

~~Q1~~ 1 산술 연산 및 논리 연산을 수행하는 CPU 구성요소?

답: 라. ALU ALU는 산술 및 논리 연산들을 수행하는 회로들로 이루어진 H/W 모듈이다.

~~Q2~~ 2 제어 유니트의 기능이 아닌것?

답: 나. 정보의 일시저장. 정보의 일시저장은 레지스터에서 담당하는 주된 기능이다.

~~Q3~~ 3 명령어 사이클에 속하는 부사이클이 아닌것?

답: 다. 제어사이클. 명령어사이클에 속하는 부사이클에는 인출, 실행, 인터럽트, 간접 사이클이 있다.

~~Q4~~ 4 기억장치로부터 인출된 명령어 코드를 제어 유니트에 의해 해독되기 전에 일시적으로 저장하고 있는 레지스터는?

답: 다. IR IR레지스터에서 임시로 OP와 오퍼랜드 값을 가지고 이를 후에 제어유닛에 보내 해독하여 실행사이클을 시작하게 한다. 실행사이클의 흐름은 제일 처음으로 IR에서 시작함으로도 알 수 있다.

~~Q5~~ 5 클럭 주파수가 2GHz 인 CPU에서 인출 사이클에 소요되는 시간은?

답: 나. 1.5ns .1초동안 2기가헤르츠 이므로 $1/2\text{기가헤르츠} = 0.5 * 10^9\text{초}$ 는 0.5ns가 소요되고 인출사이클은 여기에 3을 곱한 1.5ns가 된다.

~~Q6~~ 6 기억장치 주소가 바이트 단위로 지정되고 명령어길이는 32비트인 CPU에서 인출되는 사이클동안 PC가 얼마만큼 증가되는가?

답: 다. 4 바이트단위 지정이고 32비트는 4바이트 이다.

2.7 인터럽트 사이클 동안 수행되는 동작이 아닌 것?

답: 라. 인터럽트 플래그를 세트한다. 이건 다중 인터럽트 사이클 중에서 플래그를 세우는 것과 우선순위를 정하는 두 가지 방법 중 하나이다.

※ 8 CPU가 어떤 장치에 대한 인터럽트를 처리하는 도중에 우선순위가 더 낮은 인터럽트 요구가 들어왔다면 어떻게 되는가?

답: 다. 현재 인터럽트 서비스 루틴의 수행을 계속한다. 우선순위 방식의 다중 인터럽트는 현재 우선순위보다 더 크지 않는 이상 현재수행중인 인터럽트를 중지하지 않는다 하물며 더 낮은 우선순위가 들어오면 당연히 나중에 이루어 질 것이다.

2.9 간접 사이클 동안 어떤 동작이 수행되는가?

답: 나. 기억장치로부터 데이터의 주소를 인출한다. 간접사이클은 직접사이클과 반대로 데이터의 주소를 한번 거쳐야 완전하게 접근 할 수 있다.

2.10

4-단계 명령어 파이프라인에서 첫 번째 명령어의 실행이 완료되는 데 걸리는 시간과 두 번째 명령어의 실행에 걸리는 시간을 올바로 짹지는 것은 어느 것인가?

답: 나. 4사이클-1사이클. 4단계 파이프라인에서 첫 명령어는 4단계에 걸쳐 완료가 되고 두 번째 명령어는 첫 명령어의 두 번째 사이클에서 인출되기 시작하기에 결론적으로 첫 명령어가 완료되고 1사이클 후에 두 번째 명령어가 실행된다.

2.11

4-단계 파이프라인에서 10개의 명령어들을 실행하는 데는 모두 몇 사이클이 소요되는가?

답: 나. 13사이클. 공식에 의해 $T=K+(N-1)$ 이고 간단히 이해하면 첫 명령어는 4단계를 거치고 나머지 9개는 1사이클 씩 소요되므로 위와같은 식이 나온다.

2.12

4-way 슈퍼스칼라 프로세서에서는 매 사이클마다 몇 개씩의 명령어들이 실행될 수 있는가?

답: 다. 4개. 4-way 슈퍼스칼라의 의미는 한번에 4개의 파이프라인을 작동 한다는 것 이므로 4개이다.

2.13

직전에 실행된 산술 명령어의 결과값이 음수이면 점프하라는 명령어가 검사할 조건비트는 어느것인가?

답: 다. S플래그. S플래그는 부호를 판단하는 플래그로 0이면 양수 1이면 음수임을 판단하는 상태레지스터 중 하나이다.

2.14 서브루틴을 호출하는 CALL명령어가 실행되는 동안 수행되는 동작이 아닌 것?

답: 라. 호출할 주소를 스택으로부터 인출한다. 이것은 리턴명령어에서 수행되어야 할 동작이므로 아니다.

2.15 기억장치 주소를 나타내는 오퍼랜드 필드가 14비트라면, 이 명령어에 의해 직접 주소지정 될 수 있는 기억장치 용량은 얼마? (단어의 길이는 8비트 이다).

답: 라. 16KByte. 솔직히 뒤에붙은 조건은 중의적으로 의아하나 기억장치 용량에 집중하면 주어진 비트가 총 14비트 이므로 2^{14} 개의 주소지정이므로 $2^{10}=K$ 인 것을 생각하면 결국 2의4승인 16KByte 이다.(단어길이가 8비트 = 1바이트 이므로)

연습 문제

2.1 클록 주파수가 2GHz(주기=0.5ns)인 CPU가 ADD addr 명령어를 인출하고 실행하는데 걸리는 시간은 몇 ns? (직접주소지정 방식사용, 파이프라이닝은 되지 않은 상태일 때)

답 : 3ns. 인출 사이클에서 3번, 실행 사이클에서 3번의 마이크로 연산이 일어나므로 $0.5 \times 6 = 3$ 이다.

2.2 아래와 같은 어셈블리 프로그램에 대한 기계어 코드들이 200번지부터 저장되어 있다.

(1) 프로그램이 순차적으로 실행될 때 기억장치와 CPU 레지스터들의 내용이 어떤 값이 되는지 그림 2-6의 7,8 단계의 그림을 고쳐 그려서 표시하라.(기억장치 300번지와 301번지에는 각각 9,5가 저장되어 있다고 가정함)

아래 사진이 답이다.

아래와 같은 어셈블리 프로그램에 대한 기계어 코드들이 200번지부터 저장되어 있다.

주소	명령어	기계 코드
200	LOAD 300	1300
201	SUB 301	6301
202	STA 302	2302
203	JUMP 250	8250

답

	[LOAD] 300	[SUB] 301	[STA] 302	[JUMP] 250
PC	201	202	203	250
AC	9	4	4	4
IR	1300	6301	2302	8250

(2) SUB 301 명령어가 실행되는 중에 인터럽트 요구가 들어왔다면 인터럽트 사이클이 종료된 직후 CPU 레지스터들(PC, AC, IR, SP)에는 각각 어떤 값이 저장되어 있는가? (ISR 시작주소는 450번지, SP초기값 999 일 때)

답: PC: 450, AC: 4, IR: 6301, SP: 998. 인터럽트 사이클이 종료직후 기존의 AC, IR은 이전의 값을 가지고 있고 달라지는 것은 PC와 SP이다 PC는 ISR의 주소를 담고 있는 상태이고 SP는 999에 기존의 PC값인 202를 스택에 넣어놓고 1이 감소하고 위로 올라가 998이 된다.

2.3 인터럽트 사이클이 그림 2-9B 와 같이 종료된 다음에 처리되는 인터럽트 서비스 루틴에서 AC의 내용을 스택에 저장하였다. 그 값이 저장되는 기억장치의 주소는 몇 번지이며, SP의 내용은 어떤 값으로 변경되는가?

답: 999번지 스택에 저장이 되고 이후 SP는 998으로 변경된다. 2.2의 (2)번에서 살펴본 것과 같은 방식인데 이번엔 AC와 IR값이 손상이 되지 않게 스택에 저장이 된다는 사실을 알아내었다. 스택에 저장이 되는 것은 당연히 999번지에 주소와 함께 넣어 놓을 것이고 이후 SP는 감소한다(위치로는 올라간다)

2.4 ISR 수행도중에 더 높은 우선순위를 가진 인터럽트 요구가 들어와도 그 루틴의 수행이 중단되지 않도록 하는 방법?

답: 다중 인터럽트 사이클일 때 플래그를 설정하는 방식으로 진행하면 우선순위에 상관 없이 현재 수행중인 명령을 다 수행 후에 진행할 수 있다. 플래그는 0으로 해야 한다.

2.5

4-단계 명령어 파이프라인으로 1000개의 명령어들을 모두 실행하려면 몇 클록 주기가 걸리는가? 만약 이 파이프라인의 클록 주파수가 2 GHz라면, 그에 소요되는 전체시간은 몇ns 인가?

답 : 1003클록주기, 501.5ns. 사이클은 공식 $T=K+(N-1)$ 에서 $K=4$, $N=1000$ 으로 쉽게 구할수 있다. 다음은 소요되는 시간이다 클록 주파수가 2 GHz 이므로 1클록당 0.5ns 가 소요된다 따라서 전체 (1003클록 X 0.5)가 전체 소요시간 이다.

2.6

마이크로프로세서가 12-단계 명령어 파이프라인으로 구성되어 있고, 클록 주파수가 4 GHz일 때. 1000개의 명령어를 실행하는데 걸리는 시간과 속도 향상(Sp)과 효율($E = \text{속도향상}/\text{파이프라인 단계수}$) 구하라

답 : 걸리는 시간 = 252.75 ns, 속도향상(Sp) = 약 11.86, 효율(E) = 약 0.98

위와 같은 방식으로 걸리는 전체 클록과 각 클록마다 걸리는 시간을 곱하여 걸리는 시간을 구할 수 있다. $K=12$, $N=1000$ 이므로 총 1011 클럭이고 4GHz 이므로 각 클럭당 소요되는 시간은 0.25ns 이다 $0.25 * 1011 = 252.75$ ns이고, 속도 향상Sp 는 파이프 라인이 없을 때 전체 클록수 / 파이프 라인이 있을 때 전체 클럭 이다. $KN/K+N-1$ 을 계산하면 약 11.86이다. 효율은 문제에 나온 것 토대로 계산을 하면 0.98이 나온다.

클록 주파수가 2GHz 인 5-단계 파이프라인에 대하여

- (1) 명령어의 수 $N = 10, 100, 1000, 10000$ 개를 처리하는데 걸리는 각각의 시간구하기

답 : 7, 52, 502, 5002 ns 이다. $K=5, 1$ 클록당 걸리는시간 = 0.5ns를 이용하여 우선 전체 클록 수를 구하면 순서대로 14, 104, 1004, 10004 이다 여기에 0.5를 곱하면 답이 나온다.

- (2) 파이프라이닝 되지 않은 경우에는 각 명령어의 실행에 걸리는 시간이 $0.5 \times 5 = 2.5$ ns 가 된다. 파이프라이닝을 이용함으로써 얻게되는 속도 향상을 각 N 값에 대하여 구하라.

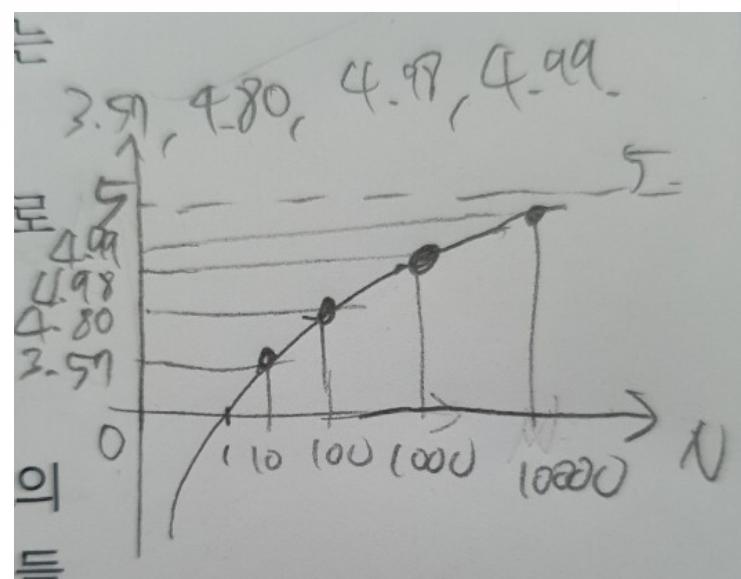
답 : 순서대로 3.57, 4.80, 4.98, 4.99 이다.

위에서 주어진 조건은 사실상 속도 향상 식에서 보면 다음과 같다.

$Sp = KN * (\text{클록당시간}) / (K+N-1) * (\text{클록당시간})$ 에서 $0.5 \times 5 = 2.5$ ns 는 결국 $K * (\text{클록당시간})$ 을 미리 계산한것과 같다. 따라서 전체 식인 $Sp = KN * (\text{클록당시간}) / (K+N-1) * (\text{클록당시간})$ 에서 공통인수인 (클록당시간)을 약분해 도 지장이 없게 된다. 즉 $Sp = KN / K+N-1$ 이 유지되고 이를 계산하면 된다. 결과로 순서대로 $50/14, 500/104, 5000/1004, 50000/10004$ 가 나온다.

- (3) 위 2번의 결과에 대한 그래프 그리기 x축은 명령어의 수 N으로 하되 log 10N의 눈금을 사용하라

답 :



CPU 명령어 파이프라인이 4단계로 이루어져 있고 1GHz의 공통 클록이 사용되고 있다.

이 CPU가 처리할 프로그램이 1000개의 명령어 들로 이루어져 있는데, 그 중 10%는 파이프라인의 두 단계만 실제 필요하고, 40%는 세 단계만 필요하며, 나머지는 네 단계 모두가 필요할 때

(1) 파이프라인을 이용하지 않은 경우, 이 프로그램을 처리하는데 걸리는 시간을 구하라

답 : 1000ns. 파이프라인을 사용하지 않을 경우 1×1000

이고 1GHz의 공통 클록이므로 1개 클럭당 1ns가 걸리므로 1000ns이다.

2.9 어떤 CPU에서 명령어 실행 과정이 네 개의 사이클들로 이루어진다고 하자.
첫 번째 사이클의 처리에 걸리는 시간이 0.5ns 두 번째와 세 번째 사이클은 1.0ns씩 그리고 마지막 사이클은 0.75ns가 각각 걸린다고 하자.

(1) 한 명령어를 실행하는 데 걸리는 시간은?

(2) 각 사이클을 처리하는 하드웨어 모듈을 독립적인 파이프라인 단계로 구현한다면, 네 개의 단계들로 이루어지는 명령어 파이프라인 구조가 된다. 파이프라인의 모든 단계들은 공통의 파이프라인 클록에 동기되어 수행된다. 파이프라인 클록의 주기[ns]와 주파수[GHz]를 각각 구하라.

(3) 이 파이프라인으로 한 개의 명령어를 실행하는 데 걸리는 시간을 구하고, (1)번의 결과와 비교하라.

(4) (1)번과 (3)번의 조건에서 100개의 명령어들을 실행하는 데 걸리는 시간을 각각 구하고, 파이프라인을 이용함으로써 얻게 되는 속도 향상(Sp)을 구하라.

$$(1) 0.5 + 1.0 + 1.0 + 0.75 = 3.25 \text{ ns}$$

$$(2) 1 \text{ ns}, 1 \text{ GHz}$$

$$(3) 4 \text{ ns}$$

$$(4) \frac{\frac{3.25}{10^3}}{10^3} = 3.15 \dots$$

$$\frac{3.25}{10^3} \times 10^3 < 10^3 \text{ ns}$$

$$4 + 99$$

(2) 파이프라인을 이용하여 이 프로그램을 처리하는데 걸리는 시간을 구하라.

답 : 1006ns. 100개의 명령어는 2단계 파이프라인, 400개는 3단계, 나머지 500개는 4단계를 따로 구한뒤 모두 더한다 각각 101, 402, 503 개의 클록이고 1클럭당 1ns이므로 모두 더하면 답이 나온다.

(3) 파이프라인으로 얻어진 속도향상과 효율을 구하라.

$\frac{9}{1} \quad \frac{3}{3}$
답 : Sp=8.93, E=2.979. 명령어마다 각각의 파이프라인이 필요한 수가 있으므로 각각의 속도향상을 구하고 더하는 방식으로 한다. 2개, 3개, 4개 파이프라인 순서대로 속도향상은 $200/101, 1200/402, 2000/503$ 를 모두 더한다. 효율은 앞서 구한 각각의 속도향상을 파이프라인수 만큼 나누면 각각 0.990, 0.995, 0.994이고 모두 더하면 2.979가 된다(근사값이다.)

2.10 그림 2-12와 같은 4-단계 명령어 파이프라인에서 각 단계의 실제 처리 시간이 아래와 같다고 하자. [참고: ps(picosecond) = 10^{-12} ns = 10^{-12} sec]

IF 단계 : 300ps
ID 단계 : 200ps
OF 단계 : 400ps
EX 단계 : 350ps

- (1) 파이프라인 클록의 주파수는 몇 [GHz]로 결정하면 되는가?
- (2) 이 파이프라인으로 1000개의 명령어들을 순차적으로 실행하는데 걸리는 시간[ns]을 구하라.
- (3) ALU의 속도를 향상시켜 EX 단계의 처리 시간을 250ps로 단축시킨다면, 전체 파이프라인의 성능을 높이는 데 도움이 되는가? 그 이유를 설명하라.
- (4) 반대로, ALU에 새로운 기능을 추가한 결과로서 EX 단계의 처리 시간이 500ps로 길어졌다면, 파이프라인의 성능에 어떤 영향을 미치는가? 명령어 1000개를 실행하는 경우에 대하여 전체 소요 시간[ns]을 계산하고, (2)번의 결과와 비교하여 설명하라.

$$(1) 400 \times 10^{-12} \quad \frac{1}{400 \times 10^{-12}}$$

$$= 2.5 \text{ GHz}$$

$$(2) T = 4 + 999 \quad 1003 \times 400 \times 10^{-12}$$

$$= 1003 \quad = 401.2 \times 10^2 \times 10^{-12}$$

$$= 401.2 [\text{ns}]$$

(3) X, OF 단계를 단축시켜야 500ps

$$(4) \text{ 영향 } \text{이 } \text{있} \text{는 } , \quad 1003 \times 500 \text{ ps} = 501.5 \text{ ns}$$

성능 20% 강화

2.1 ④-단계 파이프라인 단계들로 구성된 명령어 실행 유니트에서 12개의 명령어 들을 처리하고자 한다. 그런데 만약 다섯 번째 명령어가 분기 명령어였다면, 분기한 후에 7개의 명령어들을 더 실행하여 모두 12개의 명령어들을 실행하는 데는 전체적으로 몇 사이클이 걸리는가?

2.11

$$4 + (12-1) \times 3 \\ = 18 \text{ 사이클}$$

2.12 다섯 개의 파이프라인 단계들로 이루어진 명령어 실행 유니트가 있다.

- (1) 100개의 명령어들을 실행하는 데 소요되는 시간(클록 주기의 수)을 구하라.
- (2) 만약 파이프라인을 한 개 추가하여 2-way 슈퍼스칼라 구조로 변경한 경우에 100개의 명령어들을 실행하는 데 걸리는 시간을 구하고, (1)번 결과와 비교한 속도 향상을 구하라. 단, 동시에 실행되는 명령어들 간에 데 이터 의존성은 존재하지 않는 것으로 가정한다.

2.12

$$(1) T = 5 + (100 - 1) \times 4$$

$$(2) T = 5 + (100 - 2) / 2 = 54 \\ \frac{104}{54} = 1.9\dots$$

2.13 네 개의 5-단계 명령어 파이프라인들로 이루어진 4-way 슈퍼스칼라 프로세서를 이용한다면, 파이프라인ning도 되지 않은 기본적인 프로세서에 비하여 최대 몇 배의 속도 향상을 얻을 수 있는가?

2.13

$$5 \times 4 = 20\text{배}$$

2.14 ① 듀얼-코어 프로세서에서 하나의 독립적인 태스크로 이루어진 응용 프로그램을 처리하는 경우에는 일반적인 프로세서에 비하여 처리 시간을 별로 단축시키지 못하는 이유를 설명하라. 그리고 듀얼-코어 및 멀티-코어 프로세서가 내부 CPU 코어 수만큼의 성능 향상을 얻을 수 있는 경우는 어떤 프로그램 처리 환경인지 설명하라. ②

2.14

(1) 각 CPU 코어는 프로그램 실행을 독립적으로 수행하기 때문이다. 또한 필요할 경우에만 자료교환 할 수 있다.

(2) 응용 프로그램이 독립적으로 실행될 수 있다면, CPU 코어 수 만큼 성능 향상을 얻을 수 있다.

2.15 16-비트 컴퓨터에서 두 단어 길이의 서브루틴 호출 명령어인 'CALL 56FE' 가 215B번지와 215C번지에 저장되어 있다(모든 수는 16진수로 표기). 그리고 현재 스택 포인터(SP)의 내용은 'FFFC'이고, 기억장치 FFFD번지에는 '123A'가 저장되어 있다. 아래의 각 순간에 프로그램 카운터(PC)와 SP는 어떤 값을 가지게 되는지 구하고, 스택에 저장되는 내용들을 그림 2-18과 같은 방법으로 표시하라. 단, PC의 길이는 16비트이다.

- (1) CALL 명령어가 실행되기 전 $PC=215B, SP=FFFC$
- (2) CALL 명령어가 실행된 후 $PC=56FE, SP=FFFFB$
- (3) 서브루틴으로부터 복귀한 후 $PC=215D, SP=FFFC$
- (4) 다시 한 번 RET 명령어가 실행된 후 $PC=123A, SP=FFFFD$

2.16 다음 수식을 계산하기 위한 프로그램을 아래의 명령어들을 사용하여 작성하고, 프로그램의 길이를 비교하라. (단, 명령어들은 2.4.2절 참조)

$$X = (A + B) / (D - E \times F + G \times H)$$

16 11 6
(1) 1-주소 명령어 (2) 2-주소 명령어 (3) 3-주소 명령어

연습문제 2.16

(1) 1-주소 명령어

LOAD A → ADD B → STOR T → LOAD E → MUL F
STOR U → LOAD G → MUL H → STOR V → LOAD D
SUB U → ADD V → STOR W → LOAD T → DIV W → STOR X

(2) 2-주소 명령어

MOV R1, A → ADD R1, B → MOV R2, E → MUL R2, F → MOV R3, G
MUL R3, A → MOV R4, D → SUB R4, R2 → ADD R4, R3 → DIV R1, R4
MOV X, R1

(3) 3-주소 명령어

ADD R1, A, B → MUL R2, E, F → MUL R3, G, H → SUB R4, D, R2 → ADD R5, R4, R3 → DIV X, R1,
R5

스택을 이용한 ADD 연산

push A	$A \rightarrow TOS$
push B	$B \rightarrow TOS$
add	$A+B \quad (A+B) \rightarrow TOS$
push C	$C \rightarrow TOS$
push D	$D \rightarrow TOS$
sub	$C-D \quad (C-D) \rightarrow TOS$
mul	$(A+B) \times (C-D)$
pop X	$M[C \times J] \rightarrow TOS$

스택 명령어들(PUSH, POP)과 0-주소 산술 명령어들(ADD, SUB, MUL)을 이용하여 아래의 수식을 계산하는 프로그램을 작성하라.

$$X = (A + B) \times (C - D)$$

2.18 문제 2-17에서와 같은 스택 연산에 대하여 다음과 같은 reverse Polish(혹은 postfix) 표기가 사용될 수 있다.

$$\begin{array}{ll} a + b & \rightarrow a b + \\ a + (b \times c) & \rightarrow a b c \times + \\ (a - b) \times c & \rightarrow a b - c \times \end{array}$$

아래 식들을 reverse Polish 표기를 사용하여 변환하라.

$$\begin{array}{l} (1) A + B - C - D + E = AB + C - D - E + \\ (2) (A + B) \times (C - D) + E = AB + CD - E + \\ (3) (A \times B) + (C \times D) - E = AB * CD * E - \\ (4) (A - B) \times (((C - D \times E) / F) / G) \times E = AB - CD E * - F / G / E * \end{array}$$

2.19 16비트 명령어에서 6비트가 연산 코드 필드이고, 나머지 비트들은 주소 혹은 데이터를 나타내는 오퍼랜드가 저장된다.

- (1) 오퍼랜드 필드의 값이 주소라면, 직접 주소지정 될 수 있는 기억장치의 용량은 몇 바이트인가? 단, 기억장치는 단어(16비트) 단위로 주소가 지정된다고 가정한다. $2^{10} = 1024 \times 2 \text{ byte} = 2048 \text{ byte}$
- (2) 오퍼랜드 필드에 2의 보수로 표현되는 즉시 데이터(immediate data)가 저장되는 경우에, 이 명령어에 저장될 수 있는 데이터의 범위를 구하라.

$$-2^9 \sim 2^9 - 1 = -512 \sim 511$$

2.20 어떤 CPU가 128가지의 연산들을 수행하며, 내부 레지스터의 수는 16개이다.

이 CPU의 32비트 명령어는 연산 코드 필드와 레지스터 번호를 나타내는 오퍼랜드1, 그리고 나머지 비트들로 구성되는 주소 필드(오퍼랜드2)로 이루어진다.

- (1) 명령어 형식을 표시하라. 2-주소 명령어 형식
(2) 주소 필드(오퍼랜드 2)가 2의 보수로 표현되는 범위 주소로 사용되는 경우에, 그 범위의 범위를 구하라. $-2^{20} \sim 2^{20} - 1$
(3) 레지스터의 길이가 32비트라면, 레지스터 간접 주소지정 방식에 의해 주소지정 될 수 있는 기억장치의 용량은 몇 GByte인가? 단, 기억장치 주소는 바이트 단위로 지정된다고 가정한다.

$$2^2 \times 2^{30} = 4 \text{ Gbyte}$$

아래와 같은 주소지정 방식을 사용하는 명령어를 인출하고 실행하기 위해서는 모두 몇 번의 기억장치 액세스가 필요한가?

- (1) 즉시 주소지정 방식 0번
(2) 레지스터 주소지정 방식 0번
(3) 간접 주소지정 방식 2번
(4) 인덱스 주소지정 방식 1번

4 CPU 내부 레지스터들과 주기억장치에 그림 2-24와 같은 값들이 저장되어 있고, 명령어의 레지스터 필드에는 '1'이 저장되어 있다고 가정하자.

- (1) 레지스터 주소지정 방식이 사용된다면, 연산 처리 과정에서 어떤 데이터가 사용될 것인가? R1 = '203'
(2) 레지스터 간접 주소지정 방식이 사용된다면, 어떤 데이터가 사용될 것인가? 4450'

ps. 직접 = 1번

2.23 어떤 명령어의 주소 필드 내용을 X1이라고 하자. 그 명령어를 실행하는 데 필요한 오퍼랜드는 기억장치 X2번지에 저장되어 있으며, 인덱스 레지스터의 내용은 X3이다. 명령어가 아래와 같은 주소지정 방식을 사용하는 경우에 변수들 간의 관계식을 쓰라. 단, 기억장치 A번지의 내용은 (A)로 표시한다.

- (1) 간접 주소지정 방식 $X2 = (X1)$
(2) 인덱스 주소지정 방식 $X2 = X1 + X3$

2.24 프로그램 카운터의 내용을 X1이라 하고, 기억장치 X1번지에 저장된 명령어의 주소 필드 내용을 X2라고 하자. 그 명령어를 실행하는 데 필요한 데이터는 기억장치 X3번지에 저장되어 있으며, 인덱스 레지스터의 내용은 X4이다. 명령어가 아래와 같은 주소지정 방식을 사용하는 경우에 유효 주소(EA)에 대한 관계식을 변수들을 이용하여 쓰라.

- (1) 직접 주소지정 방식 $X3 = X2$
(2) 간접 주소지정 방식 $X3 = (X2)$
(3) 상대 주소지정 방식 $X3 = (X1 + 1) + X2$
(4) 인덱스 주소지정 방식 $X3 = X2 + (X4)$

2.25 프로그램 카운터의 내용을 X1, 기억장치 X1 번지에 저장된 명령어의 주소 부분을 X2라고 하자. 만약 $X1 = 454$ 이고, $X2 = -25$ 이고, X1에 저장되어 있는 명령어가 상대 주소지정 방식을 사용하는 분기 명령어라면. 그 명령어가 수행된 직후에 X1은 어떤 값을 가지게 되는가? 단, 기억장치는 단어-단위로 주소가 지정된다고 가정한다.

$$Ex = A + (pc), \quad -25 + (454 + 1) = 430$$

2.26 상대 주소지정 방식을 사용하는 분기 명령어가 250번지에 저장되어 있다. 명령어의 길이는 32비트이며, 그 중에서 10비트는 연산 코드이다. 기억장치의 주소는 바이트 단위로 지정된다고 할 때, 아래의 물음에 답하라.

(1) 분기 명령어가 실행되면 284번지로 분기되도록 하려면 명령어의 주소 필드에 어떤 내용을 넣어야 하는지 2의 보수로 표현하라.

(2) 232번지로 분기되도록 하는 경우에는 분기 명령어의 주소 필드에 어떤 내용을 넣어야 하는지 2의 보수로 표현하라.

$$\begin{array}{r} 284 \\ -232 \\ \hline 52 \end{array}$$

0000 0000 0000 0000 | 0000

110 111 111 111 | 0110 -19

2.27 기억장치 200번지부터 50개의 요소들로 이루어진 데이터 배열이 저장되어 있다. 인덱스 주소지정 방식을 사용하는 명령어를 이용하여 12번째 데이터 요소를 인출하려는 경우에, 명령어의 오퍼랜드 필드 및 인덱스 레지스터에 저장되어야 하는 값(10진수)은 각각 무엇인가?

200 11

2.28 레지스터-간접 주소지정 방식과 인덱스 주소지정 방식으로 결정한 유효 주소(EA)가 같아질 수 있는 조건은 무엇인가?

$$EA = (R), \quad EA = (R) + A$$

'A' 값이 '0' 이어야 함.