Problem 4

Answer the following questions on computer architecture.

(1) When the following program in C language was compiled and executed on a processor A, the output (a) was obtained. When the same program was compiled and executed on a processor B, the output (b) was obtained. Explain the reason why the difference occurred from the viewpoint of processor architecture.

```
#include <stdio.h>

union my_uni {
  int v;
  char arr[4];
};

int main(){
  union my_uni val = {0x12345678};
  int i;
  for(i=0; i<4; i++){
     printf("0x%x\n", val.arr[i]);
  }
  return 0;
}
```

```
0x12          0x78
0x34          0x56
0x56          0x34
0x78          0x12
```

Output (a)   Output (b)

(2) Explain data-hazard and control-hazard on a pipeline processor with no forwarding mechanism, using a concrete example.

(3) Consider a 4-way set-associative cache memory that stores totally 32 kibibytes (32×2^{10} bytes) of data. The address width of the cache memory is 32 bits, and the cache line size (block size) is 64 bytes. Calculate the bit width of the cache index and that of a tag of the cache memory, respectively. Calculate also the total RAM capacity (the number of bits) for storing the tags of the cache memory.

(4) Consider a processor with an instruction cache and a data cache. Suppose that the CPI (cycles per instruction) of the processor is C when there is no cache miss on both the instruction and data caches. When there is a cache miss on any of the caches, a cache miss penalty of P clock cycles is additionally imposed. Suppose that when a program was executed on the processor, the ratio of the number of load/store instructions to the total number of executed instructions was R_{ls}. Suppose also that, for that program execution, the cache miss rate of the instruction cache was R_i, the cache miss rate of the data cache was R_d, and the IPC (instructions per cycle) of the processor was I. Express I in terms of C, R_i, R_d, R_{ls}, and P.

(1) Processor A is big-endian.

Processor B is little-endian

(2) ① data hazard happens when the result of the current instruction is not ready but the following instruction needs.

eg:
```
add $3, $1, $2
sw  $3, 0($4)
```

② control hazard happens when the instructions can't be executed until the branch condition is confirmed.

eg:
```
      beq $1, $2, Else
Else: sub $3, $4, $5
```

(3) number of blocks: $\frac{32\times 2^{10} \text{ bytes}}{64 \text{ bytes}} = 2^9$

$\Rightarrow$ 2^7 blocks / set

19	7	6

cache index: 7 bit

tag of the cache memory: 19 bit

RAM Capacity: $4\times(1+19+2^9)\times 2^7$ bit

$= 66.5GB$

(4) $$\frac{1}{I} = C + CPR_{CS}[R_i + (1-R_i)R_d]$$

$$\Rightarrow I = \frac{1}{C} \cdot \frac{1}{1+PR_{CS}[R_i+(1-R_i)R_d]}$$