FPGA Firmware Development

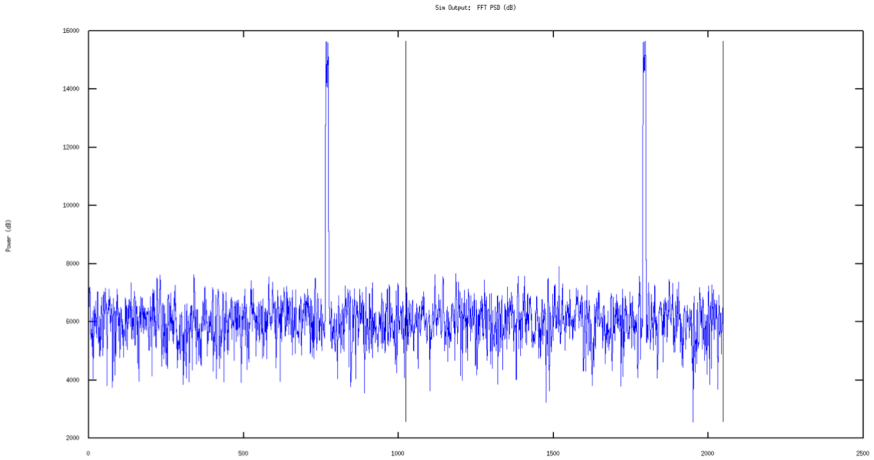
VHDL / Vivado / FINS / Jinja / Python / Octave

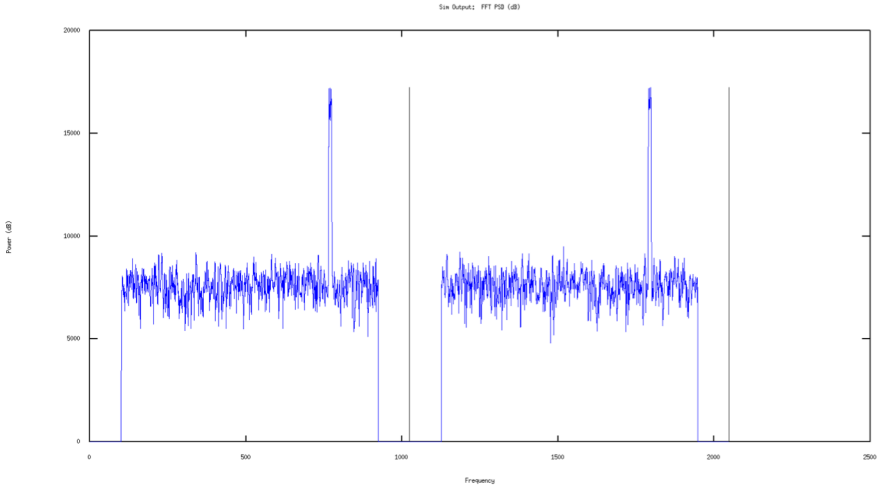
Adam

Projects

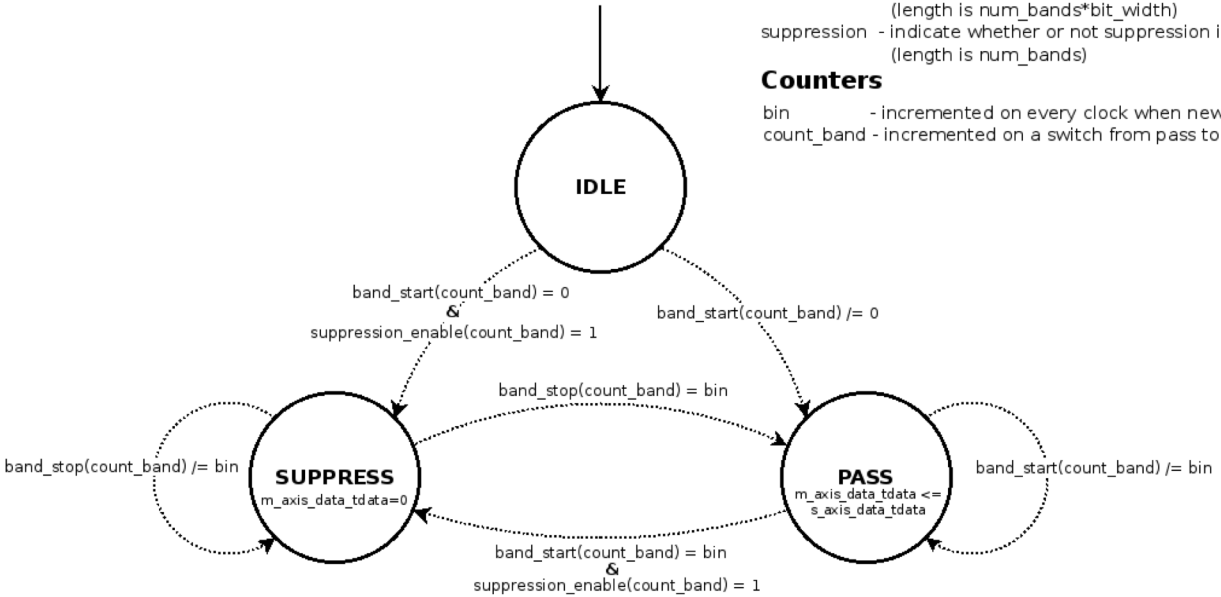
- PSD band suppressor
- FINS register verification
- Window function
- FINS schema verification
- **LFSR**
- **Curve fit**
- **Divider**

PSD Band Suppressor





Band Suppressor
(Moore State Machine)



Standard logic vectors

band_start - concatenated bin numbers for the start of each suppression band (length is num_bands*bit_width)

band_stop - concatenated bin numbers for the end of each suppression band (length is num_bands*bit_width)

suppression - indicate whether or not suppression is enabled for each band (length is num_bands)

Counters

bin - incremented on every clock when new, valid data comes in

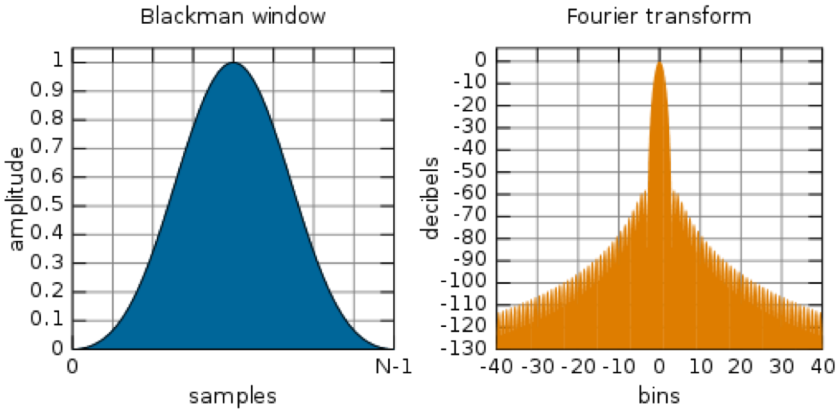
count_band - incremented on a switch from pass to suppress or suppress to pass

FINS Register Verification

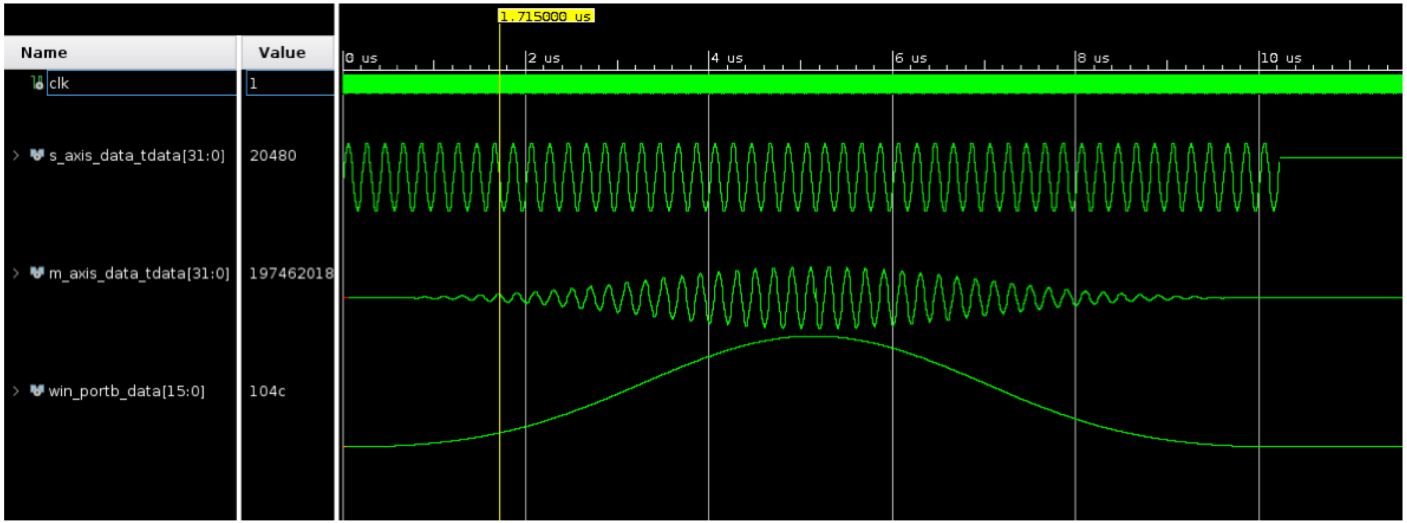
- Created Jinja template for FINS
- Testbench module to write and read from registers
- Integrated into the PSD

Window Function

- Blackman window
- Generic parameter for window source
  - ROM
  - Dual port RAM
- ROM generated from an octave script with blackman window coefficients
- DPR can be accessed using software config bus



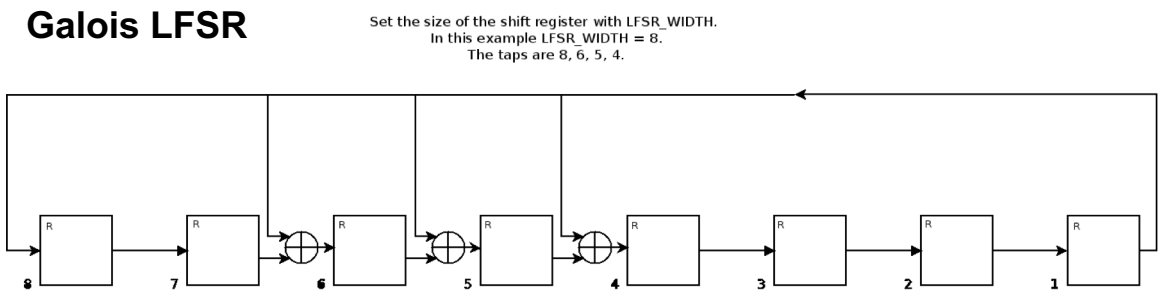
Window Function

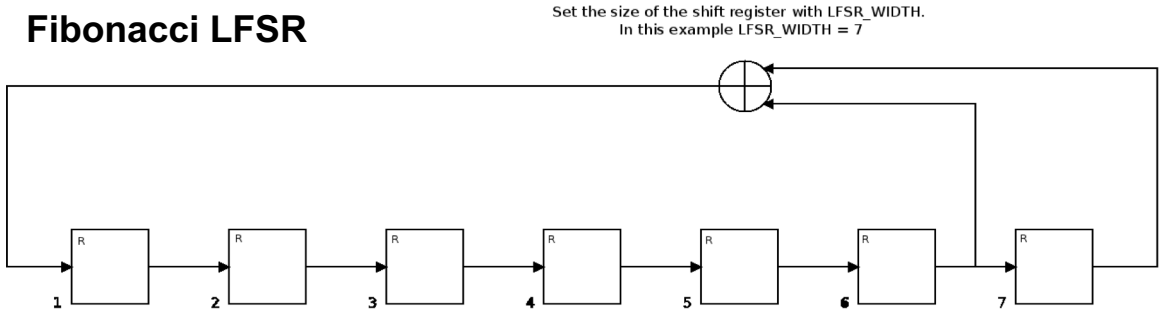


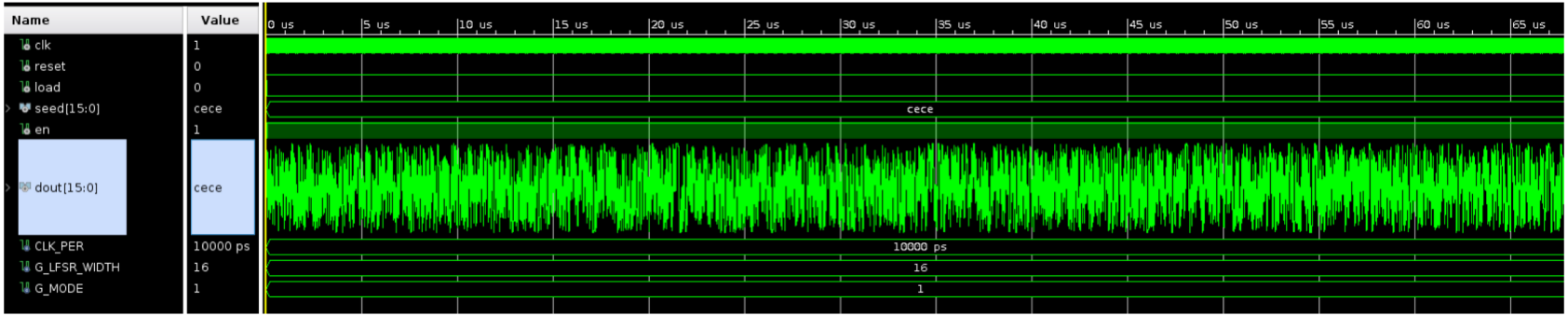
LFSR

Linear Feedback Shift Register

- Next state is a linear function of the current state
- Two types: Galois and Fibonacci
- Certain taps result in a maximal length sequence






Name
Value
clk
1
reset
0
load
0
seed[15:0]
cece
en
1
dout[15:0]
cece
CLK_PER
10000 ps
G_LFSR_WIDTH
16
G_MODE
1
0 us
5 us
10 us
15 us
20 us
25 us
30 us
35 us
40 us
45 us
50 us
55 us
60 us
65 us
cece
10000 ps
16
1

Curve Fit

- Takes 3 input samples from a parabolic curve
- Calculates a refined estimate of the x-coordinate of the parabola's peak

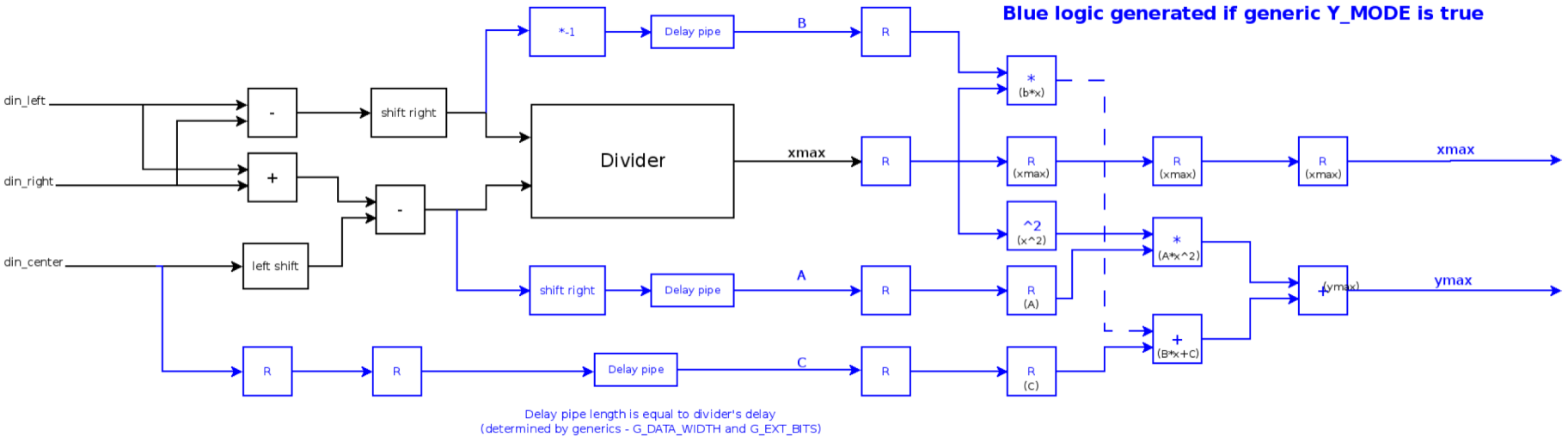
$$ax^2 + bx + c$$

$$yL = a - b + c$$
$$yC = c$$
$$yR = a + b + c$$

$$2ax + b = 0$$
$$x_{max} = -b/2a$$

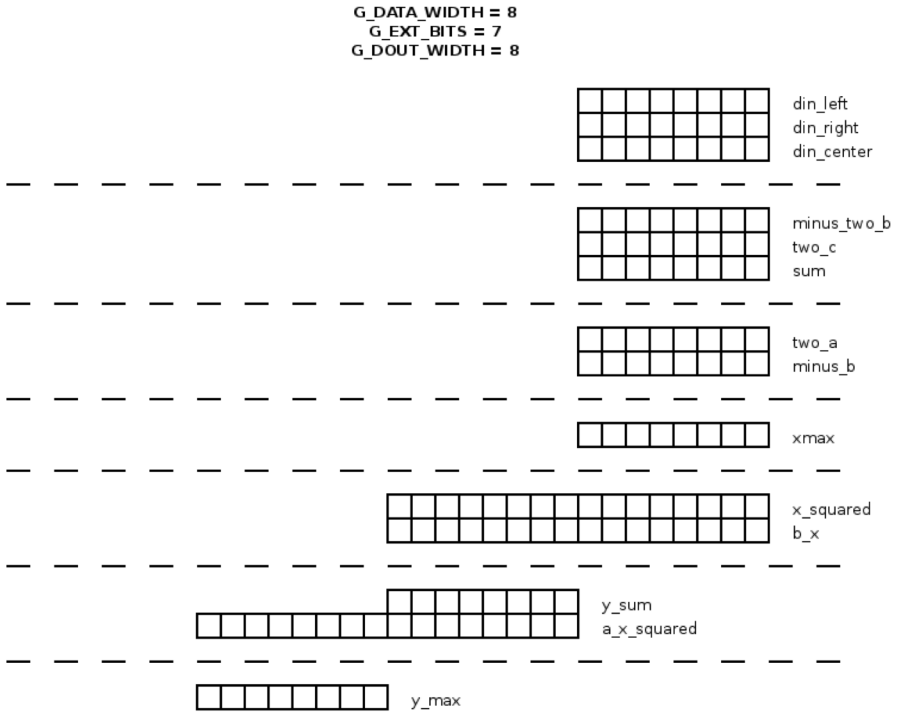
$$-b = (yL - yR)/2$$
$$2a = yL + yR - 2yC$$

Curve Fit


Blue logic generated if generic Y_MODE is true
din_left
din_right
din_center
-
+
-
shift right
left shift
*-1
Delay pipe
B
Divider
xmax
shift right
Delay pipe
A
R
R
Delay pipe
C
R
R
R
R
R
R
R
R
R
R
R
R
*
(b*x)
R
(xmax)
^2
(x^2)
R
(A)
R
(C)
R
(xmax)
R
(xmax)
R
(xmax)
xmax
*
(A*x^2)
+
(B*x+C)
+
(ymax)
ymax
Delay pipe length is equal to divider's delay
(determined by generics - G_DATA_WIDTH and G_EXT_BITS)

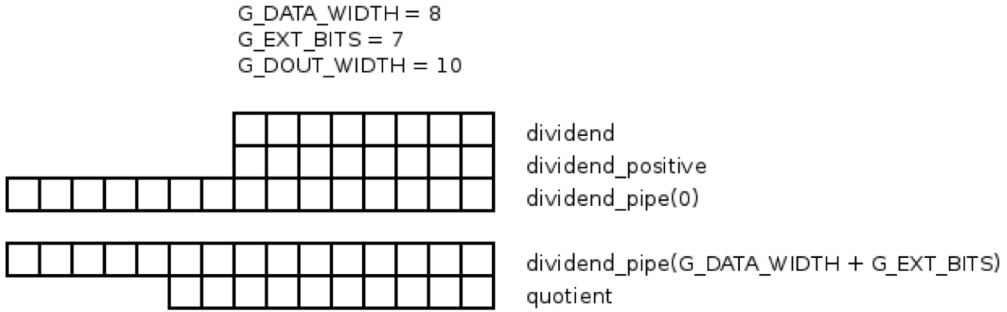
Curve Fit

- Simulation



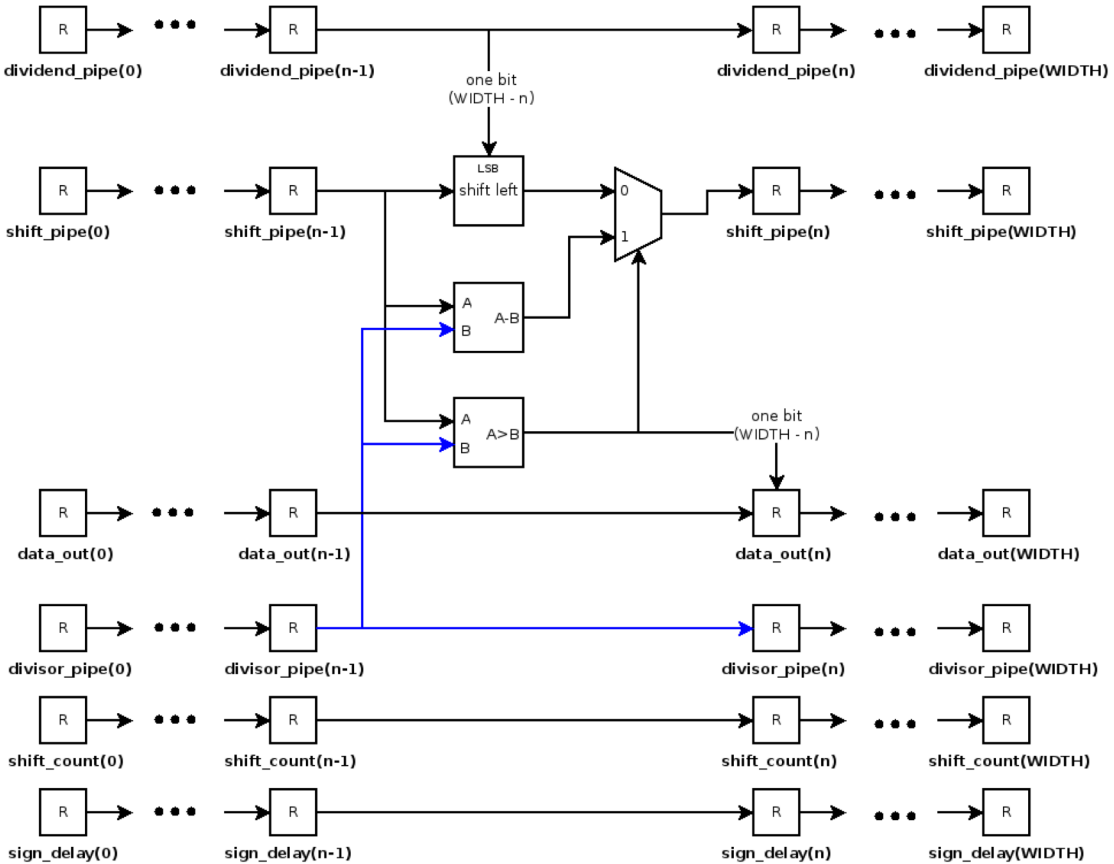
Divider

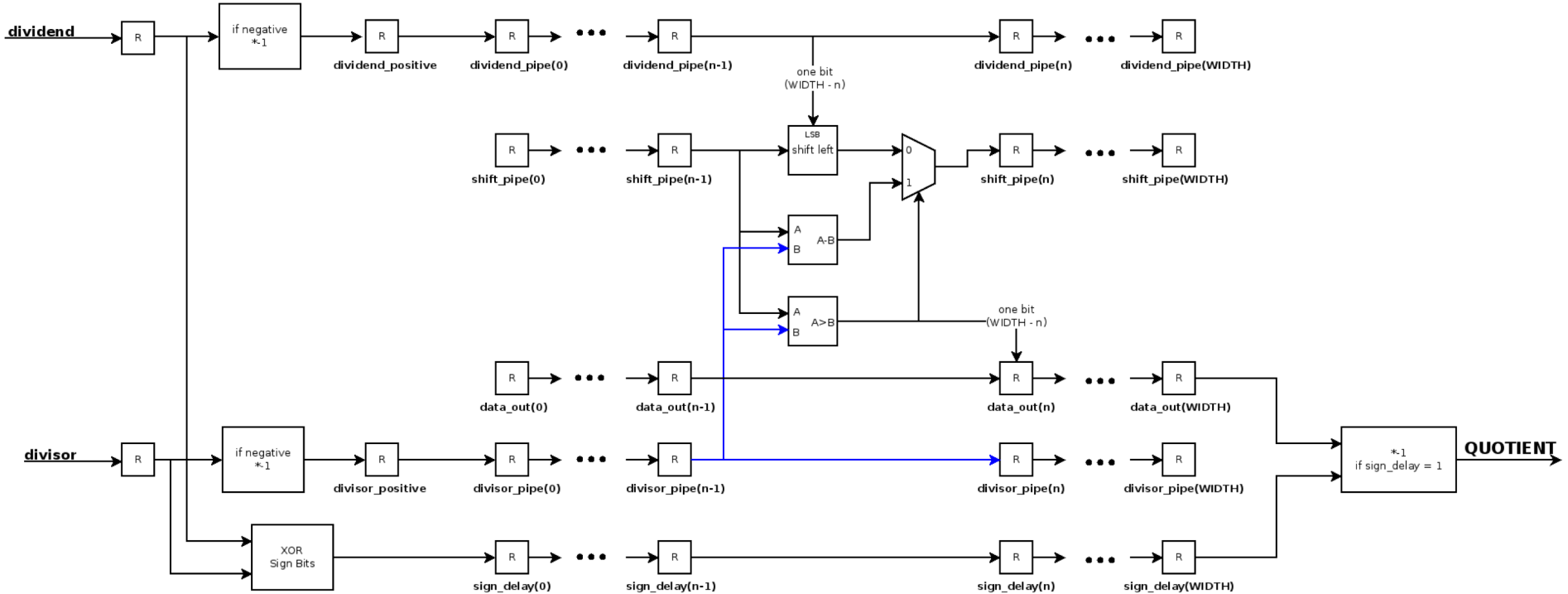
- Pipelined
- Signed binary numbers
- Handle answers less than one



Divider

- Length of pipeline is data width + desired fractional bits
- Each stage writes one bit of the output data
- Works similar to long division by hand




dividend
R
if negative
*-1
R
dividend_positive
R
dividend_pipe(0)
R
dividend_pipe(n-1)
one bit
(WIDTH - n)
R
dividend_pipe(n)
R
dividend_pipe(WIDTH)
R
shift_pipe(0)
R
shift_pipe(n-1)
LSB
shift left
0
1
R
shift_pipe(n)
R
shift_pipe(WIDTH)
A
B
A-B
A
B
A>B
one bit
(WIDTH - n)
R
data_out(0)
R
data_out(n-1)
R
data_out(n)
R
data_out(WIDTH)
divisor
R
if negative
*-1
R
divisor_positive
R
divisor_pipe(0)
R
divisor_pipe(n-1)
R
divisor_pipe(n)
R
divisor_pipe(WIDTH)
XOR
Sign Bits
R
sign_delay(0)
R
sign_delay(n-1)
R
sign_delay(n)
R
sign_delay(WIDTH)
*-1
if sign_delay = 1
QUOTIENT