INT to FP32 cast / FP32 comparator

구현 할 기능

1. INT to FP32 casting (INTtoFP32.v)
   - 정수(양수만 고려)를 FP32로 변환
2. FP32 compare value (FP32_cmp_value.v)
   - fp32인 a와 b를 입력 받아 비교하여 크거나 작은 fp32 값을 출력
3. fp32 compare (FP32_cmp.v)
   - fp32인 a와 b를 입력 받아 operator(≥, >, =, <, ≤)에 따라 true, false를 출력

INT to FP32 casting

input : 양의 정수(32bit)

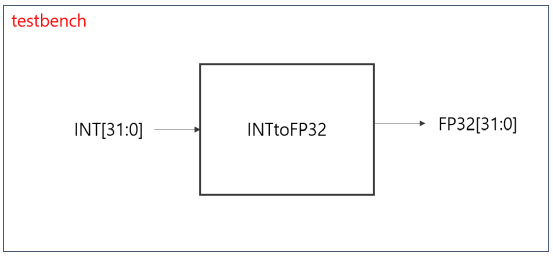
output : fp32(32bit)

3가지 결과

- 0(zero) (sign : 0 / exp : 8'b0 / man : 23'b0)
- normal
- inf(infinity) (sign : 0 / exp : 8'b1 / man : 23'b0)

  → 양의 정수 32bit를 input으로 받을 경우 inf 발생X(입력 bit의 제한이 없을 경우 inf 변환도 확인하였음)

Test

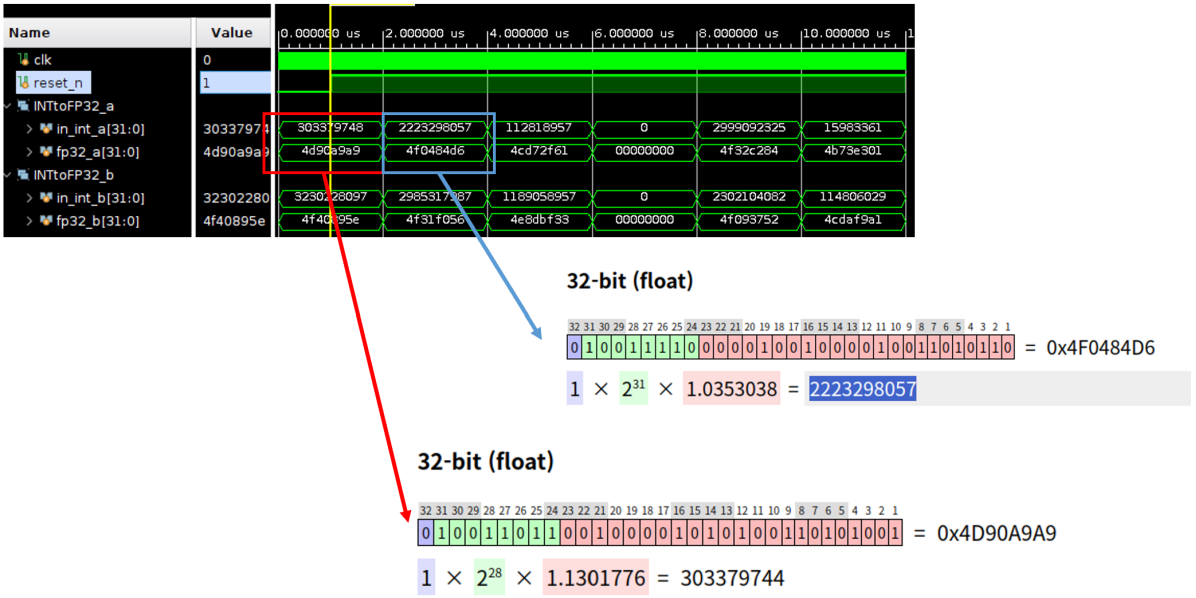



```
initial begin
    in_int_a = $random;
    in_int_b = $random;

    #2000;
    in_int_a = $random;
    in_int_b = $random;

    #2000;
    in_int_a = 0;
    in_int_b = 0;

    #2000;
    in_int_a = $random;
    in_int_b = $random;

    #2000;
    in_int_a = $random;
    in_int_b = $random;

    #2000;
    in_int_a = $random;
    in_int_b = $random;

    #2000;
    in_int_a = $random;
    in_int_b = $random;

    #2000;
    $stop;
```

임의의 양의 정수를 입력하고 fp32 출력 값 확인

→ https://evanw.github.io/float-toy/(정수, fp32, fp16 변환 사이트로 값을 확인)

(waveform / 변환 값 비교)



FP32 comparator

fp32_cmp_value

입력 값 : fp32_a, fp32_b(32 bit) / is_max(1bit - 0 : min / 1 : max)

출력 값 : 비교 후 결과 fp32 value(32bit)

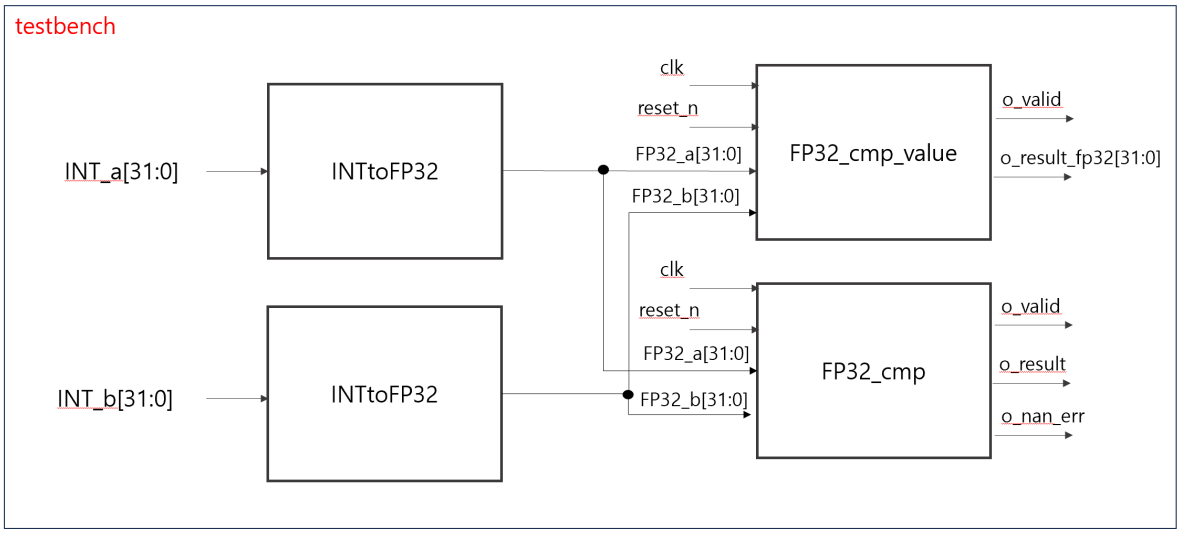
fp32_cmp

입력 값 : fp32_a, fp32_b(32 bit) / is_op(3bit - 0: ≥ / 1: > / 2: = / 3: < / 4: ≤)

출력 값 : operator에 따른 비교 결과(1bit - 0 : false / 1 : true)

Test

양의 정수 값 입력 후 결과 값 확인

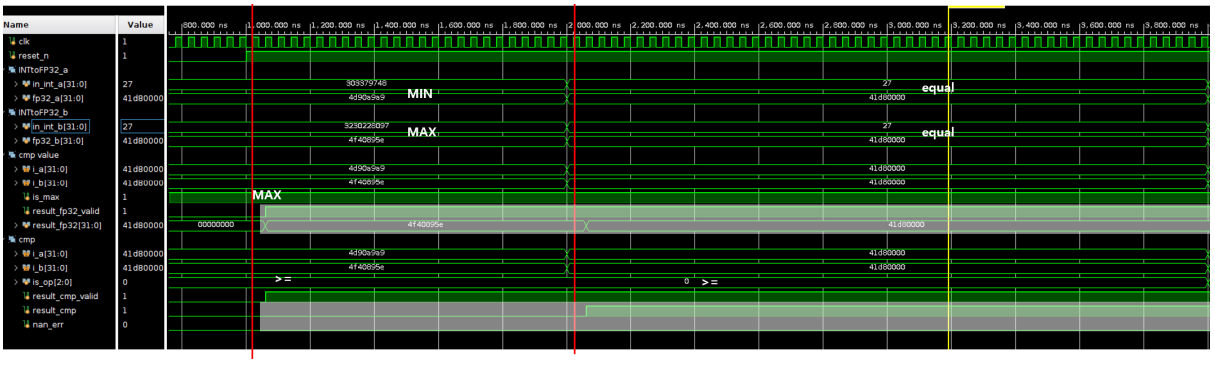

testbench
INT_a[31:0]
INTtoFP32
INT_b[31:0]
INTtoFP32
clk
reset_n
FP32_a[31:0]
FP32_b[31:0]
FP32_cmp_value
o_valid
o_result_fp32[31:0]
clk
reset_n
FP32_a[31:0]
FP32_b[31:0]
FP32_cmp
o_valid
o_result
o_nan_err

```
104     initial begin
105         in_int_a = $random;
106         in_int_b = $random;
107         is_max = `MAX;
108         is_op = `OP_GTE;
109 
110         #2000;
111         in_int_a = 27;
112         in_int_b = 27;
113 
114         #2000;
115         in_int_a = $random;
116         in_int_b = $random;
117         is_max = `MAX;
118         is_op = `OP_GT;
119 
120         #2000;
121         in_int_a = 27;
122         in_int_b = 27;
123 
124         #2000;
125         in_int_a = $random;
126         in_int_b = $random;
127         is_max = `MAX;
128         is_op = `OP_EQ;
129 
130         #2000;
131         in_int_a = 27;
132         in_int_b = 27;
133 
134         #2000;
135         in_int_a = $random;
136         in_int_b = $random;
137         is_max = `MIN;
138         is_op = `OP_LT;
139 
140         #2000;
141         in_int_a = 27;
142         in_int_b = 27;
143 
144         #2000;
145         in_int_a = $random;
146         in_int_b = $random;
147         is_max = `MIN;
148         is_op = `OP_LTE;
149 
150         #2000;
151         in_int_a = 27;
152         in_int_b = 27;
153 
154         #2000;
155         $stop;
```

case1.

is_max = max

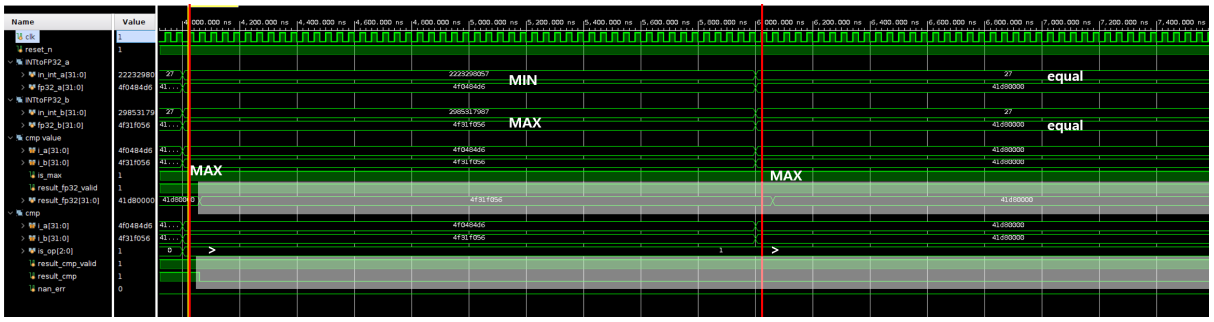
i_op ≥



case2.

is_max = max

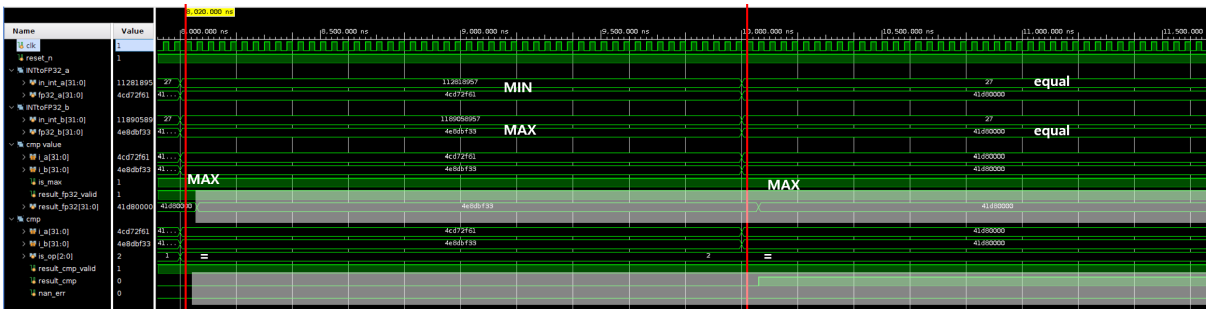
i_op >



case3.

is_max = max

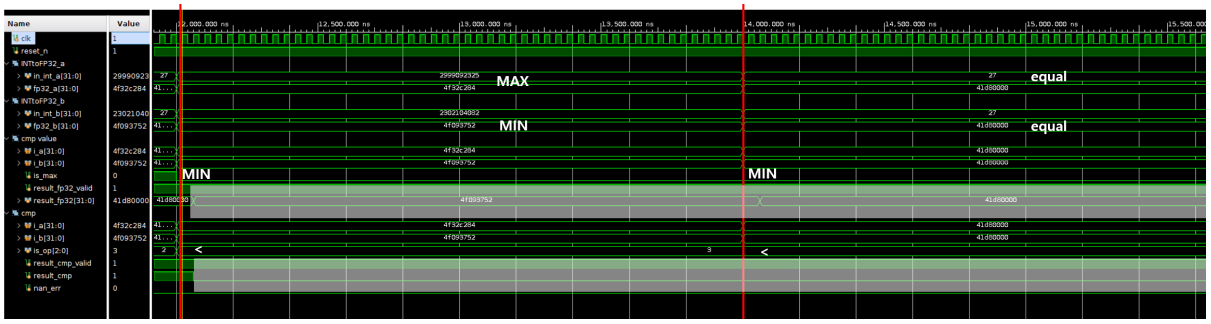
i_op =



case4.

is_max = min

i_op <

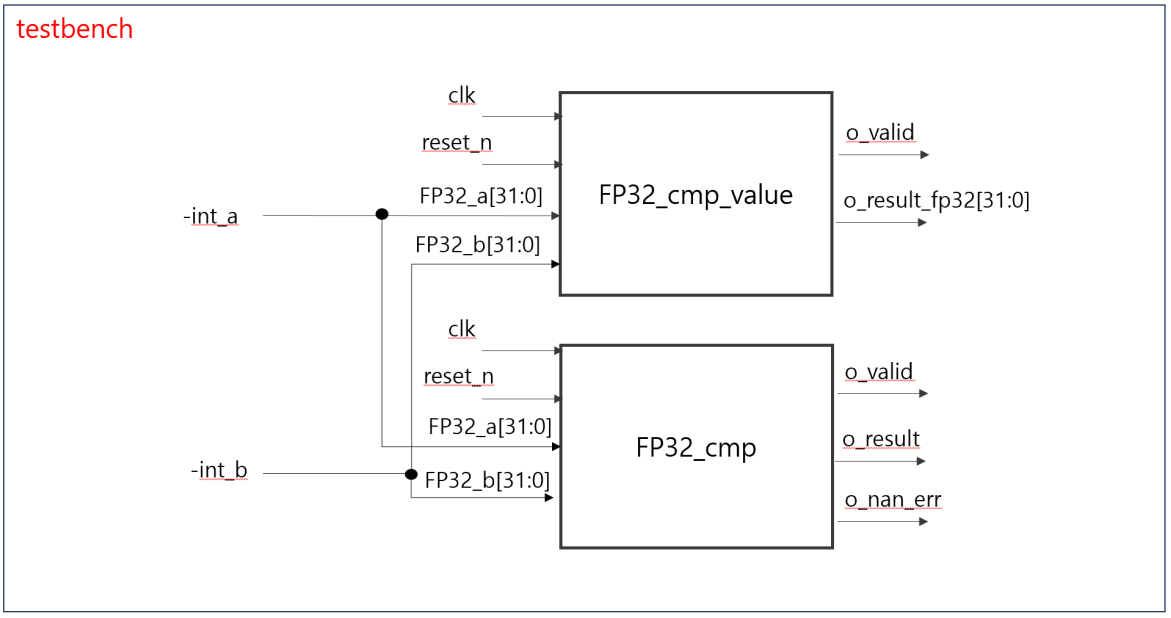


case5.

is_max = min

i_op ≤

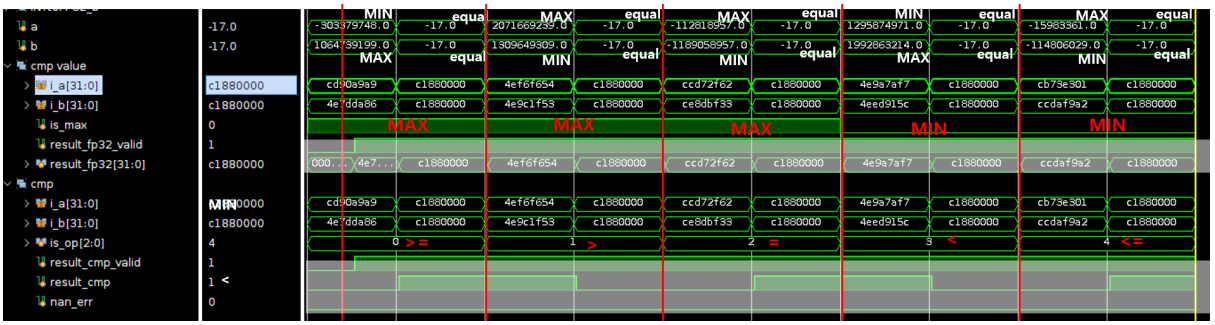
음의 정수 값 입력 후 비교 값 확인


testbench
clk
reset_n
FP32_a[31:0]
-int_a
FP32_b[31:0]
FP32_cmp_value
o_valid
o_result_fp32[31:0]
clk
reset_n
FP32_a[31:0]
-int_b
FP32_b[31:0]
FP32_cmp
o_valid
o_result
o_nan_err

```
//negative test

a = -($random);
b = -($random);
fp32_a = $shortrealtobits(a);
fp32_b = $shortrealtobits(b);
is_max = `MAX;
is_op = `OP_GTE;

#2000;
a = -17.0;
b = -17.0;
fp32_a = $shortrealtobits(a);
fp32_b = $shortrealtobits(b);

#2000;
a = -($random);
b = -($random);
fp32_a = $shortrealtobits(a);
fp32_b = $shortrealtobits(b);
is_max = `MAX;
is_op = `OP_GT;

#2000;
a = -17.0;
b = -17.0;
fp32_a = $shortrealtobits(a);
fp32_b = $shortrealtobits(b);

#2000;
a = -($random);
b = -($random);
fp32_a = $shortrealtobits(a);
fp32_b = $shortrealtobits(b);
is_max = `MAX;
is_op = `OP_EQ;

#2000;
a = -17.0;
b = -17.0;
fp32_a = $shortrealtobits(a);
fp32_b = $shortrealtobits(b);


#2000;
a = -($random);
b = -($random);
fp32_a = $shortrealtobits(a);
```

testbench에서 임의로 음수 값을 fp32로 변환하여 입력 후 결과 값을 확인

(waveform)



is_max MAX → MAX → MAX → MIN → MIN 순서

i_op ≥ → > → = → < → ≤ 순서