

Report #1

32202546 안지성

1. Find out the specifications such as Processor, memory(RAM) and OS of your computer

Processor : Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz

Memory(RAM) : DDR4 Type 16 GB

OS : 64비트 운영 체제, Windows 10

2. Explain the internal structures of IAS computer (in Figure 1.6)

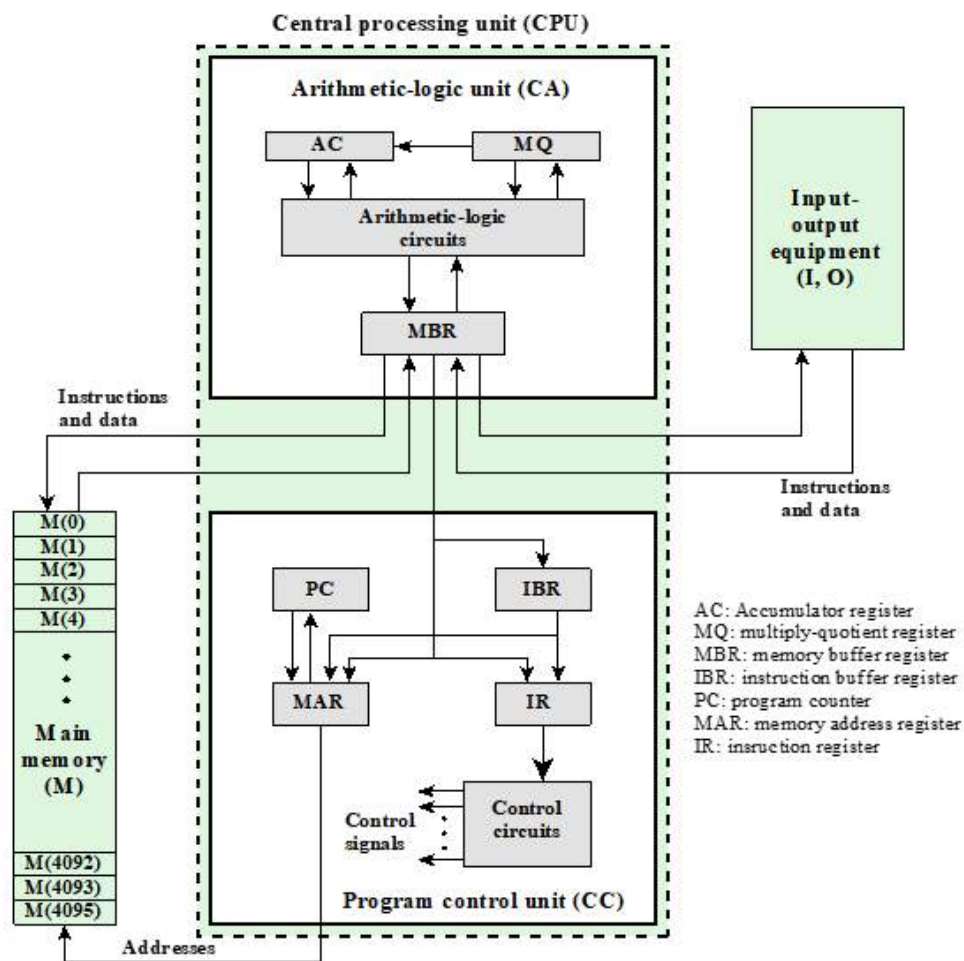


Figure 1.6 IAS Structure

-> IAS 컴퓨터는 현재 우리의 컴퓨터 구조와 매우 흡사한 구조를 가지고 있다.

우선 현재 컴퓨터에 있는 CPU와 같은 명칭은 아니지만 비슷한 구조를 가지고 있다.

Arithmetic-logic unit(CA)와 Program control unit(CC)가 합쳐져있는 형태이다.

그리고 Main memory와 Input-output equipment(I.O)로 이루어져 있다.

CA에는 우선 연산을 처리하는 ALU인 Arithmetic-logic circuits가 있다. 그리고 곱하기같은 연산을 처리하는 MQ 레지스터, 데이터를 저장하는 AC 레지스터, 메모리로부터 데이터를 가져와 저장하는 MBR 레지스터가 있다.

CC에는 제어신호를 내어서 기계 명령어들을 순차적으로 실행하게 하는 제어장치(Control circuits)와 메모리의 주소를 가져와서 저장하는 MAR 레지스터, 프로그램을 수행할 때 메모리에서 가져와야하는데 어디에서 가져올지 주소를 항상 미리 계산해서 집어넣는 PC가 있다. 또한 수행할 기계 명령어를 가져와서 저장하고 수행하기 위해서 IBR 레지스터에 하나, MAR과 IR 레지스터 한 묶음에 또 하나를 집어넣는다.

Main memory에는 주소 0번지부터 4095번지까지 4096개(4K)의 주소를 가지고 있다. 그리고 하나의 주소에는 40bit(5byte)씩 저장이 된다.

IO장치는 기계 명령어로 동작시킨 후 MBR 레지스터에 의해 데이터를 얻어온다.

3. Explain how many instructions are stored in a memory location of IAS computer. Explain the fetch process.

-> 하나의 주소에는 총 40bit가 저장되는데, 데이터를 저장할 때는 1bit는 양수, 음수를 구분하는 sign bit이고 나머지 39bit는 숫자의 크기를 나타낸다. 프로그램을 저장할 때는 한 개의 기계 명령어의 크기는 20bit이므로 명령어 2개를 저장할 수 있습니다. 한 개의 명령어 20bit를 저장할 때, 8bit는 opcode를 저장하고 12bit는 operand address를 저장한다.

fetch process는 다음과 같다.

우선 프로그램을 실행하려면 메모리에서 실행할 프로그램을 가져와야한다. 그런데 메모리 몇 번지에 가서 가져와야할지 다음 수행할 명령어의 주소를 미리 가지고 있는 PC에서 그 주소를 MAR로 옮긴다. 그 다음에 메모리에서 MAR에 저장된 주소에 있는 명령어를 요청하고 MBR에 가져온다. 그리고 MBR에 있는 메모리에서 읽어온 수행할 명령어를 수행하기 위해서 IR로 옮긴다. 그리고 명령어를 메모리에서 가져와서 수행할 때는 항상 이 작업을 반복한다. 그렇지만 이 작업을 반복하면 똑같은 명령만 수행하므로 PC에 있는 주소를 한번 반복할 때마다 다음 수행할 명령어 주소로 바꿔야 한다. 그리고 이 가져와서 수행하려는 명령어를 interpret해서 컴퓨터가 이해하는 명령어로 변환시켜 수행한다. 이 모든 작업이 fetch process이다.

4. Explain where arithmetic means and harmonic means are applied respectively, and give some examples.

-> 우선 산술평균과 조화평균은 컴퓨터의 수행능력을 체크해보는 benchmark시에 사용할 수 있다.

산술평균은 대체적으로 실제 측정하는 시간일 때 좋은데 실행시간 평균치를 구할 때 사용한다. $[AM = a+b/2]$

조화평균은 Throughput(처리율) 평균치를 구할 때 사용합니다. $[HM = 2ab/a+b = GM^2/AM]$

A에서 B까지 거리를 갈 때, 첫 번째 갈 때는 4시간이 걸리고 두 번째 갈 때는 2시간이 걸린다면 이때 평균 시간은 얼마나 걸리는지 계산할 때 산술평균을 구한다.

A에서 B까지 거리를 갈 때, 총 60km인데 첫 번째 갈 때는 4시간 걸려서 시속 15Km/h이고

두 번째 갈 때는 2시간 걸려서 시속 30Km/h인데 평균 속력은 얼마나 걸리는지 계산할 때 조화평균을 사용한다. 이때 여기서 산술평균을 사용하면 안된다. 그 이유는 단위시간당 한 일 즉, 처리율이기 때문이다. 이때는 가중치 있는 산술평균을 구해야한다. 이를 위하여 조화평균을 사용한다.

5. Suppose that 80% of the computing time is consumed by arithmetic operations in a system. If the arithmetic module of a system is speedup by a factor of 4, Then what is the maximum speedup of this system?

-> Amdahl's law를 사용하여 계산할 수 있다. 이 법칙은 컴퓨터 시스템의 일부를 개선할 때 전체적으로 얼마만큼의 최대 성능 향상이 있는지 계산할 때 사용한다. 식은 다음과 같다.

$$T = (1-f)T + fT$$

이 법칙에 따르면 어떤 시스템을 개선하여 전체 작업 중 80%의 부분에서 4배의 성능이 향상되었을 때 전체 시스템에서 최대 성능 향상은 다음과 같다.

$$\text{Speedup} = 1/((1-f)+f/n)$$

$f=0.8$, $n=4$ 를 각각 대입하면 $\text{Speedup} = 1/((1-0.8)+0.8/4) = 2.5$ 라는 값이 나온다.

따라서 최대 성능은 2.5배만큼 증가했다.

6. Generally, a computer operates by repetitively performing an instruction cycle, which consists of two sub cycles: a fetch cycle and an execute cycle. Describe in English the tasks accomplished during the fetch cycle and the execute cycle.

As mentioned above, the instruction cycle consists of two sub-cycles.

It's a Fetch cycle and an execution cycle.

In detail it is divided five processes.

► **Fetch cycle**

- **Fetch** : fetching data from the memory
- **Decode instruction** : interpreting what instruction should be performed

► **Execute cycle**

- **Fetch Operand** : fetching data required to perform the program
- **Execute** : executing instruction
- **Store Result** : storing at the memory

7. Compare a hardwired program in the customized hardware and a software program in the general purpose hardware.

-> hardwired program은 수정 시 일일이 뜯어 고쳐야 한다. 반면에 software program은 수

정이 필요할 때 program 내부만 수정하고 hardware는 뜯어고칠 필요가 없다.

8. Explain why contemporary systems rely on point-to-point interconnection rather than shared bus.

-> Bus는 공유된 선들의 집합으로 synchronous bus와 asynchronous bus로 나뉜다. synchronous는 동기식 전송이라는 뜻으로 시간을 맞춰 전송하는 방법으로 빠르다. asynchronous는 비동기식 전송으로 서로 필요한 정보를 주고받아 처리한다. 그러나 신호의 frequency가 증가하면 동기화하기가 어렵다. 따라서 point-to-point interconnection, 즉 특정한 두 지점 사이 일대일로 정보를 처리하는 방법을 사용하게 된다.

9. For a program, 10 million instructions are executed on a 2.0GHz processor. Each instruction execution requires 1 clock cycle for processing of the processor and 2 clock cycles for memory references. ($G=10^9$, million= 10^6)

1) Determine the program execution time and MIPS.

-> Program Execution time (T) = $I_c * CPI * t = I_c * (p + m * k) * t$

CPI (Cycles Per Instruction) : CPU가 하나의 instruction을 실행하는 데 필요로 하는 clock cycle의 수

I_c : 수행된 기계 명령어 개수

p : 하나의 명령어가 cpu에서 수행하는데 걸린 시간

m : 메모리 접근 횟수

k : 한번 메모리에 접근하는데 걸리는 시간

clock cycle time(t) : $1/\text{clock frequency}(f)$

f는 2.0GHz processor이므로 2이다.

$I_c=10^7$, $p=1$, $m=2$, $k=@$, $t=1/2=0.5$ 를 각각 대입하면

$CPI=1+2*=@=?$

Program Execution time (T) = $10^7 * ? * 0.5$ 이다.

(k의 값을 알지 못하여 올바른 값을 구하지 못하였다.)

MIPS (Million Instructions Per Second) : 단위 시간당 처리된 기계명령어 수

식은 다음과 같다.

$MIPS = \text{instruction count} / (\text{execution time} * 10^6) = \text{명령어 개수} / ((\text{명령어 개수} * x$

$$\text{CPI} / \text{클럭속도} \times 10^6 = \text{클럭속도} / \text{CPI} \times 10^6$$

식에 값을 대입하면

$$\mathbf{MIPS} = 10^6 / (10^7 \times 0.5) \times 10^6 = 1 / (10^7 \times 0.5) \text{이다.}$$

2) If accessing memory becomes two times faster ($T_m \Rightarrow 0.5T_m$), determine the program execution time.

강의를 들으며 열심히 풀어보았지만 마지막 문제는 제게는 어려워서 해답을 구하기 쉽지 않았습니다. 강의에서 예제 문제를 다뤄보았다면 더 접근이 쉬웠을 거 같다는 아쉬움이 있었습니다. 해설을 들으면서 다시 풀어보도록 하겠습니다. 오늘도 행복하고 평안한 하루 되세요!