A Floating Point Divider for RC Systems

Date: July 24, 2000

By: R. Jennell Rouse

Advisor: Dr. Clay Gloster

Outline

- An Overview of Floating Point Arithmetic
- IEEE Floating Point Formats
- Examples of Floating Point Division
- Examples of Floating Point Addition
- Implementation of a 32-bit Floating Point Divider
- Conclusions

Motivation

- This design is a part of the N.C. State R.A.R.E. project.
- R.A.R.E. stands for a Remote Adaptive Computing Resource on the Internet.
- With R.A.R.E., applications can execute an order of magnitude 10 times faster than on typical desktop computers.

Motivation

- Many applications use floating point division, hence there is a need to develop a floating point divider for use in R.A.R.E.
- The circuit is modeled using the Very High Speed Integrated Circuits Hardware Description Language(VHDL).
- There is a need to design a floating point divider for use in floating point operations.

Floating Point Numbers

- Floating point number representations allow us to use real numbers on a computer.
- Floating point numbers consist of a sign, exponent, mantissa, and base.

Floating Point Divider

- Floating point number representations allow us to use real numbers on a computer
- Floating point numbers consist of a sign, mantissa, base, and exponent
  - $+10.34 \times 10^{32}$
  - $+1.034 \times 10^{33}$
- Since each floating point number can be represented an infinite number of ways, we normalize the number.
  - $+1.034 \times 10^{33}$

Floating Point Numbers
IEEE Single Precision

1 bit	8 bits	23 bits
sign	exponent	mantissa

Total 32 bits

- Most computers support single (32-bit) precision formats
- Single precision format can express numbers from (-3.4 E 38 to 3.4 E 38)

Floating Point Addition (Complex)

To add two floating point numbers we:

- Align exponents while adjusting the mantissa of one operand
- Add resulting mantissas
- Compute the sign of the result based on the sign and magnitude of the two operands
- Normalize the result

Floating Point Addition

Example:

$$
\begin{array}{rr}
 & -1.25 \times 10^{2} \\
+ & \underline{+2.50 \times 10^{-1}} \\
 & -1.2475 \times 10^{2}
\end{array}
$$

Floating Point Addition

0.5 + 0.5 = ????
A + B = Q

A = 0.500000
A = 3f000000
Sign A: = 0
Exponent A: = 01111110
Mantissa A: = 00000000000000000000000
Real Exponent = -1
Mantissa = 0

B = 0.500000
B = 3f000000
Sign B: = 0
Exponent B: = 01111110
Mantissa B: = 00000000000000000000000
Real Exponent = -1
Mantissa = 0

Floating Point Addition

0.5 + 0.5 = 1.0
A + B = Q

Q = 1.000000
Q = 3f800000
Sign Q: = 0
Exponent Q: = 01111111
Mantissa Q: = 00000000000000000000000
Real Exponent = 0
Mantissa = 0

Floating Point Division

- To divide two floating point numbers we:
- Divide divisor mantissa from the dividend mantissa
- Subtract the exponent of the divisor from the dividend
- Compute the sign of the result based on the sign of the two operands
- Normalize the result

Floating Point Division

Example:

$$\begin{array}{rr} & -1.25 \times 10^{2} \\ \text{divided by} & \underline{+2.50 \times 10^{-1}} \\ & -3.125 \times 10^{2} \end{array}$$

Floating Point Division

31.367 / -0.34= ????
A + B = Q

A = 31.367
A = 41FAEF9E
Sign A: = 0
Exponent A: = 10000011
Mantissa A: = 11110101110111110011110
Real Exponent = 4
Mantissa = 8056734

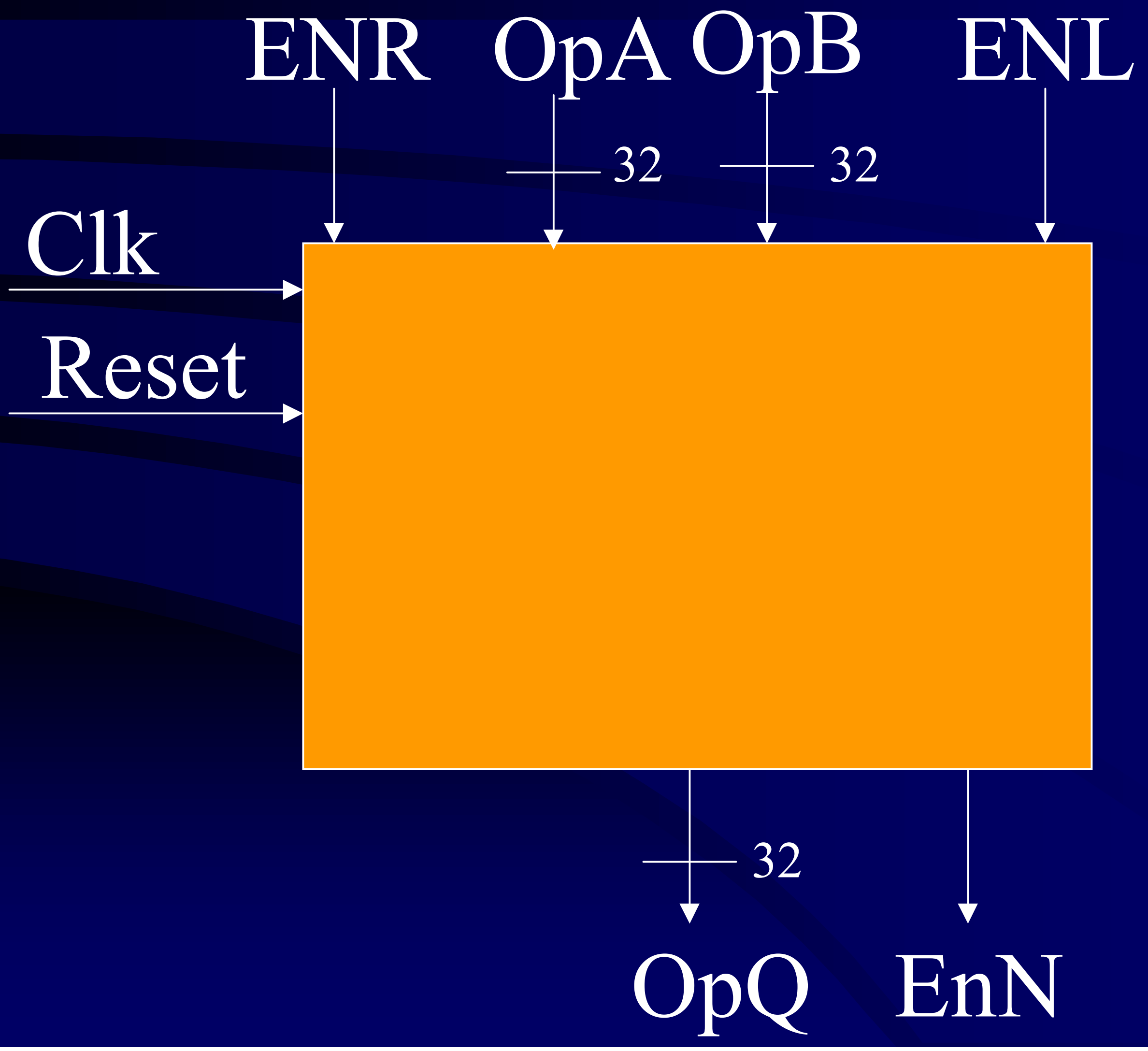
B = -0.34
B = BEAE147B
Sign B: = 1
Exponent B: = 01111101
Mantissa B: = 01011100001010001111011
Real Exponent = -2
Mantissa = 0

Floating Point Division (Contd.)

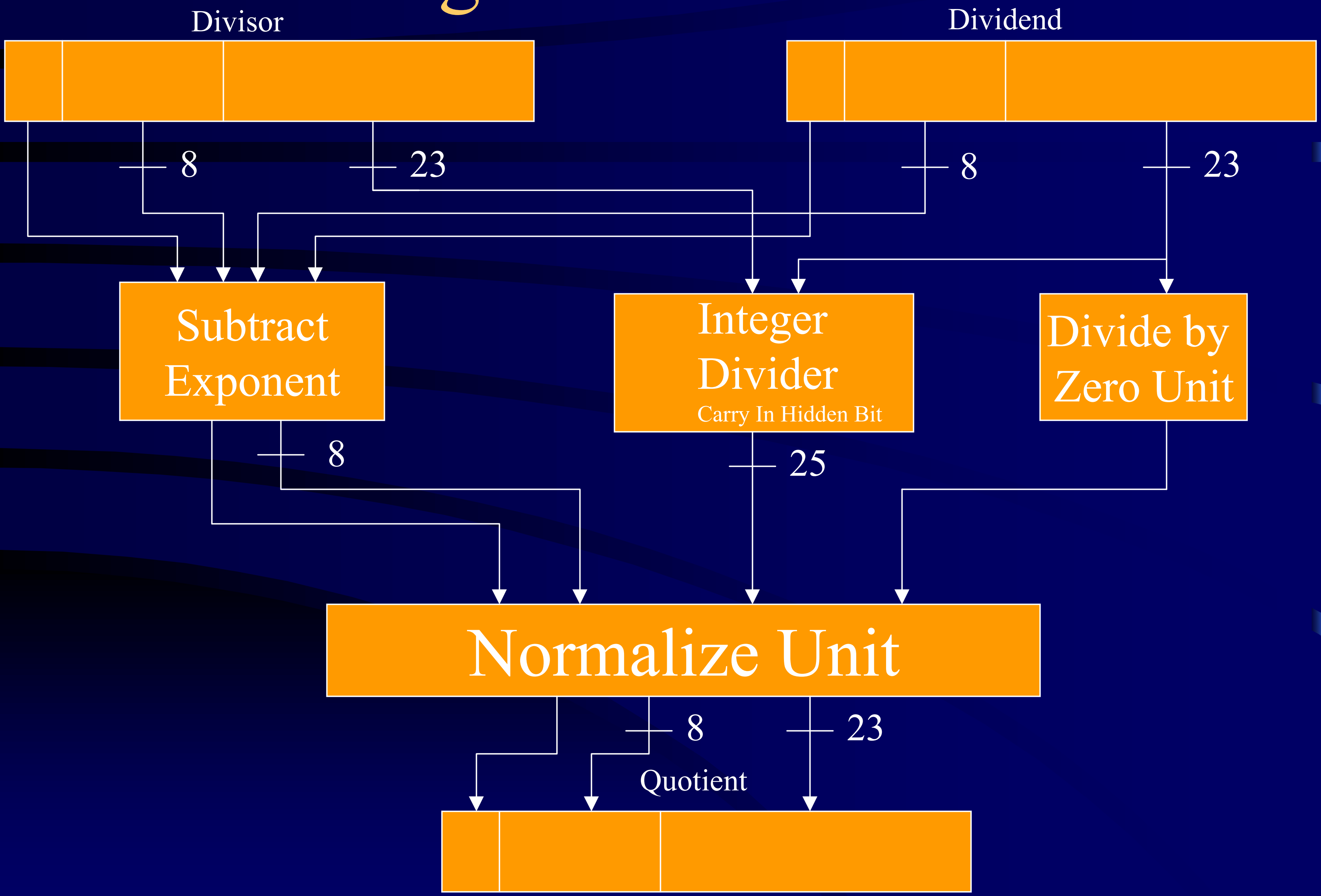
31.367 / -0.34= ????
A + B =-92.255823529

Q = -92.255823529
Q = C2B88303
Sign Q: = 1
Exponent Q: = 10000101
Mantissa Q: = 01110001000001100000011
Real Exponent = 6
Mantissa = 3703555

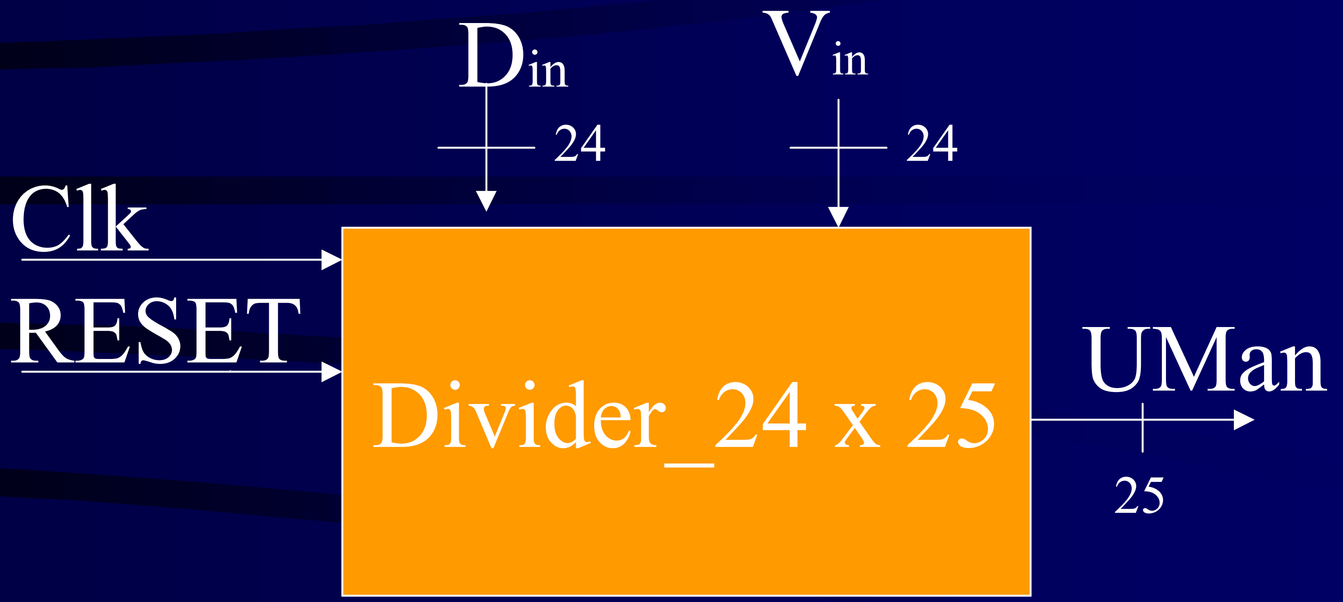
Floating Point Divider Core



Floating Point Divider Core

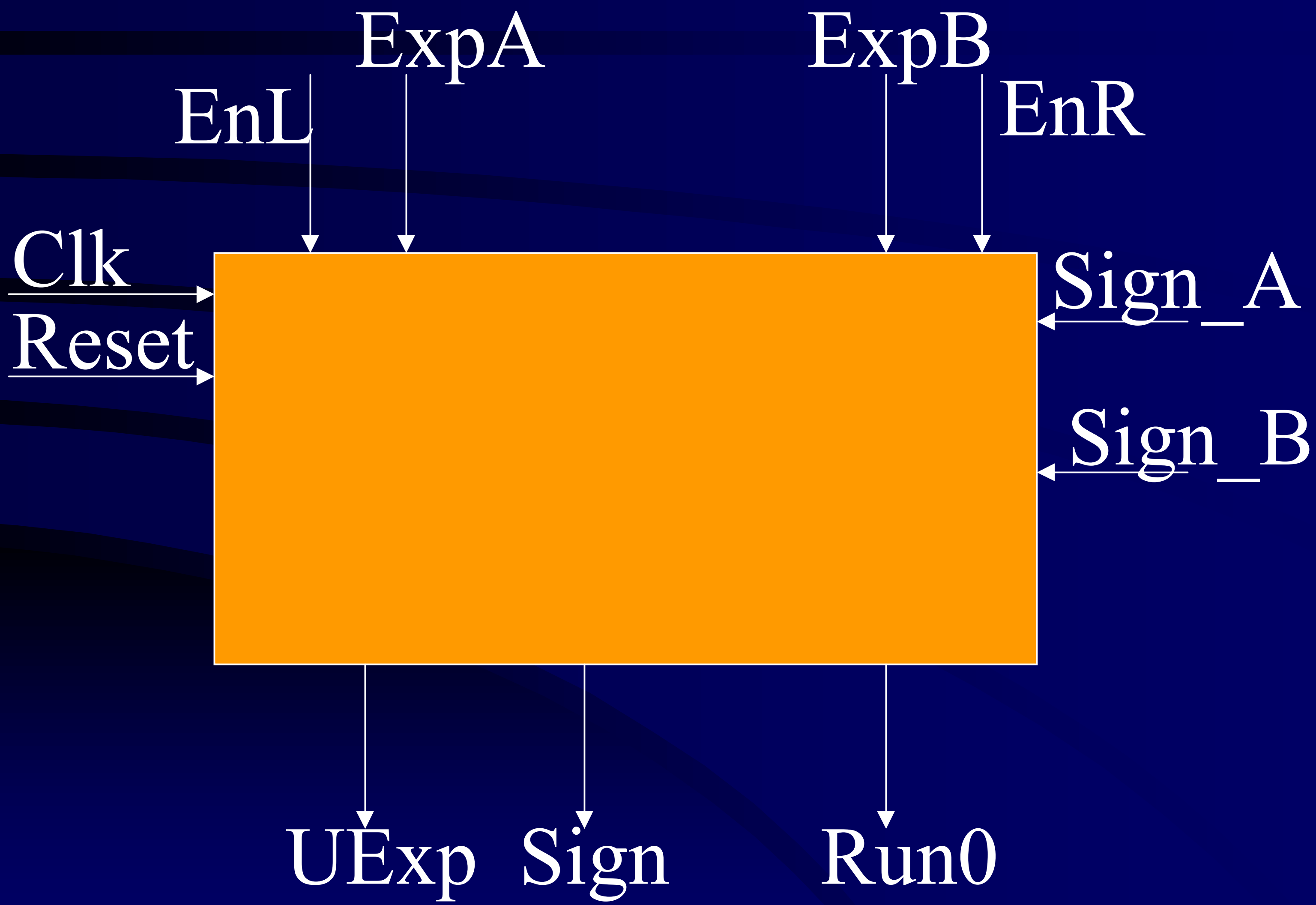


Integer Divider

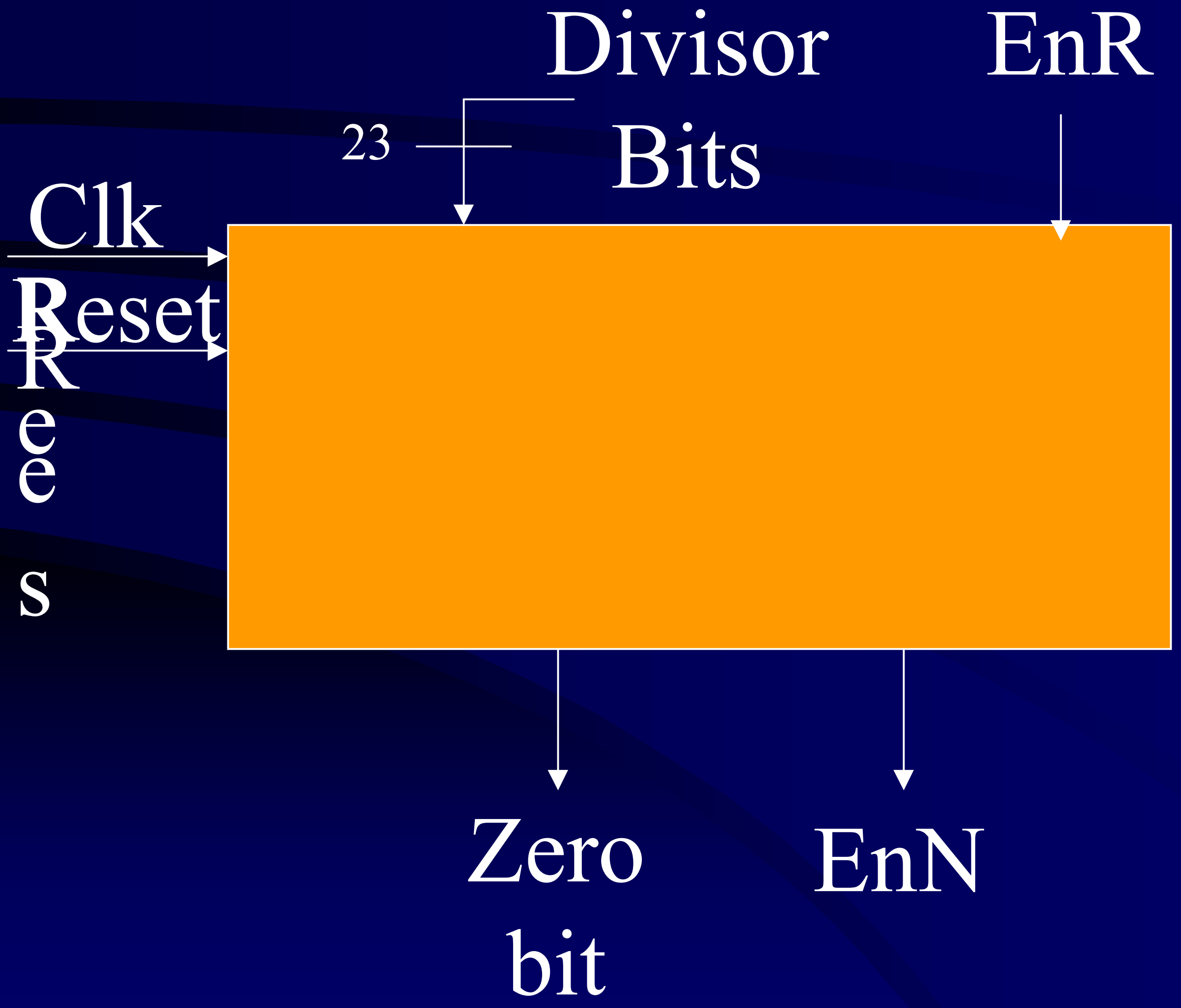


- The integer divider has a two 24 bit inputs, which are the mantissas.
- There is also a hidden CarryIn Bit into the integer divider.

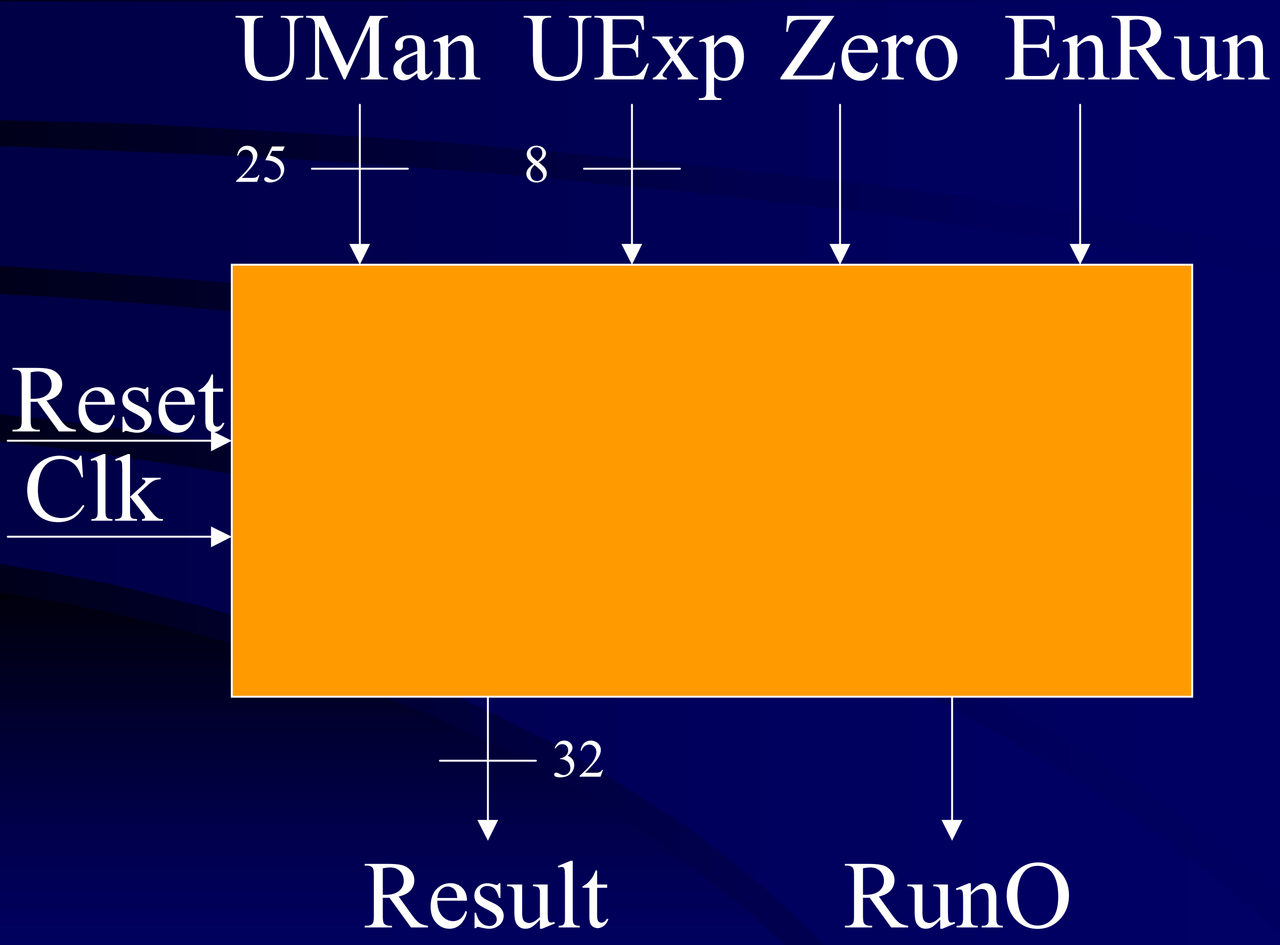
Subtract Exponent



Divide by Zero Unit

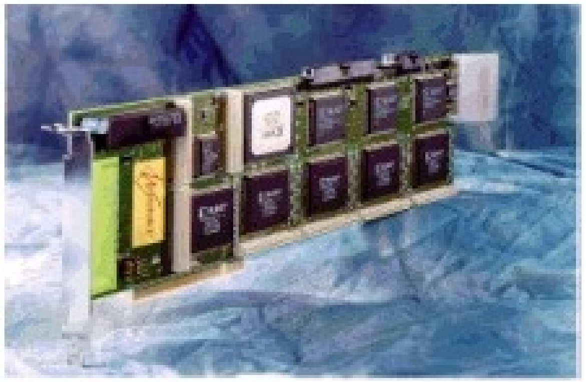


Normalize Unit



Field Programmable Gate Array FPGA

- We mapped the floating point divider, which was described in VHDL, onto a FPGA.



Reconfigurable Board

- Our current RC system utilizes an FPGA coprocessor board with five user programmable PEs.
- Each PE includes a Xilinx 4044 FPGA containing 1600 CLBs or the equivalent of up to 80K logic gates.
- Total CLBs / equivalent logic gates available (8000/400K)

Conclusions

- The implementation of a floating point division unit in an FPGA will facilitate the dramatic speedup of applications which utilize recursive floating point divisions.
- LU Decomposition utilizes forward and backward substitution in order to solve a system of simultaneous equations.

Conclusions (Cont.)

- For a system which has hundreds of unknowns division operations are utilized thousands of incidences.
- The implementation of a floating point division for reconfigurable computing can drastically speedup mathematical operations such as LU Decomposition.