CA03

- MSB를 sign bit으로 사용 (2의 보수)

Sign extention fiel   addiu도 sign extention 임!
sltiu

zero extention: andi, ori, xori

- overflow detect

+ + → -	a, b: 계산하는 수
- - → + 인 경우	s: 결과
	c: carry out

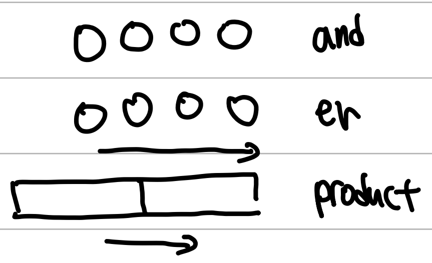
$$V = a_{n-1} \cdot b_{n-1} \cdot S_{n-1}' + a_{n-1}' \cdot b_{n-1}' \cdot S_{n-1}$$
$$= C_n \oplus C_{n-1}$$

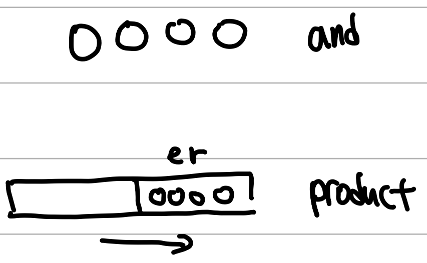
서로 다르면 overflow
같으면 정상

- multiplication

$n \times n \rightarrow 2n$ 크기

$\leftarrow$ shift left
0 0 0 0   multiplicand
0 0 0 0   multiplier
$\rightarrow$ shift right





6x5 0110
0101

0110 0101
0110 0101
0011 0010
0001 1001
+0110
0111 1001
&
0011 1100
0001 1110

and
0110 =6

multiplier
0000 0101
add 0110 0101
0011 → 0010
add 0011 0010
0001 → 1001
add 0111 1001
0011 → 1100
add 0011 1100
0001 → 1110

- division

– floating point

$$(-1)^{sign} \times (\underbrace{1}_{\text{hidden}} + \text{Fraction}) \times 2^{\text{exponent} - \underbrace{\text{bias}}_{127 \text{ or } 1023}}$$

smallest +	0	0—01	1.00—0	$= 1 \times 2^{1-127}$
largest +	0	1—10	1.11—1	$= (2 - 2^{-23}) \times 2^{254-127}$

$1.0_2 \times 2^{-1} = 0 \quad \overset{126}{01111110} \quad 1.0—0$

$0.75_{10} \times 2^4 = 0 \quad 10000010 \quad 1.10—0$

	E	F	
zero →	all zero	All zero	(denormalized)
infinity →	all 1	all zero	
NaN →	all 1	non zero	

↘ E가 all zero 면
hidden bit이 0으로 바뀜
$(-1)^S \times (0.\cdots) \times 2^{1-127}$

– rounding

round to nearest even .100 일때

guard round sticky (3번째 이하에 1이 하나라도 있는지)

guard 가 0이면 항상버림

101면 → 절반혹은 그 이상 → 완전 중간이면 짝수쪽으로 바꾸어짐.

↘ 그 이상이면 올림

- Addition

exponent 차이 → 더 작은 수를 그 차이만큼 right shift 해서 자릿수 맞춤

exponent가 같아졌으면 fraction 덧셈, 뺄셈 수행

결과 정규화, GRS로 반올림 → IEEE 형식으로 변환

) 밀려난 bit가 GRS (f 필드 바깥으로 나가버림)

히든 bit 복구

1.1100 shift right

$1.0000 + (-0.111) = 1.0000 - 0.111 = 0.001$

$0.001 \times 2^{-1} = 0.010 \times 2^{-2} \dots = 1.000 \times 2^{-4}$

이게 round 됨

hidden bit 숨기고 저장

- multiplication

$(f_1 \times 2^{e_1}) \times (f_2 \times 2^{e_2}) = f_1 \times f_2 \times 2^{(e_1 + e_2)}$

Step 0 hidden bit 복원

Step 1 bias를 고려한 Exponent 구하기 $E_1 + E_2 - 127 = E$

$-1 + (-2) = -3$, $(-1+127) + (-2+127) - 127 = 124$

$1.0000 \times 1.110 = 1.110000$

1.110000×2^{-3}

- MIPS floating point instruction

- Subword parallellism

addiu -> Sign extantion