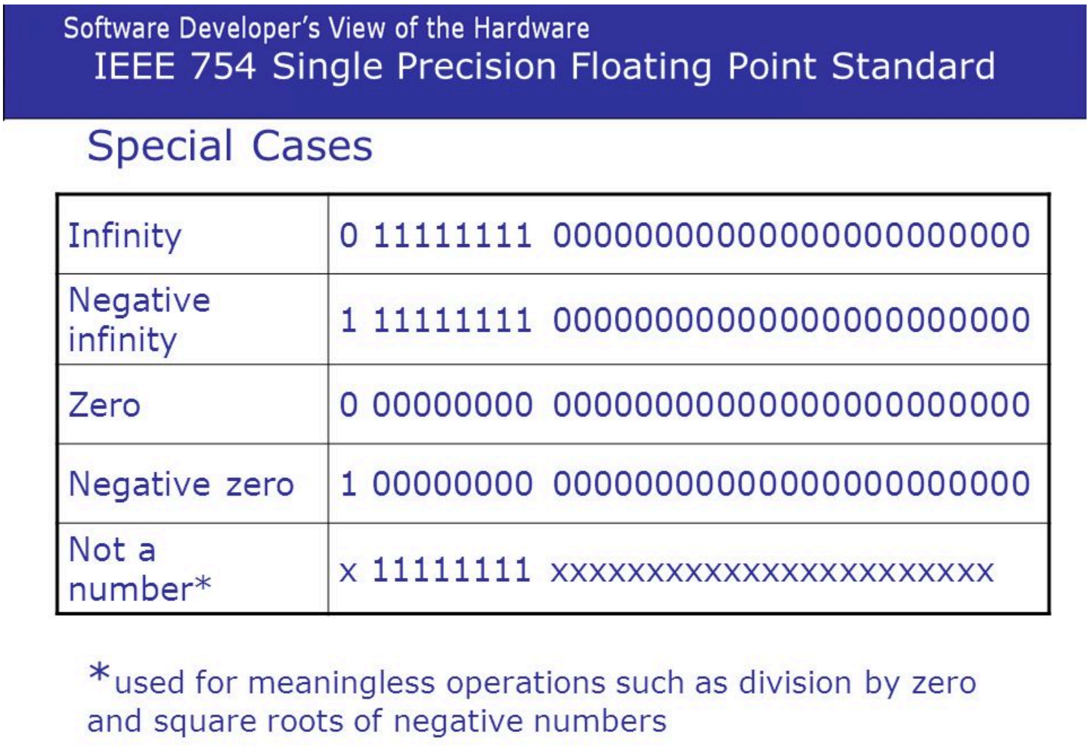
# In21-S5-EN3021 - Digital System Design

Individual Project

210351 Madugalle M.R.K.J.M.M.S.B



Software Developer's View of the Hardware

IEEE 754 Single Precision Floating Point Standard

Special Cases

| | |
|---|---|
| Infinity | 0 11111111 00000000000000000000000 |
| Negative infinity | 1 11111111 00000000000000000000000 |
| Zero | 0 00000000 00000000000000000000000 |
| Negative zero | 1 00000000 00000000000000000000000 |
| Not a number* | x 11111111 xxxxxxxxxxxxxxxxxxxxxxx |

*used for meaningless operations such as division by zero and square roots of negative numbers

**Normalized Values**

$$V = (-1)^S \times 2^{(E-127)} \times (1.F)$$

**Un-Normalized values**

$$V = (-1)^S \times 2^{(-126)} \times (0.F)$$

## Implement an adder block.

Inputs

A[31:0] - IEEE754 format number

B[31:0] - IEEE754 format number

control - one input. must trigered when positive edge is detected.

reset - must terminate the current process when positive edge detected

Outputs
out[31:0] – IEEE754 format
exception – output ‘High’ when exception is detected

Description.
Module name: Adder.v
The module start when the control signal has a positive edge.

The followings need to be done sequencialy.
step 01: The module gets two IEEE754 format numbers. And break them down to sign, mantisa, exponent in both numbers.

step 02 : get the difference of the exponents and also add two bits(‘01’) to both mantisas. Eg: for mantisa 1 add, 01 in 24th and 23th position, so that the new mantisa will be 24:0 size. With leading 01.

step 03: get the smallest number (first check exponent, if they are same, then check mantisa) right shift the bits in the mantisa of the smallest number.

step 04: add two mantisas, new exponent = maximum from two exponent

step 5: Normalize the number.
if 24th and 23rd bits are 11 then right shift the mantisa and increase the exponent.

if 24th and 23rd bits are 01 then do nothing

i if 24th and 23rd bits are 00 then leftshift until find a 1 in the 23rd bit. Decrease the exponent accordingly.

Then get thepart [22:0] from [24:0] for the output mantisa, new exponent and sign bit.

assume only 0 and 0 or 1 and 1 for the sign bit. No need to handle different sign. Only same sign is enough

· Adder substractor. Initialized.
sign bit first invert if substracting

# Implement Multiplier block.

D

Inputs

A[31:0] - IEEE754 format number

B[31:0] - IEEE754 format number

control - one input. must trigered when positive edge is detected.

reset - must terminate the current process when positive edge detected

Outputs
out[31:0] – IEEE754 format
exception – output 'High' when exception is detected

Description.
Module name: multiplier.v
The module start when the control signal has a positive edge.

The followings need to be done sequencialy.

step 01: The module gets two IEEE754 format numbers. And break them down to sign, mantisa, exponent in both numbers.

Step02: check for sign. If the same bit sign will be 0, if different bit then sign =1
output sign is that bit.

Step 03: Get the sum of the exponents and substract 127 from that. That gives the new exponent.

Step03: initialize a register to save 46 bits. Initialize with 46 zeros.([45:0]) Add 1s to strat of both mantisas. As the 23rd bit. Perform multiplication in 'partial sum' approach use the 46bit register.

Step04: Normalize the number

If 46th bit is 1 new mantisa is [44:22] round to nearest even, increase the exponent by 1.

If 46th bit and 45th bit is 01, then new mantisa is [43:21] round to nearest even

If 46th bit and 45th bit is 00, then left shift until 45th bit is 1, reduce the exponent accordingly. New mantisa will be [43:21] round to nearest even

Step05: output

# Implement Division

Similar code for ieee754 floating point divider, using SRT division algorithm for divide two mantisa.

Inputs

DD[31:0] - IEEE754 format number which is the dividend

DS[31:0] - IEEE754 format number which is the divisor

control - one input. must trigered when positive edge is detected.

reset - must terminate the current process when positive edge detected

Outputs
out[31:0] – IEEE754 format
exception – output 'High' when exception is detected

Description.
Module name: divider.v
The module start when the control signal has a positive edge.

The followings need to be done sequencialy.

Step 01: The module gets two IEEE754 format numbers. And break them down to sign, mantisa, exponent in both numbers.

Step02: check for sign. If the same bit sign will be 0, if different bit then sign =1
output sign is that bit.

Step03: exponent of the output is exponent of dividend – exponent of divisor, gives the new exponent.

Step04: divide mantisas using SRT division algorithm.

Step05: Normalize the number

Step06: output