1. 내가 사용하고 있는 LG 그램 15인치 노트북은 인텔 코어 i-5 프로세서를 사용하고 있어 코어 수는 4개이고 최대 클럭 속도는 2.40GHz까지 올라갈 수 있다. 또한 RAM 용량은 8.00GB 이며 윈도우 운영체제를 사용하고 있다.

2. IAS 컴퓨터의 내부 구조는 크게 I/O 장치들과 메인 메모리, CPU로 구성되어져 있다. I/O는 입출력 장치로 외부에서 데이터를 받아들이거나 외부로 데이터를 출력한다.

메인 메모리는 프로그램이나 데이터를 저장하는 장소인데 4096개의 주소가 있고 각 주소마다 40bit의 기계 명령어들을 저장된다.

CPU 안에는 산술 논리 연산을 처리하는 ALU(Arithmetic-logic unit)가 있고, 이 안에는 연산 처리 장치 및 ALU의 결과를 일시적으로 저장하는 산술 논리 레지스터인 AC/MQ가 있다.

또한 I/O나 메모리로부터 데이터나 명령어를 받거나, 보내기 위해 일시적을 저장하고 있는 버퍼 레지스터인 MBR이 있다.

Program control unit에는 다음 기계 명령어 2개의 메모리 주소를 미리 계산하는 PC, 데이터나 명령어의 주소를 저장하는 레지스터 MAR(이 주소의 내용이 MBR에 저장된다),

  주소를 통해 메모리로부터 읽어온 혹은 저장할 데이터나 명려어의 일부를 저장하는 버퍼 레지스터인 IBR, 8bit의 처리될 명령어 및 데이터를 저장하는 IR로 구성되어져 있다.

그리고 명령어들을 수행하는데 필요한 제어신호를 보내는 제어장치가 있다.

3. 하나의 메모리 주소는 40bit(=5byte) 크기를 가지고 있고, 여기에는 20bit 짜리 명령어 두개가 저장된다. 하나의 명령어는 8bit 의 수행할 내용과 12bit의 처리할 데이터의 주소들을 담고 있다.

 2번에서 설명했듯이 미리 수행될 명령어 두개가 담겨 있는 메모리 주소가 PC 레지스터에 저장이 되어 있고 차례가 오면 이 메모리 주소를 MAR에 넘겨준다. 그리고 이 주소에 있는 실제 명령어들을MBR에 저장된다. 그리고 명령어 두 개 중 바로 처리할 명령어8bit는 IR에 나머지 한 개의 명령어는 IBR에 저장된다. 이때 PC는 (보통)1 증가하며 다음 처리할 명령어의 주소를 저장하며 다음 수행을 준비한다. 이것이 명령어가 fetch되는 과정이다.

4. 산술평균의 경우 실제 측정한 실행시간의 평균치를 구하는데 사용이 되고 총 걸린 시간/2 로 구하고, harmonic mean은 처리율의 평균을 구하는데 사용되며 총 이동거리/총 걸린 시간으로 구한다.

예를 들어 이동거리 60km를 15km 시속으로 4시간이 걸려 갔다가, 30km 시속으로 2시간이 걸려 오는 처리과정이 있다고 가정하면, 산술평균 즉, 실행 시간 평균은 6/2=3시간이고, 처리율 평균은 2\*60/6 =20Km/h 로 구할 수 있다.

5. Amdahl's law의 speedup 공식을 사용하여 구할 수 있다. 1/(0.2+0.8/4) = 10/4 =5/2 로 최대 5/2배가 빨라진다.

6. Approximately fetch cycle is to bring instruction from CPU for needed operation and execute cycle is to execute instruction.

 In fetch cycle progress, bring instruction address be calculated from PC before and then store/ load instruction.

 Next progress is to decode instruction and fetch other needed operand.

 In execute cycle progress, operate data and for result store calculate address and store result finally.

7.  산술 기능을 하는 프로그램을 설계했다고 가정했을 때 Hardwired 프로그램의 경우 처음 설계한 순서 혹은 logic 대로만 처리가 가능하다. 즉 중간에 내가 산술 처리 방식을 + 에서 - 로 바꾸고 싶다고 해도 하드웨어 전체를 뜯어내 재설계 하지 않는 이상 불가능하다. 하지만 소프트웨어 프로그램의 경우 그때그때 필요한 명령어를 요청하면 명령어를 해석하여 하드웨어에 해당 처리에 대한 신호를 주는 방식으로 실행할 수 있다. 따라서 수정이 거의 필요없는 한정되고 고정된 특정 프로그램 수행이 필요하다면 Hardwired 프로그램이 그렇지 않다면 소프트웨어 프로그램이 적절하다.

8.  데이터가 이동할 수 있는 속도는 빨라졌지만 다양한 장치들이 함께 공유하는 기존의 시스템 버스에서는 이때 주고 받는 신호의 동기화나 순서 등을 서로 맞추기가 어려워지는 문제가 생기기 시작하며 실제 전송하는 데이터양을 증가시키거나 지연시간을 감소시키는 일에 대해 어려움에 봉착했다. 따라서 직접 장치들을 연결하여 많은 데이터를 빠르게 주고 받는 point to point 방식이 선호되고 있다.

9. clock cycle frequency = 1초에 몇 번 진동이 반복되는지, clock cycle time = 한 번 진동하는데 몇 초가 걸리는지, clock cycle = 진동 한 번을 말함

clock cycle frequency 는 1초에 한 번 진동하는게 1hz를 말한. 이 문제의 경우, 2ghz라고 했으니 1초에 2\*10^9 = 2억 번 진동하게 되고, 한 번 진동하는데 걸리는 시간은 1/2\*10^9

9-1. 프로그램 수행 시간 = 수행된 명령어 개수 \* 평균적으로 하나의 명령서 수행시 걸리는 시간 = 10^7 \* (1\*1/{2\*10^9} + 2\*1/{2\*10^9}) = 10^7 \*(3/2\*10^9) = 3/200 초의 시간이 걸린다.

 MIPS = 처리된 명령어 수(million 단위)/걸린 시간(초 단위) = 10/(3/200) = 2000/3 이다.

9-2. 메모리 접근 속도가 2배 빨라졌다면, 10^7 \* (1\*1/{2\*10^9} + 2\*1/{2\*10^9}\*1/2) 처럼 한 번 메모리를 접근하는 시간을 2로 나눠준다. 이 식을 계산하면 수행시간은 1/100 초가 나온다.