

1. When a processor accepts an Interrupt event from an external device, how does the processor find the corresponding interrupt service routine to process the interrupt?

>> 먼저 프로세서는 인터럽트 처리 프로그램(Interrupt handling program)의 시작 주소를 PC에 적재한다. 컴퓨터 구조와 운영체제의 설계에 따라 인터럽트 유형에 대하여 또는 각 장치에 대하여 한 개씩의 프로그램이 존재하기도 하지만 만약 한 개 이상의 인터럽트 처리 루틴들이 있다면, 프로세서는 그들 중에서 어느 것을 실행해야 하는지를 결정해야 한다. 이에 대한 정보는 원래의 인터럽트 신호에 포함되어 있는 경우도 있고 프로세서가 인터럽트를 발생한 장치로 요구(request)를 보내어 필요한 정보를 포함하고 있는 응답을 받기도 한다.

2. A DMA module is transferring characters to memory using cycle stealing, from a device transmitting at 9600bps. The processor is fetching instructions at the rate of 1 MIPS. By how much will the processor be slowed down due to the DMA activity? Assume the processor only access memory to fetch instructions.

>> 프로세서가 IF를 위해서만 메모리에 접근한다고 가정했을 때, 프로세서는 매 *ms* 당 메인 메모리에 한번 씩 접근해야 한다. DMA 모듈은 매초  $9600/8 = 1200$ 개의 문자들을 전송하거나 (1MIPS) 당 하나씩 문자를 전송한다. 따라서 DMA는 833번째 cycle 마다 "steal"을 한다. 따라서 프로세서의 속도는 대략  $\frac{1}{833} \times 100\% = 0.12\%$  늦어진다.

3. Convert the decimal numbers (25.25) to binary. Convert the binary number (101.101) to decimal and hexadecimal number.

32163006 이진옥  
3번  
(1) 25.25를 2진수로.  
 $25 \rightarrow \begin{array}{r} 25 \div 2 = 12 \dots 1 \\ 12 \div 2 = 6 \dots 0 \\ 6 \div 2 = 3 \dots 0 \\ 3 \div 2 = 1 \dots 1 \\ 1 \div 2 = 0 \dots 1 \end{array} = 11001$   
 $0.25 \rightarrow 0.25 \times 2 = 0.5 \rightarrow 0$   
 $0.5 \times 2 = 1.0 \rightarrow 1$   
 $\therefore 25.25 \Rightarrow 11001.01$

(2) 이진수 101.101을 10진수와 16진수로.  
① 10진수로.  
 $101 \rightarrow 4+1=5, 0.101 \rightarrow \frac{1}{2} + \frac{1}{8} = \frac{5}{8} = 0.625$   
 $\Rightarrow 101.101 \rightarrow 5.625$   
② 16진수로.  
 $101.101 \Rightarrow \underbrace{0101}_5.\underbrace{1010}_{10} \Rightarrow 5.A$

4. Represent the following decimal numbers in twos complements using 8bits: +23, -23

32163006 이진수  
4번>

① +23  $\Rightarrow 10111$       ② -23

$\Rightarrow$   $\begin{array}{r} 23 \dots 1 \\ 211 \dots 1 \\ \underline{215 \dots 1} \\ 22 \dots 0 \end{array}$        $\Rightarrow +23 = 00010111$   
 $\Rightarrow 00010111$   
 $\Rightarrow 11101000$   
 $\Rightarrow 11101001$

$\Rightarrow 00010111$

5. Assume numbers are represented in 5-bit 2's complement representation. Show the calculation of the followings:

- 1)  $6 + 10$       2)  $-6 + 10$       3)  $6 - 10$       4)  $-6 - 10$

32163006 이진수  
5번>

1)  $6 + 10$       2)  $-6 + 10$        $C_1=1, C_2=1 \Rightarrow \text{Overflow X}$

$6 \Rightarrow 00110$        $6 \Rightarrow 00110$   
 $10 \Rightarrow 01010$        $-6 \Rightarrow 11001$   
 $\Rightarrow$   $\begin{array}{r} 00110 \\ +01010 \\ \hline 10000 \end{array}$        $\begin{array}{r} 00110 \\ +11001 \\ \hline 11010 \\ +01010 \\ \hline 100100 \end{array}$

3)  $6 - 10 = 6 + (-10)$       4)  $-6 - 10 = (-6) + (-10)$

$6 \Rightarrow 00110$        $-6 \Rightarrow 11010$   
 $-10 \Rightarrow 10110$        $-10 \Rightarrow 10110$   
 $\Rightarrow$   $\begin{array}{r} 00110 \\ +10110 \\ \hline 11100 \end{array}$        $\Rightarrow$   $\begin{array}{r} 11010 \\ +10110 \\ \hline 110000 \end{array}$   
 $\Rightarrow$   $\begin{array}{r} 110000 \\ +10110 \\ \hline 110000 \end{array}$

6. Express the following numbers in IEEE 32bit floating-point format

- 1) -1.5      2)  $1/16$

32163006 이진수

1) -1.5

$-1.5 \Rightarrow -1.1 \times 2^0 \rightarrow 0 + 127 = 127$

$\Rightarrow 10111111100 \dots 00$

2)  $1/16$

$1/16 = 0.0625$

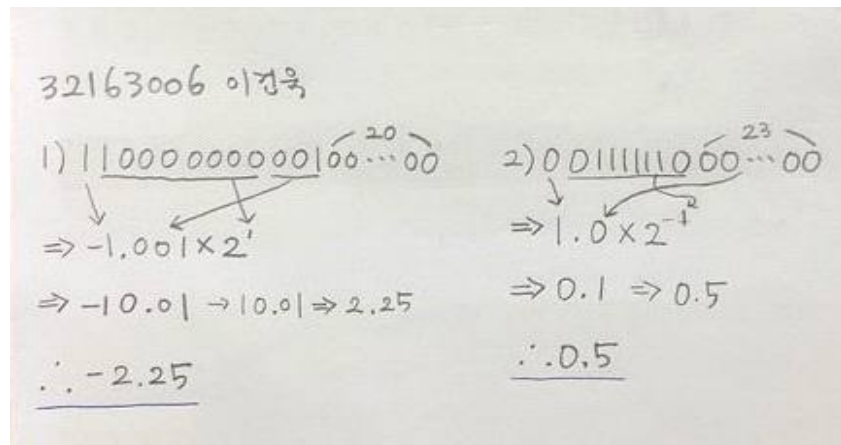
$\Rightarrow 0.0001 \Rightarrow 1.0 \times 2^{-4} \rightarrow -4 + 127 = 123$

$\Rightarrow 00111101100 \dots 00$

7. What is the equivalent decimal value of the following IEEE 32bit floating-point representation?

1) 1 10000000 001000000000000000000000

2) 0 01111110 000000000000000000000000



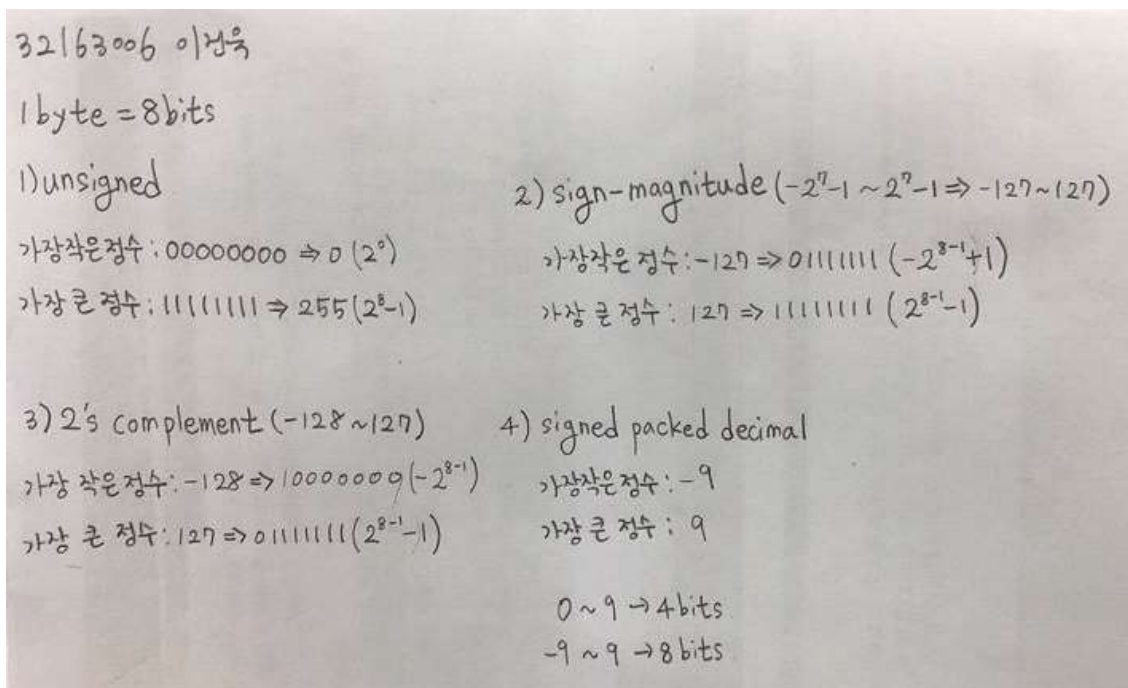
8. A given processor has words of 1 bytes. What is the smallest and largest integer that can be represented in the following representations:

1) unsigned

2) sign-magnitude

3) 2's complement

4) signed packed decimal



9. Do the calculation of adding -8 and -9. Assume numbers are represented in 5-bit 2's complement representation. Show the following flags after the addition.

- 1) C                      2) O                      3) S                      4) Z

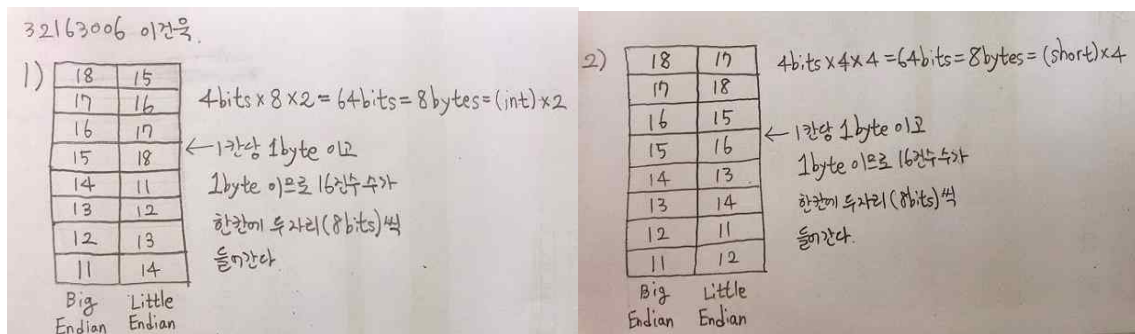
>> 2의 보수로 -8은 11000, -9는 10111이다. -8과 -8를 더하는 연산을 수행하면  $11000 + 10111 = 101111$ 이 나와 범위를 벗어난 1(Carry)로 인해 1) Carry Flag가 1로 설정되고 연산결과에서 overflow가 발생했으므로 2) Overflow Flag 역시 1로 설정된다. 음수와 음수를 더하는 연산하기 때문에 결과가 음수가 나와 산술 논리 연산의 결과가 음수일 경우 1로 설정되는 3) Sign Flag 역시 1로 설정되어야 하지만 Overflow 발생으로 인해 음수값이 아닌 양수값이 나왔으므로 3) Sign Flag는 켜지지 않는다. 마지막으로 4) Zero Flag 역시 결과값이 0은 아니므로 켜지지 않는다.

∴ 1) C : 1                      2) O : 1                      3) S : 0                      4) Z : 0

10. For the following data structure, draw the big-endian and little-endian layouts.

- 1) struct {  
     int i; //0x11121314  
     int j; //0x15161718  
 } s1;

2) struct {  
     short i; //0x1112  
     short j; //0x1314  
     short k; //0x1516  
     short l; //0x1718  
 } s2;



11. Compare the numeric data formats of x86 and ARM processors.

#### 1) Data Types

x86은 unsigned integer(부호가 없는 정수), integer(정수), floating point(부동소수)를 모두 표현하는 반면 ARM은 전력과 공간 절약을 위해 floating point(부동소수) 하드웨어를 제공하지 않는다.

#### 2) Endian

x86은 little-endian 형식을 사용하여 최하위 바이트가 가장 낮은 주소에 저장되도록 하고 ARM은 시스템 설계자들을 위해 효율적인 동적 데이터 적재/저장이 가능하도록 E-비트가 어떤 endian(E-bit = 0 > little-endian, E-bit = 1 > big-endian)으로 데이터를 적재 혹은 저장하는지를 정의해준다.

## 12. List and explain five important fundamental issues in designing Instruction Set.

- 연산 종류(Operation repertoire) : 얼마나 많은 그리고 어떤 연산들이 제공되는지, 그리고 얼마나 복잡한 연산들이 제공되는지
- 데이터 유형(Data types) : 연산이 수행될 데이터의 여러 가지 유형들
- 명령어 형식(Instruction format) : 명령어의 길이(비트 수), 주소의 개수, 여러 필드들의 크기 등
- 레지스터(Register) : 명령어들에 의해 참조될 레지스터들의 수 및 그들의 용도
- 주소지정(Addressing) : 오퍼랜드의 주소가 지정되는 방식들