



과목명	컴퓨터 구조
담당교수	송인식 교수님
학과	소프트웨어학과
학번	32200185
이름	곽다은
제출일자	2021.09.26

Implementing Saturating Addition

Saturating Arithmetic

Saturating arithmetic은 연산의 결과값이 최소값과 최대값 사이의 고정 범위로 제한된 산술이다. 연산 결과가 최대보다 큰 경우 결과는 고정된 최대값으로 설정되고, 최소값보다 작은 경우 고정된 최소값으로 설정된다.

예를 들어 유효한 값의 범위가 -100에서 100 사이인 경우, Saturating 연산의 결과는 다음과 같다.

- $60 + 30 \rightarrow 90$
- $60 + 70 \rightarrow 100$
- $60 - 200 \rightarrow -100$

Bit-level integer coding rules

Bit-level에서의 정수를 코딩할 때 지켜지는 규칙이 있는데, 이는 C언어에서의 비트, 논리 및 산술 연산을 더 잘 이해하도록 하기 위함이라고 한다. 여러가지 규칙들 중, 금지된 항목으로 조건문, 반복문, switch문, 함수호출 등이 있다. 허용되는 연산으로는 bit-level과 논리 연산자들, shift연산 등이 있고, 특정 상수는 INT_MIN과 INT_MAX로 표현한다.

위의 내용들을 참고하여 Saturating Addition을 구현해 보았다.

구현 코드

```
#include <stdio.h>
#include <limits.h>
int saturating_add(int x, int y) {

    int result = x + y;

    unsigned sign_x = (unsigned)x >> 31;
    unsigned sign_y = (unsigned)y >> 31;
    unsigned sign_result = (unsigned)result >> 31;

    // x>0 y>0 result<0 result=MAX
    (sign_x == 0) && (sign_y == 0) && (sign_result == 1) && (result = INT_MAX);
    // x<0 y<0 result>0 result=MIN
    (sign_x == 1) && (sign_y == 1) && (sign_result == 0) && (result = INT_MIN);
```

```

    return result;
}
int main() {
    int x = 0, y = 0;
    printf("정수 x를 입력하세요 : ");
    scanf_s("%d", &x);
    printf("정수 y를 입력하세요 : ");
    scanf_s("%d", &y);
    printf("%d", saturating_add(x, y));
}

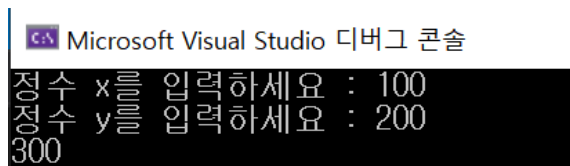
```

- C언어에서 제공하는 <limits.h> 라이브러리의 INT_MAX, INT_MIN을 통해 integer 형식 변수의 최대와 최소값을 설정해주었다.
- shift 연산자를 사용하여 MSB를 확인해 음수 / 양수를 판단하였고, 연산의 결과값을 통해 INT_MAX와 INT_MIN을 리턴할지를 판단해주었다.

실행 결과

1. 연산 결과가 범위 내에 있을 경우

- $100 + 200 = 300$



Microsoft Visual Studio 디버그 콘솔

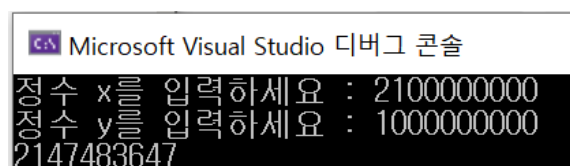
```

정수 x를 입력하세요 : 100
정수 y를 입력하세요 : 200
300

```

2. 연산 결과가 INT_MAX를 넘어선 경우

- $2100000000 + 1000000000 = 2147483647$ ($3100000000 \rightarrow 2147483647$)



Microsoft Visual Studio 디버그 콘솔

```

정수 x를 입력하세요 : 2100000000
정수 y를 입력하세요 : 1000000000
2147483647

```