address stream (continuously update). Translate the given virtual address to physical address (either binary or hexadecimal). Circle either TLB hit or miss and either Page hit or miss. If a page fault occurs and replacement is needed, find a victim page (virtual page #) and its physical page #.

a) Virtual address: 0011 0001 0010 0011
Physical address: _____
TLB: Hit / Miss
Page: Hit / Fault
Page Replacement? (VPN:__, PPN:__)

Valid	Dirty	Ref	Tag	Physical Page #

b) Virtual address: 0010 0101 0110 0111
Physical address: _____
TLB: Hit / Miss
Page: Hit / Fault
Page Replacement? (VPN:__, PPN:__)

Valid	Dirty	Ref	Tag	Physical Page #

c) Virtual address: 0111 0110 0101 0100
Physical address: _____
TLB: Hit / Miss
Page: Hit / Fault
Page Replacement? (VPN:__, PPN:__)

Valid	Dirty	Ref	Tag	Physical Page #

d) Virtual address: 0000 1010 1011 1100
Physical address: _____
TLB: Hit / Miss
Page: Hit / Fault
Page Replacement? (VPN:__, PPN:__)

Valid	Dirty	Ref	Tag	Physical Page #