

System program HW1

2020310328 신진건

* Sfp가 표현할 수 있는 수의 범위 : $-65504 \leq \text{int or float} \leq 65504$

* Sfp 가 31744이면 +무한, 64512이면 -무한

* Sfp 가 NaN인 범위 : $31744 < \text{sfp} < 32768$, $64512 < \text{sfp} < 65335$

1. sfp float2sfp(float input)

- float 과 unsigned int 가 있는 union을 선언한다.
- union 안의 float에 input을 저장한다. Input의 비트를 연산하기 위해 union 안의 unsigned int 사용한다.
- 먼저 비트연산자를 통해 부호를 float에서 sfp로 옮겨준다.
- 그 다음 반복문을 통해 float의 exp와 E를 구하고 sfp를 만족하는 형태로 다시 바꿔준다.
- 반복문과 비트연산자를 통해 exp를 sfp에 맞는 비트로 옮겨준다.
- float의 소수점 아래 10자리를 비트연산자와 반복문을 통해 sfp의 frac으로 옮긴다. (round to zero 이기 때문에 float의 남은 frac은 버린다.
- input이 0 이면 0을 반환, input이 65504보다 크면 31744를 반환, input이 -65504보다 작으면 64512를 반환하게 함으로써 special case 처리한다.

2. sfp int2sfp (int input)

- int를 float형으로 바꾸기.
- 그 다음은 float2sfp 함수와 동일한 방식으로 작성한다.

3. float sfp2float (sfp float)

- 비트연산을 통해 부호를 저장. 마지막에 부호를 곱해준다.
- 반복문과 비트연산을 통해 exp와 E를 저장

- 반복문과 비트연산을 통해 hidden bit를 포함해서 frac 부분을 10진수로 계산하여 저장.
- E가 0보다 크다면 앞에서 구한 frac 부분에서 2^E 를 곱하고 E가 0보다 작다면 나눈다.
- $E = -15$ 이고 앞에서 계산한 frac이 1이라면 Denormalized value로 0을 반환한다.
- sfp가 무한의 형태라면 조건문을 통해 float의 무한 형태를 출력하도록 함.
- sfp가 NaN 이라면 조건문을 통해 float의 NaN 형태를 출력하도록 함

4. int sfp2int (sfp input)

- sfp2float과 같은 방법으로 한다.
- round to zero 이므로 마지막 반환값을 int로 바꿔준다.
- special case는 sfp의 범위와 조건문을 통해 처리해준다.

5. sfp sfp_add(sfp a, sfp b)

- 먼저 가장 앞에서 조건문을 통해 special case를 처리한다.
- 부호가 다를 때와 같을 때로 나누고 그 안에서 또 a의 exp가 b의 exp 보다 클 때와 작을 때로 나뉘어서 구한다.
- 비트 연산을 통해 a와 b 각각의 exp와 frac을 구한다. (비트를 int 형에 저장) (hidden bit 까지 저장)
- 부호가 같을 때는 2개의 frac의 덧셈이고, 다를 때는 2개의 frac의 뺄셈이다.
- 덧셈의 예시처럼 더 작은 exp 값을 가진 sfp를 큰 exp 바꿔준다. 이때 round to even을 위해 버려지는 비트를 rest에 따로 저장한다.
- 부호가 같아서 덧셈을 할 경우에는 frac의 마지막 bit가 1이고 rest가 1000... 일 때 $frac+1$ 을 해준다. 반면 마지막 bit가 0이고 rest가 1000... 일 때는 1을 더하지 않는다. (round to even)
- 부호가 같아서 뺄셈을 할 경우에는 frac의 마지막 bit가 1이고 rest가 1000... 일 때 $frac-1$ 을 해준다. 반면 마지막 bit가 0이고 rest가 1000... 일 때는 1을 빼지 않는다. (round to even)
- 계산을 한 후에 hidden bit 가 0 이라면 frac 부분에서 있는 첫 번째 1을 hidden bit로 조작해주고 exp도 그에 맞게 바꿔준다.
- 지금까지 구한 부호와 exp, frac을 비트연산을 통해 정답을 설정한 후 반환한다.

6. sfp sfp_mul(sfp a, sfp b)

- 먼저 가장 앞에서 조건문을 통해 special case를 처리한다.
- 지수는 지수끼리 frac은 frac끼리 곱셈을 해야한다.
- 부호는 따로 저장한 후 마지막에 결과값에 저장시킨다.
- a와 b의 exp를 각각 구한 후에 더한 다음 15를 뺀 값이 결과값의 exp이다. (bias 중복이라 15뺀)
- frac 부분의 비트 계산을 위해 hidden 비트를 포함한 frac을 int에 저장한다.
- 각각의 frac을 곱하면 22비트 혹은 21비트의 값이 나온다.
- 앞의 10개의 비트는 결과값에 저장을 하고 나머지는 round to even을 위해 다른 변수에 저장시킨다.
- 나머지 비트 값이 1000... 보다 크다면 올림, 1000... 보다 작다면 버린다.
- 1000... 이고 결과값의 frac의 마지막 비트가 1이라면 frac에 1을 더해주고 아니라면 버린다.
- 지수가 30을 넘는다면 조건문을 통해 무한을 출력하도록 한다.

7. char* sfp2bits(sfp result)

- malloc을 통해 char 배열 메모리 할당
- 첫번째 비트를 비트 연산자를 통해 배열에 저장(첫번째 비트는 배열에서 가장 마지막에 위치)
- 반복문과 비트연산자를 통해 나머지를 배열에 저장(비트의 위치와 배열의 인덱스는 반대)
- char 배열 반환