|  |
| --- |
| **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

**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

24 = 16

n = 4

**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.

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

**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 : -(27 -1) ~ +(27 -1) , -127 ~ +127

③ 1의 보수 방식 : -(27-1) ~ +(27-1) , -127 ~ +127

④ 2의 보수 방식 : -(27) ~ +(27-1) , -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

**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.

**6장**

**01. 부동 소수점 연산에서 가수는 동일하지만 지수가 다른 두 수를 덧셈할 때 다음 중 적절한 것은?**

① 높은 지숫값에 일치시킨다.

② 낮은 지숫값에 일치시킨다.

③ 두 지숫값의 평균값에 일치시킨다.

④ 상관없다.

**02. 다음 중 연산장치를 구성하는 요소가 아닌 것은?**

① 산술장치

② 논리장치

③ 시프터

④ 레지스터

**03. 다음 중 플래그 레지스터 Z 비트와 가장 관련 있는 것은?**

① AND 게이트

② OR 게이트

③ XOR 게이트

④ NOR 게이트

Z 비트는 입력값이 0이면 1, 아니면 0을 가진다. 모든 비트가 0이어야 0으로 취급, 즉 하나라도 1일 경우 Z 비트에 0을 주어야 하므로 OR 게이트에 NOT을 붙인 NOR 게이트를 사용해야 한다.

**04. 다음 중 정수의 덧셈 연산에서 오버플로우 점검과 가장 관련 있는 것은?**

① AND 게이트

② OR 게이트

③ XOR 게이트

④ NOR 게이트

부호가 같은 정수끼리 덧셈할 때, 잉여비트와 부호 비트가 동일할 경우 무시할 수 있다.

XOR 게이트는 비트가 같으면 0, 다르면 1이 나온다.

**05. 다음 중 무부호 곱셈 연산 0101(2) \* 0100(2)에서 부분곱에 해당하는 것은?**

① 0101(2)

② 0100(2)

③ 01010(2)

④ 010100(2)

0101

\* 0100

--------------------

0000

0000

0101 ← 부분곱 010100

0000

--------------------

0010100

= 10100(2)

**06. 다음 중 시프트-덧셈 방식의 곱셈 연산과 관계없는 것은?**

① 승수의 MSB 검사

② 누적 부분곱과 피승수를 왼쪽 시프트

③ 누적 부분곱 ← 0

④ 누적 부분곱 ← 부분곱 + 누적 부분곱

승수의 LSB(최하위 비트)를 검사한다.

**07. 다음 중 Booth 알고리즘과 관계 없는 것은?**

① 곱셉 연산에 적용된다.

② 승숫값에 연속된 1이나 0이 많이 포함될 때 효과적이다.

③ 승수의 LSB 오른쪽에 1비트를 추가한 후 2비트씩 점검한다.

④ 피승숫값에 0이 많을 경우 적합하다.

**08. 다음 중 무부호 정수의 나눗셈을 위한 비복원 알고리즘과 관계 없는 것은?**

① 부분 나머지가 음수이면 1비트 오른쪽으로 시프트된 제수를 더한다.

② 부분 나머지가 0이면 대응하는 몫의 비트가 0이다.

③ 부분 나머지가 양수이면 1비트 오른쪽으로 시프트된 제수를 뺀다.

④ 부분 나머지를 부분적으로 복원한다.

복원 알고리즘은 부분 나머지가 몇이던 간에 일단 제수를 빼고, 부분 나머지가 음수일 경우 제수를 다시 더한 다음 시프트해 다음 계산으로 넘어간다.

비복원 알고리즘은 일단 뺀 후 부분 나머지가 음수일 경우 1비트 오른쪽으로 시프트된 제수를 더한다.

부분 나머지가 음수일 경우 대응하는 몫의 비트가 0이며, 음수가 아닐 경우(0, 양수) 1이다.

**09. 다음 중 부동 소수점의 덧셈과 관계 없는 것은?**

① 지수 정렬

② 가수 덧셈

③ 오버플로우 점검

④ 언더플로우 점검

부동 소수점 덧셈 과정

1. 지수 정렬

2. 가수 덧셈

3. 정규화

4. 오버플로우 점검

5. 자리 맞춤

**10. 다음 중 부동 소수점의 나눗셈과 관계 없는 것은?**

① 지수 뺄셈

② 정규화

③ 부호 결정

④ 초과 코드를 조정하기 위해 지수에서 127을 뺄셈

부동 소수점 나눗셈 과정

1. 지수 뺄셈

2. 가수 나눗셈

3. 정규화

4. 오버플로우 점검

5. 자리 맞춤

6. 부호 결정

**​**

127초과 코드는 IEEE-754에서 부동 소수점 수의 지수를 표현할 때 사용된다. 지수 뺄셈 과정에서 초과 코드로 조정해주기 위해 127을 한 번 더해줘야 한다

**11. 부동 소수점 연산의 정밀도는 ( 가드 디지트 )를 사용하여 개선할 수 있다.**

정규화 과정에서 오른쪽 시프트할 때 발생하는 손실을 막기 위해 사용된다.

**12. 플래그 레지스터에 연산 결과가 양수인지 아닌지를 나타내는 상태 비트 P를 추가하려고 한다. [그림 6-4]와 같은 연산장치일 때 P 비트의 입력을 구현하라.**

양수일때는 1, 그 외에는 0의 값을 가지는 상태 비트 P를 추가한다.

최상위 비트 (MSB)를 검사한다.

최상위 비트 0(양수) → NOT 게이트 → 1

1(음수) → NOT 게이트 → 0

**13. 2의 보수 덧셈 연산에서 어떤 경우에 잉여 비트를 무시할 수 있는가?**

가수와 피가수의 부호(MSB)가 같을 때 잉여비트와 부호비트가 같은 경우, 가수와 피가수의 부호가 다를 경우 무시할 수 있다.

ex) 1101 (-3)

+ 1100 (-4)

----------------------

11001

**14. [그림 6-12]는 6 X (-5)에 Booth 알고리즘을 적용한 것이다. (-6) X (-5)에 Booth 알고리즘을 적용하여 [그림 6-12]의 (a)와 같이 나타내라.**

-6 = 1010(2)

-5 = 1011(2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | Q | Q-1 | M | Action |
| 0000 0000  - 1010 0000  ------------------  0110 0000 시프트  0011 0000 | 1011 | 0 (추가 비트) | 1010 | 10이므로 A-M하고 시프트 |
| 0011 0000 시프트  0001 1000 | 1011 | 1 | 1010 | 11이므로 시프트만 진행 |
| 0001 1000  + 1010 0000  -----------------  1011 1000 시프트  1101 1100 | 1011 | 1 | 1010 | 01이므로 A+M하고 시프트 |
| 1101 1100  - 1010 0000  ------------------  0011 1100 시프트  0001 1110 | 1011 | 0 | 1010 | 10 이므로 A-M하고 시프트 |
| 11110(2) = 30 | | | | |

**7장**

**01. 다음 중 다중 사이클 방식의 데이터 경로에 대한 설명으로 틀린 것은?**

① 다음 사이클을 위해 현재 내용을 임시 저장소에 저장해야한다.

② 메모리에 두 번 접근하기 위해 데이터 메모리와 명령어 메모리로 분할해야 한다.

③ 단일 사이클 방식에 비해 일반적으로 클록 사이클 시간이 짧다.

④ 단일 사이클 방식에 비해 파이프라이닝 기법에 더 적합하다.

**02. 데이터 경로의 일부 컴퓨팅 자원은 2개 이상의 데이터가 입력될 수도 있는데, 이러한 경우 명령어가 제대로 실행될 수 없다. 이와 같은 문제점을 해결하려면 무엇이 필요한가?**

① 시프터

② 버퍼

③ 멀티플렉서

④ 디멀티플렉서

**03. 다음 중 명령어 인출 과정과 가장 관련 있는 것은?**

① 프로그램 계수기

② 레지스터 파일

③ 데이터 메모리

④ 연산장치

**04. 명령어의 정의에 따라 데이터를 처리하기 위해 데이터를 진행시키는 경로를 ( 데이터경로 ) 라고 한다.**

**05. “하나의 명령어를 실행할 때 유사한 컴퓨팅 자원을 여러 개 배치하지 않고 다시 사용하려면 단일 사이클 방식을 사용해야한다.” 이는 맞는가, 틀린가?**

틀리다. 다중 사이클 방식을 사용해야 한다.

**06. 다음과 같은 picoMIPS 명령어를 실행할 때 주어진 데이터 경로, 레지스터 파일, 메모리에 적절한 내용을 2진법으로 나타내라.**

lw r3, r2, #1

① 010 rs

② 011 rt

③ 000001 imm

④ 0000 0000 0000 0001 r3

⑤ 0000 0000 0000 0100 rs + rt

⑥ 0000 0000 1000 0010 r2

⑦ 0000 0000 0000 0010

⑧ 0000 0000 1000 0010

**07. picoMIPS 아키텍쳐에 mov라는 이동 명령어를 추가하고자 한다. 예를 들면 mov r1, r2는 r2의 내용을 r1에 저장하는 명령이다. (1) 이와 같은 이동 명령을 picoMIPS에 추가하려면 단일 사이클 데이터 경로를 어떻게 수정해야 하는가? (힌트 : add r1, r2, r3에서 r3가 0이면 mov r1, r2와 동일한 효과를 나타낸다.) (2) 이동 명령어의 데이터 경로를 그림으로 나타내라(단, 명령어 인출 단계와 레지스터 읽기 단계는 필요 없음).**

**08. 다음과 같은 picoMIPS 명령어를 수행할 때 주어진 데이터 경로, 레지스터 파일, 메모리에 적절한 내용은 무엇인가? 레지스터 파일의 내용은 [그림 7-15] (a)의 명령어 실행 전과 동일하다.**

**어느 주소의 메모리 내용이 어떤 내용으로 수정되는가?**

sw r3, 1(r2)

op rs rt imm

0101 010 011 000001

① 0000 0000 0000 0010

② 0000 0000 0000 0001

③ 0000 0000 0000 0010

④ 0000 0000 0000 0001

⑤ 0000 0000 0000 0100

⑥ 0000 0000 0000 0100

⑦ 0000 0000 0000 0001

**09. 다중 사이클 방식 데이터 경로의 기본 골격에서 조건 분기 명령어 beq를 실행하려면 2개의 공통 사이클 외에 2개의 사이클이 더 필요하다. 이때 추가적으로 필요한 사이클의 데이터 흐름을 데이터 경로를 사용하여 나타내라.**

만약 문제에서 말한 2개의 공통 사이클이 명령어 인출 사이클과 명령어 해독\*읽기 사이클이라면,

beq 명령어는 if ( a == b ) PC = PC + 2(명령어의 크기) + imm \* 2 라고 표현할 수 있다.

이때 a == b를 계산하는 사이클 하나와

나머지 부분 PC = PC + 2(명령어의 크기) + imm \* 2를 계산하는 사이클 하나로 나눌 수 있다.

**8장**

**01. 다음 중 제어장치의 역할과 관계없는 것은?**

① 명령어 해독

② 데이터 연산

③ 제어 신호 생성

④ 제어 신호 인가

**02. 메모리에서 가져온 명령어를 해독한 후 시스템 전체에 제어 신호를 제공하여 모든 기능장치가 명령어의 지시대로 동작하게 하는 컴퓨터의 구성 요소는?**

① 제어장치

② 기능장치

③ 디코더

④ ALU

**03. 다음 중 제어장치의 입력과 가장 거리가 먼 것은?**

① 명령어

② 데이터

③ 프로세서의 상태

④ 클록 신호

**04. 다음 중 마이크로프로그래밍 방식과 가장 거리가 먼 것은?**

① 펌웨어 방식

② 다른 종류의 명령어 집합을 에뮬레이션하기 쉽다.

③ 컴퓨터의 구성, 타이밍, 기술의 변화에 적응하기 쉽다.

④ 조합 논리회로를 사용하여 구현한다.

④ 조합 논리회로를 사용하는 것은 고정결선식 방식이다. 마이크로프로그래밍 방식은 PROM을 사용한다.

① 마이크로프로그래밍 방식은 펌웨어 방식, 마이크로 코드 방식이라고도 한다.

②③ 마이크로프로그래밍의 장점 :

명령어 집합의 변경이 가능하다.

다른 종류의 명령어 집합을 에뮬레이션할 수 있다.

다수의 명령어 집합을 구현할 수 있다.

컴퓨터의 구성, 타이밍, 기술의 변화에 쉽게 적응할 수 있다.

**05. 다음 중 제어점과 관계 없는 것은?**

① 레지스터 읽기, 쓰기 신호를 적용하는 지점

② 연산장치에 연산의 종류를 인가하는 지점

③ 제어 신호를 생성하는 지점

④ 멀티플렉서를 이용하여 기능장치를 연결하는 지점

제어점은 제어 신호가 적용되는 지점이다.

**06. 다음 중 마이크로 명령어의 구성 요소가 아닌 것은?**

① 마이크로프로그램 계수기

② 마이크로 연산 필드

③ 분기 조건

④ 마이크로 명령어 주소

마이크로프로그램 계수기 (µPC)는 마이크로 명령어를 순차적으로 읽기 위해 사용된다.

**07. 다음 중 PROM에 대한 설명으로 틀린 것은?**

① AND 게이트 배열의 출력을 OR 게이트 배열의 입력과 연결하는 조합 논리회로이다.

② 마이크로프로그래밍 방식의 제어 저장소로 사용될 수 있다.

③ 필요 없는 최소항을 위한 공간도 있어야 하므로 주어진 칩 공간을 낭비할 수 있다.

④ 내부에 디코더 회로를 포함한다.

PLA : AND 게이트 배열의 출력을 OR 게이트 배열의 입력과 연결하는 조합 논리회로

PROM : 디코더로 모든 종류의 최소항을 생성

**08. 다음 중 수직적 마이크로 명령어와 관계 없는 것은?**

① 하드웨어가 복잡하다.

② 제어 저장소의 용량이 비교적 작다.

③ 지연 시간이 없어서 속도가 빠르다.

④ 디코더를 사용해야 한다.

수직적 마이크로 명령어는 마이크로 연산 필드를 인코더로 압축하여 마이크로 명령어의 길이를 줄인 방식이다. 명령어의 길이가 짧아 수평적 마이크로 명령어에 비해 작은 제어 저장소가 필요하다.

디코더로 연산 필드의 내용을 해독하는 과정을 거쳐야 하므로 속도가 느리다.

**09. 다음 명령어의 위치는 현재 명령어에 직접 명시될 수도 있고 암묵적으로 명시될 수도 있다. 후자의 경우 ( PC 프로그램 계수기 ) 라는 레지스터를 사용하여 다음에 수행할 명령어의 위치를 나타낸다.**

**10. 오늘날의 컴퓨터에서 사용하는 제어장치 구현 방식에는 마이크로프로그래밍 방식 외에 조합 논리회로를 사용하여 구현하는 ( 고정결선식 방식 )이 있다.**

**11. 다중 사이클 방식의 데이터 경로에서 R-형식 명령어와 I-형식 적재 명령어는 마지막 단계가 레지스터 쓰기 사이클이다. 동일한 레지스터 쓰기 사이클이지만 제어 신호는 서로 다른데, 어떤 신호가 다르며 그 원인은 무엇인가?**

Reg Addr, Reg Data 신호가 다르다.

Reg Addr 신호는 rt, rd필드 중에 어느 필드를 레지스터 쓰기 단자의 주소로 사용하느냐를 의미한다.

Reg Data 신호는 DR레지스터, C버퍼 중에 어느 것을 레지스터 쓰기 단자의 데이터로 사용하느냐를 의미한다.

R-형식 명령어는 rt, rs 의 데이터를 처리한 결과를 rd 주소에 저장한다.

I-형식 명령어는 rs, imm의 데이터를 처리한 결과를 rt에 저장한다.