



2회 ▶ 05-4, 99-1

인터럽트 비트(interrupt bits) 10010과 마스크 비트(mask bits) 01110을 상호 AND 하였을 때의 출력 비트는?

- ① 11100
- ② 00011
- ③ 11101
- ④ 00010

핵심이론

AND

- 대응되는 비트들 간에 AND 연산을 수행한다.
- 마스크 연산(Masking Operation)이라고도 한다.

유사문제

1회 ▶ 산 03-1

1. A 레지스터 내용이 11010100이고, B 레지스터 내용이 10101100일 때 A와 B의 AND 연산 결과는?

- ① 11010100
- ② 10101100
- ③ 10000100
- ④ 11111100

1회 ▶ 산 12-3

2. 8비트 데이터 11011010에 11110000을 논리적으로 AND(즉, MASK)하면 그 결과는?

- ① 11010000
- ② 00001010
- ③ 00100101
- ④ 00000000

1회 ▶ 산 13-1

3. 2개의 2진수 10110110과 11010111을 AND 연산한 결과의 값은?

- ① 10010110
- ② 01101001
- ③ 11110111
- ④ 10001101

1회 ▶ 산 00-2

4. A의 내용이 1010, B의 내용이 1100이다. masking operation 후의 A 내용은?

- ① 1000
- ② 0010
- ③ 1110
- ④ 0110

[정답] 핵심문제 ④ / 유사문제 1. ③ 2. ① 3. ① 4. ①



1회 ▶ 산 13-1

다음 중 AND 연산을 이용하여 어느 비트(문자)를 지울 것인가를 결정하는 자료가 되는 것은?

- ① CARRY(캐리)
- ② 플립플롭
- ③ 패리티(parity) 비트
- ④ 마스크(mask) 비트

핵심이론

Mask 연산

- 데이터 내 특정 비트들의 값을 0으로 리셋(Reset)시키기 위한 동작이다.
- 대응시키는 0비트를 마스크 비트(Mask Bits)라고 한다.
- AND 회로를 활용한다.

예) A 레지스터에 '11001010'이 저장되어 있을 때 상위 4비트를 0으로 리셋(Reset)하는 연산

→ B 레지스터에 0으로 리셋하고자 하는 4비트를 0으로, 나머지 비트를 1로 만들어 00001111을 저장하여 AND 연산을 수행한다.

A : 1 1 0 0 1 0 1 0 (연산 전)

B : 0 0 0 0 1 1 1 1

A : 0 0 0 0 1 0 1 0 (연산 후)

유사문제

1회 ▶ 산 04-2

1. 비수치 데이터에서 마스크를 이용하여 불필요한 부분을 제거하기 위한 연산은?

- ① OR
- ② XOR
- ③ AND
- ④ NOT

1회 ▶ 산 99-1

2. 어느 자료의 일부분 또는 자료 전체를 지울 때 사용하는 연산은?

- ① Move
- ② AND
- ③ OR
- ④ Complement

1회 ▶ 산 11-2

3. 특정의 비트 또는 특정의 문자를 삭제하기 위해 가장 필요한 연산은?

- ① OR 연산
- ② MOVE 연산
- ③ Complement 연산
- ④ AND 연산

2회 ▶ 06-4, 산 13-2

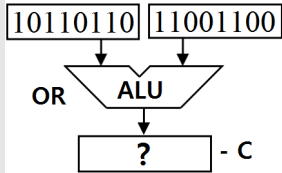
4. 다음 중 AND 마이크로 동작과 유사한 것은?

- ① insert 동작
- ② OR 동작
- ③ packing 동작
- ④ mask 동작



2회 ▶ 05-2, 산 99-3

다음 그림과 같이 A, B 2개의 레지스터에 있는 자료에 대해 ALU가 OR 연산을 행하면 그 결과의 출력 레지스터 C의 내용은? (단, A는 10110110 B는 11001100)



- ① 11101110
- ② 11111110
- ③ 10000000
- ④ 10110110

핵심이론

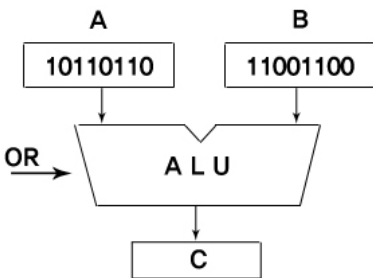
OR

- 대응되는 비트들 간에 OR 연산을 수행한다.

유사문제

1회 ▶ 산 13-3

1. [그림]에서와 같이 A, B 레지스터에 있는 2개의 자료에 대하여 ALU에 의해 OR 연산이 이루어졌을 때 그 결과가 출력되는 C 레지스터의 내용은?



- ① 11101110
- ② 10110110
- ③ 10000000
- ④ 11111110



2회 ▶ 04-1, 00-1

논리마이크로 연산에 있어서 레지스터 A와 B의 값이 단서와 같이 주어졌을 때 selective-set 연산을 수행하면 어떻게 되는가?(단, A는 프로세서 레지스터이고, B는 논리 오퍼랜드, A=1010, B=0011)

- ① 1100
- ② 1011
- ③ 0011
- ④ 1010

핵심이론

Selective-Set 연산(선택적 세트 연산)

- 데이터 내 특정 비트를 1로 세트해 주기 위한 동작이다.
- OR 회로를 활용한다.

예) A 레지스터에 '11001010'이 저장되어 있을 때 상위 4비트를 1로 세트하는 연산

→ B 레지스터에 1로 세트하고자 하는 4비트를 1로, 나머지 비트를 0으로 만들어 11110000을 저장하여 OR 연산을 수행한다.

A : 1 1 0 0 1 0 1 0 (연산 전)

B : 1 1 1 1 0 0 0 0

A : 1 1 1 1 1 0 1 0 (연산 후)

유사문제

2회 ▶ 13-2, 03-1

1. 두 개의 데이터를 혼합하거나 일부에 삽입하는데 사용되는 연산은?

- ① AND 연산
- ② OR 연산
- ③ MOVE 연산
- ④ Complement 연산

1회 ▶ 산 06-1

2. 2개 이상의 자료를 섞을 때(문자삽입 등)의 사용에 편리한 연산자는?

- ① MOVE 연산
- ② 보수 연산
- ③ AND 연산
- ④ OR 연산

1회 ▶ 산 02-1

3. 레지스터에 저장되어 있는 몇 개의 비트를 1로 하기 위해서는 그 장소에 x를 가진 데이터를 y연산을 하면 된다. 이 때 x와 y는?

- ① $x=0, y \rightarrow \text{AND}$
- ② $x=1, y \rightarrow \text{AND}$
- ③ $x=1, y \rightarrow \text{OR}$
- ④ $x=0, y \rightarrow \text{OR}$

1회 ▶ 산 09-1

4. 레지스터에 저장되어 있는 비트들을 모두 1로 만들기 위해 해당 레지스터에 데이터 A를 연산 B로 계산할 때 옳은 것은?

- ① A : ff, B : OR
- ② A : 00, B : AND
- ③ A : 00, B : OR
- ④ A : ff, B : AND

1회 ▶ 산 11-2

5. 논리 마이크로 연산을 수행하기 위하여 다음과 같은 식이 주어졌을 때 옳지 않은 것은?

$$P+Q : R1 \leftarrow R2 \vee R3$$

- ① P가 1이면 R1의 내용은 변할 수도 있다.
- ② P 또는 Q가 1이면 데이터 전송이 일어난다.
- ③ "V"는 논리 마이크로 연산 OR를 나타낸다.
- ④ "+"는 덧셈 마이크로 연산을 나타낸다.



1회 ▶ 05-1

논리 마이크로 동작 중 Exclusive-OR과 같은 동작을 하는 것은?

- ① Selective-set 동작
- ② mask 동작
- ③ compare 동작
- ④ selective-clear 동작

핵심이론

XOR(Selective-Complement)

- 대응되는 비트들 간에 XOR 연산을 수행한다.
- 자료의 특정 비트들을 반전시키는 연산이다.
- 비교(Compare) 동작과 같은 동작을 한다.
- 반전시킬 부분의 비트들을 1과 XOR 연산하면 된다.

예)
$$\begin{array}{r} 1001\ 0101 \\ \text{XOR } 0000\ 1111 \\ \hline 1001\ 1010 \end{array}$$

유사문제

3회 ▶ 산 10-2, 03-1, 01-1

1. 비교(compare) 동작과 같은 동작을 하는 논리 연산은?

- ① 마스크 동작
- ② OR 동작
- ③ XOR 동작
- ④ AND 동작

1회 ▶ 12-2

2. 두 데이터를 비교하는 연산(compare)과 같은 동작을 하는 논리 연산은?

- ① EX-OR 연산
- ② AND 연산
- ③ OR 연산
- ④ NOT 연산

1회 ▶ 산 00-4

3. 이항(Binary) 연산의 하나로 자료의 특정 비트를 반전시키고자 하는 경우에 사용하는 연산은?

- ① AND 연산
- ② OR 연산
- ③ XOR 연산
- ④ NOT(Complement) 연산

2회 ▶ 산 04-1, 01-3

4. 컴퓨터의 연산장치에서 2개의 자료 11011101, 01101101을 Exclusive-OR 연산하였을 때의 결과는?

- ① 01001111
- ② 10110000
- ③ 11111101
- ④ 01001101

1회 ▶ 05-4

5. 다음 진리표와 같은 연산을 하는 gate는?

입 력		출 력
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

- ① OR gate
- ② AND gate
- ③ EXCLUSIVE OR gate
- ④ NAND gate

1회 ▶ 산 11-1

6. 레지스터에 저장되어 있는 몇 개의 비트를 반전하기 위하여 그 장소에 x를 가진 데이터를 y 연산하면 된다. 이 때 x와 y는?

- ① x = 1, y = EX-OR
- ② x = 0, y = OR
- ③ x = 1, y = AND
- ④ x = 0, y = AND



1회 ▶ 산 05-1추

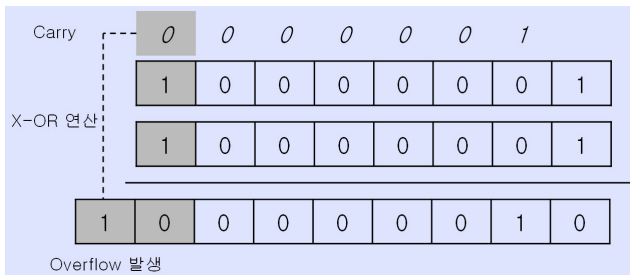
산술 연산에서 overflow가 발생했을 경우 이것을 검출해야 하는데 이 때 사용되는 논리 게이트는?

- ① NOR
- ② OR
- ③ Exclusive-OR
- ④ NAND

핵심이론

X-OR 회로를 사용한 Overflow 판단

- $(-126) - (+126) = (-126) + (-126) = (-252)$



※ Overflow 판단 : 최종 자리올림수(Carry) 두 비트를 비교하여 서로 다른 경우에는 오버플로우(Overflow)로 판단
→ X-OR 회로 사용

유사문제

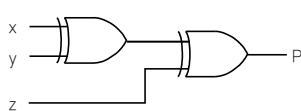
1회 ▶ 산 13-3

1. 다음 중 패리티 비트를 검사하려면 어떤 게이트를 사용하는 것이 가장 좋은가?

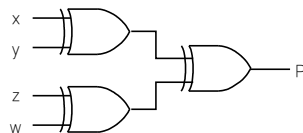
- ① AND
- ② NAND
- ③ NOR
- ④ EX-OR



유수의 조언



▲ 3변수 짝수 패리티 생성기



▲ 4변수 짝수 패리티 생성기

1회 ▶ 08-2

2. 논리 연산 명령을 사용해서 기억영역을 clear 시킬 수 없는 것은?

- ① exclusive OR 연산 한다.
- ② 0(zero)으로 mask 씌운 AND 연산한다.
- ③ 원하는 비트 수만큼 왼쪽으로 rotate 한다.
- ④ 원하는 비트 수만큼 왼쪽으로 논리 shift 한다.



2회 ▶ 14-1, 08-4

연산 명령 자체로 특수한 곱셈과 나눗셈을 수행하거나 혹은 곱셈과 나눗셈에 보조적으로 이용되는 것은?

- ① 산술적 shift
- ② 논리적 shift
- ③ ADD
- ④ rotate

핵심이론

산술 시프트(Arithmetic Shift)

- 부호 비트를 제외한 나머지 비트만 Shift하는 연산이다.
- 연산 명령 자체로 특수한 곱셈과 나눗셈을 수행하거나 혹은 곱셈과 나눗셈에 보조적으로 이용된다.
- 산술 좌측 시프트(Arithmetic Shift-left)
 - 왼쪽으로 n Bit Shift하면 원래 자료에 2^n 을 곱한 값과 같다.
 - 예) 8-bit 시프트 레지스터의 0000 1000(=10진수 8)을 왼쪽으로 3비트만큼 산술 시프트하면 0100 0000(=10진수 64)이 된다.
- 산술 우측 시프트(Arithmetic Shift-right)
 - 오른쪽으로 n Bit Shift하면 원래 자료를 2^n 으로 나눈 값과 같다.
 - 예) 8-bit 시프트 레지스터의 0000 1000(=10진수 8)을 오른쪽으로 3비트만큼 산술 시프트하면 0000 0001(=10진수 1)이 된다.

유사문제

2회 ▶ 산 12-1, 07-1

1. Shift register에 있는 binary number가 여섯(6)번 Shift-left 되었을 때의 값은?(단, Shift register는 충분히 크다고 가정한다.)

- ① number \times 6 ② number - 6
- ③ number \times 64 ④ number \div 64

1회 ▶ 산 00-4

2. 시프트 레지스터(shift register)의 내용을 오른쪽으로 두 번 시프트하면 원래의 data는 어떻게 변화하는가?

- ① 원래 data의 1/2배
- ② 원래 data의 1/4배
- ③ 원래 data의 2배
- ④ 원래 data의 4배

1회 ▶ 산 12-3

3. 8개의 플립플롭으로 된 시프트 레지스터(Shift Register)에 10진수로 64가 기억되어 있을 때 이를 오른쪽으로 3비트만큼 산술 시프트하면 그 값은?

- ① 4 ② 8
- ③ 12 ④ 24

1회 ▶ 산 13-2

4. 시프트 레지스터에서 왼쪽으로 한 번 시프트시키면 원래의 데이터는?

- ① 1/2로 된다.
- ② 1/4로 된다.
- ③ 2배로 된다.
- ④ 4배로 된다.

1회 ▶ 산 09-2

5. 8bit register의 데이터가 00101001이다. 이 데이터를 4배 증가시키려고 할 때 취하는 연산 명령은?

- ① Shift Left 4회
- ② Shift Left 2회
- ③ Shift Right 4회
- ④ Shift Right 2회

1회 ▶ 산 14-2

6. 산술적 시프트(shift)에 대한 설명으로 옳은 것은?

- ① 산술적 시프트는 오른쪽 방향으로 한 비트씩 이동시킨다.
- ② 산술적 시프트는 Rotate와 동일하다.
- ③ 산술적 시프트는 논리적 시프트와 동일하다.
- ④ 산술적 시프트는 곱셈과 나눗셈의 보조역할에 이용된다.

[정답] 핵심문제 ① / 유사문제 1. ③ 2. ② 3. ② 4. ③ 5. ② 6. ④



108 Shift 연산 고급문제 1

1회 ▶ 00-1

다음은 이동(shift)의 경우이다. 8-비트로 구성된 레지스터 7번의 내용이 11011001일 때 SRA 7,3을 실행하고 난 후의 레지스터 7의 내용은? (단, SRA 7, 3은 레지스터 7 번을 우측으로 산술 이동(Arithmetic Shift to the right) 3회 수행함을 뜻한다.)

- ① 1111101 ② 00011011 ③ 11111011 ④ 01111011

핵심이론

산술 시프트 연산시 패딩 비트

- 자리를 이동 후 생긴 빈자리에 입력되는 비트인 패딩 비트(Padding Bit)는 다음과 같다.

구분	수치 표현법	음수	양수
산술 좌측 시프트	부호화 절대치	• 패딩 비트 : 0 예) 1 0 1 1 0 0 1 1 (-51) ↓ ↓ ↓ ↓ ↓ ↓ ↓ 1 1 1 0 0 1 1 0 (-102)	
	1의 보수	• 패딩 비트 : 1 예) 1 1 0 0 1 1 0 0 (-51) ↓ ↓ ↓ ↓ ↓ ↓ ↓ 1 0 0 1 1 0 0 1 (-102)	
	2의 보수	• 패딩 비트 : 0 예) 1 1 0 0 1 1 0 1 (-51) ↓ ↓ ↓ ↓ ↓ ↓ ↓ 1 0 0 1 1 0 1 0 (-102)	
산술 우측 시프트	부호화 절대치	• 패딩 비트 : 0 예) 1 0 1 1 0 0 1 1 (-51) ↓ ↓ ↓ ↓ ↓ ↓ ↓ 1 0 1 1 0 0 1 1 (-25)	• 패딩 비트 : 0
	1의 보수	• 패딩 비트 : 1 예) 1 1 0 0 1 1 0 0 (-51) ↓ ↓ ↓ ↓ ↓ ↓ ↓ 1 1 1 0 0 1 1 0 (-25)	
	2의 보수	• 패딩 비트 : 1 예) 1 1 0 0 1 1 0 1 (-51) ↓ ↓ ↓ ↓ ↓ ↓ ↓ 1 1 1 0 0 1 1 0 (-26)	

유사문제

1회 ▶ 08-2

1. 10110101이라는 이진 자료가 2's complement 방식으로 표현되어 있다. 이를 우측으로 3비트만큼 산술적 이동(Arithmetic shift) 하였을 때의 결과는?

- ① 11110110 ② 11010110 ③ 10000110 ④ 00010110

1회 ▶ 06-4

2. 74라는 수가 8비트의 레지스터에 기록되어 있다. 그중 가장 좌측비트는 부호를 나타내고, 나머지 7비트는 절대값을 나타낸다. 이 레지스터를 우측으로 한 비트 산술적 이동(arithmetic shift)을 한 결과는?

- ① 35 ② 36 ③ 37 ④ 38

2회 ▶ 산 14-2, 08-2

3. 8비트로 표현되는 부호와 절대치의 방식에서 -50을 1비트 우측으로 시프트(shift) 했을 때 옳은 것은?

- ① 10011000 ② 11011000 ③ 11011001 ④ 10011001

1회 ▶ 00-2

4. 8비트 부호와 2의 보수로 나타낸 수 -77을 오른쪽으로 두 비트 산술 시프트 수행한 결과는?

- ① OVERFLOW ② -20 ③ -19.5 ④ +19

1회 ▶ 산 08-1

5. $(-24)_{10}$ 을 부호화 절대치 방법에서의 1비트 좌측 시프트할 경우 옳바른 것은? (단, 표현은 8 비트로 한다.)

- ① 11011110 ② 01011110 ③ 10110000 ④ 01010111

[정답] 핵심문제 ③ / 유사문제 1. ① 2. ③ 3. ④ 4. ② 5. ③



1회 ▶ 09-2

다음은 산술 시프트(Arithmetic shift)에 관한 설명이다. 옳지 않은 것은?

- ① 레지스터의 값을 우측으로 쉬프트할 때 새로운 입력 비트는 1의 보수, 2의 보수 모두 0이 입력된다.
- ② 레지스터의 값을 좌측으로 쉬프트할 때 새로운 입력 비트는 1의 보수 경우 부호 비트가 입력되고, 2의 보수의 경우 무조건 0이 입력된다.
- ③ 레지스터의 값을 n비트 우측으로 시프트하면 2^n 으로 나누는 효과를 갖는다.
- ④ 1의 보수 표현 방식으로 레지스터에 저장된 값이 최상위 비트인 부호비트와 최하위 비트인 LSB가 서로 다를 때 우측 시프트를 수행하면 잘림 에러(Truncation Error)가 발생한다.

유 사 문 제

2회 ▶ 13-3, 10-2

1. shift 명령을 수행한 후 빈 공간에 채워지는 내용이 다른 것은?

- ① 왼쪽으로 논리 shift한 결과
- ② 오른쪽으로 논리 shift한 결과
- ③ 2의 보수법으로 왼쪽으로 산술 shift한 결과
- ④ 오른쪽으로 산술 shift한 결과

1회 ▶ 산 14-3

2. 다음 중 소수점 이하를 잃어버리는 절단(truncation) 현상인 것은?

- ① $(10001111)_2$ 을 좌측으로 시프트
- ② $(10001111)_2$ 을 우측으로 시프트
- ③ $(11110000)_2$ 을 좌측으로 시프트
- ④ $(11110000)_2$ 을 우측으로 시프트

2회 ▶ 산 11-2, 00-3

3. 1의 보수에 의해 표현된 수를 좌측으로 1bit 산술shift 하는 경우 입력되는 비트는?

- ① 1
- ② 0
- ③ sign bit
- ④ LSB(Least Significant Bit)

1회 ▶ 산 08-2

4. 다음 설명 중 옳은 것은?

- ① 10101100의 오른쪽 4비트만 0(zero)으로 하려면 마스크 내용을 00001111로 OR 연산을 한다.
- ② 0(zero)은 부동소수점으로 표현할 수 없다.
- ③ 논리적 시프트(왼쪽이나 오른쪽 모두)는 시프트된 공간에 항상 0(zero)이 들어온다.
- ④ 산술 시프트(왼쪽이나 오른쪽 모두)는 시프트된 공간에 항상 0(zero)이 들어온다.



1회 ▶ 산 99-1

문자의 위치 변환에 이용하는데 가장 효율적인 동작은?

- ① 로테이트(rotate) 동작
- ② 산술 시프트(shift)
- ③ 논리 시프트
- ④ 좌측 및 우측 시프트

핵심이론

Rotate 연산

- 논리 Shift에서 밀려 나가는 비트의 값을 반대편 값으로 입력하는 연산이다.
- 문자 위치 변환에 사용된다.
- 순환 시프트(Circular Shift)라고도 한다.

유사문제

1회 ▶ 산 10-1

1. n bit의 레지스터 $A(A_{n-1}A_{n-2}\cdots A_1A_0)$ 와 $B(B_{n-1}B_{n-2}\cdots B_1B_0)$ 에 대해 다음의 마이크로 오퍼레이션(micro-operation)을 n 번 수행하였다. 이 때, shr은 오른쪽 시프트(right shift), cir은 오른쪽 회전 시프트(rotate right)이다. 어떤 기능을 수행한 것인가?

cir B, shr A, $A_{n-1} \leftarrow B_0$

- ① A의 내용을 B로 직렬 전송(serial transfer)
- ② B의 내용을 A로 직렬 전송(serial transfer)
- ③ A와 B의 내용을 교환
- ④ B의 내용을 2로 나눈 나머지를 A에 저장