|  |
| --- |
| **컴퓨터구조 과제 리포트**  5\_8 Chapter 연습문제 풀이 |



|  |  |  |  |
| --- | --- | --- | --- |
| 제출일 | 2021. 06. 14 | 전공 | 컴퓨터정보공학전공 |
| 과목 | 컴퓨터구조 | 학번 | 2123405 |
| 담당교수 | 이수철 | 이름 | 최창환 |

|  |
| --- |
|  |

**컴퓨터구조 과제 리포트**

**5장**

**01. 0.5를 IEEE-754의 단일 정밀도 방식으로 표현할 경우 가수 부분 23비트는?**

**① 00000000000000000000000**

**② 01000000000000000000000**

**③ 10000000000000000000000**

**④ 11000000000000000000000**

**우선 0.5를 이진수로 표현한다.**

**0.5 \* 2 = 1.0**

**0.5(10) = 0.1(2)**

**가수 필드는 00000000000000000000000이 된다.**

**0 / 0111 1110 / 0000 0000 0000 0000 0000 000**

**부호 지수 가수**

**02. 다음 중 특성이 다른 것은?**

**① 멀티플렉서**

**② 레지스터**

**③ 가산기**

**④ 디코더**

**03. IEEE-754의 부동 소수점 표현에서 사용하는 잠복 비트의 목적은?**

**① 연산 속도를 향상하기 위해**

**② 숫자를 유일하게 만들기 위해**

**③ 유효 숫자를 늘리기 위해**

**④ 정규화하기 위해**

**04. 다음 중 순차 논리회로와 관계없는 것은?**

**① 플립플롭과 같은 기억소자를 포함한다.**

**② 입력 신호와 내부 상태에 의해 출력이 결정된다.**

**③ 디코더**

**④ 레지스터**

**05. 레지스터 주소 비트를 사용하여 레지스터 파일에 포함된 레지스터 중 하나를 명시하는 데 필요한 논리회로는?**

**① 디코더**

**② 인코더**

**③ 멀티플렉서**

**④ 디멀티플렉서**

**06. 다음 중 언더플로우와 가장 관계 깊은 것은?**

**① 조합 회로에 미세한 전류가 흐르는 경우**

**② 인코더가 입력으로 받아들이기에 데이터가 너무 작은 경우**

**③ -9.0 \* 10^100과 같이 숫자가 너무 작아서 주어진 부동 소수점 형식으로 표현할 수 없는 경우**

**④ 실숫값이 0에 너무 근접하여 0으로 취급할 수밖에 없는 경우**

**소수 넷째 자리까지 표현할 수 있는 프로그램에서**

**0과 0.0001 사이에 있는 수, 0.00001같은 수를 표현하려고 하면 양수 언더플로우가 발생한다.**

**0과 -0.0001 사이의 수를 표현할 때도 발생한다. 이때는 음수 언더플로우라고 한다.**

**07. 레지스터 파일이 32비트 크기의 레지스터 16개로 구성되어 있다. 이 레지스터 파일의 주소 단자를 구성하는 비트 수는?**

**① 5**

**② 4**

**③ 32**

**④ 16**

**2^n = 16**

**n = 4**

**4비트가 있으면 16개의 레지스터 주소를 나타낼 수 있다.**

**08. 다음 중 수치 데이터에 대한 설명으로 틀린 것은?**

**① 수치 데이터의 영역은 범위와 정밀도를 의미한다.**

**② 정밀도는 연속적인 데이터 값 2개 사이의 거리를 의미한다.**

**③ 범위는 가수를 위한 유효 자릿수에 의해 주로 좌우된다.**

**④ IEEE-754 표준안은 범위와 정밀도 측면에서 두 가지 부동 소수점 형식을 제공한다.**

**범위는 가수가 아니라 지수에 의해 결정**

**09. "부동 소수점 표현에서 2가 아닌 기수를 사용할 경우 잠복 비트라는 기법을 사용할 수 없다." 이는 맞는가, 틀린가?**

**맞다**

**소수점의 왼쪽, 즉 정수 부분을 항상 한가지 수로 고정해서 따로 가수 필드에 저장할 필요가 없게 해주는 게 잠복 비트이다.**

**2진수에서는 0과 1만 사용하는데 정수 부분이 1로 고정되므로 항상 값이 같지만,**

**8진수나 10진수 같은 경우에는 0을 제외한 수가 1만 있는 것이 아니므로 잠복 비트를 사용할 수 없다.**

**​**

**10. ( ) 는 다국적 알파벳을 대부분 포함하기 위해 하나의 문자를 16비트 체계로 구성한 코드로 만국 공통의 국제 문자 부호 체계를 의미한다.**

**유니코드**

**​**

**11. 하나의 바이트(8비트)로 구성된 워드를 사용하는 마이크로프로세서가 있다. 무부호 표현, 부호 표현, 1의 보수 방식, 2의 보수 방식으로 나타낼 수 있는 최솟값과 최댓값은 얼마인가?**

**① 무부호 표현 : 0 ~ 2^8-1 , 0 ~ 255**

**② 부호 표현 : -(2^7 -1) ~ +(2^7 -1) , -127 ~ +127**

**③ 1의 보수 방식 : -(2^7 -1) ~ +(2^7 -1) , -127 ~ +127**

**④ 2의 보수 방식 : -(2^7) ~ +(2^7 -1) , -128 ~ +127**

**​**

**12. 비정규화된 부동 소수점 수의 존재 이유는 무엇인가?**

**하나의 실수가 무한한 방법으로 표현된다는 문제점이 있다. 즉 하나의 실수에 대한 고유한 표현이 없다.**

**​**

**13. 단일 정밀도 IEEE-754의 표현에서 다음 수의 10진수 값은 얼마인가?**

**부호**

**지수**

**가수**

**10진수 값**

**1**

**1000 0011**

**1100 0000 0000 0000 0000 000**

**(1)**

**0**

**0111 1110**

**1010 0000 0000 0000 0000 000**

**(2)**

**0**

**1000 0000**

**0000 0000 0000 0000 0000 000**

**(3)**

**① -28**

**② 0.8125**

**③ 2**

**①**

**부호는 1이므로 음수이다.**

**지수의 값은 131인데 127초과 코드 이므로 4이다.**

**가수는 11**

**잠복비트로 인해 1.11이 되고**

**전체는 (-1) \* 1.11 \* 2^4 가 된다.**

**1.11의 소수점을 오른쪽으로 4번 시프트한다.**

**11100(2) = 28, 따라서 -28이 된다.**

**​**

**​**

**②**

**부호는 0이므로 양수이다.**

**지수의 값은 126인데 127초과 코드 이므로 -1이다.**

**가수는 101**

**잠복비트로 인해 1.101이 되고**

**전체는 (1) \* 1.101 \* 2^-1가 된다.**

**1.101의 소수점을 왼쪽으로 1번 시프트한다.**

**0.1101(2) = 0.5 + 0.25 + 0.0625 = 0.8125, 따라서 0.8125가 된다.**

**​**

**​**

**③**

**부호는 0이므로 양수이다.**

**지수의 값은 128인데 127초과 코드이므로 1이다.**

**가수는 0**

**잠복비트로 인해 1.0이 되고**

**전체는 (1) \* 1.0 \* 2^1가 된다.**

**1.0의 소수점을 오른쪽으로 1번 시프트한다.**

**10(2) = 2, 따라서 2가 된다.**

**14. 대부분의 아키텍처에서 정수를 표현하는 데 부호-크기 방식이나 1의 보수 방식보다 2의 보수 방식을 선호하는 이유는 무엇인가?**

**양수 음수 전환 과정이 다른 방식보다는 번거롭지만, 0이 하나이고 덧셈 뺄셈을 쉽게 구현할 수 있기 때문이다.**

**2장**

**01. 다음 중 폰노이만 아키텍처와 관계없는 것은?**

① 데이터와 명령어를 모두 메모리에 저장

② 유니버설 기계

③ 프로그램 내장형 컴퓨터

④ 명령어를 데이터와 동일한 방식으로 표현

**02. 다음 중 알고리즘의 영향을 받는 것은?**

① 프로그램의 크기

② CPI

③ 클록률

④ 클록 속도

**03. 다음 중 컴파일러나 운영체제와 같은 개념이 도입된 컴퓨터 세대는?**

① 1세대 컴퓨터

② 2세대 컴퓨터

③ 3세대 컴퓨터

④ 4세대 컴퓨터

**04. 경험적 관찰에 의해 예측한 것으로, 단일 마이크로칩에 포함된 트랜지스터의 수가 18개월마다 약 2배씩 증가한다는 법칙은?**

① 황의 법칙

② 폰노이만의 법칙

③ 무어의 법칙

④ 암달의 법칙

**05. 다음 용어 중 사용되는 범주가 다른 것은?**

① CISC

② MIPS

③ MFLOPS

④ KLIPS

**06. 다음 성능 척도 중 MIPS 값을 계산하는 데 사용되지 않은 것은?**

① 명령어 개수

② 명령어의 종류

③ 실행 시간

④ 클록 속도

**07. 다음 중 클록 속도에 영향을 미치는 것은?**

① 프로그램의 크기

② 알고리즘

③ 반도체 제조 기술

④ 명령어 배합

**08. 다음 중 CPI를 줄일 수 있는 방법과 거리가 먼 것은?**

① RISC 아키텍처 사용

② 병렬화

③ 반도체 제조 기술

④ 파이프라이닝

**09. 다음 중 프로그램 내장식 컴퓨터와 관계없는 것은?**

① 고정결선식 컴퓨터

② 폰노이만 모델

③ 프린스턴 아키텍쳐

④ 데이터와 프로그램을 동일 형태로 메모리에 저장

**10. 다음 중 잘못된 설명은?**

① 무어의 법칙 : 단일 마이크로칩에 포함된 트랜지스터의 수가 1.5년마다 약 2배씩 증가한다.

② 폰노이만 아키텍처 : 컴퓨터는 CPU, 메모리, 입출력장치로 구성된다.

③ 처리율 : 프로그램의 시작부터 종료까지의 시간을 의미한다.

④ 3세대 컴퓨터 : 집적회로를 사용한 컴퓨터를 말한다.

**11. "클록 속도가 빠를수록 컴퓨터의 성능이 우수하다" 라는 판단 기준은 맞는가, 틀린가?**

틀리다

**12. 프로그램 내장식 컴퓨터의 경우, CPU와 메모리 사이에 발생하는 트래픽이 성능에 큰 영향을 미친다. 따라서 CPU와 메모리를 연결하는 부분을 (** 폰노이만 병목 **)이라고 한다.**

**13. 하버드 아키텍처의 목적은 무엇인가?**

폰노이만 병목 현상을 해결하기 위해

**14. 프로그램에서 부동 소수점 연산 명령어가 차지하는 비율이 10%이며, 부동 소수점 연산 명령어를 2배 빠르게 개선했다. 이 시스템의 전반적인 성능 향상은 어느 정도인가?**

약 1.05배

**15. 어떤 프로세서에서 프로그램 실행 시간이 100ms이며, 부동 소수점 덧셈에 60ms, 부동 소수점 곱셈에 30ms가 소요된다. 설계 팀이 동일한 비용으로 부동 소수점 덧셈을 1.6배 개선하거나 부동 소수점 곱셈을 3배 개선할 수 있는 프로세서 모델을 제안했을 때, 어느 방식이 더 좋은가?**

덧셈 방식이 더 좋다.

**3장**

**01. 다음 중 어셈블리어와 관계없는 것은?**

① 연상 부호

② 컴파일러

③ 매크로

④ 레이블

**02. 다음 중 피연산자의 수를 줄이는 방법과 거리가 먼 것은?**

① 스택

② 누산기

③ 목적지 피연산자의 위치를 근원지 피연산자와 중첩

④ 레지스터

**03. 다음 분기 명령어 중 가장 많은 작업을 수행하는 연산은?**

① bgt r1, r2, label

② brz label

③ beq label

④ jmp label

**04. CPU 내부에 기억장치를 추가할 때 나타나는 효과와 관계없는 것은?**

① 메모리를 다시 참조할 필요성을 줄여줌

② 명령어의 길이 축소

③ 데이터의 길이 축소

④ CPU와 메모리 사이의 트래픽 감소

**05. 다음 중 폰노이만형 컴퓨터 연산자의 기능과 거리가 먼 것은?**

① 제어 기능

② 연산 기능

③ 참조 기능

④ 전달 기능

**06. 명령어 실행 사이클 중 간접 사이클 동안에 수행되는 것은?**

① 명령어를 해독한다.

② 데이터의 주소를 읽는다.

③ 데이터를 읽는다.

④ 인터럽트를 처리한다.

**07. 메모리를 참조할 때 데이터의 주소를 명시하기 위한 버퍼는?**

① Acc

② MDR

③ MBR

④ MAR

**08. 다음 중 고정 길이 명령어와 관계 없는 것은?**

① 명령어 해독 과정이 짧다.

② 하드웨어가 단순하다.

③ 프로그램의 크기가 증가할 수 있다.

④ 강력한 명령어를 도입할 수 있다.

**09. 다음 명령어 중 종류가 다른 것은?**

① SUB

② HLT

③ CAL

④ RET

**10. 명령어의 길이가 2바이트라고 가정할 때 프로시저 호출 명령 'cal proc'의 의미는?**

① IR ← M[PC]; PC ← PC + 2; TOS ← proc

② IR ← M[PC]; PC ← PC + cal; TOS ← proc + 2

③ IR ← M[PC]; TOS ← PC + 2; TOS ← proc

④ IR ← M[PC]; TOS ← cal + 2; PC ← proc

⑤ IR ← M[PC]; TOS ← PC + 2; PC ← proc

**11. 다음에 수행할 명령어의 주소를 가리키는 레지스터의 이름은 무엇인가?**

프로그램 계수기

(프로그램 카운터)

Program Counter

**12. 5절의 메모리-메모리 컴퓨터와 가정이 동일할 때, y = ax^2 + b를 연산하는데 발생하는 총 메모리 트래픽은 얼마인가?**

64byte

교재 5절의 메모리-메모리 컴퓨터

● 연산 부호 크기는 8비트, 즉 1바이트

● 연산마다 최대 2개의 피연산자 필드

● 첫 번재 피연산자 필드는 근원지 및 목적지 겸용

● 메모리 주소는 16비트, 즉 2바이트

● 데이터 크기는 32비트, 즉 4바이트

​

명령어 의미

mov y, a y ← M[a]

mul y, x y ← M[a] \* M[x]

mul y, x y ← M[a] \* M[x] \* M[x]

add y, b y ← M[a] \* M[x] \* M[x] + M[b]

데이터 전송 트래픽 : 8 + (12 \* 3) = 44byte

총 트래픽은 20byte + 44byte = 64byte

​

​

**13. 프로시저를 호출할 때 복귀 주소를 저장해야 한다. 복귀 주소를 위한 저장 공간으로 특정 레지스터를 이용할 때 발생하는 문제는 무엇인가?**

복귀 주소를 레지스터나 프로시저의 시작 부분에 저장할 경우,

프로시저를 중첩 호출하거나 재진입 가능한 프로시저를 호출할 수 없다.

**14. 프로그램과 컴퓨터 하드웨어 사이의 인터페이스에 대한 완전한 정의 혹은 명세를 무엇이라고 하는가?**

명령어 집합 구조(ISA)

**15. 정수를 표현하기 위한 부호-크기 방식은 0이 2개 존재한다. 이는 좋은 명령어 집합 구조를 위한 (** 적절성 **) 에 위반된다.**

**16. [표 3-3]에서 104번지와 106번지의 기계어를 각각 4500, 2500으로 바꿀 때 대응하는 어셈블리어는 무엇인가? 가상 컴퓨터로 수정한 프로그램을 실행하면 메모리와 레지스터의 최종 내용은 무엇인가?**

SUB 500, STA 500

M[500]에 데이터 0002가 저장된다.

데이터는 메모리 500번지와 502번지에 각각 0001과 0002로 저장되어 있다고 가정한다.

0100 LDA 500 누산기에 M[500]의 데이터를 적재 // 누산기 : 0001

0102 ADD 502 누산기의 내용과 M[502]의 데이터를 덧셈 // 누산기 : 0001 + 0002

0104 SUB 500 누산기의 내용과 M[500]의 데이터를 뺄셈 // 누산기 : 0002

0106 STA 500 누산기의 데이터 0002를 M[500]에 저장

M[500]에 데이터 0002가 저장된다.

**17. 명령어 인출 사이클에서 발생하는 작업을 기호로 사용하며 순서대로 나타내되 MAR과 MBR에 대한 경유 과정도 포함하라. 단, 명령어의 길이는 4바이트이다.**

MAR ← M[PC]

MBR ← M[MAR]

IR ← MBR;

PC ← PC + 4;

**4장**

**01. 다음 중 즉치 주소 지정 방식과 관계 없는 것은?**

① 상수 주소 지정 방식이라고도 한다.

② 명령어의 일부를 사용하여 데이터를 표현한다.

③ 고급언어에서 상수를 명시할 빈도수가 매우 낮기 때문에 효용성이 떨어진다.

④ picoMIPS에서 I-형식 명령어의 imm 필드가 사용한다.

**02. 다음 주소 지정 방식 중 속도가 가장 빠른 것은?**

① 즉치 주소 지정

② 레지스터 직접 주소 지정

③ 메모리 간접 주소 지정

④ 색인 주소 지정

**03. 다음 중 0-주소 명령어를 사용하는 컴퓨터 구조는?**

① 누산기 컴퓨터

② 스택 컴퓨터

③ 마이크로컴퓨터

④ 범용 레지스터 컴퓨터

**04. 다음 중 주소 지정 방식이 아닌 것은?**

① 직접 주소 지정

② 버퍼 주소 지정

③ 레지스터 주소 지정

④ 즉시 주소 지정

**05. 다음 중 0-주소 명령어 컴퓨터와 가장 관련 있는 것은?**

① 스택

② 큐

③ 베이스 레지스터

④ 색인 레지스터

**06. 다음 중 명령어 설계 과정과 가장 거리가 먼 것은?**

① 연산 부호의 종류

② 주소 지정 방식

③ 메모리의 대역폭

④ 워드의 크기

**07. 다음 중 연산을 수행한 후에 입력 데이터가 모두 보존되는 방식은?**

① 0-주소 명령어 컴퓨터

② 1-주소 명령어 컴퓨터

③ 2-주소 명령어 컴퓨터

④ 3-주소 명령어 컴퓨터

**08. 스택 컴퓨터에서 스택의 최상위 데이터는 (** 묵시 **) 주소 지정 방식이다.**

**09. "CISC 구조는 복잡하고 강력한 연산을 단순 명령어의 조합으로 구현한다." 이는 맞는가, 틀린가?**

틀리다.

**10. 피연산자 필드를 짧게 명시하는 것이 중요한 이유는 무엇이며, 오늘날의 컴퓨터는 어떤 방법을 사용하여 피연산자 필드를 짧게 하는가?**

피연산자 필드를 짧게 명시하면 명령어의 주소가 짧아지는데, 인출 트래픽이 줄어 폰노이만 병목을 줄일 수 있다. 오늘 날의 컴퓨터는 CPU 내부에 다수의 데이터를 임시로 저장하는 범용 레지스터 컴퓨터 방식을 사용한다. 레지스터 주소는 메모리 주소보다 짧기 때문에 범용 레지스터 컴퓨터의 명령어 길이는 메모리-메모리 컴퓨터의 명렁어 길이보다 짧다.

**11. RISC 아키텍처의 주된 아이디어는 무엇인가?**

단순하고 더 빨리 실행되는 여러 개의 명령어를 사용하는 아키텍처

**12. 대부분의 RISC 아키텍처에서 3-주소 명령어를 주로 사용하는 이유는 무엇인가?**

단순한 명령어를 여러 개 사용하기 위해서이다.

**13. 주소 해상도는 무엇인가? 명령어와 데이터를 위해 동일한 주소 해상도를 사용해야 하는가? 비트 단위 주소 해상도를 사용하지 않는 이유는 무엇인가?**

주소 해상도는 아키텍처가 직접 명시할 수 있는 정보의 최소 단위이다. 일관성과 간결성을 위해 동일한 주소 해상도를 사용해야 한다. 비트 단위까지 주소를 분해하면 주소를 나타내는 데 많은 비트가 필요할 뿐만 아니라 데이터 정렬에 추가 시간이 필요하기 때문이다. 그래서 오늘날의 아키텍처는 대부분 8비트(1바이트), 16비트(2바이트), 혹은 그 배수 단위의 주소 해상도를 사용한다.

**14. 레지스터 직접 주소 방식이 흔히 사용되는 이유는 무엇인가?**

레지스터 주소가 메모리 주소보다 짧아서 비교적 짧은 피연산자를 요구할 뿐만 아니라 데이터 접근 속도도 빠르기 때문이다.

**15. 데이터 1234(16)를 빅 엔디언과 리틀 엔디언으로 표시하라.**

빅 엔디언 : 1234

리틀 엔디언 : 3412

**16. 4장 1.3절과 동일한 가정하에 3-주소 명령어를 사용할 때 범용 레지스터 컴퓨터에서 발생하는 메모리 트래픽을 분석하라.**

16개의 레지스터 사용 적재. 저장 명령어 컴퓨터

y = ax^2 + bx + c

**인출 트래픽**

load, store 명령어 : (연산부호 + 레지스터 주소 + 메모리 주소)3.5byte로 구성.

그러나 컴퓨터의 최소 단위는 바이트 단위이므로 load, store 명령어의 길이는 4byte가 된다.

mul, add 명령어 : (연산부호 + 레지스터 주소 + 레지스터 주소 + 레지스터 주소) 2.5byte로 구성되지만 위와 같은 이유로 3byte가 된다.

인출 트래픽은 (4byte \* 5) + (3byte \* 5) = 35byte

**데이터 전송 트래픽**

load, store 명령어 : 하나의 데이터를 다른 장소로 이동시킨다. 4byte

mull, add 명령어 : 레지스터에 있는 데이터만 사용하므로 메모리에 접근할 필요가 없으니 데이터 트래픽이 발생하지 않는다.

데이터 전송 트래픽은 4byte \* 5 = 20byte

총 트래픽 : 35byte + 20byte = 55byte