



(컴퓨터구조) 과제 #1

201312845 김일식 | 201511183 김영서 | 201511237 허승리

제출일 : 2016 년 10 월 27 일

(컴퓨터 구조) 과제 #1

1) 문제 정의

부호를 갖는 임의의 두 정수와 실수에 대한 사칙연산을 수행하는 ALU 시뮬레이터를 구현한다.

덧셈과 뺄셈은 병렬가감산기의 동작원리에, 곱셈과 나눗셈은 Booth 알고리즘 및 나눗셈 알고리즘에 따라서 구현한다.

실수에 대한 연산은 32 비트의 IEEE754 표준에 따른 부동 소수점 형태로 표현하고, 4 가지의 예외 규칙을 적용한다.

2) Idea 및 알고리즘

// ALU.h : 헤더 파일

// main_int.cpp : main 및 정수 연산 함수 구현

// float.cpp : 실수 연산 함수 구현

■정수의 덧셈

사용자에게 입력 받은 두 10 진수 정수를 이진수로 표현하여 크기가 32 인 int 형 배열에 저장한다. 두 이진 배열의 덧셈을 위해 adder_subtractor()함수를 구현한다.

Adder_subtractor() 함수에 덧셈을 수행 할 두 배열과 sign 값이 인자로 전달된다. 덧셈 연산은 전가산기의 알고리즘을 따른다. 초기 carry 값은 sign 의 값(덧셈 = 0)이 저장된다. 전가산기에서 sum 은 두 입력값과 carry 값의 XOR 연산으로 결정되고, carry 값은 ((입력값 1 과 입력값 2 의 XOR 연산)*이전 carry 값 + 입력값 1*입력값)이 저장된다. 이를 코드로 표현하면 아래와 같다.

```
sum[i] = a[i] ^ b[i] ^ c;  
c = (a[i] ^ b[i])*c + a[i] * b[i];
```

이 연산을 i 를 증가시키며 32 번 반복하여 결과 배열 sum 을 얻는다. i==30 일 때의 carry 값을 저장해두어 마지막 carry 값과 XOR 연산을 하여 오버플로우를 나타내는 v 플래그 값을 구하고, 결과 배열의 값을 검사하여 모두 0 일 경우 z 플래그를 1 로 set 한다. s 플래그는 결과 배열의 마지막 방의 값이 된다.

■정수의 뺄셈

뺄셈의 경우에도 `adder_subtractor()` 함수를 사용한다. `Adder_subtracrer()` 배열에 피감수 배열과 감수 배열이 인자로 넘어가고, `sign` 인자에는 1 이 넘어간다. 감(가)수 배열에 `sign` 과 감(가)수배열의 비트의 XOR 연산값을 저장하는 알고리즘을 통해 비트가 반전된다(1 의 보수). 덧셈의 경우에는 `sign` 이 0 이기 때문에 비트가 반전되지 않는다. 또, 초기 `carry` 값에 1 이 저장된 후에 `sum` 을 구하는 연산을 진행하므로 감수의 2 의 보수 표현도 만족한다. 이후의 알고리즘은 덧셈과 동일하다.

■정수의 곱셈

Booth 알고리즘을 그대로 적용한다. `M`, `A`, `Q` 레지스터에 피승수, 승수 등을 초기값으로 세팅한다. `Q-1`, `Q0` 을 검사하여 순서대로 0, 1 이면 `A` 에서 `M` 을 뺀다. 순서대로 1, 0 이면 더한다. `A0` 과 `Q31` 을 넘어가는 비트로 대입한다. `Q0` 을 `Q-1` 로 저장한다. 그 후에 오른쪽으로 쉬프트한다. 그리고 `Q31` 을 쉬프트 전 값으로 세팅한다. 이 것을 비트 수 만큼 반복한다.

■정수의 나눗셈

Booth 나눗셈 알고리즘을 그대로 적용한다. `M`, `A`, `Q` 레지스터에 피제수, 제수 등을 초기값으로 세팅한다. 일단 `A` 의 비트 패턴을 `temp` 에 저장한다. 나중에 복원용으로 사용한다. 피제수가 음수이면 `A` 를 1 로 초기화한다. 제수와 피제수의 부호가 다르면 그 내용을 `check` 에 저장한다. 그 후 `A`, `Q` 를 왼쪽으로 쉬프트한다. 넘어가는 비트를 변수들에 저장한다. `temp` 에 `A` 의 비트 패턴을 저장한다. `A` 와 `M` 의 부호가 같으면 두 레지스터를 더하고, 부호가 다르면 두 레지스터를 뺀다. 연산 전의 `A` 의 부호와 연산 후의 `A` 의 부호가 같으면 성공, 다르면 실패이다. 같다면 `Q0` 을 1 로 세트한다. `A` 가 0 이어도 `Q0` 을 1 로 세트한다. 아니면 `temp` 를 이용해 `A` 를 복원한다. 이것을 비트 수 만큼 반복한다. 처음 제수와 피제수의 비트가 다르다면 결과에 보수를 취해준다.

■IEEE754 표준화

사람이 실수를 IEEE754 로 표준화하는 논리를 최대한 적용한다. 실수를 입력 받으므로 정수부에 대한 2 진수 값, 소수부에 대한 2 진수 값을 담은 각각의 배열을 선언하고 0 으로 초기화한다. 이때 IEEE754 표준에 의해 최대 들어올 수 있는 정수는 약 130 자리를, 소수부는 정수부에서 shift 되고 남은 부분에 대해 값을 대입하므로 최대 23 자리를 할당한다. IEEE754 표준에 맞게 32 비트를 1 비트의 부호, 8 비트의 지수, 23 비트의 가수 필드로 구분하여 각각의 크기에 맞게 배열을 선언하고 배열정보는 0 으로 초기화한다.

*먼저 입력 받은 실수의 양/음을 판단한다. 음수일 경우 부호 정보를 담은 배열의 값을 1 로 변경시켜준다.

*부호정보를 처리하고 난 후 실수를 정수부와 소수부로 나누어 각각

2 진수화한다.(정수부는 2 를 나누며, 소수부는 2 를 곱하면서. 각각 dec_to_bin1(), dec_to_bin2() 함수 사용)

*앞서 선언했던 정수부 배열과 소수부 배열을 가수배열에 대입해야 하는데 이때 입력한 실수의 절댓값이 1 이상일 경우 정수부 배열에서 가장 먼저 1 이 나오는 인덱스의 다음부터 차례차례 가수배열에 대입하고 대입하고 남은 부분에 대해 소수부 배열의 앞자리부터 대입한다. 반면 입력한 실수의 절댓값이 1 미만일 경우 오로지 소수부 배열만을 가수배열에 대입한다. 이때 가장 먼저 1 이 나오는 인덱스 다음부터 대입 한다. 두가지 경우 모두 대입을 하다가 가수배열의 인덱스 23 을 넘어서는 값에 대해선 버림을 실행한다.

*가수배열에 대입하는 과정에서 가장 먼저 발견된 1 에 대한 정수부 혹은 소수부 배열의 인덱스는 다음과 같이 처리한다.

해당 인덱스를 소수점에 대한 기준으로 삼고 정수부에서 첫번째 1 을 제외하고 남은 배열 정보의 수만큼 지수비트에 전달하고, 소수부에서 첫번째 1 이 나오기까지의 인덱스를 지수비트에 전달한다.

ex) 1101.101₍₂₎ → 1.101101₍₂₎, (+)3 // 0.0011₍₂₎ → 1.1₍₂₎, (-)3

*전달받은 수(정수부에서 넘어온 경우 양수, 소수부에서 넘어온 경우 음수 처리)만큼 2 의 지수승을 해준다. 이때 IEEE754 표준은 127 바이어스된 값이므로 지수비트에 들어갈 값은 전달받은 수와 127 의 합에 대한 2 진수 값이 들어간다.

*바이어스된 지수배열의 10 진수 값이 128, 0 인 경우에 대해 가수 값이 0 이거나 0 이 아닌 경우는 각각 NaN, overflow, zero, underflow 예외처리를 해준다.(입력 값에서부터 처리)

■실수의 덧셈

2 개의 IEEE754 표준으로 표현된 32 비트 배열의 지수부는 크기가 8 인 지수부 배열에, 가수부는 크기가 25 인 가수부 배열에 각각 저장한다. 이 때, 병렬가감산함수나 2 의보수함수는 인덱스가 0 인 방이 오른쪽에 위치한다고 가정하지만, 실수 배열은 인덱스가 0 인 방이 왼쪽에 위치하므로 배열에 역순으로 저장한다. 가수부 배열의 크기가 25 인 이유는 IEEE754 표준에서 생략된 1 과 가수의 덧셈에서 생기는 carry 값을 저장하기 위해 23 에 2 을 더해주었기 때문이다.

피가수가 0 인 경우는 결과값이 가수의 값과 같고, 반대인 경우는 피가수의 값과 같다. 둘 다 0 인 경우는 결과값도 0 이 된다. 이 세 경우에는 연산을 종료한다.

나머지 경우는 이어서 연산을 진행한다. 피가수를 a, 가수를 b, 결과를 c 로 표현하자. 먼저 지수를 조정해주기 위해 두 배열의 지수의 차를 구하여 count 에 저장한다. Count 가 양수이면 b 를 조정해준다. 결과 지수는 a 의 지수로 맞추고 b 의 가수부는 count 만큼 우측으로 shift 연산을 진행한다. 결과값의 부호는 a 의 부호가 된다. Count 가 음수인 경우는 a 를 조정해준다. 결과 지수는 b 의 지수로, a 의 가수부는 -count 만큼 우측 shift 를 해준다. 결과값의 부호는 b 의 부호가 된다. 두 수의 지수가 같은 경우는 결과 지수를 a 의지수로

맞춰주고(b 도 가능) 결과값의 부호는 a 와 b 의 가수를 비교하여 절댓값이 큰 수의 부호를 따라간다.

입력 배열이 음수인 경우에는 가수부에 2 의 보수를 취해준 후에 가수의 덧셈연산을 진행한다. 덧셈의 결과가 0 인 경우는 결과값에 0 을 출력한다. 0 이 아닌 경우, 가수 배열의 맨 마지막 방을 검사한다. Carry 가 발생하는 경우는 2 의 보수를 취해준다. 23~0 번째 방을 검사하여 첫번째 1 이 나타나는 인덱스를 count 에 저장하고, 지수에서 count-1 을 빼준다. 최종 결과 배열의 가수부에는 c 의 가수부배열 23-count 번째 방의 값부터 저장해주고, 지수부에는 c 의 지수부배열을 저장해준다.

■실수의 뺄셈

덧셈과 기본 로직은 동일하다. 가수의 덧셈만 뺄셈으로 바꾸어준다.

■실수의 곱셈

실수의 곱셈은 지수부와 가수부를 다르게 진행한다. 먼저 두 수의 지수부를 각각 8 비트 배열로 저장한다. 그리고 이 각각의 배열에서 127, 즉 bias 값을 뺀다. 그 후 병렬가감산기 함수로 두 배열의 비트를 더한다. 가수부는 가수부의 저장 방식과 원래 이진수의 저장 방식이 조금 다르므로 그것을 원래 이진수 저장 로직으로 넣는다. 이 것을 두 개의 23 비트 배열로 비트 패턴을 저장한다. 그 후에 두 개의 가수부 배열을 23 비트로 곱셈 계산을 실행한다. 이 계산은 정수 곱셈 계산 알고리즘을 그대로 사용한다.

■실수의 나눗셈

실수의 나눗셈은 실수의 곱셈과 같이 지수부와 가수부를 다르게 진행한다. 먼저 두 수의 지수부를 각각 8 비트 배열로 저장한다. 그리고 이 각각의 배열에서 127, 즉 bias 값을 뺀다. 그 후 병렬가감산기 함수로 두 배열을 뺀다. 가수부는 가수부의 저장 방식과 원래 이진수의 저장 방식이 조금 다르므로 그것을 원래 이진수 저장 로직으로 넣는다. 그 후에 두 개의 가수부 배열을 23 비트로 나눗셈 계산을 실행한다. 이 계산은 정수 나눗셈 계산 알고리즘을 그대로 사용한다.

3) 수행결과

결과 1 - 입력 : 0, 0

덧셈 0, 뺄셈 0, 곱셈 0 을 출력한다. 나눗셈은 0 으로 나눌 수 없다는 메시지와 함께 실행하지 않는다.

```

C:\Windows\system32\cmd.exe
연산의 종류를 선택하세요<1: 정수, 2: 실수> : 1
↳ 2개의 정수 입력 : 0 0
=====덧셈=====
1. 이진수로 표현
피가수 : 0000 0000 0000 0000 0000 0000 0000 0000
가수 : 0000 0000 0000 0000 0000 0000 0000 0000
2. 덧셈 연산
0000 0000 0000 0000 0000 0000 0000 0000
+ 0000 0000 0000 0000 0000 0000 0000 0000
-----
0000 0000 0000 0000 0000 0000 0000 0000
C : 0, S : 0, Z : 1, U : 0
3. 10진수 결과값 : 0
=====뺄셈=====
1. 이진수로 표현
피감수 : 0000 0000 0000 0000 0000 0000 0000 0000
감수 : 0000 0000 0000 0000 0000 0000 0000 0000
2. 뺄셈 연산
0000 0000 0000 0000 0000 0000 0000 0000
- 0000 0000 0000 0000 0000 0000 0000 0000
-----
0000 0000 0000 0000 0000 0000 0000 0000
C : 1, S : 0, Z : 1, U : 0
3. 10진수 결과값 : 0
=====곱셈=====
#초기값
          a                                Q                                Q-1                                M
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#1
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#2
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#3
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#4
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#5
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#6
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#7
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#8
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#9
SHIFT

```

```
C:\Windows\system32\cmd.exe

0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#16 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#17 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#18 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#19 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#20 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#21 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#22 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#23 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#24 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#25 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#26 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#27 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#28 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#29 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#30 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#31 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000
#32 SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 0000

이진수 결과 값 : 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
10진수 결과 값 : 0

-----나눗셈-----
0으로 나눌 수 없습니다

계속하려면 아무키나 입력하세요(종료: q)
_
```


결과 2 - 입력 : 10, -100

덧셈 -90, 뺄셈 110, 곱셈 -1000, 나눗셈 몫 10 을 각각 출력한다.

```
C:\Windows\system32\cmd.exe
연산의 종류를 선택하세요<1: 정수, 2: 실수> : 1
  2개의 정수 입력 : 10 -100
-----덧셈-----
1. 이진수로 표현
피가수 : 0000 0000 0000 0000 0000 0000 0000 1010
가수   : 1111 1111 1111 1111 1111 1111 1001 1100
2. 덧셈 연산
0000 0000 0000 0000 0000 0000 0000 1010
+ 1111 1111 1111 1111 1111 1111 1001 1100
-----
1111 1111 1111 1111 1111 1111 1010 0110
C : 0, S : 1, Z : 0, V : 0
3. 10진수 결과값 : -90
-----뺄셈-----
1. 이진수로 표현
피감수 : 0000 0000 0000 0000 0000 0000 0000 1010
감수   : 1111 1111 1111 1111 1111 1111 1001 1100
2. 뺄셈 연산
0000 0000 0000 0000 0000 0000 0000 1010
- 1111 1111 1111 1111 1111 1111 1001 1100
-----
0000 0000 0000 0000 0000 0000 0110 1110
C : 0, S : 0, Z : 0, V : 0
3. 10진수 결과값 : 110
-----곱셈-----
#초기값
      A                      Q                      Q-1                      M
0000 0000 0000 0000 0000 0000 0000 0000 | 1111 1111 1111 1111 1111 1001 1100 | 0 | 0000 0000 0000 0000 0000 0000 1010
#1
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0111 1111 1111 1111 1111 1100 1110 | 0 | 0000 0000 0000 0000 0000 0000 1010
#2
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0011 1111 1111 1111 1111 1110 0111 | 0 | 0000 0000 0000 0000 0000 0000 1010
#3
A ← A-M
1111 1111 1111 1111 1111 1111 1111 0110 |
SHIFT
1111 1111 1111 1111 1111 1111 1011 | 0001 1111 1111 1111 1111 1111 0011 | 1 | 0000 0000 0000 0000 0000 0000 1010
#4
SHIFT
1111 1111 1111 1111 1111 1111 1101 | 1000 1111 1111 1111 1111 1111 1001 | 1 | 0000 0000 0000 0000 0000 0000 1010
#5
SHIFT
1111 1111 1111 1111 1111 1111 1110 | 1100 0111 1111 1111 1111 1111 1100 | 1 | 0000 0000 0000 0000 0000 0000 1010
#6
A ← A+M
0000 0000 0000 0000 0000 0000 0000 1000 |
SHIFT
0000 0000 0000 0000 0000 0000 0100 | 0110 0011 1111 1111 1111 1111 1110 | 0 | 0000 0000 0000 0000 0000 0000 1010
#7
SHIFT
```

```
C:\Windows\system32\cmd.exe

1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 1111 1100 0001 1000 1111 1111 | 1 | 0000 0000 0000 0000 0000 0000 0000 1010
#25
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 1111 1110 0000 1100 0111 1111 | 1 | 0000 0000 0000 0000 0000 0000 0000 1010
#26
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 1111 1111 0000 0110 0011 1111 | 1 | 0000 0000 0000 0000 0000 0000 0000 1010
#27
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 1111 1111 1000 0011 0001 1111 | 1 | 0000 0000 0000 0000 0000 0000 0000 1010
#28
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 1111 1111 1100 0001 1000 1111 | 1 | 0000 0000 0000 0000 0000 0000 0000 1010
#29
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 1111 1111 1110 0000 1100 0111 | 1 | 0000 0000 0000 0000 0000 0000 0000 1010
#30
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 1111 1111 1111 0000 0110 0011 | 1 | 0000 0000 0000 0000 0000 0000 0000 1010
#31
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 1111 1111 1111 1000 0011 0001 | 1 | 0000 0000 0000 0000 0000 0000 0000 1010
#32
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 1111 1111 1111 1100 0001 1000 | 1 | 0000 0000 0000 0000 0000 0000 0000 1010

이진수 결과 값 : 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1100 0001 1000
10진수 결과 값 : -1000

=====나눗셈=====
#초기값
A Q M
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 1010 | 1111 1111 1111 1111 1111 1111 1001 1100
#1
SHIFT
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0001 0100 |
A ← A+M
1111 1111 1111 1111 1111 1111 1001 1100 |
복원
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0001 0100 | 1111 1111 1111 1111 1111 1111 1001 1100
#2
SHIFT
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0010 1000 |
A ← A+M
1111 1111 1111 1111 1111 1111 1001 1100 |
복원
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0010 1000 | 1111 1111 1111 1111 1111 1111 1001 1100
#3
SHIFT
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0101 0000 |
A ← A+M
1111 1111 1111 1111 1111 1111 1001 1100 |
복원
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0101 0000 | 1111 1111 1111 1111 1111 1111 1001 1100
#4
SHIFT
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 1010 0000 |
A ← A+M
1111 1111 1111 1111 1111 1111 1001 1100 |
복원
```

```
C:\Windows\system32\cmd.exe

#26
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0010 1000 0000 0000 0000 0000 0000 0000 |
A←A+M
1111 1111 1111 1111 1111 1111 1001 1100 |
복원
0000 0000 0000 0000 0000 0000 0000 0000 | 0010 1000 0000 0000 0000 0000 0000 0000 | 1111 1111 1111 1111 1111 1001 1100
#27
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0101 0000 0000 0000 0000 0000 0000 0000 |
A←A+M
1111 1111 1111 1111 1111 1111 1001 1100 |
복원
0000 0000 0000 0000 0000 0000 0000 0000 | 0101 0000 0000 0000 0000 0000 0000 0000 | 1111 1111 1111 1111 1111 1001 1100
#28
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 1010 0000 0000 0000 0000 0000 0000 0000 |
A←A+M
1111 1111 1111 1111 1111 1111 1001 1100 |
복원
0000 0000 0000 0000 0000 0000 0000 0000 | 1010 0000 0000 0000 0000 0000 0000 0000 | 1111 1111 1111 1111 1111 1001 1100
#29
SHIFT
0000 0000 0000 0000 0000 0000 0000 0001 | 0100 0000 0000 0000 0000 0000 0000 0000 |
A←A+M
1111 1111 1111 1111 1111 1111 1001 1101 |
복원
0000 0000 0000 0000 0000 0000 0000 0001 | 0100 0000 0000 0000 0000 0000 0000 0000 | 1111 1111 1111 1111 1111 1001 1100
#30
SHIFT
0000 0000 0000 0000 0000 0000 0000 0010 | 1000 0000 0000 0000 0000 0000 0000 0000 |
A←A+M
1111 1111 1111 1111 1111 1111 1001 1110 |
복원
0000 0000 0000 0000 0000 0000 0000 0010 | 1000 0000 0000 0000 0000 0000 0000 0000 | 1111 1111 1111 1111 1111 1001 1100
#31
SHIFT
0000 0000 0000 0000 0000 0000 0000 0101 | 0000 0000 0000 0000 0000 0000 0000 0000 |
A←A+M
1111 1111 1111 1111 1111 1111 1010 0001 |
복원
0000 0000 0000 0000 0000 0000 0000 0101 | 0000 0000 0000 0000 0000 0000 0000 0000 | 1111 1111 1111 1111 1111 1001 1100
#32
SHIFT
0000 0000 0000 0000 0000 0000 0000 1010 | 0000 0000 0000 0000 0000 0000 0000 0000 |
A←A+M
1111 1111 1111 1111 1111 1111 1010 0110 |
복원
0000 0000 0000 0000 0000 0000 0000 1010 | 0000 0000 0000 0000 0000 0000 0000 0000 | 1111 1111 1111 1111 1111 1001 1100

이진수 몫 : 0000 0000 0000 0000 0000 0000 0000 0000
10진수 몫 : 0
이진수 나머지 : 0000 0000 0000 0000 0000 0000 0000 1010
10진수 나머지 : 10

계속하려면 아무키나 입력하세요<종료: q>
```

결과 3 - 입력 : 13 6

결과로 덧셈 19, 뺄셈 7, 곱셈 78, 나눗셈 몫 2 를 각각 출력한다.

```
C:\Windows\system32\cmd.exe
연산의 종류를 선택하세요(1: 정수, 2: 실수) : 1
2개의 정수 입력 : 13 6
=====덧셈=====
1. 이진수로 표현
피가수 : 0000 0000 0000 0000 0000 0000 0000 1101
가수 : 0000 0000 0000 0000 0000 0000 0000 0110
2. 덧셈 연산
0000 0000 0000 0000 0000 0000 0000 1101
+ 0000 0000 0000 0000 0000 0000 0000 0110
-----
0000 0000 0000 0000 0000 0000 0001 0011
C : 0, S : 0, Z : 0, U : 0
3. 10진수 결과값 : 19
=====뺄셈=====
1. 이진수로 표현
피가수 : 0000 0000 0000 0000 0000 0000 0000 1101
감수 : 0000 0000 0000 0000 0000 0000 0000 0110
2. 뺄셈 연산
0000 0000 0000 0000 0000 0000 0000 1101
- 0000 0000 0000 0000 0000 0000 0000 0110
-----
0000 0000 0000 0000 0000 0000 0000 0111
C : 1, S : 0, Z : 0, U : 0
3. 10진수 결과값 : 7
=====곱셈=====
#초기값
          A          Q          Q-1          M
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0110 | 0 | 0000 0000 0000 0000 0000 0000 1101
#1
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0011 | 0 | 0000 0000 0000 0000 0000 0000 1101
#2
A←A-M
1111 1111 1111 1111 1111 1111 1111 0011 |
SHIFT
1111 1111 1111 1111 1111 1111 1111 1001 | 1000 0000 0000 0000 0000 0000 0001 | 1 | 0000 0000 0000 0000 0000 0000 1101
#3
SHIFT
1111 1111 1111 1111 1111 1111 1111 1100 | 1100 0000 0000 0000 0000 0000 0000 | 1 | 0000 0000 0000 0000 0000 0000 1101
#4
A←A+M
0000 0000 0000 0000 0000 0000 0000 1001 |
```

```
C:\Windows\system32\cmd.exe

#29
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0010 0111 0000 | 0 | 0000 0000 0000 0000 0000 0000 0000 1101
#30
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0001 0011 1000 | 0 | 0000 0000 0000 0000 0000 0000 0000 1101
#31
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 1001 1100 | 0 | 0000 0000 0000 0000 0000 0000 0000 1101
#32
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0100 1110 | 0 | 0000 0000 0000 0000 0000 0000 0000 1101

이진수 결과 값 : 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 1110
10진수 결과 값 : 78

=====나눗셈=====
#초기값
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 1101 | 0000 0000 0000 0000 0000 0000 0000 0110
#1
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0001 1010 |
A←A-M
1111 1111 1111 1111 1111 1111 1111 1010 |
복원
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0001 1010 | 0000 0000 0000 0000 0000 0000 0000 0110
#2
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0011 0100 |
A←A-M
1111 1111 1111 1111 1111 1111 1111 1010 |
복원
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0011 0100 | 0000 0000 0000 0000 0000 0000 0000 0110
#3
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0011 1000 |
A←A-M
1111 1111 1111 1111 1111 1111 1111 1010 |
복원
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0011 1000 | 0000 0000 0000 0000 0000 0000 0000 0110
#4
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 1101 0000 |
A←A-M
1111 1111 1111 1111 1111 1111 1111 1010 |
복원
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 1101 0000 | 0000 0000 0000 0000 0000 0000 0000 0110
#5
SHIFT
```

```
C:\Windows\system32\cmd.exe

0000 0000 0000 0000 0000 0000 0000 0000 | 0011 0100 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0110
#27
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0110 1000 0000 0000 0000 0000 0000 0000 |
A←A-M
1111 1111 1111 1111 1111 1111 1111 1010 |
복원
0000 0000 0000 0000 0000 0000 0000 0000 | 0110 1000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0110
#28
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 1101 0000 0000 0000 0000 0000 0000 0000 |
A←A-M
1111 1111 1111 1111 1111 1111 1111 1010 |
복원
0000 0000 0000 0000 0000 0000 0000 0000 | 1101 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0110
#29
SHIFT
0000 0000 0000 0000 0000 0000 0000 0001 | 1010 0000 0000 0000 0000 0000 0000 0000 |
A←A-M
1111 1111 1111 1111 1111 1111 1111 1011 |
복원
0000 0000 0000 0000 0000 0000 0000 0001 | 1010 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0110
#30
SHIFT
0000 0000 0000 0000 0000 0000 0000 0011 | 0100 0000 0000 0000 0000 0000 0000 0000 |
A←A-M
1111 1111 1111 1111 1111 1111 1111 1101 |
복원
0000 0000 0000 0000 0000 0000 0000 0011 | 0100 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 0000 0000 0110
#31
SHIFT
0000 0000 0000 0000 0000 0000 0000 0110 | 1000 0000 0000 0000 0000 0000 0000 0000 |
A←A-M
0000 0000 0000 0000 0000 0000 0000 0000 |
00 = 1
0000 0000 0000 0000 0000 0000 0000 0000 | 1000 0000 0000 0000 0000 0000 0000 0001 | 0000 0000 0000 0000 0000 0000 0000 0110
#32
SHIFT
0000 0000 0000 0000 0000 0000 0000 0001 | 0000 0000 0000 0000 0000 0000 0000 0010 |
A←A-M
1111 1111 1111 1111 1111 1111 1111 1011 |
복원
0000 0000 0000 0000 0000 0000 0000 0001 | 0000 0000 0000 0000 0000 0000 0000 0010 | 0000 0000 0000 0000 0000 0000 0000 0110

이진수 몫 : 0000 0000 0000 0000 0000 0000 0000 0010
10진수 몫 : 2
이진수 나머지 : 0000 0000 0000 0000 0000 0000 0000 0001
10진수 나머지 : 1
```

결과 4 - 입력 : 8000123 12260

결과값으로 덧셈 8012383, 뺄셈 7987863, 곱셈 오버플로우, 나눗셈 몫 652 를 각각 출력한다.

```
C:\Windows\system32\cmd.exe
연산의 종류를 선택하세요<1: 정수, 2: 실수> : 1
  2개의 정수 입력 : 8000123 12260
=====덧셈=====
1. 이진수로 표현
피가수 : 0000 0000 0111 1010 0001 0010 0111 1011
가수   : 0000 0000 0000 0000 0010 1111 1110 0100
-----
2. 덧셈 연산
  0000 0000 0111 1010 0001 0010 0111 1011
+  0000 0000 0000 0000 0010 1111 1110 0100
-----
  0000 0000 0111 1010 0100 0010 0101 1111
C : 0, S : 0, Z : 0, V : 0
3. 10진수 결과값 : 8012383
=====뺄셈=====
1. 이진수로 표현
피감수 : 0000 0000 0111 1010 0001 0010 0111 1011
감수   : 0000 0000 0000 0000 0010 1111 1110 0100
-----
2. 뺄셈 연산
  0000 0000 0111 1010 0001 0010 0111 1011
-  0000 0000 0000 0000 0010 1111 1110 0100
-----
  0000 0000 0111 1001 1110 0010 1001 0111
C : 1, S : 0, Z : 0, V : 0
3. 10진수 결과값 : 7987863
=====곱셈=====
#초기값
      A                                Q                                Q-1                                M
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0010 1111 1110 0100 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#1
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0001 0111 1111 0010 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#2
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0000 0000 0000 1011 1111 1001 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#3
A ← A-M
1111 1111 1000 0101 1110 1101 1000 0101 |
SHIFT
1111 1111 1100 0010 1111 0110 1100 0010 | 1000 0000 0000 0000 0000 0101 1111 1100 | 1 | 0000 0000 0111 1010 0001 0010 0111 1011
#4
A ← A+M
0000 0000 0011 1101 0000 1001 0011 1101 |
SHIFT
0000 0000 0001 1110 1000 0100 1001 1110 | 1100 0000 0000 0000 0000 0010 1111 1110 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#5
SHIFT
0000 0000 0000 1111 0100 0010 0100 1111 | 0110 0000 0000 0000 0000 0001 0111 1111 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#6
A ← A-M
1111 1111 1001 0101 0010 1111 1101 0100 |
SHIFT
```

```
C:\Windows\system32\cmd.exe

#25
SHIFT
0000 0000 0000 0000 0000 1011 0110 1011 | 0000 1110 1000 0101 0100 0110 0000 0000 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#26
SHIFT
0000 0000 0000 0000 0000 0101 1011 0101 | 1000 0111 0100 0010 1010 0011 0000 0000 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#27
SHIFT
0000 0000 0000 0000 0000 0010 1101 1010 | 1100 0011 1010 0001 0101 0001 1000 0000 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#28
SHIFT
0000 0000 0000 0000 0000 0001 0110 1101 | 0110 0001 1101 0000 1010 1000 1100 0000 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#29
SHIFT
0000 0000 0000 0000 0000 1011 0110 | 1011 0000 1110 1000 0101 0100 0110 0000 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#30
SHIFT
0000 0000 0000 0000 0000 0101 1011 | 0101 1000 0111 0100 0010 1010 0011 0000 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#31
SHIFT
0000 0000 0000 0000 0000 0010 1101 | 1010 1100 0011 1010 0001 0101 0001 1000 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011
#32
SHIFT
0000 0000 0000 0000 0000 0001 0110 | 1101 0110 0001 1101 0000 1010 1000 1100 | 0 | 0000 0000 0111 1010 0001 0010 0111 1011

이진수 결과 값 : 0000 0000 0000 0000 0000 0000 0001 0110 1101 0110 0001 1101 0000 1010 1000 1100
10진수 결과 값 : -2147483648

=====나눗셈=====
#초기값
          A          Q          M
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 0111 1010 0001 0010 0111 1011 | 0000 0000 0000 0000 0010 1111 1110 0100
#1
SHIFT
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 1111 0100 0010 0100 1111 0110 |
A←A-M
1111 1111 1111 1111 1101 0000 0001 1100 |
복원
0000 0000 0000 0000 0000 0000 0000 | 0000 0000 1111 0100 0010 0100 1111 0110 | 0000 0000 0000 0000 0010 1111 1110 0100
#2
SHIFT
0000 0000 0000 0000 0000 0000 0000 | 0000 0001 1110 1000 0100 1001 1110 1100 |
A←A-M
1111 1111 1111 1111 1101 0000 0001 1100 |
복원
0000 0000 0000 0000 0000 0000 0000 | 0000 0001 1110 1000 0100 1001 1110 1100 | 0000 0000 0000 0000 0010 1111 1110 0100
#3
SHIFT
0000 0000 0000 0000 0000 0000 0000 | 0000 0011 1101 0000 1001 0011 1101 1000 |
A←A-M
1111 1111 1111 1111 1101 0000 0001 1100 |
복원
0000 0000 0000 0000 0000 0000 0000 | 0000 0011 1101 0000 1001 0011 1101 1000 | 0000 0000 0000 0000 0010 1111 1110 0100
#4
SHIFT
0000 0000 0000 0000 0000 0000 0000 | 0000 0111 1010 0001 0010 0111 1011 0000 |
A←A-M
1111 1111 1111 1111 1101 0000 0001 1100 |
복원
```



```
C:\Windows\system32\cmd.exe

SHIFT
0000 0000 0000 0000 0000 1001 0110 0001 | 1110 1100 0000 0000 0000 0000 1010 |
A←A-M
1111 1111 1111 1111 1101 1001 0111 1101 |
복원
0000 0000 0000 0000 0000 1001 0110 0001 | 1110 1100 0000 0000 0000 0000 1010 | 0000 0000 0000 0000 0010 1111 1110 0100
#27
SHIFT
0000 0000 0000 0000 0001 0010 1100 0011 | 1101 1000 0000 0000 0000 0000 0001 0100 |
A←A-M
1111 1111 1111 1111 1110 0010 1101 1111 |
복원
0000 0000 0000 0000 0001 0010 1100 0011 | 1101 1000 0000 0000 0000 0000 0001 0100 | 0000 0000 0000 0000 0010 1111 1110 0100
#28
SHIFT
0000 0000 0000 0000 0010 0101 1000 0111 | 1011 0000 0000 0000 0000 0000 0010 1000 |
A←A-M
1111 1111 1111 1111 1111 0101 1010 0011 |
복원
0000 0000 0000 0000 0010 0101 1000 0111 | 1011 0000 0000 0000 0000 0000 0010 1000 | 0000 0000 0000 0000 0010 1111 1110 0100
#29
SHIFT
0000 0000 0000 0000 0100 1011 0000 1111 | 0110 0000 0000 0000 0000 0000 0101 0000 |
A←A-M
0000 0000 0000 0000 0001 1011 0010 1011 |
Q0 = 1
0000 0000 0000 0000 0001 1011 0010 1011 | 0110 0000 0000 0000 0000 0000 0101 0001 | 0000 0000 0000 0000 0010 1111 1110 0100
#30
SHIFT
0000 0000 0000 0000 0011 0110 0101 0110 | 1100 0000 0000 0000 0000 0000 1010 0010 |
A←A-M
0000 0000 0000 0000 0000 0110 0111 0010 |
Q0 = 1
0000 0000 0000 0000 0000 0110 0111 0010 | 1100 0000 0000 0000 0000 0000 1010 0011 | 0000 0000 0000 0000 0010 1111 1110 0100
#31
SHIFT
0000 0000 0000 0000 0000 1100 1110 0101 | 1000 0000 0000 0000 0000 0001 0100 0110 |
A←A-M
1111 1111 1111 1111 1101 1101 0000 0001 |
복원
0000 0000 0000 0000 0000 1100 1110 0101 | 1000 0000 0000 0000 0000 0001 0100 0110 | 0000 0000 0000 0000 0010 1111 1110 0100
#32
SHIFT
0000 0000 0000 0000 0001 1001 1100 1011 | 0000 0000 0000 0000 0000 0010 1000 1100 |
A←A-M
1111 1111 1111 1111 1110 1001 1110 0111 |
복원
0000 0000 0000 0000 0001 1001 1100 1011 | 0000 0000 0000 0000 0000 0010 1000 1100 | 0000 0000 0000 0000 0010 1111 1110 0100

이진수 몫 : 0000 0000 0000 0000 0000 0010 1000 1100
10진수 몫 : 652
이진수 나머지 : 0000 0000 0000 0000 0001 1001 1100 1011
10진수 나머지 : 6603

계속하려면 아무키나 입력하세요<종료: q>
```

결과 5 - 입력 : -1000023 -21000

결과값으로 덧셈 -10021023, 뺄셈 -9979023, 곱셈 오버플로우, 나눗셈 몫 476 을 각각 출력한다.

```
C:\Windows\system32\cmd.exe
연산의 종류를 선택하세요<1: 정수, 2: 실수> : 1
2개의 정수 입력 : -1000023 -21000
=====덧셈=====
1. 이진수로 표현
피가수 : 1111 1111 0110 0111 0110 1001 0110 1001
가수 : 1111 1111 1111 1111 1010 1101 1111 1000

2. 덧셈 연산
1111 1111 0110 0111 0110 1001 0110 1001
+ 1111 1111 1111 1111 1010 1101 1111 1000
-----
1111 1111 0110 0111 0001 0111 0110 0001

C : 1, S : 1, Z : 0, U : 0
3. 10진수 결과값 : -10021023
=====뺄셈=====
1. 이진수로 표현
피감수 : 1111 1111 0110 0111 0110 1001 0110 1001
감수 : 1111 1111 1111 1111 1010 1101 1111 1000

2. 뺄셈 연산
1111 1111 0110 0111 0110 1001 0110 1001
- 1111 1111 1111 1111 1010 1101 1111 1000
-----
1111 1111 0110 0111 1011 1011 0111 0001

C : 0, S : 1, Z : 0, U : 0
3. 10진수 결과값 : -9979023
=====곱셈=====
#초기값
      A                Q                Q-1                M
0000 0000 0000 0000 0000 0000 0000 0000 | 1111 1111 1111 1111 1010 1101 1111 1000 | 0 | 1111 1111 0110 0111 0110 1001 0110 1001
#1
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0111 1111 1111 1111 1101 0110 1111 1100 | 0 | 1111 1111 0110 0111 0110 1001 0110 1001
#2
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0011 1111 1111 1111 1110 1011 0111 1110 | 0 | 1111 1111 0110 0111 0110 1001 0110 1001
#3
SHIFT
0000 0000 0000 0000 0000 0000 0000 0000 | 0001 1111 1111 1111 1111 0101 1011 1111 | 0 | 1111 1111 0110 0111 0110 1001 0110 1001
#4
A ← A-M
0000 0000 1001 1000 1001 0110 1001 0111 |
SHIFT
0000 0000 0100 1100 0100 1011 0100 1011 | 1000 1111 1111 1111 1111 1010 1101 1111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#5
SHIFT
0000 0000 0010 0110 0010 0101 1010 0101 | 1100 0111 1111 1111 1111 1101 0110 1111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#6
SHIFT
0000 0000 0001 0011 0001 0010 1101 0010 | 1110 0011 1111 1111 1111 1110 1011 0111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#7
SHIFT
0000 0000 0000 1001 1000 1001 0110 1001 | 0111 0001 1111 1111 1111 1111 0101 1011 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#8
```

```
C:\Windows\system32\cmd.exe

SHIFT
0000 0000 0000 0000 1100 0011 1001 0100 | 0000 0100 0100 1010 1110 0011 1111 1111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#23
SHIFT
0000 0000 0000 0000 0110 0001 1100 1010 | 0000 0010 0010 0101 0111 0001 1111 1111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#24
SHIFT
0000 0000 0000 0000 0011 0000 1110 0101 | 0000 0001 0001 0010 1011 1000 1111 1111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#25
SHIFT
0000 0000 0000 0000 0001 1000 0111 0010 | 1000 0000 1000 1001 0101 1100 0111 1111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#26
SHIFT
0000 0000 0000 0000 0000 1100 0011 1001 | 0100 0000 0100 0100 1010 1110 0011 1111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#27
SHIFT
0000 0000 0000 0000 0000 0110 0001 1100 | 1010 0000 0010 0010 0101 0111 0001 1111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#28
SHIFT
0000 0000 0000 0000 0000 0011 0000 1110 | 0101 0000 0001 0001 0010 1011 1000 1111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#29
SHIFT
0000 0000 0000 0000 0001 1000 0111 | 0010 1000 0000 1000 1001 0101 1100 0111 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#30
SHIFT
0000 0000 0000 0000 0000 1100 0011 | 1001 0100 0000 0100 0100 1010 1110 0011 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#31
SHIFT
0000 0000 0000 0000 0000 0000 0110 0001 | 1100 1010 0000 0010 0010 0101 0111 0001 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001
#32
SHIFT
0000 0000 0000 0000 0000 0000 0011 0000 | 1110 0101 0000 0001 0001 0010 1011 1000 | 1 | 1111 1111 0110 0111 0110 1001 0110 1001

이진수 과 값 : 0000 0000 0000 0000 0000 0000 0000 0011 0000 1110 0101 0000 0001 0001 0010 1011 1000
10진수 과 값 : -2147483648

=====나눗셈=====
# 초기값
A Q M
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1111 0110 0111 0110 1001 0110 1001 | 1111 1111 1111 1111 1010 1101 1111 1000
#1
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1110 1100 1110 1101 0010 1101 0010 |
A ← A-M
0000 0000 0000 0000 0101 0010 0000 0111 |
복원
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1110 1100 1110 1101 0010 1101 0010 | 1111 1111 1111 1111 1010 1101 1111 1000
#2
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1101 1001 1101 1010 0101 1010 0100 |
A ← A-M
0000 0000 0000 0000 0101 0010 0000 0111 |
복원
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1101 1001 1101 1010 0101 1010 0100 | 1111 1111 1111 1111 1010 1101 1111 1000
#3
SHIFT
1111 1111 1111 1111 1111 1111 1111 1111 | 1111 1011 0011 1011 0100 1011 0100 1000 |
A ← A-M
0000 0000 0000 0000 0101 0010 0000 0111 |
복원
```

```
C:\Windows\system32\cmd.exe

Q0 = 1
1111 1111 1111 1111 1100 0100 1110 1010 | 1101 0010 0000 0000 0000 0000 0011 | 1111 1111 1111 1111 1010 1101 1111 1000
#26
SHIFT
1111 1111 1111 1111 1000 1001 1101 0101 | 1010 0100 0000 0000 0000 0000 0110 |
A ← A-M
1111 1111 1111 1111 1101 1011 1101 1101 |
Q0 = 1
1111 1111 1111 1111 1101 1011 1101 1101 | 1010 0100 0000 0000 0000 0000 0111 | 1111 1111 1111 1111 1010 1101 1111 1000
#27
SHIFT
1111 1111 1111 1111 1011 0111 1011 1011 | 0100 1000 0000 0000 0000 0000 1110 |
A ← A-M
0000 0000 0000 0000 0000 1001 1100 0011 |
복원
1111 1111 1111 1111 1011 0111 1011 1011 | 0100 1000 0000 0000 0000 0000 1110 | 1111 1111 1111 1111 1010 1101 1111 1000
#28
SHIFT
1111 1111 1111 1111 0110 1111 0111 0110 | 1001 0000 0000 0000 0000 0000 1100 |
A ← A-M
1111 1111 1111 1111 1100 0001 0111 1110 |
Q0 = 1
1111 1111 1111 1111 1100 0001 0111 1110 | 1001 0000 0000 0000 0000 0000 1101 | 1111 1111 1111 1111 1010 1101 1111 1000
#29
SHIFT
1111 1111 1111 1111 1000 0010 1111 1101 | 0010 0000 0000 0000 0000 0000 0011 1010 |
A ← A-M
1111 1111 1111 1111 1101 0101 0000 0101 |
Q0 = 1
1111 1111 1111 1111 1101 0101 0000 0101 | 0010 0000 0000 0000 0000 0000 0011 1011 | 1111 1111 1111 1111 1010 1101 1111 1000
#30
SHIFT
1111 1111 1111 1111 1010 1010 0000 1010 | 0100 0000 0000 0000 0000 0000 0111 0110 |
A ← A-M
1111 1111 1111 1111 1111 1100 0001 0010 |
Q0 = 1
1111 1111 1111 1111 1111 1100 0001 0010 | 0100 0000 0000 0000 0000 0000 0111 0111 | 1111 1111 1111 1111 1010 1101 1111 1000
#31
SHIFT
1111 1111 1111 1111 1111 1000 0010 0100 | 1000 0000 0000 0000 0000 0000 1110 1110 |
A ← A-M
0000 0000 0000 0000 0100 1010 0010 1100 |
복원
1111 1111 1111 1111 1111 1000 0010 0100 | 1000 0000 0000 0000 0000 0000 1110 1110 | 1111 1111 1111 1111 1010 1101 1111 1000
#32
SHIFT
1111 1111 1111 1111 1111 0000 0100 1001 | 0000 0000 0000 0000 0000 0001 1101 1100 |
A ← A-M
0000 0000 0000 0000 0100 0010 0101 0001 |
복원
1111 1111 1111 1111 1111 0000 0100 1001 | 0000 0000 0000 0000 0000 0001 1101 1100 | 1111 1111 1111 1111 1010 1101 1111 1000

이진수 : 0000 0000 0000 0000 0000 0001 1101 1100
10진수 : 476
이진수 나머지 : 1111 1111 1111 1111 1111 0000 0100 1001
10진수 나머지 : -4023

계속하려면 아무키나 입력하세요<종료: q>
```

4) 토의사항

정수 연산의 곱셈 나눗셈을 하는 과정에서 가장 크게 느꼈던 문제점은, 물론 덧셈과 뺄셈 알고리즘도 완벽히 구현되어야 하지만, 추가로 각 상황에 일어나는 모든 과정을 확실하게 코드로 옮겨야 한다는 점이였다. 단 한 가지라도 빼먹으면 오류가 났고 원하는 결과값이 도출되지 않았다. 특히 나눗셈에서 제수가 음수일 때 A 레지스터의 모든 값을 1로 초기화하는 것을 착각하여 오류가 났고 이로 인해 시간이 많이 걸렸다. 실수 연산을 겸하여 다양한 비트 크기의 이진수 배열들을 연산하는 과정이 굉장히 헛갈리고 힘들었다. 또한 여러 가지 예외 사항에 대해서 생각해 보는 것도 굉장히 까다로웠다. 이것들을 해결하는 과정에서 이진수 배열 계산을 좀 더 정확하게 알게 되었고 특히 실수 연산을 더욱 더 깊게 알게 되는 계기가 되었다.

실수 연산을 위한 IEEE754 표준화를 하는 과정에서 예외처리에 대해 확실하게 해결을 하지 못했다. float 형의 범위와 IEEE754 표준의 범위가 같다고 설명을 들었는데 실제 범위의 끝 값들을 대입했을 경우 코드 논리의 오류인지는 확실히 모르겠지만 부정확한 값이 들어가게 되면서 많은 고민을 했다. 뿐만 아니라 float 형의 수를 2진수화 하는 과정에서 나머지 연산자(%)를 사용했어야 했는데 이 연산자가 int 형만을 취급하여 2진수화를 하는 함수를 설계함에 있어서도 많은 시간을 할애했다.

dec_to_bin() 함수에서 음수가 들어오는 경우,

음수에 절대값을 취하고 그 값을 2진수로 변환한 뒤 2의 보수를 취해주었다.

최솟값인 -2^{31} 을 2진수로 표현하는 과정에서 절대값인 2^{31} 은 32bit의 데이터 표현 범위를 벗어나게 되어 런타임 에러가 발생하였다. INT_MIN 입력이 들어오는 경우는 따로 예외처리를 해주어 해결하였다.