정보처리기사

제2과목 전자계산기 구조

01 부울 대수

문혜영교수

목차

부울대수

- 기본
  - AND
  - OR
  - NOT

- 불 대수의 기본 공식
  - ✓교환법칙 : $A+B = B+A$, $A \cdot B = B \cdot A$
  - ✓결합법칙 : $A+(B+C) = (A+B)+C$, $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
  - ✓분배법칙 : $A \cdot (B+C) = A \cdot B + A \cdot C$, $A+B \cdot C = (A+B) \cdot (A+C)$
  - ✓멱등법칙 : $A+A = A$, $A \cdot A = A$
  - ✓보수법칙 : $A+A' = 1$, $A \cdot A' = 0$
  - ✓항등법칙 : $A+0 = A$, $A+1 = 1$, $A \cdot 0 = 0$, $A \cdot 1 = A$
  - ✓콘센시스 : $AB+BC+CA' = AB+CA'$, $(A+B)(B+C)(C+A') = (A+B)(C+A')$
  - ✓드모르강 : $(A+B)' = A' \cdot B'$, $(A \cdot B)' = A'+B'$
  - ✓복원법칙 : $A'' = A$

- 논리식의 간소화

- 논리식 Y = AB+AB′+A′B를 최소화 시킨 것은?
  ① AB
  ② A+B
  ③ A+B′
  ④ AB′

정답 2

- 카르노 맵

정보처리기사

제2과목 전자계산기 구조

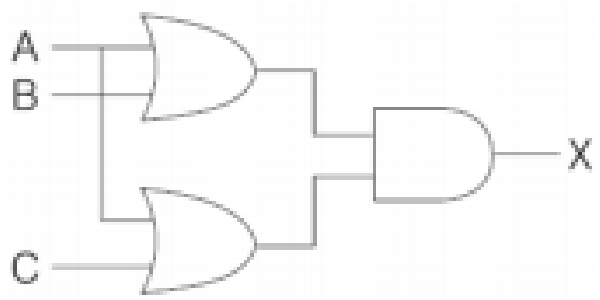
02 논리 게이트

문혜영교수

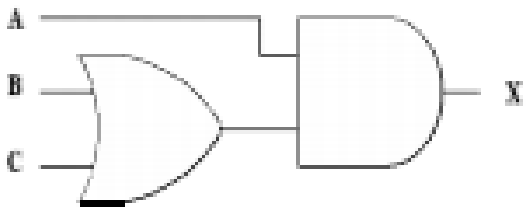
목차

- 논리 게이트
  - AND
  - OR
  - NOT
  - BUFFER
  - XOR

  - NAND
  - NOR
  - XNOR

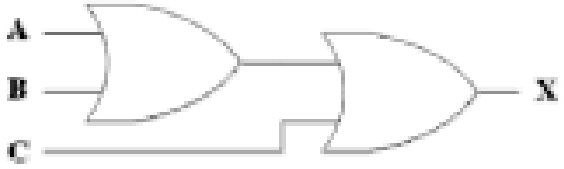


A+B·C



A·(B+C)

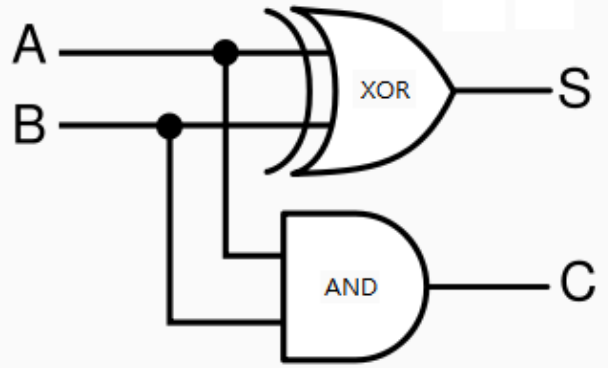
(A·B)·C



(A+B)+C

- 논리회로
  - 조합논리회로
  - 순서논리회로
- 조합논리회로의 종류
  - ✓반가산기, 전가산기, 병렬가산기, 반감산기, 전감산기, 디코더, 인코더, 멀티플렉서, 디멀티프렉서, 다수결회로, 비교기 등

- 반가산기(HA, Half Adder)
  - ✓2진수 두 개를 덧셈하여 합(S)과 자리올림수(C)를 구한다.
  - ✓논리회로

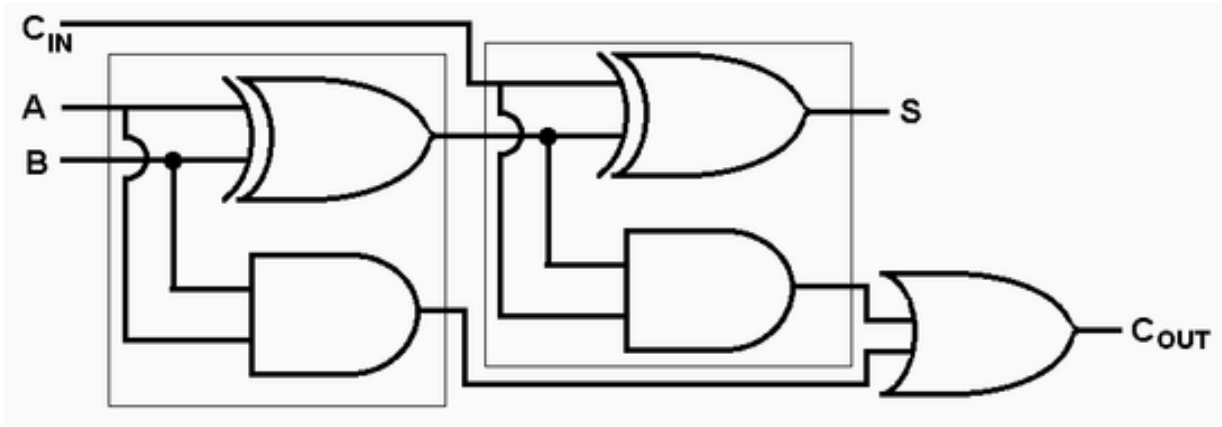


- 전가산기(FA, Full Adder)
  - ✓2진수 3자리를 더하여 합(S)과 자리올림수(C)를 구하는 회로이다.

  - ✓논리식 :

$$S = A \oplus B \oplus C_{in},\ C_{out} = (A \cdot B) + C_{in}(A \oplus B)$$

  - ✓회로 : 전가산기는 두 개의 반가산기와 한 개의 OR Gate로 구성된다.

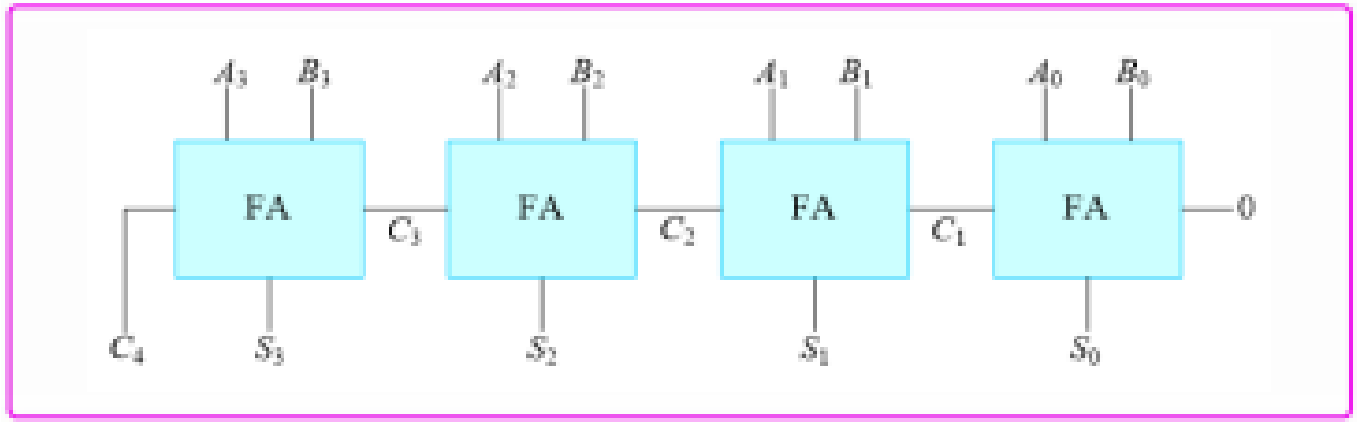


문제

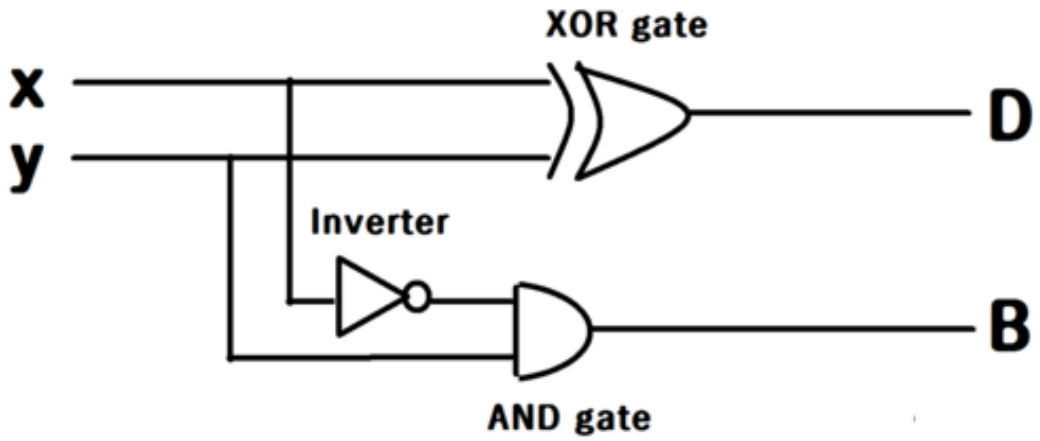
- 전가산기를 구성하기 위하여 필요한 소자를 바르게 나타낸 것은?
  ① 반가산기 2개, AND 게이트 1개
  ② 반가산기 1개, AND 게이트 2개
  ③ 반가산기 2개, OR 게이트 1개
  ④ 반가산기 1개, OR 게이트 2개

정답 3

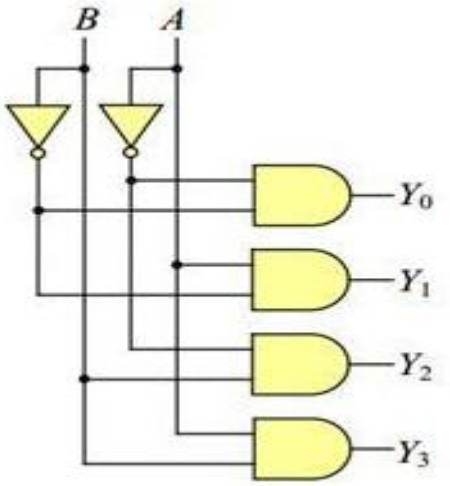
- 병렬가산기(PA, Parallel Adder)
  - ✓n Bit로 된 2진수 A, B에 대한 덧셈을 n개의 전가산기(FA)를 이용하여 구현한 실질적인 가산기이다.



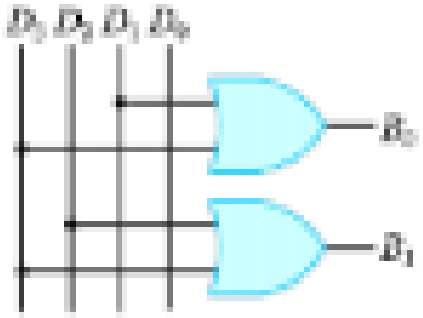
- 반감산기(HS, Half Subtrack)
  - ✓1Bit짜리 2진수 2자리에 대한 감산을 하는 회로이다.
  - ✓회로



- 디코더(Decoder)
  - ✓n Bit의 입력- 2^n개의 출력
  - ✓해독기
  - ✓$n \times 2^n$ 디코더는 2^n개의 AND 게이트가 소요된다.



✓인코더



- 멀티플렉서(MUX, Multiplexer)
  - ✓2^n의 입력선 중 1개를 선택한다.
  - ✓2^n개의 입력선 중 1개의 선을 선택하기 위해 n개의 선택선(Select Line)을 이용한다.

- 디멀티플렉서(DeMUX, DeMultiplecxer)
  - ✓1개의 입력선으로 들어오는 정보를 2^n개의 출력선 중 1개를 선택하여 출력하는 회로이다.
  - ✓2^n개의 출력선 중 1개의 선을 선택하기 위해 n개의 선택선(Select Line)을 이용한다.

문제

- N개의 입력 데이터에서 입력선을 선택하여 단일 채널로 송신하는 것은?
  ① 인코더
  ② 감산기
  ③ 전가산기
  ④ 멀티플렉서

정답 4

정보처리기사

제2과목 전자계산기 구조

03 순서논리회로

문혜영교수

목차

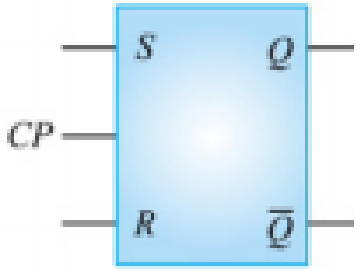
- 순서논리회로
  - ✓출력이 입력신호와 바로직전의 상태에 의해 결정된다.
  - ✓대표적인 순서논리회로에는 플립플롭, 카운터, 레지스터 등이 있다.

- 플립플롭(FF, Flip-Flop)의 특징
  - ✓1비트의 정보를 기억한다.
  - ✓레지스터, RAM, 카운터 등을 구성하는 기본 소자이다.

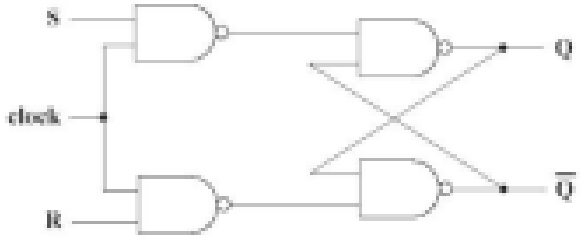
- RS 플립플롭(Reset-Set FF)

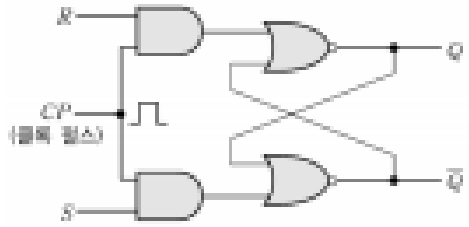
S	R	Q(t+1)	의미
0	0	Q(t)	전 상태 유지
0	1	0	RESET
1	0	1	SET
1	1	?	불능(허용안함)

진리표



논리기호



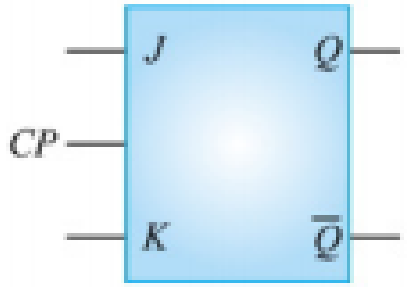


회로도

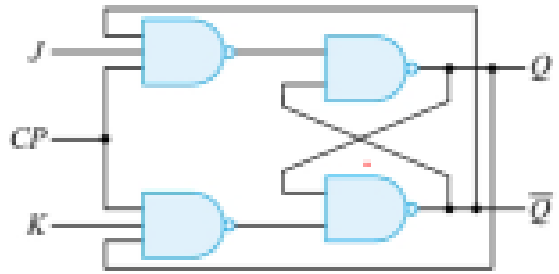
- JK 플립플롭

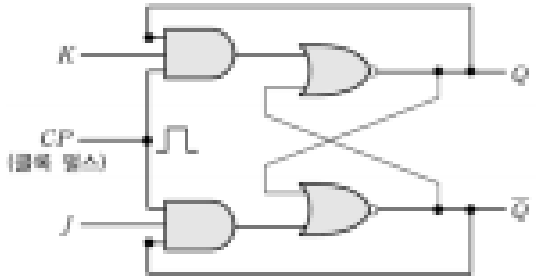
J	K	Q(t+1)	의미
0	0	$Q_{(t)}$	전 상태 유지.
0	1	0	RESET.
1	0	1	SET.
1	1	$\overline{Q_{(t)}}$	반전

진리표



논리기호



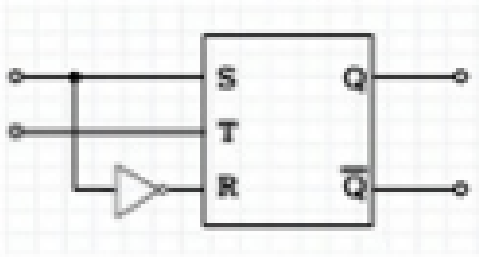


회로도

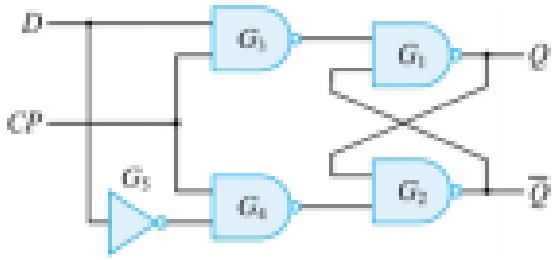
- D 플립플롭

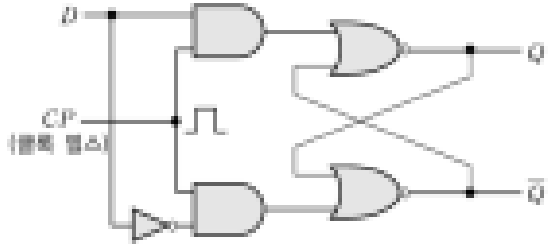
D	Q(t+1)	의미
0	0	RESET
1	1	SET

진리표



논리기호



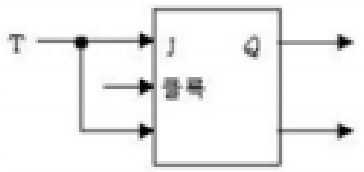


회로도

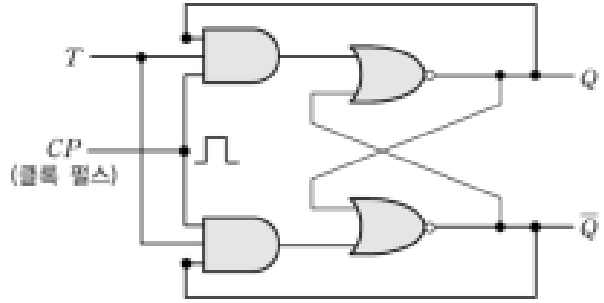
- T 플립플롭

T	Q(t+1)	의미
0	Q(t)	전 상태 유지
1	Q(t)	반전

진리표



논리기호



회로도

- 마스터-슬레이브 플립플롭(주종 플립플롭)
  - ✓출력 측의 일부가 입력 측에 궤환(FeedBack)되어 유발하는 레이스 현상을 없애기 위해 고안된 플립플롭이다.

문제

- JK 플립플롭의 트리거 입력과 상태 전환 조건을 설명한 것 중 옳지 않은 것은?
  ① J=O, K=0일 때 반전되지 않는다.
  ② J=0, K=1일 때 0으로 되돌아간다.
  ③ J=1, K=0일 때는 1로 된다.
  ④ J=1, K=0일 때는 반전된다.

정답 4

정보처리기사

제2과목 전자계산기 구조

04 자료의 표현

문혜영교수

목차

- 자료 구성의 단위
  - ✓비트(Bit) : n Bit로 2^n가지 표현 가능
  - ✓니블(Nibble) : 4Bit
  - ✓바이트(Byte)
    - ✓KB, MB, GB, TB, PB
  - ✓워드(Word)

  - ✓필드(Field)
  - ✓레코드(Record)
  - ✓블록(Block), 물리 레코드(Physical Record)
  - ✓파일(File)
  - ✓데이터베이스(Database)

문제

- 64가지의 각기 다른 자료를 나타내려고 하면 최소한 몇 개의 비트(Bit) 필요한가?
  ① 1
  ② 3
  ③ 5
  ④ 6

정답 4

- 수치 정보 표현 시 만족시켜야 할 조건
  - ✓ 기억장치의 공간을 적게 차지해야 한다.
  - ✓ 데이터 처리 및 CPU 내에서 이동이 용이해야 한다.
  - ✓ 10진수와 상호 변환이 용이해야 한다.
  - ✓ 한정된 수의 비트로 정밀도가 높게 표현해야 한다.

- 진법변환
  - ✓ 수의 표현
  - ✓ 10진수-> 2, 8, 16진수
  - ✓ 2, 8, 16진수-> 10진수
  - ✓ 2, 8, 16진수 상호 변환

문제

- 2진법의 수 1101.11을 10진법으로 표현하면?
  ① 11.75
  ② 13.55
  ③ 13.75
  ④ 15.3

정답 3

정보처리기사

제2과목 전자계산기 구조

05 보수

문혜영교수

목차

- 보수
  - ✓컴퓨터가 기본적으로 수행하는 덧셈 회로를 이용하여 뺄셈을 수행하기 위해 사용한다.
  - ✓r진법에는 r의 보수와 r-1의 보수가 있다.
    - ✓1의 보수는 주어진 각 자리값을 0일 때는 1로, 1일 때는 0으로 변환한다.
  - ✓보수를 이용한 뺄셈
    - ✓1의 보수 이용
    - ✓2의 보수 이용

문제

- 2진수 $(1001011)_2$의 2의 보수(2's Complement)는?
  ① 0110100
  ② 1110100
  ③ 1110101
  ④ 0110101

정답 4

- 고정 소수점 표현
  - ✓2진 연산
    - ✓음수의 표현 방법
      - ✓부호와 절대치
      - ✓부호와 1의 보수
      - ✓부호와 2의 보수
    - ✓표현 범위
      - ✓부호와 절대치 : $-(2^{n-1}-1) \sim +2^{n-1}-1$
      - ✓부호와 1의 보수 : $-(2^{n-1}-1) \sim +2^{n-1}-1$
      - ✓부호와 2의 보수 : $-2^{n-1} \sim +2^{n-1}-1$
  - ✓10진 연산
    - ✓언팩(Unpack) 연산 : 연산이 불가능하고, 데이터의 입·출력에 사용된다.
    - ✓팩(Pack) 연산 : 연산이 가능하고, 데이터의 입·출력이 불가능하다.

❖ 2의 보수 표현법이 널리 사용되는 이유
- ❖ 1의 보수 표현에 비해 덧셈 연산이 간단하다.
- ❖ 표현 범위가 넓다.

문제

- 2의 보수 표현방식으로 8Bit의 기억공간에 정수를 표현할 때 표현 가능한 범위는?
  ① $-2^7 \sim +2^7$
  ② $-2^8 \sim +2^8$
  ③ $-2^7 \sim +2^7 - 1$
  ④ $-2^8 \sim +2^8 - 1$

정답 3

정보처리기사

제2과목 전자계산기 구조

06 부동소수점

문혜영교수

목차

- 부동 소수점 방식의 특징
  ✓아주 큰 수나 작은 수, 매우 정밀한 수를 표현하는 데 적합하다.

❖ MFLOPS : 1초간에 실행되는 부동 소수점 연산의 수를 100만을 단위로 하여 나타낸 수.

❖ 정규화 : 유효 자릿수를 최대로 하여 수의 정밀도를 높이기 위한 것.

- ✓ 부동 소수점 수의 연산 방법
  - ✓덧셈, 뺄셈
    - ✓0인지 여부 조사,
    - ✓기수의 위치 조정(지수가 큰 쪽에 맞춤),
    - ✓기수부 값끼리 더하거나 뺀다,
    - ✓결과를 정규화한다.
  - ✓곱셈
    - ✓0인지 여부 조사,
    - ✓지수를 더한다,
    - ✓가수를 곱한다,
    - ✓결과를 정규화한다.
  - ✓나눗셈
    - ✓0인지 여부 조사,
    - ✓부호 결정, 피제수가 제수보다 작게 피제수의 위치를 조정한다,
    - ✓지수의 뺄셈을 한다,
    - ✓가수의 나눗셈을 한다.

문제

- Floating Point Number에서 저장 비트가 필요 없는 것은?
  ① 부호
  ② 지수
  ③ 소수점
  ④ 소수(가수)

정답 3

- 자료의 외부적 표현
  - ✓BCD(Binary Coded Decimal, 2진화 10진 코드)
  - ✓ASCII 코드
    - ✓2^7 = 128가지의 문자를 표현
    - ✓통신 제어용 및 마이크로컴퓨터의 기본 코드로 사용한다.
  - ✓EBCDIC(확장 2진화 10진 코드)
    - ✓8Bit 코드로 IBM에서 개발하였다
    - ✓2^8 = 256가지의 문자를 표현할 수 있다.

문제

- 다음 중 ASCII 문자에 해당하지 않는것은?
  ① 제어 문자
  ② 영문자
  ③ 로마 숫자
  ④ 아라비아 숫자

정답 3

정보처리기사

제2과목 전자계산기 구조

07 코드

문혜영교수

목차

- 기타 자료의 표현 방식
  - ✓BCD 코드
    - ✓8421코드라고도 한다.
    - ✓대표적인 가중치 코드이다.
  - ✓Excess-3 코드(3초과 코드)
    - ✓BCD 코드에 $3_{10}(= 0011_2)$을 더하여 만든 코드이다.
    - ✓대표적인 자기보수 코드이며, 비가중치 코드이다.
  - ✓Gray 코드
    - ✓BCD 코드의 인접하는 비트를 X-OR 연산하여 만든 코드이다.
    - ✓A/D 변환기에 사용한다.
    - ✓1Bit만 변환시켜 다음 수치로 증가시키기 때문에 하드웨어적인 오류가 적다.

- ✓패리티 검사 코드
  - ✓1Bit의 오류만 검출할 수 있다.
  - ✓1의 개수에 따라 짝수(Even, 우수)패리티와 홀수(Odd, 기수) 패리티 방법이 있다.
- ✓해밍 코드
  - ✓오류를 스스로 검출하여 교정이 가능한 코드이다.
  - ✓해밍코드 중 1, 2, 4, 8, 16, … , 2^n번째는 오류 검출을 위한 패리티 비트이다.

- ✓코드의 분류
  - ✓가중치 코드 : BCD(8421), 2421, 84-2-1, Biquinary(5043210), 51111, Ring-Counter(9876543210)
  - ✓비가중치 코드 : 3 초과(Excess-3), Gray, Jonson, 2-out-of-5, 3-out-of-5
  - ✓자기 보수 코드 : 3 초과(Excess-3), 2421, 51111, 84-2-1
  - ✓오류 검출용 코드 : 해밍 코드, 패리티 검사 코드, Biquinary, Ring-Counter, 2-out-of-5, 3-out-of-5

문제

- 10진수 956에 대한 BCD 코드(Binary Coded Decimal)는?
  - ① 1001 0101 0110
  - ② 1101 0110 0101
  - ③ 1000 0101 0110
  - ④ 1010 0110 0101

정답 1

정보처리기사

제2과목 전자계산기 구조

08 중앙처리장치

문혜영교수

목차

- 중앙처리장치 : 제어장치, 연산장치, 레지스터, 버스
- 제어장치
  - ✓모든 장치들의 동작을 지시하고 제어하는 장치이다.

- 연산장치
  - ✓산술 연산, 논리 연산, 관계 연산, 이동(Shift) 등을 수행한다.
  - ✓가산기, 누산기, 보수기, 데이터 레지스터등
  - ✓누산기(AC, Accumulator) : 연산 결과를 기억하는 장치이다.

- 레지스터
  - ✓CPU 내부에서 처리할 명령어나 연산의 중간 결과값 등을 일시적으로 기억하는 임시 기억장치이다.
  - ✓레지스터 간의 자료 전송
    - ✓직렬 전송 : 직렬 시프트 마이크로 오퍼레이션을 뜻하며, 병렬 전송에 비해 전송 속도가 느리다.
    - ✓병렬 전송 : 하나의 클록 펄스 동안에 레지스터 내의 모든 비트, 즉 워드가 동시에 전송되는 전송 방식이다.
    - ✓버스 전송 : 병렬 전송에 비해 결선의 수를 줄일 수 있다는 장점이 있다.

- ✓레지스터의 종류 및 기능
  - ✓프로그램 카운터 : 다음에 실행할 명령어의 번지를 기억하는 레지스터
  - ✓명령 레지스터 : 현재 실행 중인 명령의 내용을 기억하는 레지스터
  - ✓누산기(AC, Accumulator) : 연산된 결과를 일시적으로 저장하는 레지스터
  - ✓PSWR(상태 레지스터, 플래그 레지스터) : 시스템 내부의 순간순간의 상태를 기록하고 있는 정보, 오버플로, 언더플로, 자리올림등
  - ✓MAR(메모리 주소 레지스터) : 데이터의 번지를 기억하는 레지스터
  - ✓MBR(메모리 버퍼 레지스터) : 데이터가 잠시 기억되는 레지스터
  - ✓인덱스 레지스터(Index Register) : 주소의 변경, 서브루틴 연결 및 프로그램에서의 반복 연산의 횟수를 세는 레지스터
  - ✓데이터 레지스터(Data Register)
  - ✓시프트 레지스터(Shift Register)
  - ✓메이저 스테이터스 레지스터(Major Status Register)

- 버스
  - ✓버스를 통해 전달되는 제어 신호, 어드레스 신호 및 데이터 신호의 상호 시간적 관계가 잘 유지되어야 한다.

  - ✓전송하는 정보에 따른 분류
    - ✓주소 버스(Address Bus)
    - ✓자료 버스(Data Bus)
    - ✓제어 버스(Control Bus)

  - ✓버스 위치에 따른 분류
    - ✓내부 버스
    - ✓외부 버스

문제

- 누산기(Accumulator)란?
  ① 연산장치에 있는 레지스터(Register)의 하나로 연산 결과를 기억하는 장치이다.
  ② 기억장치 주변에 있는 회로인데 가감승제 계산 및 논리연산을 행하는 장치이다.
  ③ 일정한 입력 숫자들을 더하여 그 누계를 항상 보관하는 장치이다.
  ④ 정밀 계산을 위해 특별히 만들어 두어 유효 숫자의 개수를 늘리기 위한 것이다.

정답 1

- 명령어의 구성
  - ✓연산자부(Operation Code부)
  - ✓모드(Mode)부
  - ✓자료(Operand)부
- 명령어 설계 시 고려사항
  - ✓연산자의 종류
  - ✓명령어 형식
  - ✓주소지정방식
  - ✓데이터 구조
  - ✓인스트럭션 세트의 효율성을 높이기 위하여 고려할 사항 : 기억 공간, 사용 빈도, 주소지정 방식

- 연산자(Operation Code)의 기능
  - ✓함수 연산 기능
    - ✓수치적인 산술 연산 : ADD, SUB, MUL, DIV, 산술 Shift 등
    - ✓비수치적인 논리 연산 : NOT, AND, OR, XOR, 논리적 Shift, Rotate, Complement, Clear 등
  - ✓자료 전달 기능
    - ✓Load : 기억장치에 기억되어 있는 정보를 CPU(레지스터)로 꺼내오는 명령
    - ✓Stor : CPU(레지스터)에 있는 정보를 기억장치에 기억시키는 명령
    - ✓Move, Push, PoP
  - ✓제어 기능
    - ✓무조건 분기 명령, 조건 분기 명령, Call(서브루틴 호출), Return(복귀)
  - ✓입·출력 기능 : INPUT, OUTPUT

- ✓ 단항 연산자(Unary Operator) : NOT, Complement, Shift, Rotate, Move 등
- ✓ 이항 연산자(Binary Operator) : 사칙연산, AND, OR, XOR, XNOR 등
- ✓ 연산자 우선 순위 : 산술 연산자 > 관계 연산자 > 논리 연산자

문제

- 레지스터의 내용을 메모리에 전달하는 기능을 무엇이라 하는가?
  ① Fetch
  ② Store
  ③ Load
  ④ Transfer

정답 2

- 연산(Operation)
  - ✓AND(Masking Operation)
    - ✓특정 문자 또는 특정 비트를 삭제(Clear)시키는 연산으로, Masking 연산이라고도 한다.
    - ✓삭제할 부분의 비트를 0과 AND시켜서 삭제하는데, 대응시키는 0인 비트를 Mask Bit라 한다.
  - ✓OR(Selective-Set)
    - ✓Selective-Set 연산이라고도 한다.
    - ✓삽입하거나 세트시킬 비트에 삽입할 문자 코드 또는 1을 OR시킨다.
  - ✓XOR(Compare, 비교)
    - ✓두 개의 데이터를 비교하거나 특정 비트를 반전시킬 때 사용한다.

정보처리기사

제2과목 전자계산기 구조

09 연산

문혜영교수

목차

- ✓연산
  - ✓AND
  - ✓OR
  - ✓XOR
  - ✓NOT
  - ✓논리 Shift
  - ✓Rotate
  - ✓산술 Shift
    - ✓2^n으로 곱하거나 나눌 때 사용한다.
    - ✓왼쪽으로 n Bit Shift하면 원래 자료에 2^n을 곱한 것과 같다.
    - ✓오른쪽으로 n Bit Shift하면 원래 자료를 2^n으로 나눈 것과 같다.
    - ✓홀수를 오른쪽으로 한 번 Shift하면 0.5의 오차가 발생한다.

문제

- A의 내용이 1010, B의 내용이 1100이다. Masking Operation 후의 A 내용은?
  ① 1000
  ② 0010
  ③ 1110
  ④ 0110

정답 1

정보처리기사

제2과목 전자계산기 구조

10 명령어

문혜영교수

목차

- 명령어 형식
  - ✓3주소 명령어
    - ✓연산의 결과는 주로 Operand 1에 기록한다.
    - ✓연산 시 원래의 자료를 파괴하지 않는다.
  - ✓2주소 명령어
    - ✓가장 일반적으로 사용되는 명령어 형식이다.
    - ✓실행 속도가 빠르고 기억 장소를 많이 차지하지 않는다.
    - ✓원래의 자료가 파괴된다.

✓1주소 명령어
  ✓누산기를 이용

✓0주소 명령어
  ✓스택 머신이라고도 한다.
  ✓인스트럭션 수행시간이 짧다.
  ✓스택을 사용한 컴퓨터에서 수식을 계산하기 위해서는 수식을 Postfix 형태로 변경해야 한다.

- Stack의 응용 분야
  - 부 프로그램 호출 시 복귀주소
  - 재귀(Recursion) 프로그램의 순서 제어

문제

- 주소 부분이 하나밖에 없는 1-주소 명령 형식에서 결과 자료를 넣어두는데 사용하는 레지스터는?
  ① 누산기(Accumulator)
  ② 스택(Stack)
  ③ 인덱스(Index) 레지스터
  ④ 범용 레지스터

정답 1

- 주소 설계 시 고려 사항
  - ✓표현의 효율성
  - ✓사용의 편리성
  - ✓주소공간과 기억공간의 독립성

- 주소지정방식(Addressing Mode)의 종류
  - ✓묵시적(Implied), 즉시적(Immediate), 직접(Direct), 간접(Indirect), 계산에 의한 주소지정방식이 있다.

- ✓암시적(묵시적) 주소지정방식(Implied Mode)
- ✓즉시 주소지정방식(Immediate Mode)
  - ✓명령어 자체에 실제 데이터를 가지고 있는 방식이다.
  - ✓별도의 기억장소를 액세스하지 않고 실행 속도가 빠르다.
- ✓직접 주소지정방식(Direct Mode)
  - ✓Operand부에 실제 사용할 데이터의 주소를 표현
  - ✓주소 길이에 제약을 받는다.
- ✓간접 주소지정방식(Indirect Mode)
  - ✓Operand부에 실제 데이터의 주소가 저장된 곳의 주소를 표현
  - ✓최소한 주기억장치를 두 번 이상 접근
  - ✓긴 주소에 접근 가능한 방식이다.

✓계산에 의한 주소지정방식

- ✓상대 주소 : 명령어의 주소 부분 + PC
- ✓베이스 레지스터 : 명령어 주소 부분 + Base Register
- ✓인덱스 레지스터 : 명령어의 주소 부분 + Index Register

문제

- 명령어의 주소 부분과 프로그램 카운터(PC)의 값을 더해서 유효 주소를 결정하는 주소 모드는?
  ① Implied 모드
  ② Relative Address 모드
  ③ Index Address 모드
  ④ Register Indirect 모드

정답 2

정보처리기사

제2과목 전자계산기 구조

11 마이크로 오퍼레이션

문혜영교수

목차

- 마이크로 오퍼레이션의 정의
  - ✓한 개의 Clock 펄스 동안 실행되는 기본 동작으로 모든 마이크로 오퍼레이션은 CPU의 Clock 펄스에 맞춰 실행된다.

- 마이크로 사이클 타임(Micro Cycle Time)
  - ✓한 개의 Micro Operation을 수행하는데 걸리는 시간

- Micro Cycle Time 부여 방식
  - ✓동기 고정식(Synchronous Fixed)
    - 모든 마이크로 오퍼레이션 중에서 동작시간이 가장 긴 마이크로 오퍼레이션의 동작시간을 Micro Cycle Time이라 한다.
    - 모든 마이크로 오퍼레이션의 동작시간이 비슷할 때 유리한 방식이다.
  - ✓동기 가변식(Synchronous Variable)
    - 각 그룹별로 서로 다른 Micro Cycle Time을 정의하는 방식이다.
    - 마이크로 오퍼레이션 수행시간이 현저한 차이를 나타낼 때 사용한다.
  - ✓비동기식(Asynchrorous)

문제

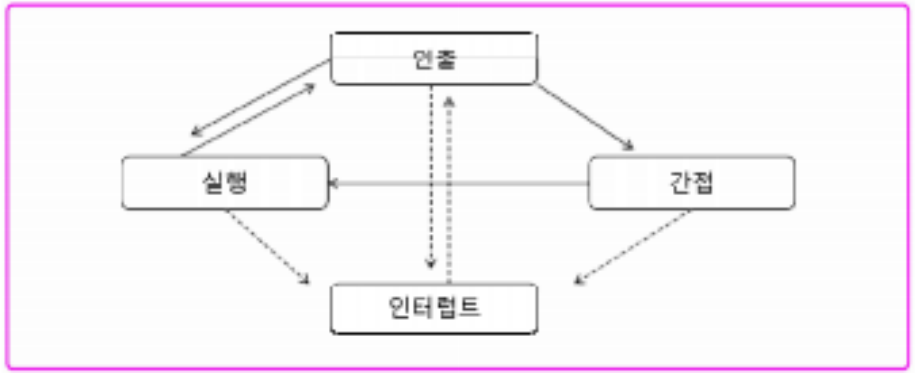
- 동기 가변식(Synchronous Variable) 동작에 대한 설명 중 옳지 않는 것은?
  ① 각 마이크로 오퍼레이션의 사이클 타임이 현저한 차이를 나타낼 때 사용한다.
  ② 모든 마이크로 오퍼레이션의 수행 시간이 유사한 경우에 사용한다.
  ③ 중앙처리장치의 시간을 효율적으로 이용할 수 있다.
  ④ 마이크로 오퍼레이션에 대하여 서로 다른 사이클을 정의할 수 있다.

정답 2

- CPU의 Major상태

플립플롭상태		메이저 상태
F	R	
0	0	Fetch(인출)
0	1	Indirect(간접)
1	0	Execute(실행)
1	1	Interrupt(인터럽트)

- 명령어 처리과정



- 인출 단계(Fetch Cycle)

Micro Operation		의미
T0	MAR ← PC	
T1	MBR ← M(MAR), PC ← PC+1	메모리에서 MAR이 지정하는 위치의 값을 MBR에 전송 PC값을 1증가시킴
T2	IR ← MBR	MBR에 있는 명령어를 명령레지스터에 전송
T3	F ← 1 또는 R ← 1	F플립플롭에 1을 전송하여 실행 사이클로 넘어감 또는 R플립플롭에 1을 전송하여 간접 사이클로 넘어감

- 간접 단계(Indirect Cycle)

Micro Operation		의미
T0	MAR ← MBR(ADDR)	MBR에 있는 명령어의 주소를 MAR에 전송
T1	MBR ← M(MAR)	메모리에서 MAR이 지정하는 위치의 값을 MBR에 전송
T2	NOP	동작 없음
T3	F ← 1, R ← 0	F에 1, R에 0을 전송하여 Excute Cycle로 이동

- 실행단계

Micro Operation		의미
T0	MAR ← MBR(ADDR)	MBR에 있는 명령어의 주소를 MAR에 전송
T1	MBR ← M(MAR)	메모리에서 MAR이 지정하는 위치의 값을 MBR에 전송
T2	AC ← AC+MBR	AC의 값과 MBR의 값을 더해 AC에 전송
T3	F ← 0 또는 R ← 1	Execute Cycle은 F=1, R=0인 상태이다. 이때 F에 0을 전송하면 결과 F=0, R=0이 되어 Fetch단계로 이동하게 된다. R에 1을 전송하면 결과 F=1, R=1이 되어 Interrupt단계로 이동한다.

- 인터럽트단계

Micro Operation		의미
T0	MBR(AD) ← PC, PC ← 0	PC에 저장되어 있는 번지를 MBR의 주소부분으로 전송 복귀주소를 저장할 0번지를 PC에 전송
T1	MAR ← PC, PC ← PC+1	PC에 저장되어 있는 번지를 MAR에 전송 인터럽트 벡터의 위치를 지정하기 위하여 PC값을 1증가시킴
T2	M(MAR) ← MBR, IEN ← 0	MBR를 메모리의 MAR이 지정하는 위치(0번지)로 전송 인터럽트단계가 끝날 때까지 다른 인터럽트가 방해하지 않게 IEN에 0을 전송
T3	F ← 0, R ← 0	F에 0, R에 0을 전송하여 Fetch Cycle로 이동

문제

- 간접 사이클(Indirect Cycle) 동안에 컴퓨터는 무엇을 하는가?
  ① 명령을 읽는다.
  ② 오퍼랜드(Operand)를 읽는다.
  ③ 인터럽트(Interrupt)를 처리한다.
  ④ 오퍼랜드(Operand)의 어드레스(Address)를 읽는다.

정답 4

정보처리기사

제2과목 전자계산기 구조

12 제어기의 구현

문혜영교수

목차

- LDA

T0	MAR ← MBR(ADDR)	MBR에 있는 명령어의 주소를 MAR에 전송
T1	MBR ← M(MAR), AC ← 0	메모리의 MAR이 지정하는 위치의 값을 MBR에 전송, AC를 0으로 초기화
T2	AC ← AC+MBR	AC의 값과 MBR의 값을 더한 후 누산기에 전송
T3	F←0, R←0	인출주기로 전이

- STA

Micro Operation		의미
T0	MAR ← MBR(ADDR)	MBR에 있는 명령어의 주소를 MAR에 전송
T1	MBR ← AC	AC값을 MBR에 전송
T2	M(MAR) ← MBR	MBR의 값을 메모리의 MAR이 지정하는 위치에 전송
T3	F←0, R←0	인출주기로 전이

- BUN
  (Branch UNconditionally)

PC에 특정한 주소를 전송하여 실행명령의 위치를 변경하는 무조건 분기명령이다.

PC ← MBR(ADDR)

- BSA
  (Branch and Save Return Address)

복귀주소를 저장하고 부프로그램을 호출하는 명령이다

문제

- 다음의 마이크로 오퍼레이션과 관련 있는 것은?

MAR ← MBR[ADDR]
MBR ← M[MAR]
AC ← AC + MBR

① AND
② ADD
③ JMP
④ BSA

정답 2

- 제어 데이터: 제어신호를 발생하기 위한 자료
- 제어 데이터 종류
  - ✓메이저 스테이트 사이의 변천을 제어하는 데이터
  - ✓중앙처리장치의 제어점을 제어하는 데이터
  - ✓인스트럭션의 순서를 결정하는데 필요한 제어 데이터
- 제어기의 구현
  - ✓고정배선 제어장치(Hard-wired Control Unit)
    - ✓조합논리회로를 설계하여 해당 제어점에 연결하는 방식이다.
    - ✓Hardware적인 구성 방법이다. 고속. 회로 구성이 복잡하다.
  - ✓마이크로 프로그래밍 기법(Micro Programmed Control Unit)
    - ✓내부 제어 신호를 여러 가지 마이크로 인스트럭션으로 작성하는 것이다.
    - ✓Software적인 구성 방법이다. 펌웨어(Firmware)를 이용하는 방식이다. 저속

- 마이크로 프로그램에서 제어 메모리의 번지 결정
  - ✓다음에 실행할 제어 메모리의 번지를 결정하는 정보
    - ✓제어 주소 레지스터(CAR)
    - ✓명령 레지스터(IR)
    - ✓상태 레지스터(SR)

- 마이크로 명령의 형식
  - ✓수평 마이크로 명령(Horizontal Micro Instruction)
  - ✓수직 마이크로 명령(Vertical Micro Instruction)
  - ✓나노 명령(Nano Instruction)

문제

- 마이크로 프로그램(Micro Program)에 대한 설명 중 옳지 않은 것은?
  ① 마이크로 프로그램은 보통 RAM에 저장한다.
  ② 마이크로 프로그램은 각종 제어 신호를 발생시킨다.
  ③ 마이크로 프로그램은 마이크로 명령으로 형성되어 있다.
  ④ 마이크로 프로그램은 CPU 내의 제어장치를 설계하는 프로그램이다.

정답 1

- 입·출력장치의 구성
  - ✓입·출력 제어장치
    - ✓입출력장치와 컴퓨터사이의 자료 전송을 제어한다.
    - ✓데이터 버퍼 레지스터를 이용하여 두 장치 간의 속도 차를 조절한다.
  - ✓입·출력 인터페이스
    - ✓내부장치(주기억장치, CPU)와 외부 입출력장치사이에 정보를 원활하게 전송하기 위한 방법이다.
  - ✓입·출력 버스
    - ✓데이터버스, 주소버스, 제어버스

- 기억장치와 입·출력장치의 동작 차이

비교 항목	입·출력장치	기억장치
동작의 속도	느리다	빠르다
정보의 단위	Byte(문자)	Word
착오 발생률	많다	적다

- 스풀링(Spooling, Simultaneous Peripheral Operation On-Line)
  - ✓디스크에 입·출력할 데이터를 모았다가 나중에 한꺼번에 입·출력한다.
  - ✓여러 작업을 병행 수행할 수 있도록 하여 다중 프로그래밍 시스템의 성능 향상을 가져온다.
  - ✓디스크 일부를 매우 큰 버퍼처럼 사용하며, 큐 방식의 입·출력을 수행한다.

구분	버퍼링	스풀링
저장 위치	주기억장치	보조기억장치
운영 방식	단일 연산	다중 작업
구현 방식	하드웨어	소프트웨어

문제

- I/O 효율을 높이기 위해 I/O의 내용을 디스크 등에 모아 두었다가 처리하는 방식은?
  ① Overlapping
  ② Pipelining
  ③ Spooling
  ④ Relocation

정답 3

정보처리기사

제2과목 전자계산기 구조

13 인터럽트

문혜영교수

목차

- 입·출력 제어 방식

제어 방식	CPU 관여 여부	특징
Program의 의한 I/O	○	가장 원시적인 방식
Interrupt에 의한 I/O	○	
DMA에 의한 I/O	×	소형 컴퓨터에서 이용
Channel에 의한 I/O	×	대형 컴퓨터에서 이용

- Programmed I/O
  - ✓CPU가 계속 Flag를 검사

✓Interrupt I/O
  - ✓CPU가 계속 Flag를 검사하지 않고,
  - ✓입·출력장치의 요구가 있을 때 데이터를 전송하는 제어방식이다.

- DMA(Direct Memory Access)에 의한 I/O
  - ✓CPU를 거치지 않고 메모리와 입·출력장치가 직접 통신
  - ✓Cycle Steal 방식을 이용하여 데이터를 전송한다.

✓DMA의 데이터 전송 절차

- ✓버스 사용 요구-버스 사용 허가-데이터 전송-완료되면 인터럽트 신호를 보냄

- Cycle Steal
  - ✓채널과 CPU가 주기억장치를 동시에 Access할 때 우선순위를 데이터 채널에게 주는 방식이다.
  - ✓한 번에 한 데이터 워드를 전송하고 버스와 제어를 CPU에게 돌려줍니다.
  - ✓CPU(중앙처리장치)는 메모리 참조가 필요 없는 오퍼레이션을 계속 수행합니다.

- 채널 제어기에 의한 I/O
  - ✓CPU 관여 없이 주기억장치와 입·출력장치 사이에서 입·출력을 제어
  - ✓DMA 방법으로 입·출력을 수행한다.

✓채널의 종류

- ✓Selector Channel : 고속 입·출력 장치, 1개의 장치를 관리,(자기 디스크, 자기 테이프, 자기드럼)
- ✓Multiplexer Channel : 저속 입·출력 장치(카드 리더, 프린터), 동시에 여러 개의 입·출력 장치를 제어한다.
- ✓Block Multiplexer Channel

문제

- 다음 중 DMA의 설명이 옳지 않는 것은?
  ① DMA는 Direct Memory Access의 약자이다.
  ② DMA는 기억장치와 주변장치 사이의 직접적인 데이터 전송을 제공한다.
  ③ DMA는 블록으로 대용량의 데이터를 전송할 수 있다.
  ④ DMA는 입·출력 전송에 따른 CPU의 부하를 증가시킬 수 있다.

정답 4

- 인터럽트의 정의
  - ✓프로그램을 실행하는 도중에 예기치 않은 상황이 발생한 경우
  - ✓현재 실행 중인 작업을 잠시 중단하고 발생된 상황을 우선 처리한 후 실행 중이던 작업으로 복귀하는 것을 말한다.
- 인터럽트 종류 및 발생 원인
  - ✓외부 인터럽트
    - ✓전원 이상 인터럽트
    - ✓기계 착오 인터럽트
    - ✓외부 신호 인터럽트(타이머, 인터럽트 키)
    - ✓입·출력 인터럽트
  - ✓내부 인터럽트
    - ✓프로그램 검사 인터럽트(0으로 나눌 때, Overflow 또는 Underflow, 불법적인 명령)
  - ✓소프트웨어 인터럽트
    - ✓SVC 인터럽트

- 인터럽트 발생 시 CPU가 확인할 사항
  - ✓프로그램 카운터의 내용
  - ✓레지스터의 내용
  - ✓상태조건의 내용(PSW)
- 인터럽트의 동작 원리(수행 순서)
  - ✓인터럽트 요청 신호 발생
  - ✓현재 수행중인 명령을 완료하고, 상태를 기억시킨다.
  - ✓인터럽트를 요청한 장치를 식별
  - ✓인터럽트 서비스 루틴을 실행한다.
  - ✓보존한 프로그램 상태를 복귀
  - ✓중단된 프로그램 실행 재개

문제

- Interrup 발생 원인이 아닌 것은?
  ① 정전
  ② 기억장치 내 허용되지 않는 곳에의 접근 시도
  ③ Operator의 조작
  ④ 임의의 부 프로그램에 대한 호출

정답 4

- 인터럽트 우선순위

  전원 이상 → 기계 착오 → 외부 신호 → 입·출력 → 명령어 잘못 → 프로그램 → SVC

- 인터럽트 우선순위
  - 하나 이상의 인터럽트가 발생하였을 때 먼저 서비스할 장치를 결정

  ✓인터럽트 우선순위 체제의 기능
  - ✓각 장치에 우선순위를 부과하는 기능
  - ✓인터럽트를 요청한 장치의 우선순위를 판별하는 기능
  - ✓우선순위가 높은 것을 먼저 처리할 수 있는 기능

- 소프트웨어적인 인터럽트 우선순위 판별 방법 : Polling
  - ✓인터럽트 요청 플래그를 차례로 검사.
  - ✓속도가 느리지만, 경제적이고 회로가 간단하며 융통성이 있다.

- 하드웨어적인 인터럽트 우선순위 판별 방법 : Vectored Interrupt
  - ✓반응 속도가 빠르다, 비싸고, 회로가 복잡하고 융통성이 없다.

  - ✓직렬(Serial) 우선순위 부여 방식 : 데이지 체인(Daisy-Chain) 방식
    - ✓한 개의 회선에 직렬로 연결한다. 우선순위가 높은 장치를 선두에 위치시킨다.
  - ✓병렬(Parallel) 우선순위 부여 방식
    - ✓Mask Register를 사용한다.
    - ✓마스크 레지스터는 우선순위가 높은 것이 서비스 받고 있을 때 우선순위가 낮은 것을 비활성화시킬 수 있다.

문제

- 우선순위 인터럽트 가운데 소프트웨어의 처리 기법은?
  ① 스트로브(Strobe) 방법
  ② 폴링(Polling) 방법
  ③ 병렬 우선순위(Paralled Priority) 방법
  ④ 데이지-체인(Daige-Chain) 방법

정답 2

정보처리기사

제2과목 전자계산기 구조

14 기억장치

문혜영교수

목차

- 기억장치의 분류
  - ✓주기억장치
    - ✓반도체
      - ✓RAM : SRAM, DRAM
      - ✓ROM : Mask ROM, PROM, EPROM, EEPROM
    - ✓자기(Magnetic) : 자기 코어, 자기 박막 필름
  - ✓보조기억장치
    - ✓DASD : 자기 디스크, 자기 드럼, 하드디스크, 플로피 디스크, 광 디스크
    - ✓SASD : 자기테이프
  - ✓특수 기억장치
    - ✓복수 모듈 기억장치, 연관기억장치, 캐시 기억장치, 가상기억장치

- 기억장치의 특성을 결정하는 요소
  - ✓기억 용량
  - ✓Access Time
    - ✓읽기 요청이 발생한 시간부터 정보를 꺼내서 사용 가능할 때까지의 시간이다.
  - ✓Cycle Time
    - ✓기억장치에 읽기 신호를 보낸 후 다시 읽기 신호를 보낼 수 있을 때까지의 시간
    - ✓Cycle Time이 Access Time보다 길거나 같다
  - ✓Bandwidth(대역폭, 전송률)
    - ✓메모리로부터 또는 메모리까지 1초 동안 전송되는 최대한의 정보량

- 기억장치의 특성에 따른 구분
  - ✓내용의 보존 여부
    - ✓파괴성 메모리 : 자기 코어
    - ✓비파괴성 메모리 : 자기 코어를 제외한 모든 기억장치
  - ✓전원 단절 시 내용 소멸 여부
    - ✓휘발성 메모리
    - ✓비휘발성 메모리
  - ✓재충전(Refresh) 여부
    - ✓정적 메모리(SRAM)
    - ✓동적 메모리(DRAM)
  - ✓접근 방식
    - ✓순차접근저장 매체(SASD)
    - ✓직접접근저장 매체(DASD)

문제

- 주기억장치는 하드웨어 특정상 주기억장치가 제공할 수 있는 정보 전달 능력에 한계가 있는데, 이 한계를 무엇이라 하는가?
  ① 주기억장치 전달
  ② 주기억장치 접근폭
  ③ 주기억장치 밴드 폭
  ④ 주기억장치 정보 전달폭

정답 3

- ROM(Read Only Memory)
  - ✓읽기만 가능하고 쓰기는 불가능한 기억장치
  - ✓ROM의 종류와 특징
    - ✓Mask ROM
    - ✓PROM
    - ✓EPROM
    - ✓EEPROM
    - ✓EAROM

- RAM(Random Access Memory)
  - ✓DRAM과 SRAM의 특징

구분	동적 램(DRAM)	정적 램(SRAM)
구성 소자	콘덴서	플립플롭
재충전시간	재충전(Refresh)시간이 필요	불필요
전력 소모	적음	많음
접근 속도	느림	빠름
집적도(밀도)	높음	낮음
가격	저가	고가
용도	일반적인 주기억장치	캐시 메모리

- 자기 코어
  - ✓데이터를 읽으면 읽은 내용이 지워지는 파괴 메모리
  - ✓저장을 위해 1Bit마다 하나의 코어 플레인(Core Plane)이 필요하다.

✓주기억장치의 용량 계산

- 주소선의 수 = MAR(AR) = PC
- Data Bus의 비트 수 = MBR(DR) = IR

문제

- 기억장치의 총 용량이 4,096워드이고 워드 길이가 16Bit일 때 프로그램 카운트(PC), 주소 레지스터(AR), 데이터 레지스터(DR)의 크기로서 바른 것은?

	PC	AR	DR
①	12,	12,	16
②	12,	12,	8
③	8,	8,	16
④	16,	8,	16

정답 1

정보처리기사

제2과목 전자계산기 구조

15 기타 기억장치

문혜영교수

목차

- 보조기억장치의 일반적인 특징
  - ✓중앙처리장치와 직접 자료 교환이 불가능하다.
  - ✓접근 시간(Access Time)이 오래 걸린다.

- 자기 테이프(Magnetic Tape)
  - ✓순차 처리(SASD)만 가능하다.
  - ✓블록 단위로 데이터를 전송하며, 블록과 블록 사이에는 GAP이 있다.

- 자기 디스크(Magnetic Disk)
  - ✓자기 디스크의 구조
    - ✓트랙
    - ✓섹터
    - ✓실린더
  - ✓ Access Time
    - ✓Access Time = Seek Time + Latency Time + Transmission Time
- 자기 드럼(Magnetic Drum)

문제

- 랜덤(Random) 처리가 되지 않는 기억장치는?
  ① 자기 드럼
  ② 자기 디스크
  ③ 자기 테이프
  ④ 자기코어

정답 3

- 연관기억장치(Associative Memory)
  - ✓기억장치에서 자료를 찾을 때 주소에 의해 접근하지 않고, 기억된 내용의 일부를 이용하여 Access할 수 있는 기억장치이다.
- 복수 모듈 기억장치
  - ✓주기억장치와 CPU의 속도 차이 문제점을 개선한다.
  - ✓기억장치의 버스를 시분할하여 사용한다.
  - ✓기억장소의 접근을 보다 빠르게 한다.
- 메모리 인터리빙(Memory Interleaving)
  - ✓기억장치를 각 모듈이 순차적으로 번갈아 가며 접근하는 방법이다.
  - ✓대역폭을 효율적으로 높일 수 있다.
  - ✓캐시 기억장치, 고속 DMA 전송 등에서 많이 사용된다.

문제

- 메모리 인터리빙(Interleaving)의 설명이 아닌 것은?
  ① 저속의 블록 단위 전송이 가능하다.
  ② 캐시 기억장치, 고속 DMA 전송 등에서 많이 사용된다.
  ③ 기억장치의 접근시간을 효율적으로 높일 수 있다.
  ④ 각 모듈을 번갈아가면서 접근(Access)할 수 있다.

정답 1

- 캐시 메모리(Cache Memory)
  - ✓CPU의 처리 속도와 주기억장치의 접근 속도 차이를 줄이기 위해 사용한다.
  - ✓주기억장치를 접근하는 횟수가 줄어듦으로써 컴퓨터의 처리 속도가 향상된다.

✓매핑 프로세스(Mapping Process)
- ✓직접 매핑: 적중률이 낮아질 수 있다.
- ✓어소시에이티브 매핑(Associative Mapping, 연관 매핑)
- ✓세트-어소시에이티브 매핑(Set-Associative Mapping)

✓쓰기 정책
- ✓Write-Through
- ✓Write-Back, Write-Once

✓캐시의 적중률 = $\frac{\text{적중 횟수}}{\text{총 접근횟수}}$

문제

- 주기억장치의 속도가 CPU의 속도에 비해 현저히 늦다. 명령어의 수행 속도를 CPU의 속도와 유사하도록 하고자 할 때 사용되는 기억장치는?
  ① Cache 기억장치
  ② Virtual 기억장치
  ③ Segment 기억장치
  ④ 복수 모듈 기억장치

정답 1

정보처리기사

제2과목 전자계산기 구조

16 병렬 컴퓨터

문혜영교수

목차

- 가상기억장치(Virtual Memory)
  - ✓기억 용량이 작은 주기억장치를 마치 큰 용량을 가진 것처럼 사용할 수 있어 사용자는 프로그램의 크기에 제한 받지 않고 프로그램 실행이 가능하다.
- 가상기억장치의 특징
  - ✓소프트웨어적인 방법으로 보조기억장치를 주기억장치처럼 사용
  - ✓보조기억장치 중 디스크와 같은 DASA 장치에서 가능
- 가상기억장치의 관리 기법
  - ✓페이징 기법, 세그먼트 기법
- 기억장치의 관리 전략
  - ✓반입(Fetch) 전략, 배치(Placement) 전략, 교체(Replacement) 전략

문제

- 가상(Virtual)기억장치에 대한 설명이 아닌 것은?
  ① 주목적은 컴퓨터의 속도를 향상시키기 위한 방법이다.
  ② 주기억장치를 확장한 것과 같은 효과를 제공한다.
  ③ 실제로는 보조기억장치를 사용하는 방법이다.
  ④ 사용자가 프로그램 크기에 제한받지 않고 실행이 가능하다.

정답 1

- 병렬 컴퓨터의 특징
  - ✓일부 하드웨어 오류가 발생하더라도 전체 시스템은 동작할 수 있다.
  - ✓프로그램 작성이 어려워진다.
  - ✓기억장치를 공유할 수 있다.
- 병렬 컴퓨터의 분류
  - ✓플린(Flynn)의 분류
    - ✓SISD(Single Instruction stream Single Data stream)
    - ✓SIMD(Single Instruction stream Multi Data stream)
    - ✓ MISD(Multi Instruction stream Single Data stream)
    - ✓MIMD(Multi Instruction stream Multi Data stream)
  - ✓팽(Feng)의 분류 : 컴퓨터의 구조를 병렬수행의 정도에 따라 분류
    - ✓WSBS , WPBS, WSBP, WPSP

문제

- Flynn의 분류법 중 여러 개의 처리기에서 수행되는 인스트럭션(Instruction)들은 각기 다르나 전체적으로 하나의 데이터 스트링을 가지는 형태는?
  ① SISD
  ② SIMD
  ③ MISD
  ④ MIMD

정답 3

- 병렬 처리 기법
  - ✓파이프라인 프로세서(Pipeline Processor)
    - ✓명령어를 수행하기 위한 각각의 스테이지를 병렬로 구동시켜 명령어의 처리 흐름이 끊이지 않도록 하여 마치 수도관을 흐르는 물처럼 끊임 없인 명령어가 처리된다고 하여 이러한 이름이 붙었다.4단계 명령어 파이프라인의 수행 순서
    - ✓인출 → 해독 → 오퍼랜드 인출 → 명령 실행

- 파이프라인 해저드(Pipeline Hazard)
  - 파이프라인의 성능을 저해하는 요인
  - 파이프라인 프로세서 의존성으로 발생할 수 있는 문제로 다음 명령어가 다음 클럭 사이클에 실행 될 수 없는 상황
  - 파이프라인 해저드는 다중 파이프라인에서만 발생되며 기본적인 해결책은 지연(stall)이다.
- 파이프라인 해저드 유형
  - 구조 해저드
  - 데이터 해저드
  - 제어 해저드
- √ 파이프라인 해저드의 해결방법
  - 구조 해저드 : 파이프라인을 위한 명령어 설계
  - 데이터 해저드 : 전방전달, 지연(stall)
  - 제어 해저드 : 분기예측, 지연(stall)