Efficient Floating-Point Logarithm Unit for FPGAs

Nikolaos Alachiotis, Alexandros Stamatakis

The Exelixis Lab,
Dept. of Computer Science,
TUM, Munich, Germany

PRESENTATION OVERVIEW

- Introduction
- Approximation Strategy
- Reconfigurable Architecture
- Performance Evaluation
- Conclusion and Future Work

INTRODUCTION

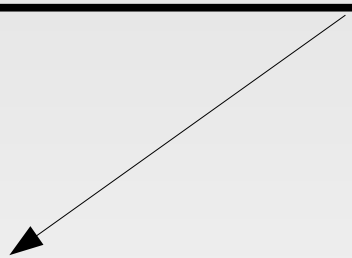
- The Project:

Design of HW accelerators for Phylogenetic Inference Programs

INTRODUCTION

- The Project:

Design of HW accelerators for Phylogenetic Inference Programs



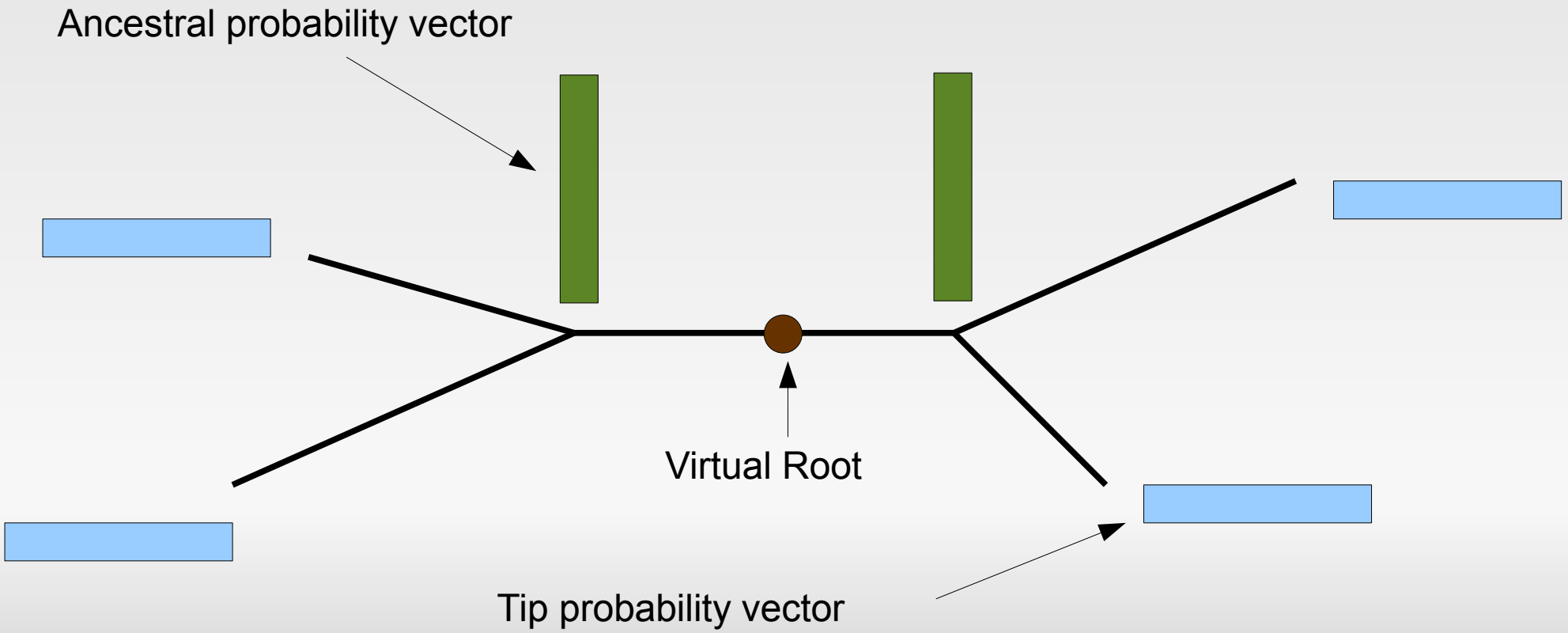
Calculation of evolutionary relationships between organisms

core function: the Phylogenetic Likelihood Function

INTRODUCTION

- The Project:

Design of HW accelerators for Phylogenetic Inference Programs



INTRODUCTION

- The Project:

Design of HW accelerators for Phylogenetic Inference Programs

  - The Phylogenetic Likelihood Function:

    85% of total execution time

  - Log-Likelihood Scores:

    2% of total execution time

INTRODUCTION

- The Project:

Design of HW accelerators for Phylogenetic Inference Programs

  - The Phylogenetic Likelihood Function:

    85% of total execution time

  - Log-Likelihood Scores:

    2% of total execution time

Need for a **resource-efficient** logarithm function

APPROXIMATION STRATEGY

"A Hardware-Independent Fast Logarithm Approximation with Adjustable Accuracy," by O. Vinyals, G. Friedland. *Tenth IEEE Inter. Symposium on Multimedia*, pp. 61–65, 2008.

Open source C implementation: ICSILog 0.6 BETA

Floating-Point number in IEEE-754 standard

$$\text{Number} = \text{sign} * 2^{\text{exponent}} * \text{mantissa}$$

APPROXIMATION STRATEGY

$$\text{Number} = \text{sign} * 2^{\text{exponent}} * \text{mantissa}$$

Logarithm defined only for positive values

$$\text{LOG(Number)} = \text{LOG}\left(2^{\textit{exponent}} * \textit{mantissa}\right)$$

Multiplicative property of logarithm

$$= \text{LOG}\left(2^{\textit{exponent}}\right) + \text{LOG}(\textit{mantissa})$$

$$= \textit{exponent} * \text{LOG}(2) + \text{LOG}(\textit{mantissa})$$

APPROXIMATION STRATEGY

$$\text{Number} = \text{sign} * 2^{\text{exponent}} * \text{mantissa}$$

Logarithm defined only for positive values

$$\text{LOG(Number)} = \text{LOG}\left(2^{\textit{exponent}} * \textit{mantissa} \right)$$

Multiplicative property of logarithm

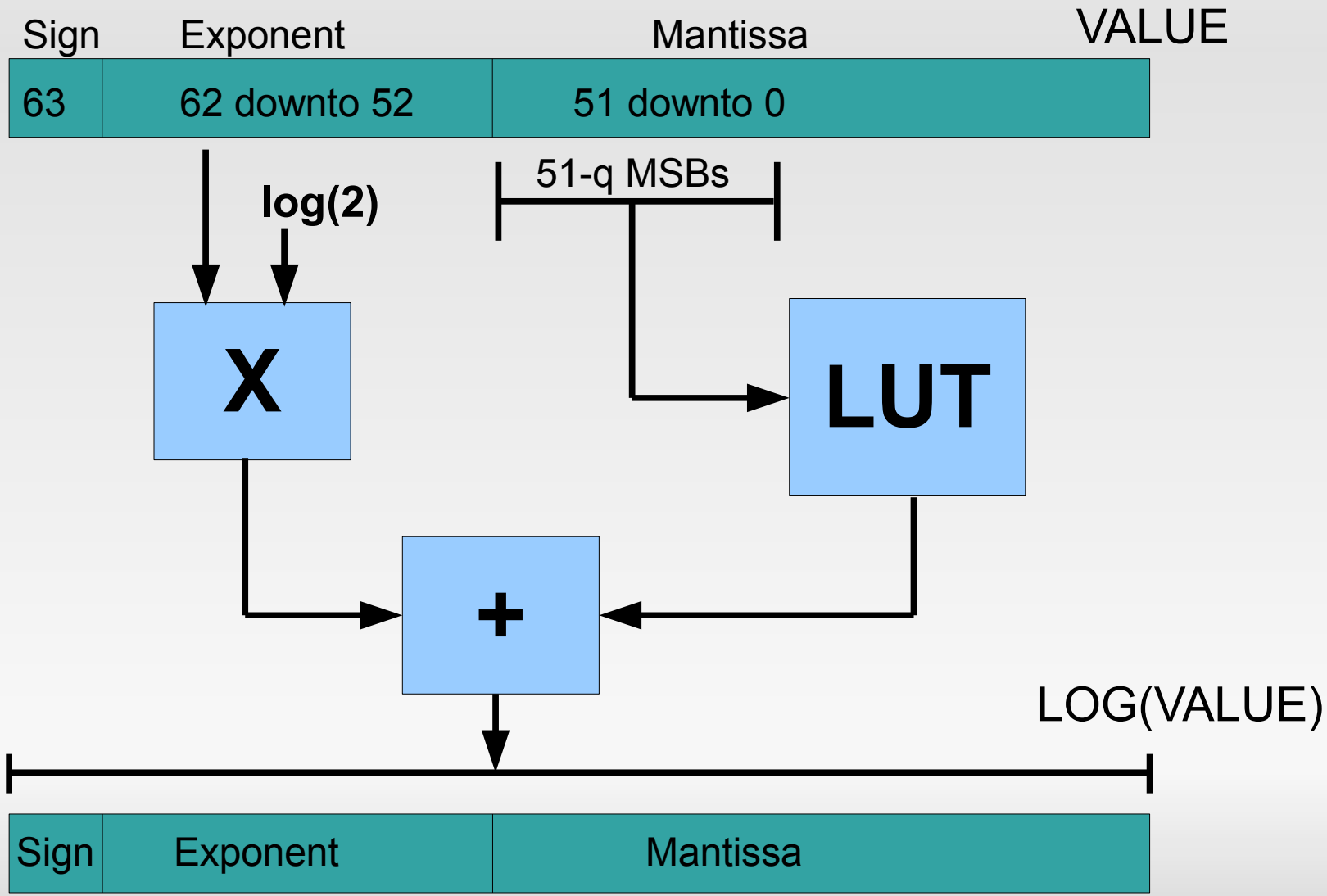
$$= \text{LOG}\left(2^{\textit{exponent}} \right) + \text{LOG}(\textit{mantissa})$$

$$= \textit{exponent} * \text{LOG}(2) + \boxed{\text{LOG}(\textit{mantissa})}$$

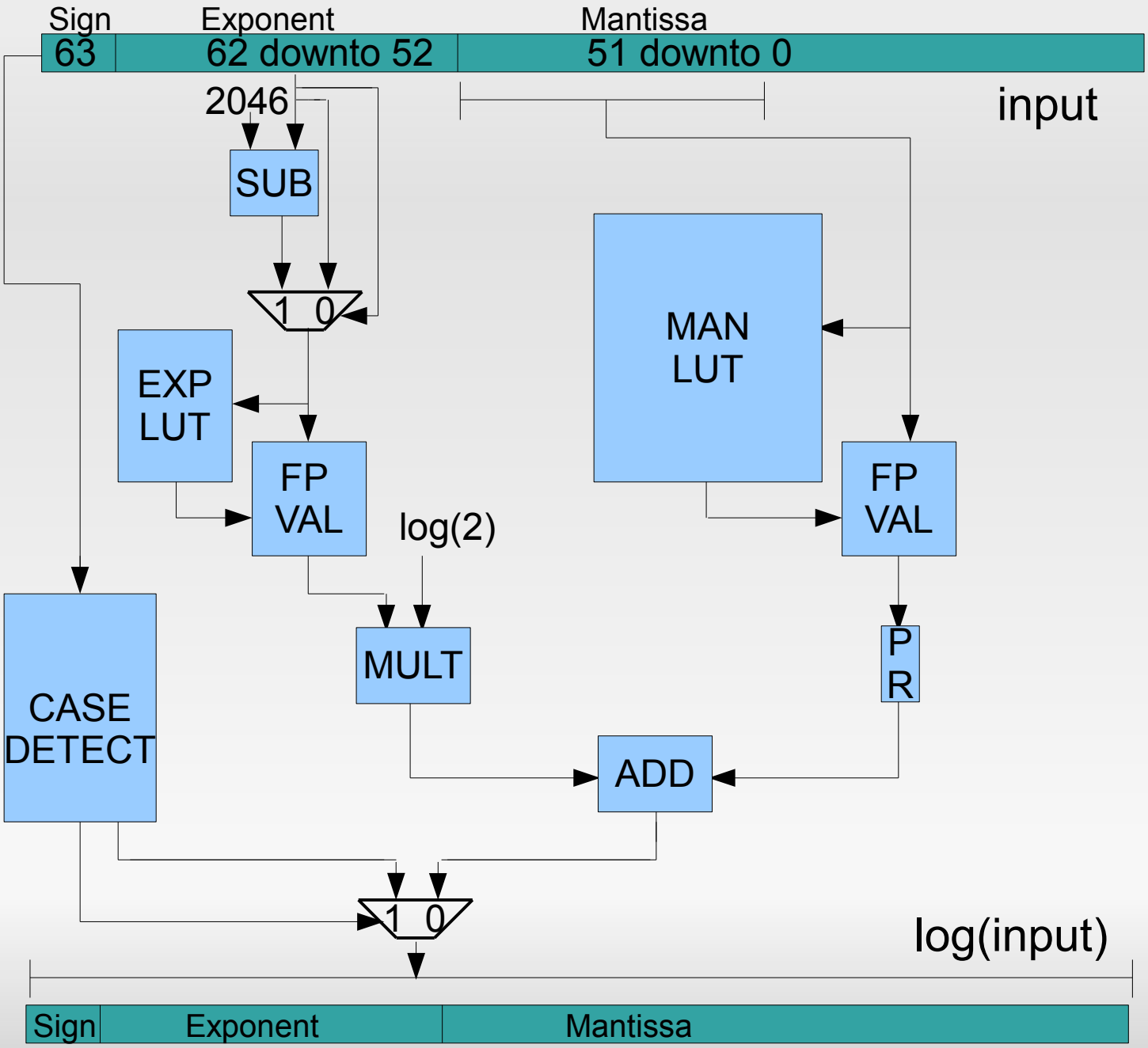
Lookup Table

APPROXIMATION STRATEGY

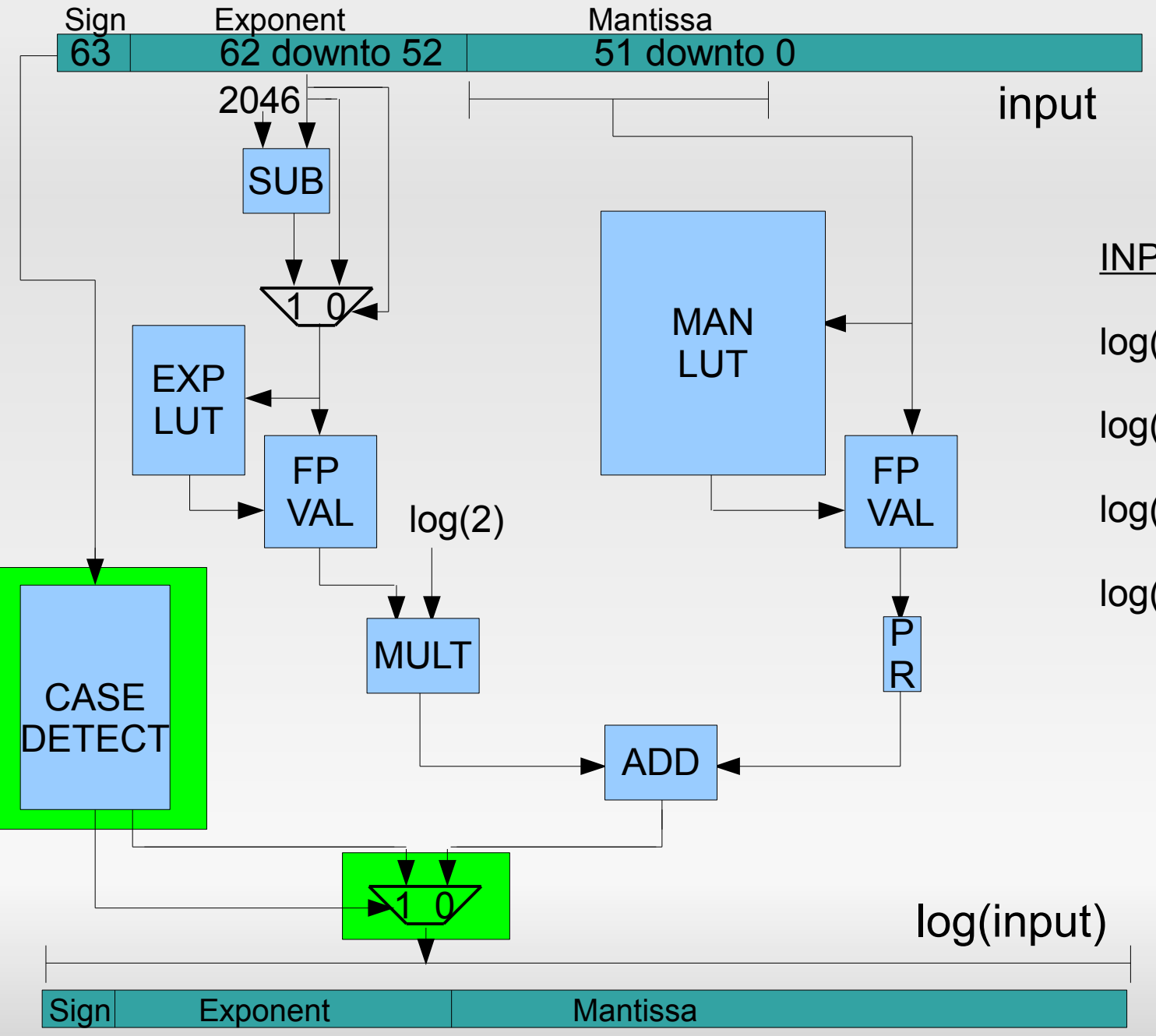
$$LOG(Value) = exponent * LOG(2) + LOG(mantissa)$$



LOGARITHM APPROXIMATION UNIT (LAU) ARCHITECTURE



LOGARITHM APPROXIMATION UNIT (LAU) ARCHITECTURE



INPUT CASE DETECTION

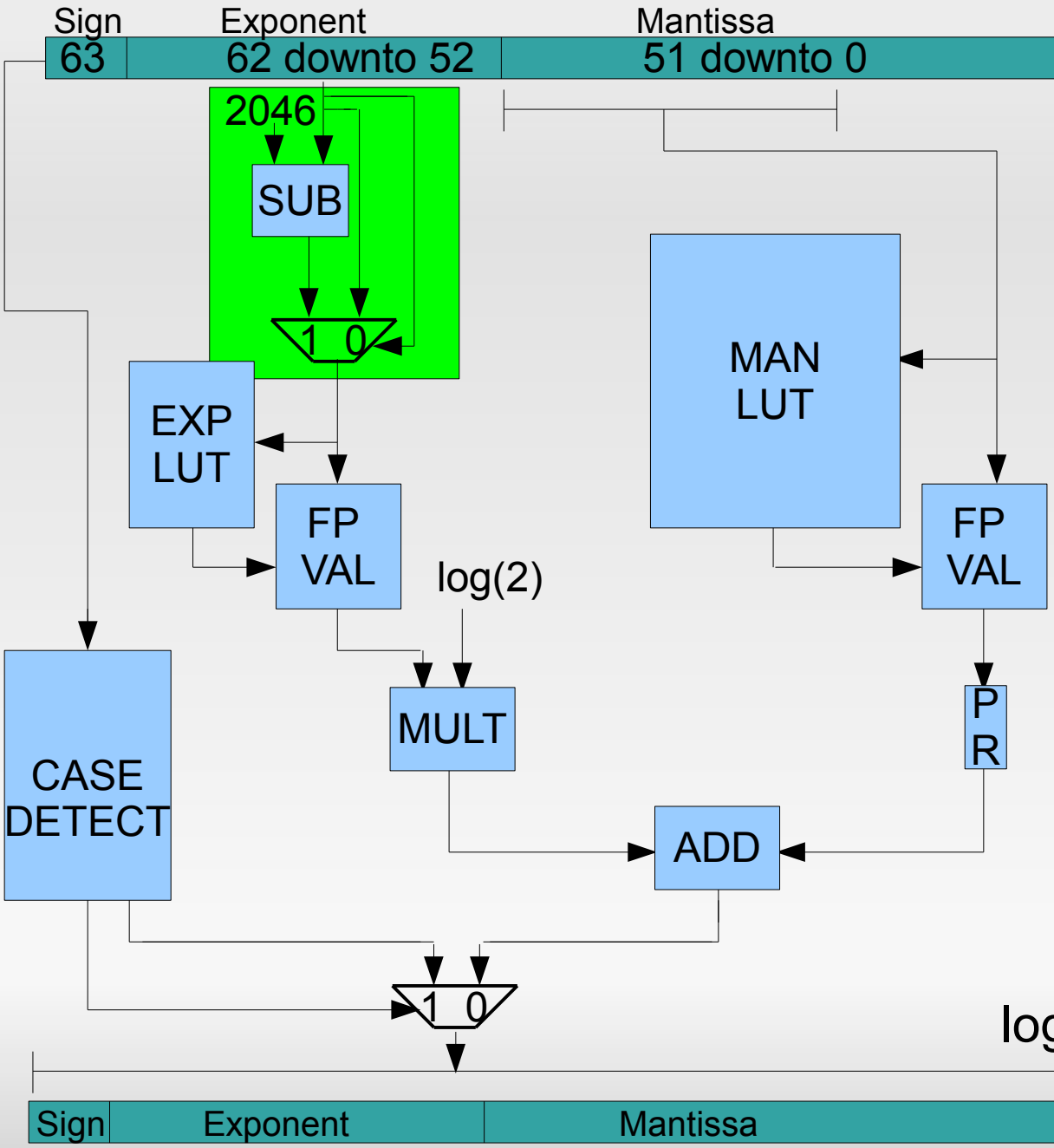
log(Negative number)=nan

log(Nan)=nan

log(Inf)=Inf

log(-Inf)=nan

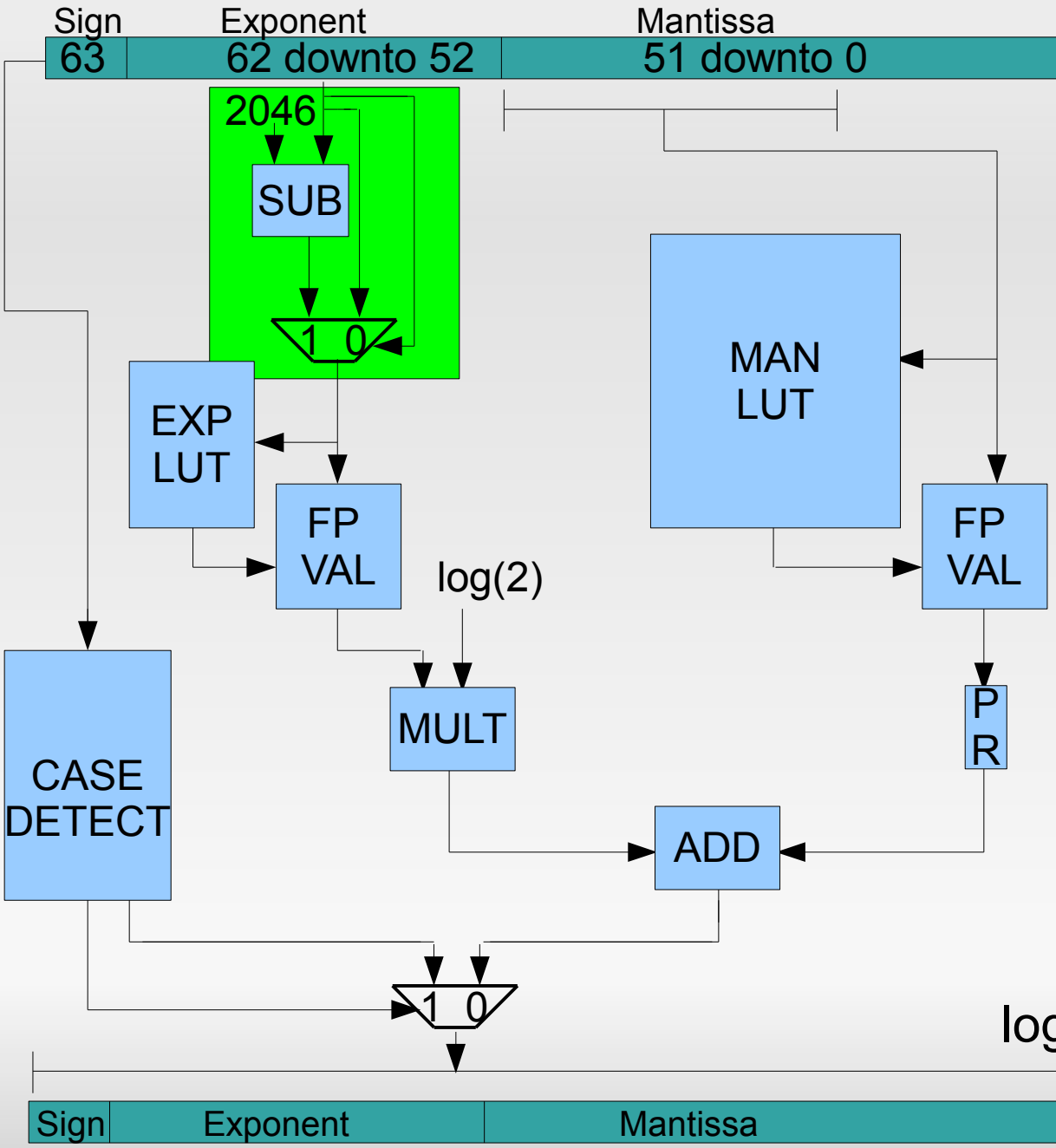
LOGARITHM APPROXIMATION UNIT (LAU) ARCHITECTURE



CREATE THE EXPLUT INDEX

Decimal value	Exponent
0	-1023
1	-1022
…	…
1022	-1
1023	0
1024	1
…	...
2046	1023

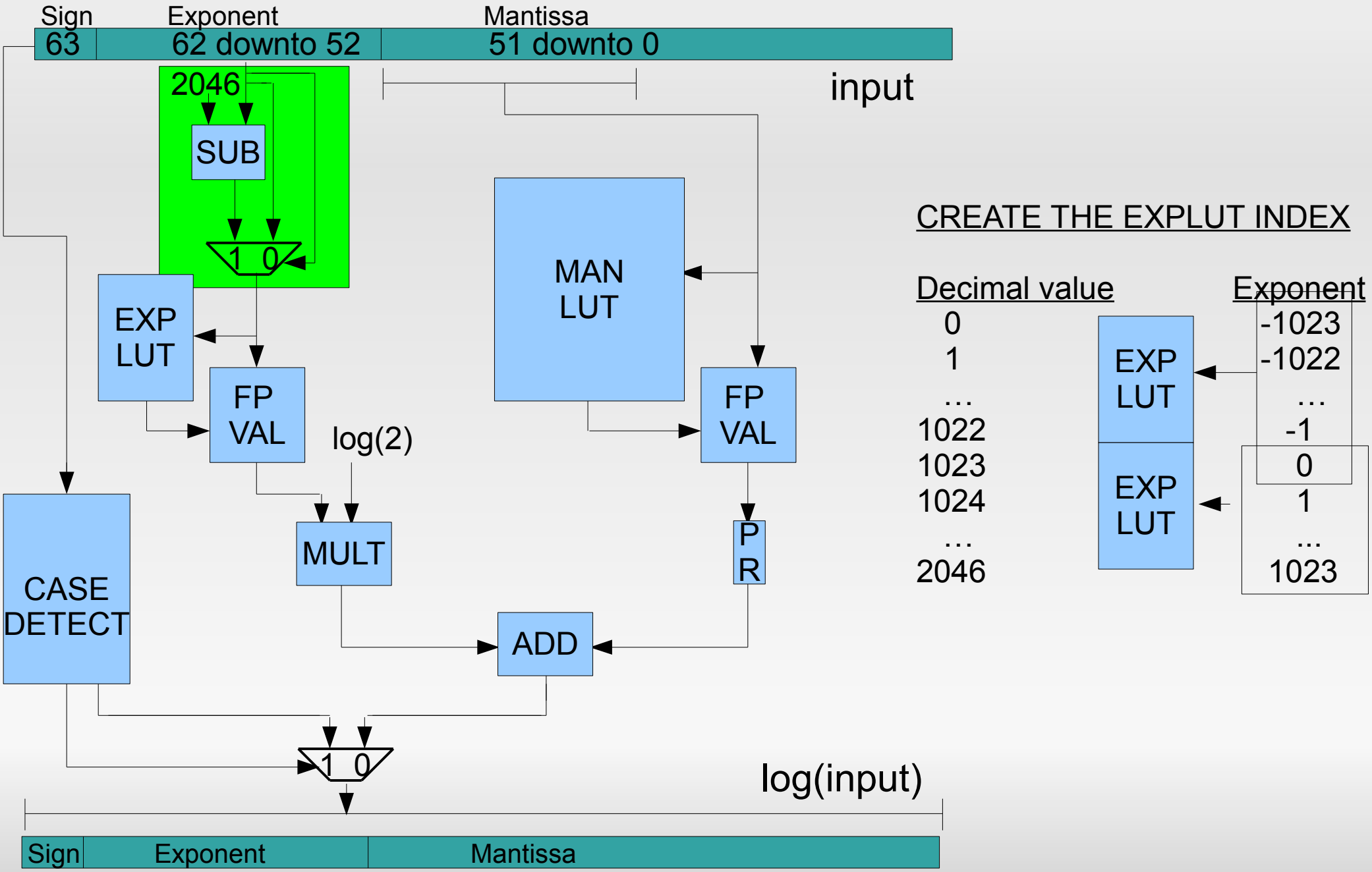
LOGARITHM APPROXIMATION UNIT (LAU) ARCHITECTURE



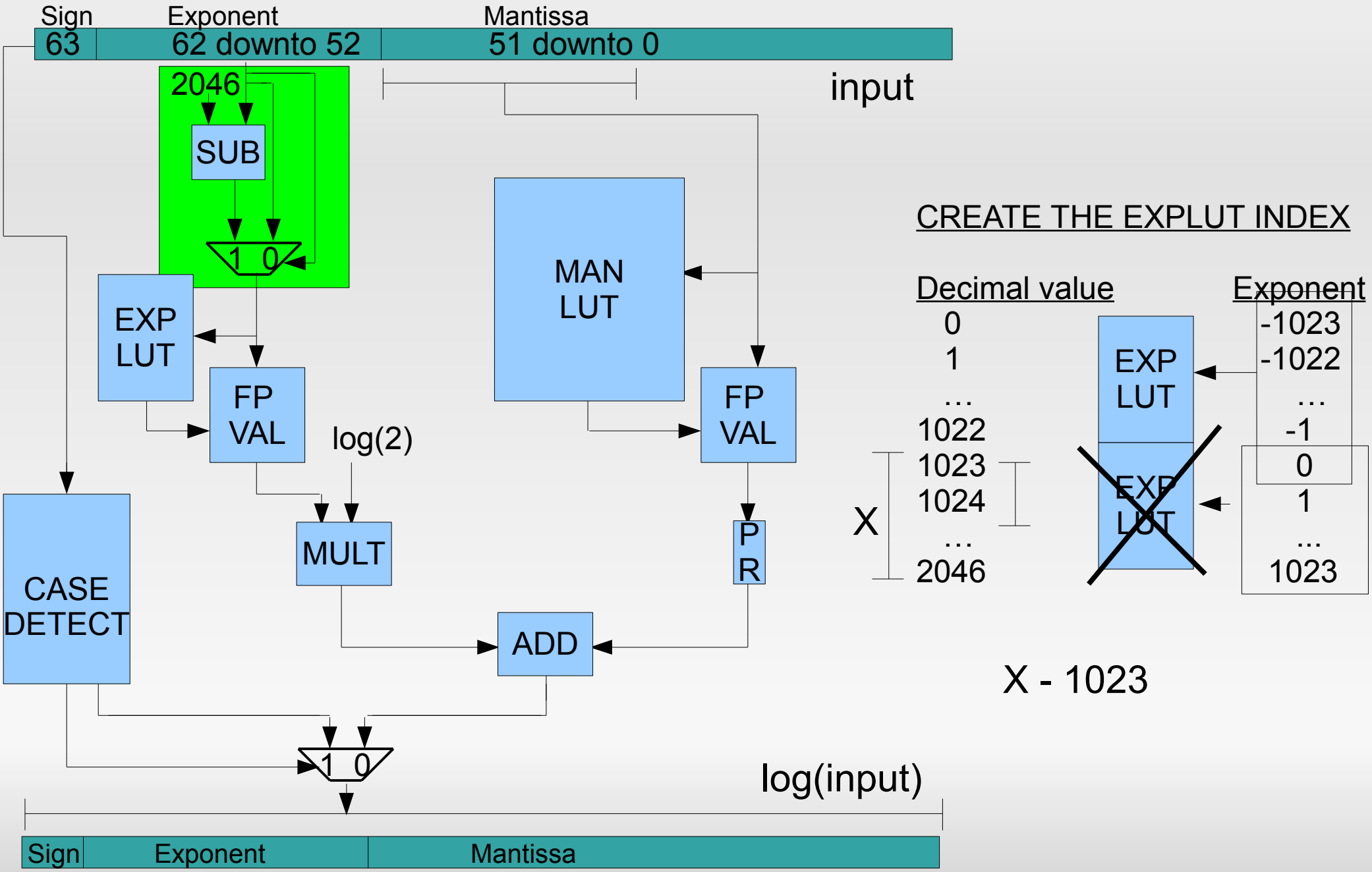
CREATE THE EXPLUT INDEX

Decimal value	Exponent
0	-1023
1	-1022
…	…
1022	-1
1023	0
1024	1
…	…
2046	1023

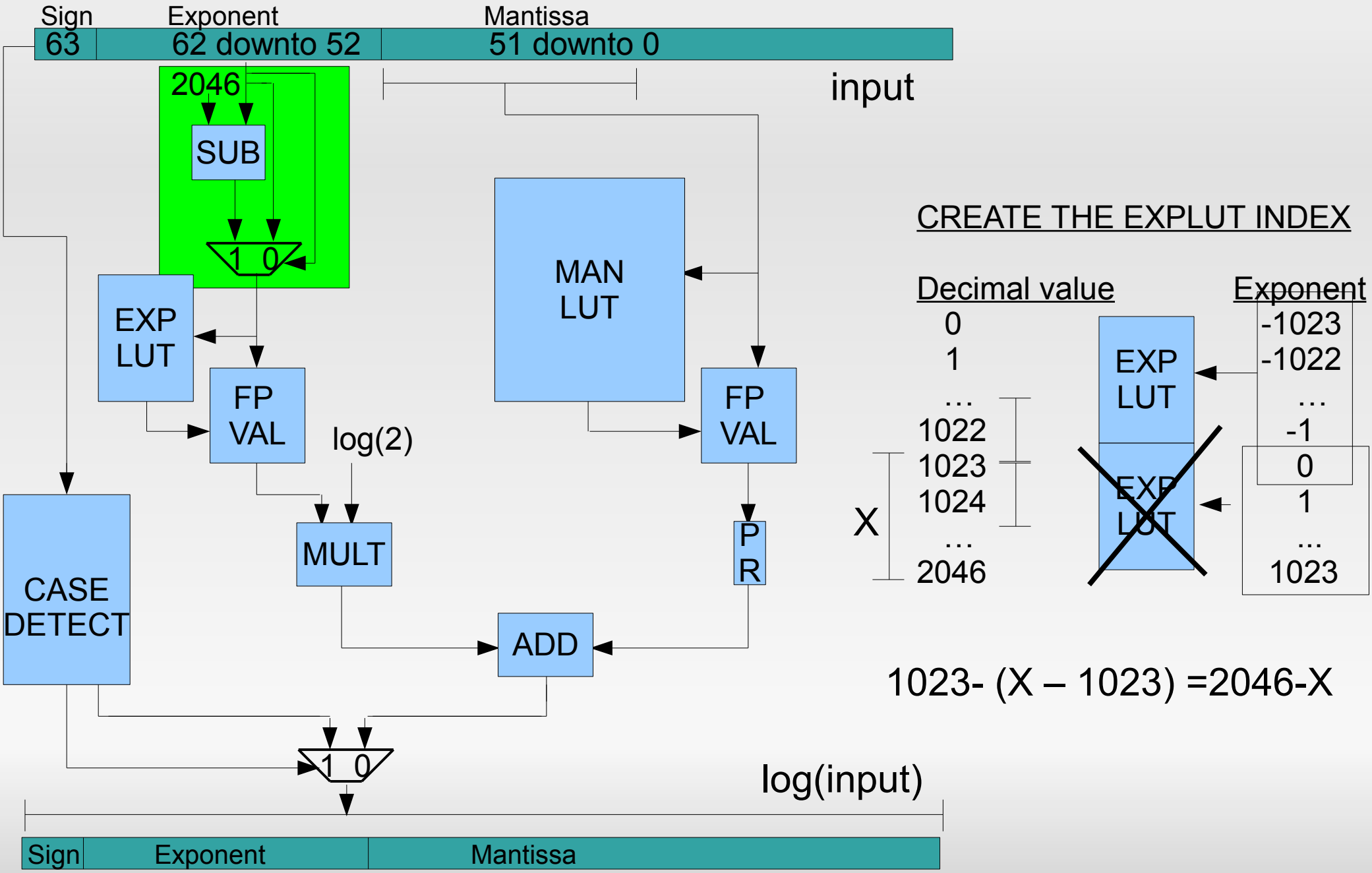
LOGARITHM APPROXIMATION UNIT (LAU) ARCHITECTURE



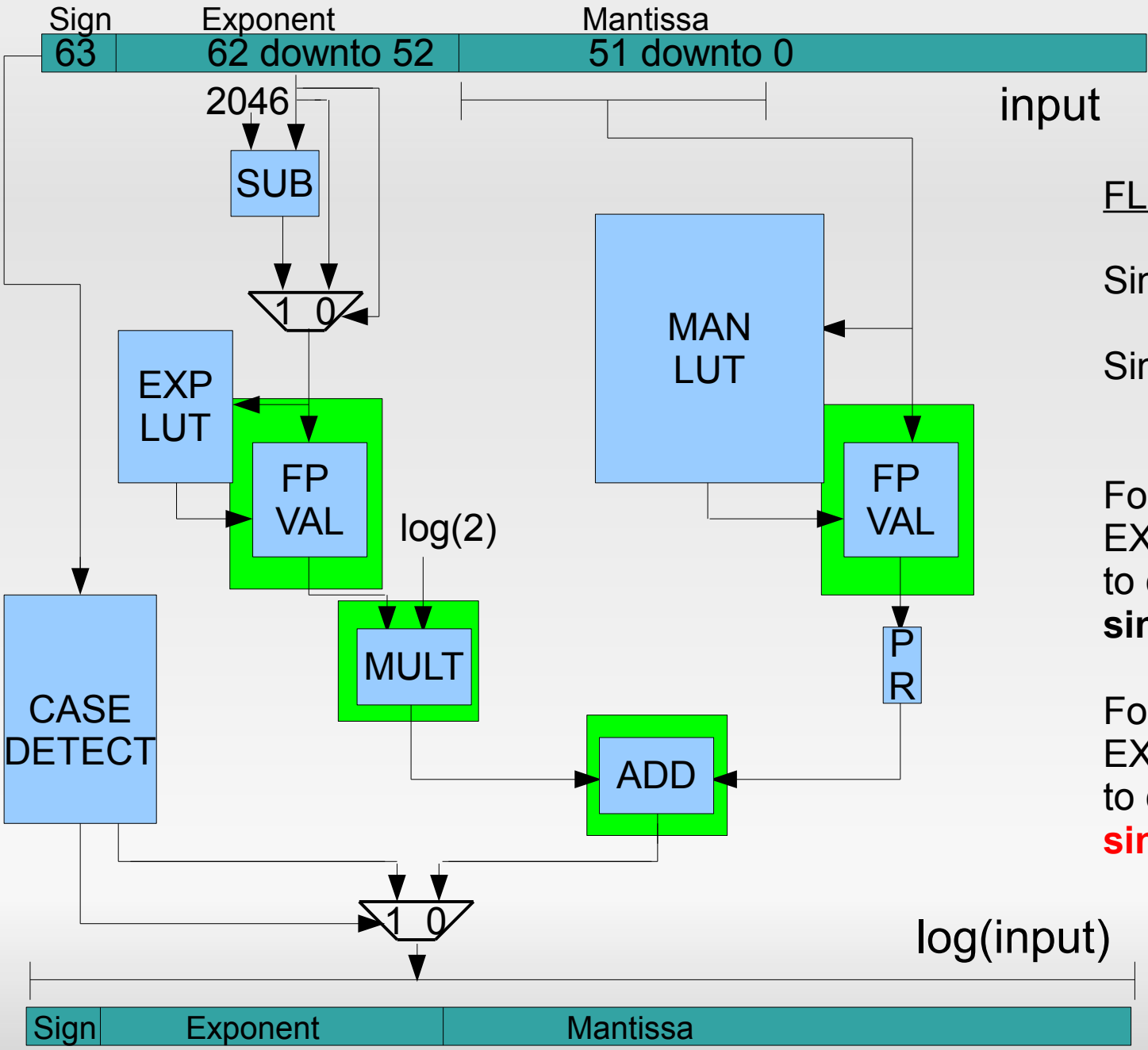
LOGARITHM APPROXIMATION UNIT (LAU) ARCHITECTURE



LOGARITHM APPROXIMATION UNIT (LAU) ARCHITECTURE



LOGARITHM APPROXIMATION UNIT (LAU) ARCHITECTURE



FLOATING-POINT VALUE

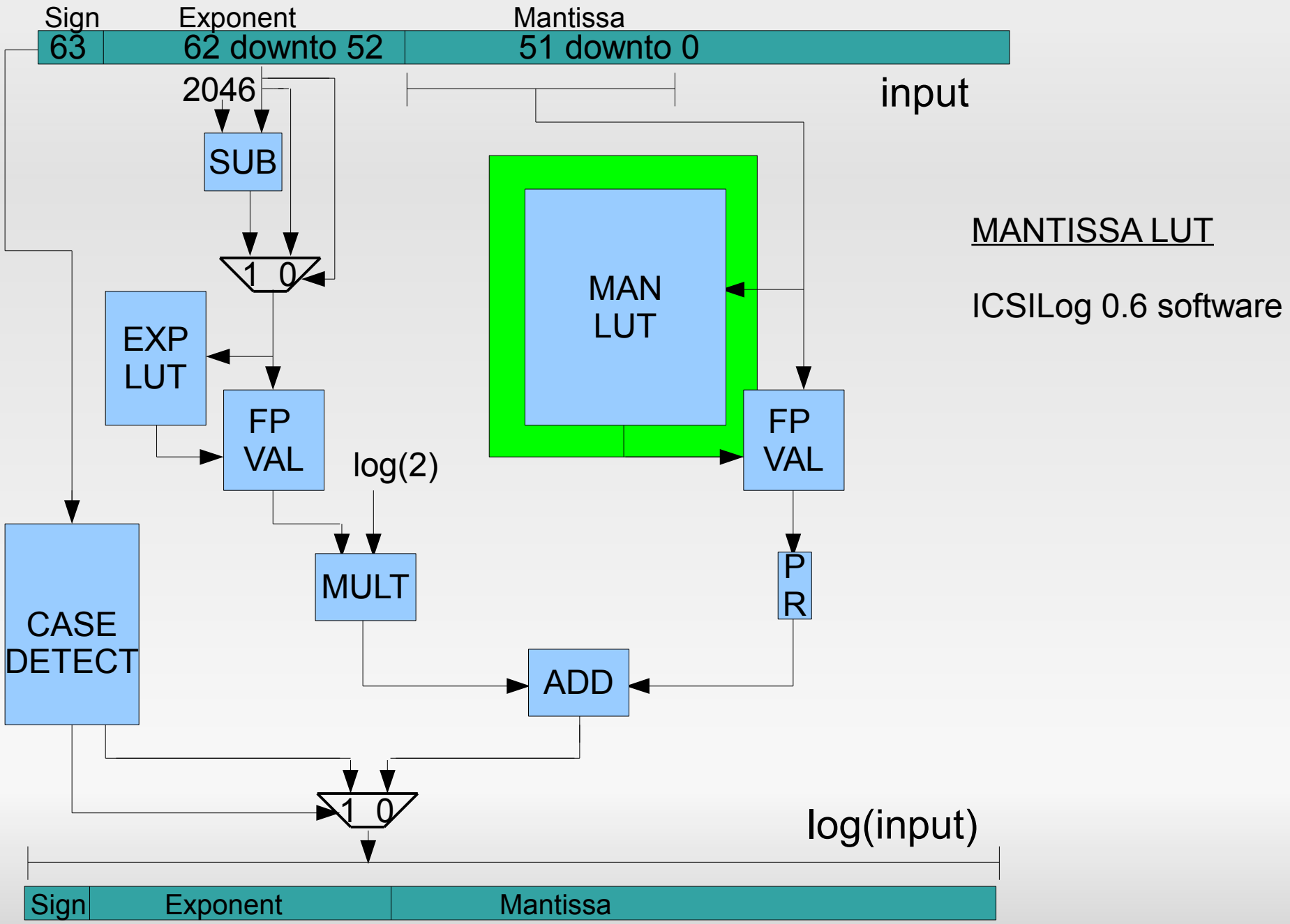
Single-precision values

Single-precision MULT and ADD

For **single-precision** inputs EXPLUT containts 128 entries to construct a **single-precision** value

For **double-precision** inputs EXPLUT contains 1024 entries to construct a **single-precision** value

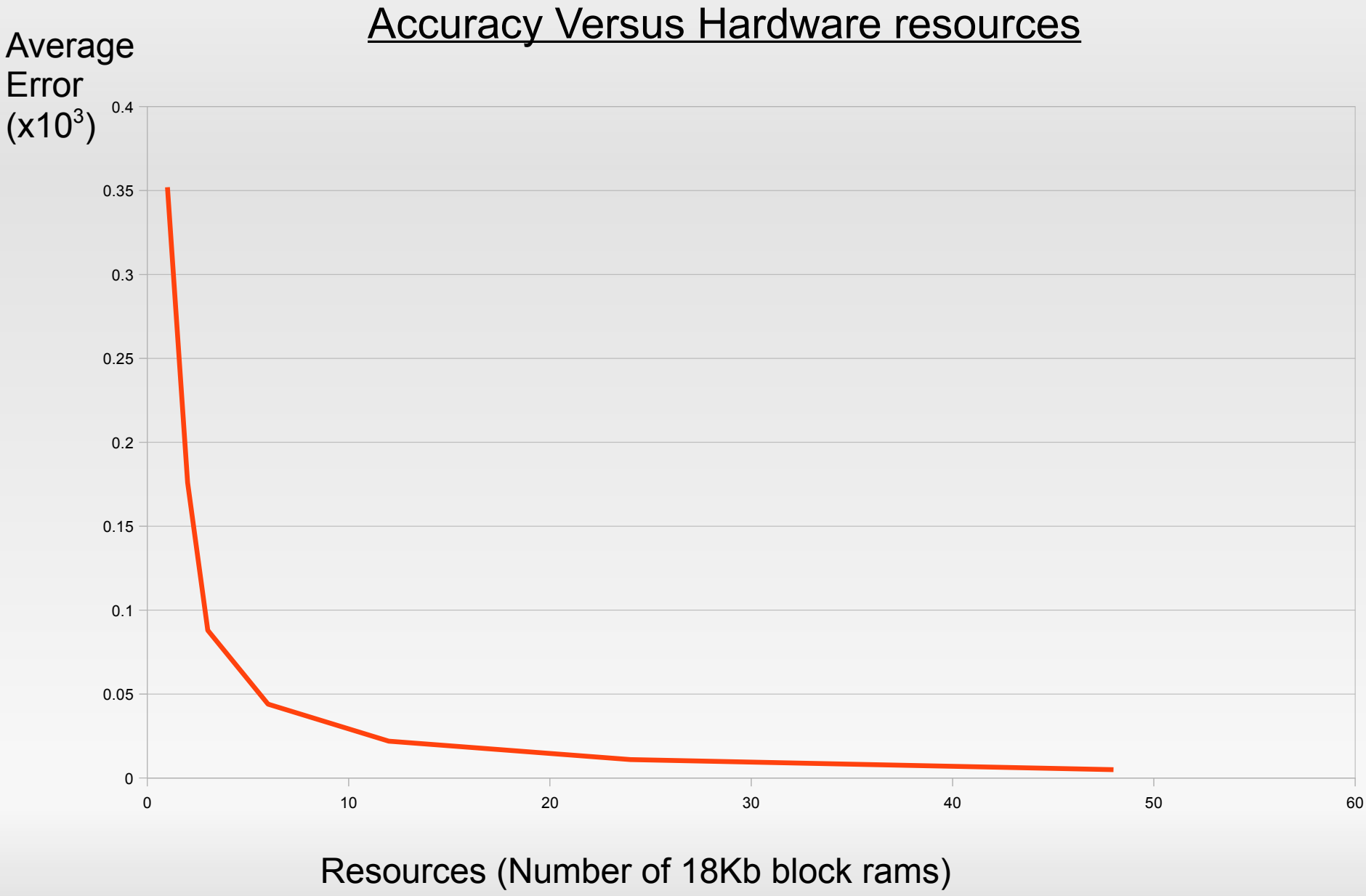
LOGARITHM APPROXIMATION UNIT (LAU) ARCHITECTURE



MANTISSA LUT

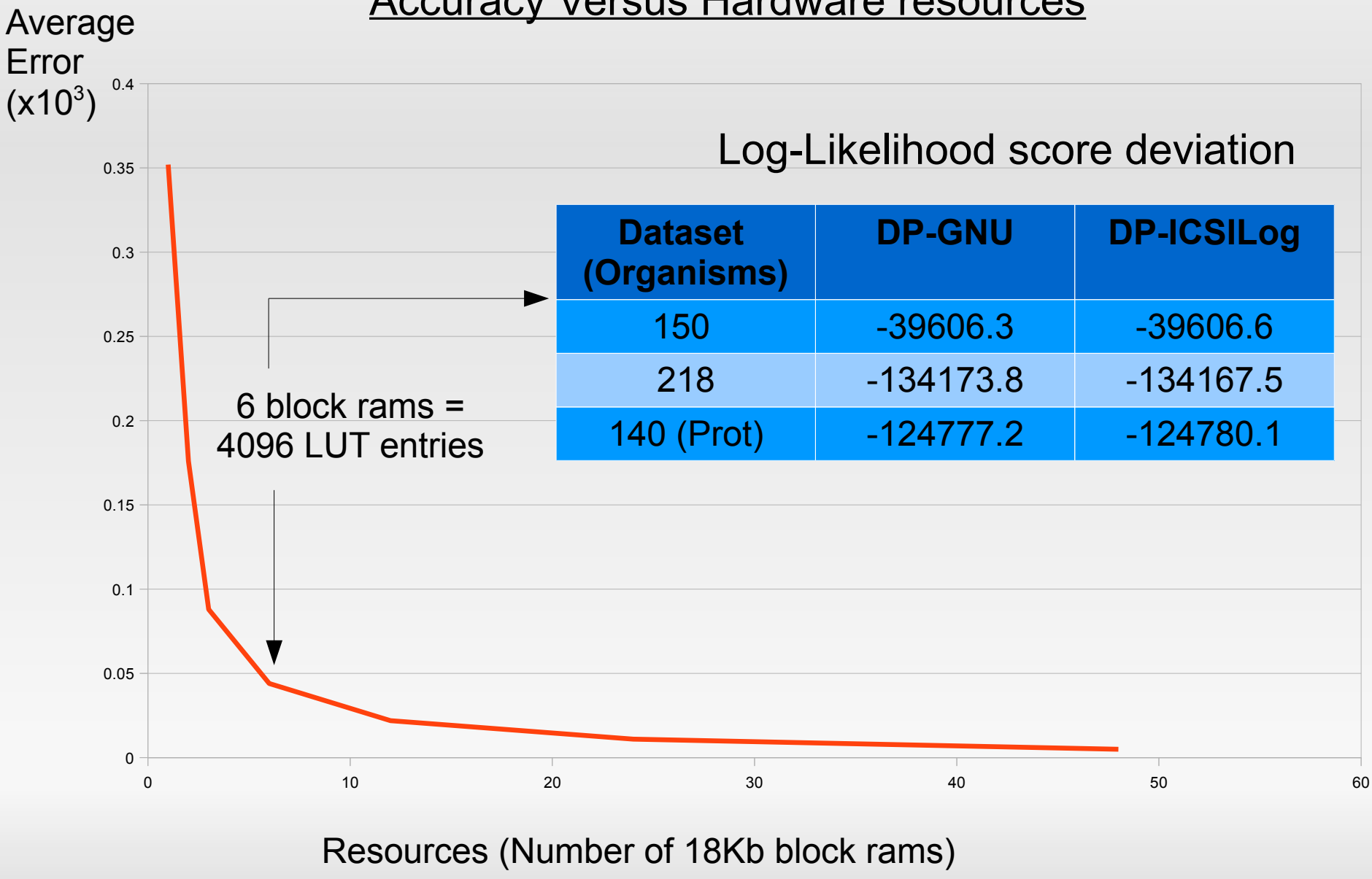
ICSILog 0.6 software

PERFORMANCE EVALUATION



PERFORMANCE EVALUATION

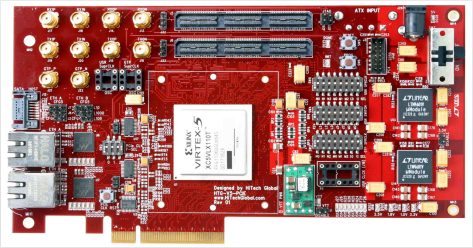
Accuracy Versus Hardware resources



Log-Likelihood score deviation

Dataset (Organisms)	DP-GNU	DP-ICSILog
150	-39606.3	-39606.6
218	-134173.8	-134167.5
140 (Prot)	-124777.2	-124780.1

PERFORMANCE EVALUATION

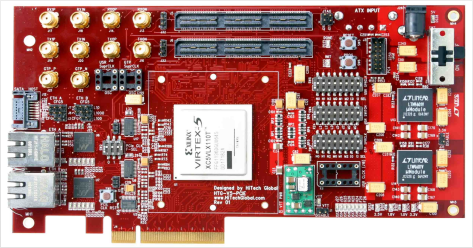


VIRTEX 5 SX95T for mapping and verification

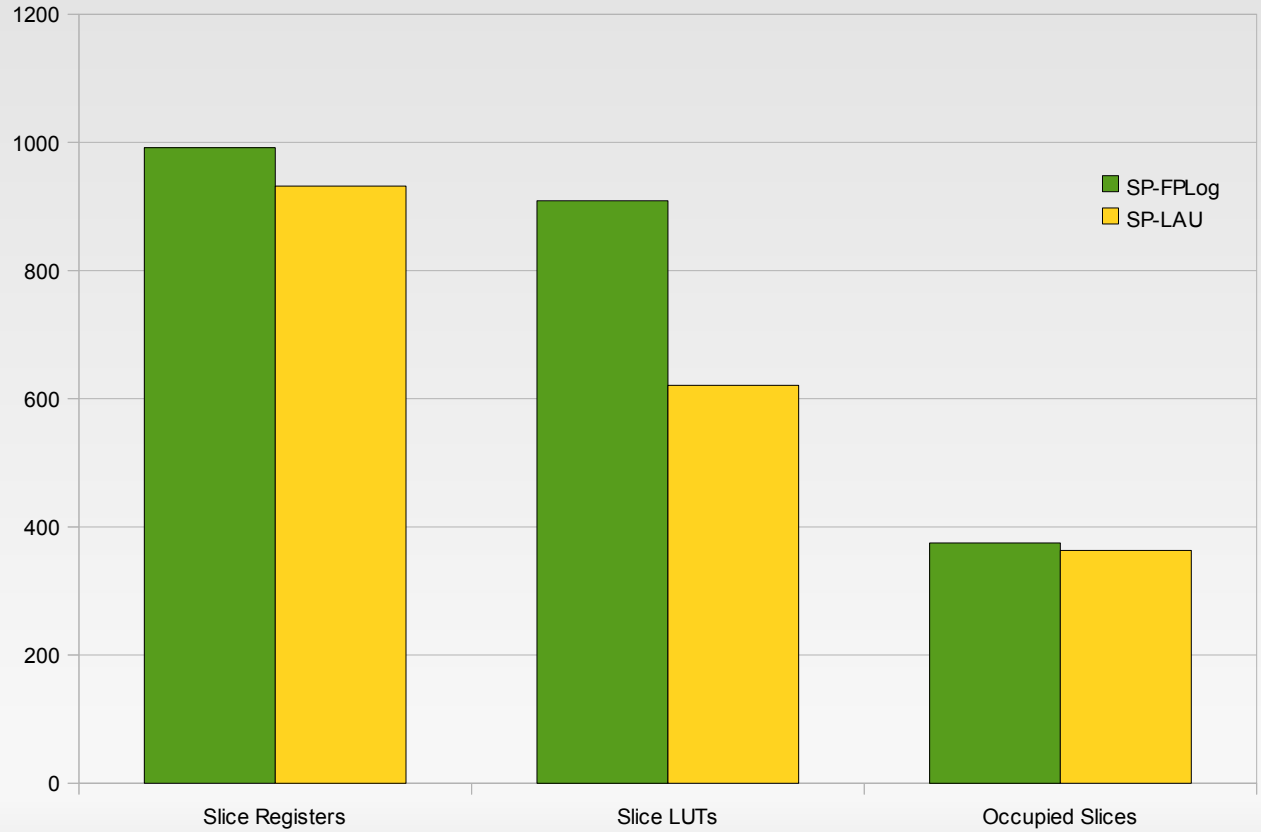
XILINX ISE 10.1 and CHIPSCOPE Pro Analyzer

F. de Dinechin, C. Klein, B. Pasca,
“Generating high-performance custom floating-point pipelines,” *Proc. of FPL 2009*.

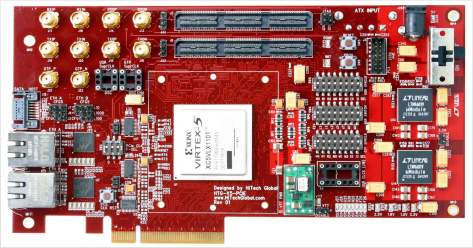
PERFORMANCE EVALUATION



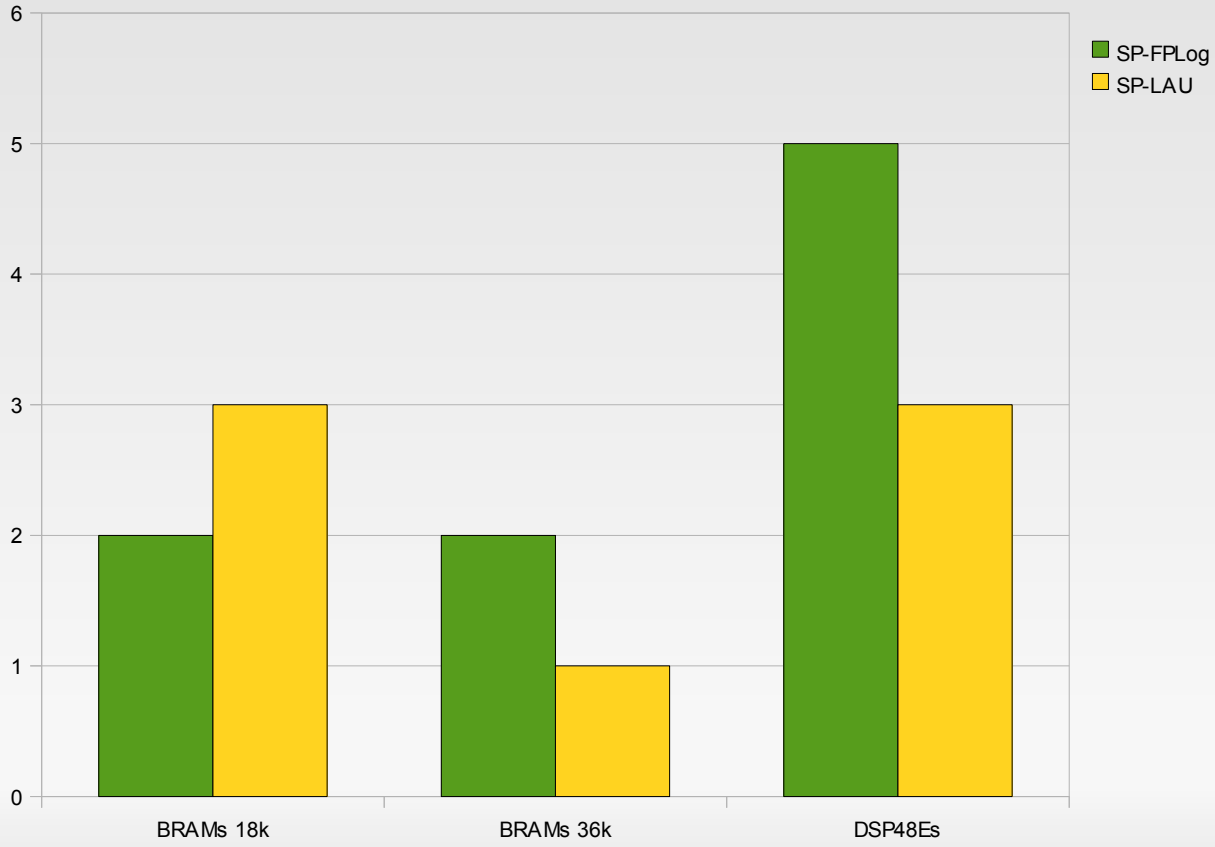
Resource Utilization and Performance: Single Precision



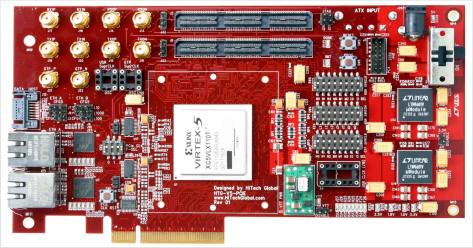
PERFORMANCE EVALUATION



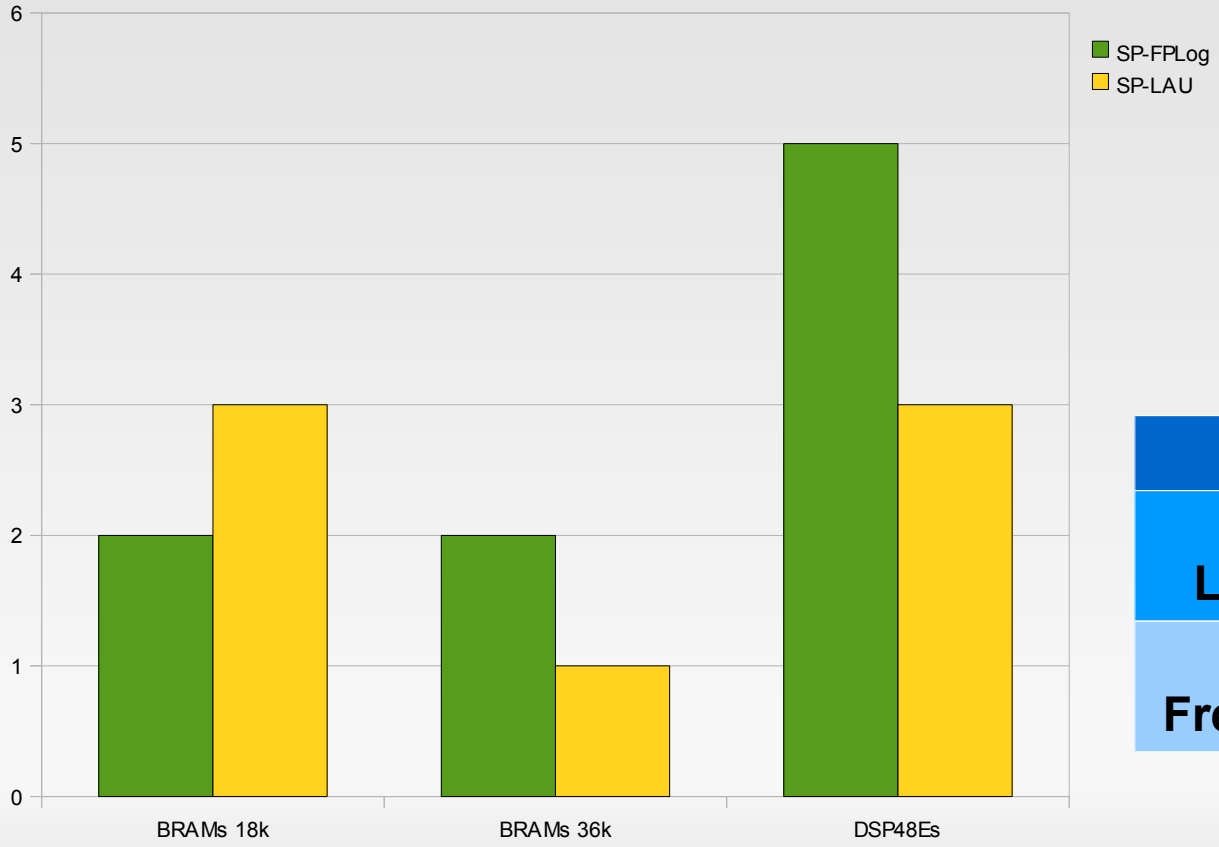
Resource Utilization and Performance: Single Precision



PERFORMANCE EVALUATION

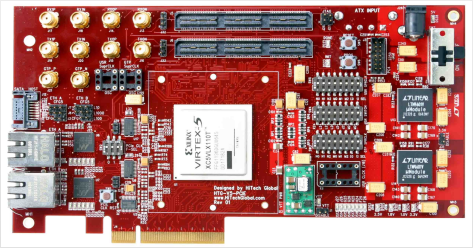


Resource Utilization and Performance: Single Precision

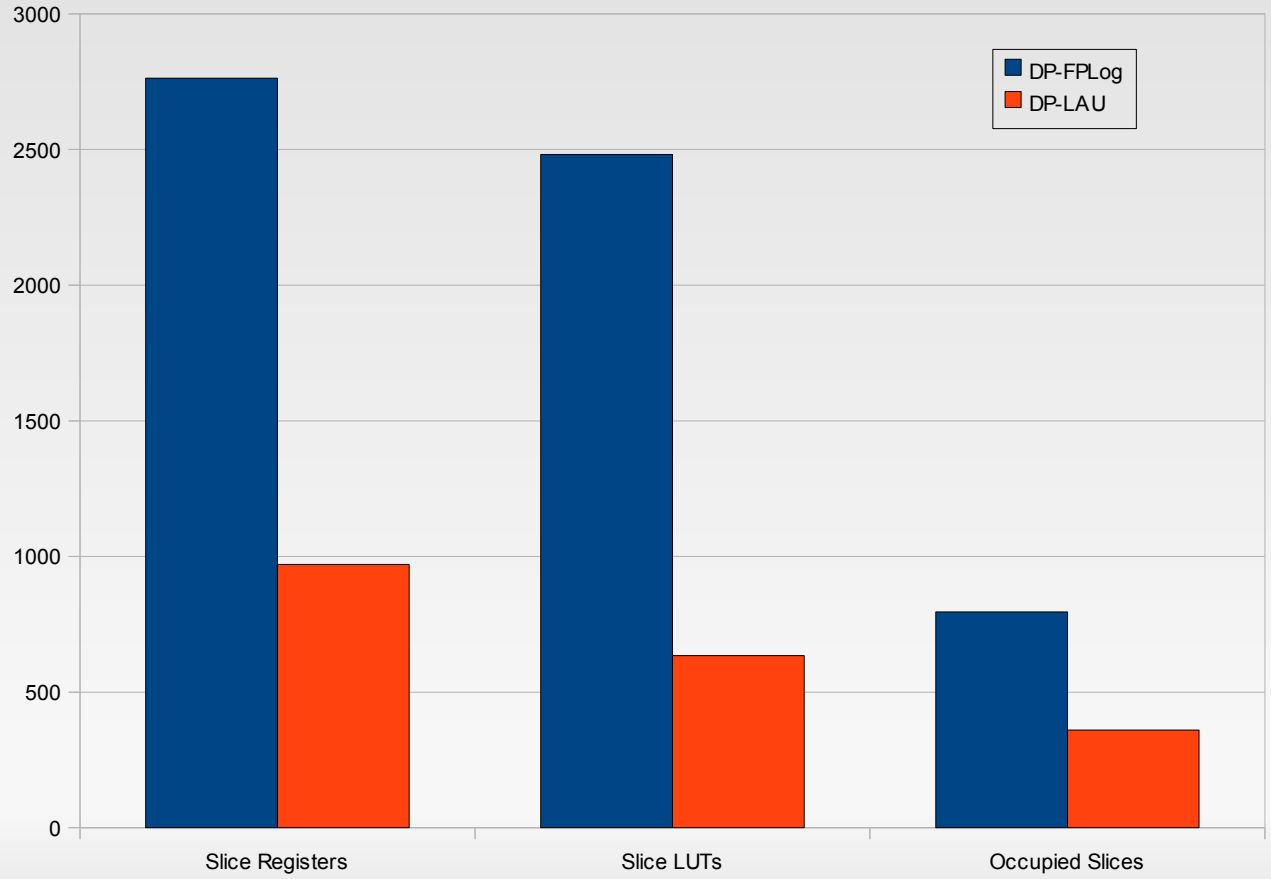


	FPLog	LAU
Clock Latency	20	22
Max Frequency	244.7	353.5

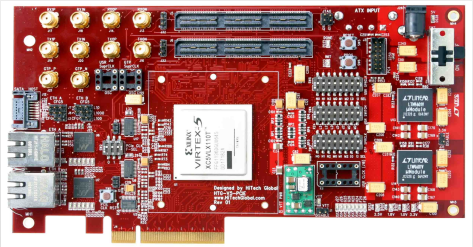
PERFORMANCE EVALUATION



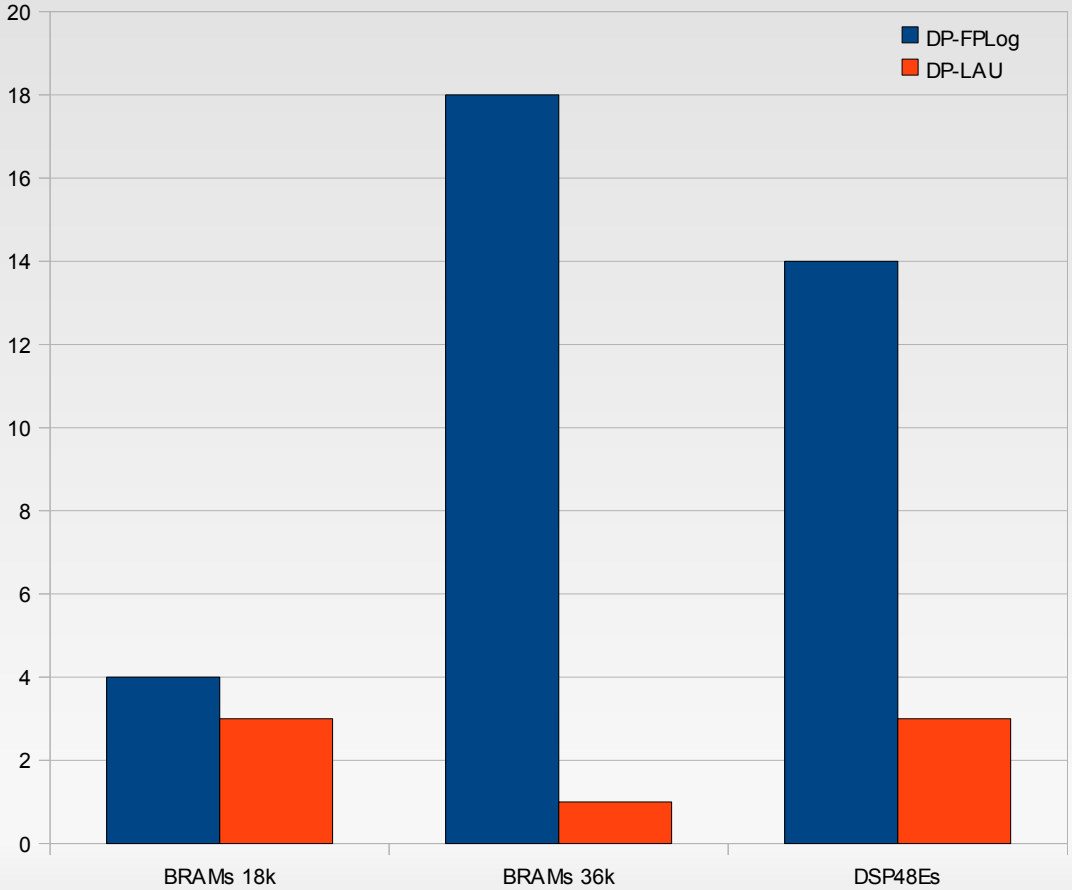
Resource Utilization and Performance: Double Precision



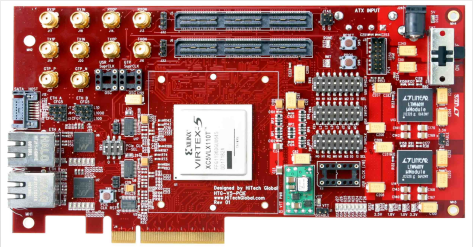
PERFORMANCE EVALUATION



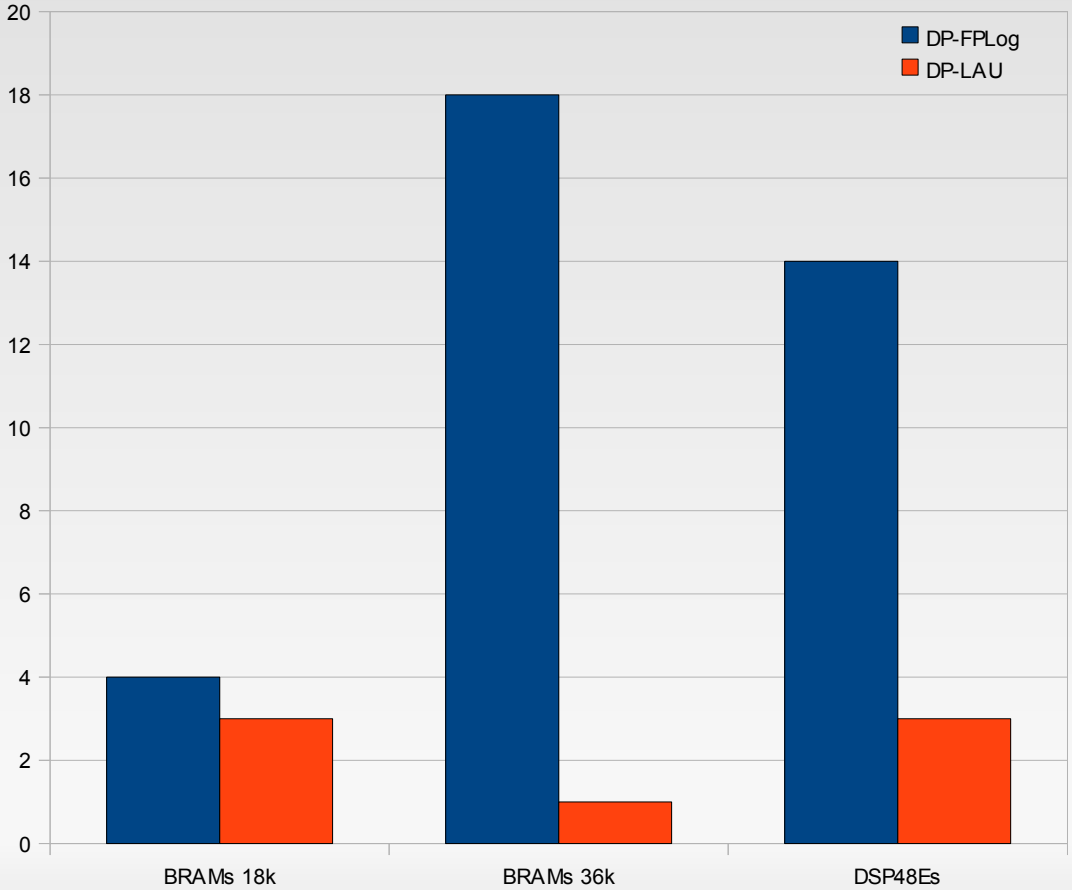
Resource Utilization and Performance: Double Precision



PERFORMANCE EVALUATION

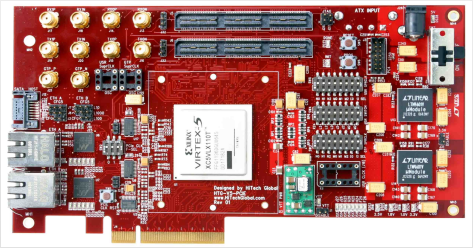


Resource Utilization and Performance: Double Precision



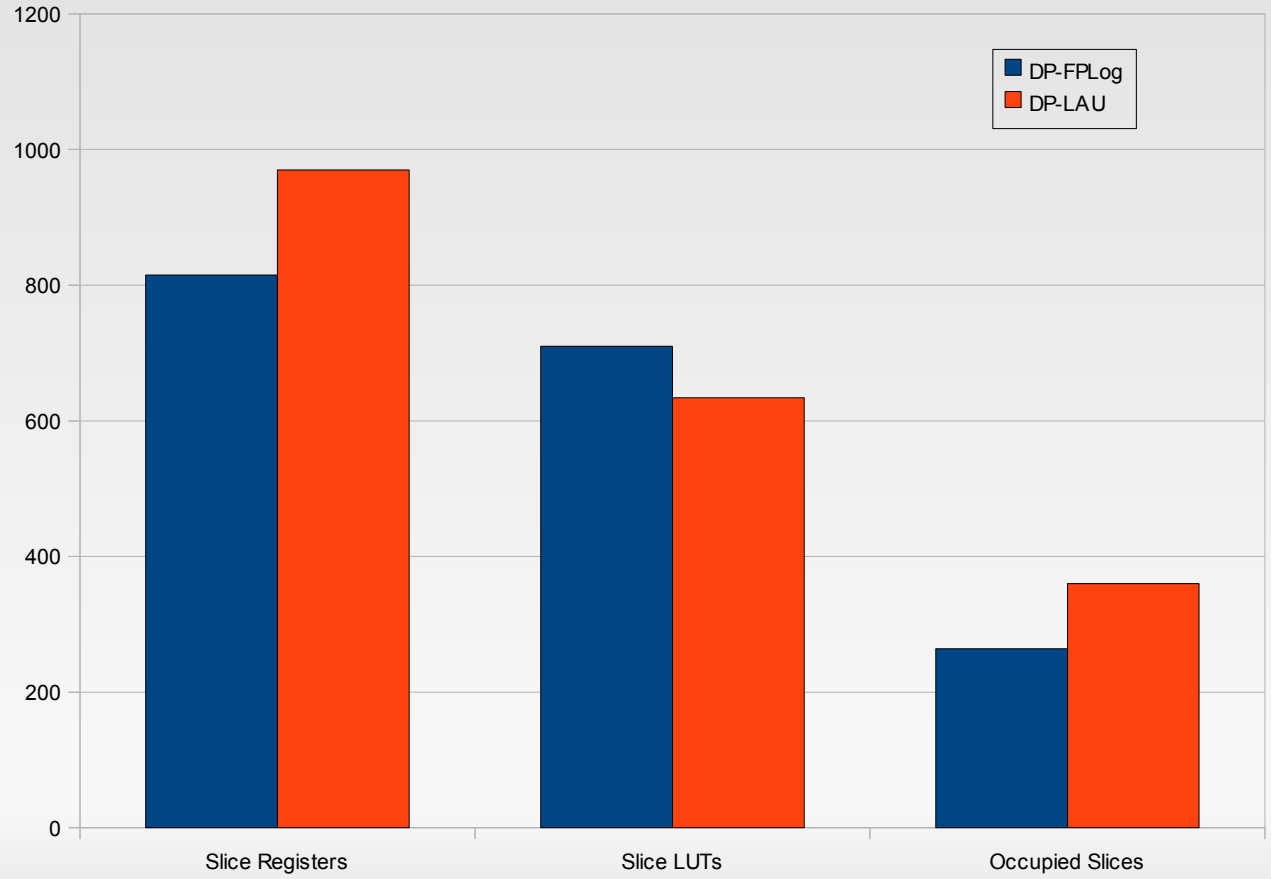
	FPLog	LAU
Clock Latency	34	22
Max Frequency	192.3	320.6

PERFORMANCE EVALUATION

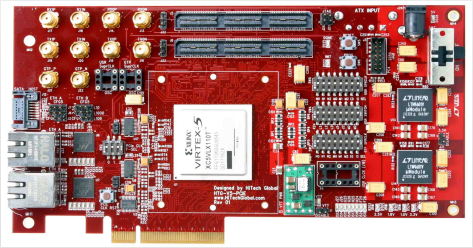


Resource Utilization and Performance: Double Precision

DP-FPLog with same accuracy as DP-LAU

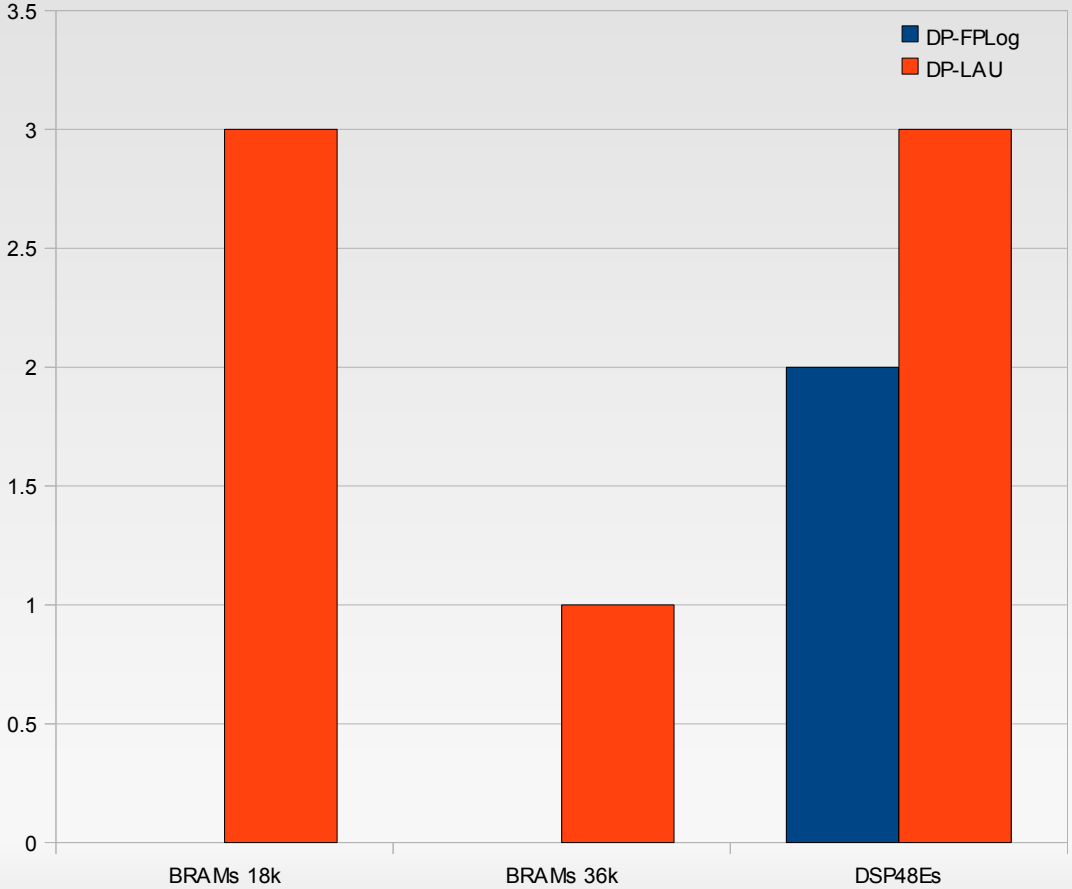


PERFORMANCE EVALUATION

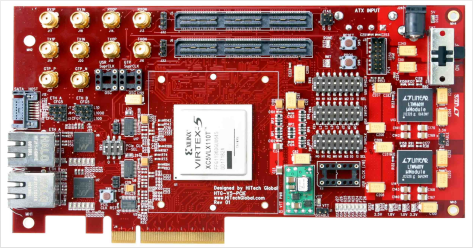


Resource Utilization and Performance: Double Precision

DP-FPLog with same accuracy as DP-LAU

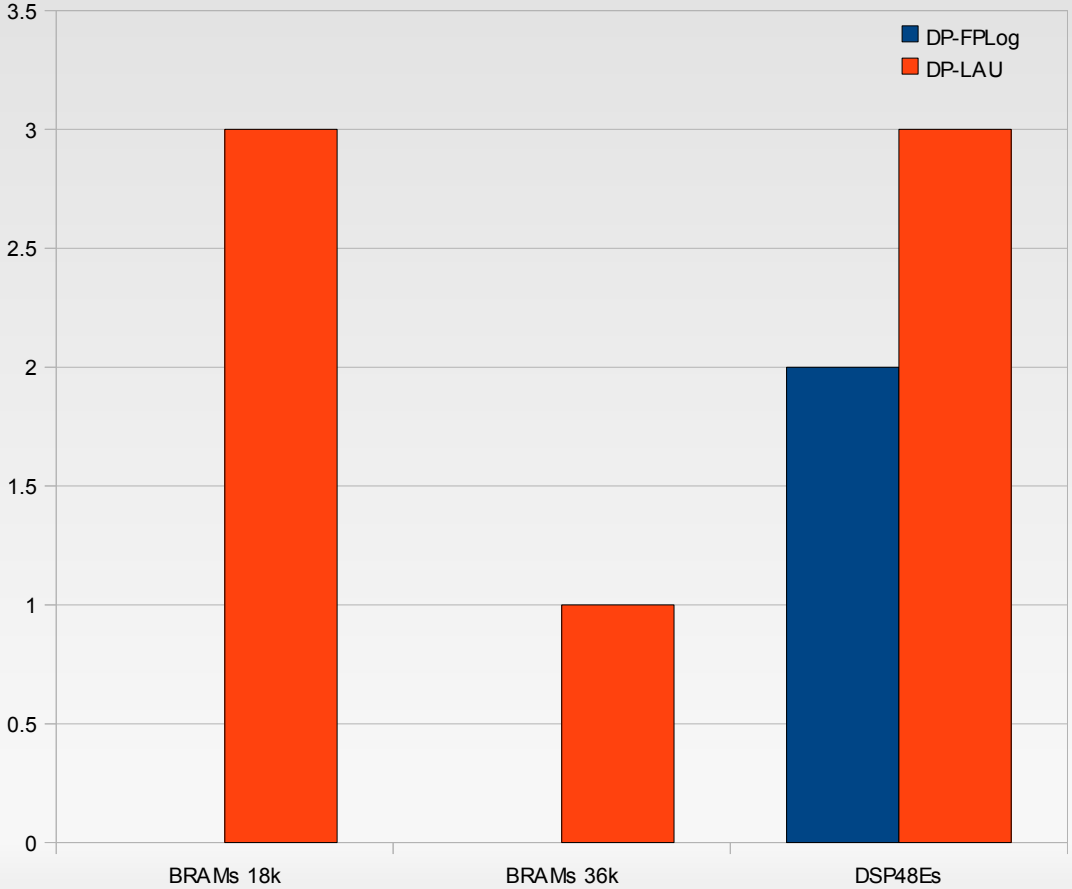


PERFORMANCE EVALUATION



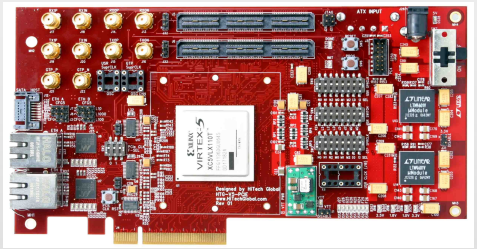
Resource Utilization and Performance: Double Precision

DP-FPLog with same accuracy as DP-LAU



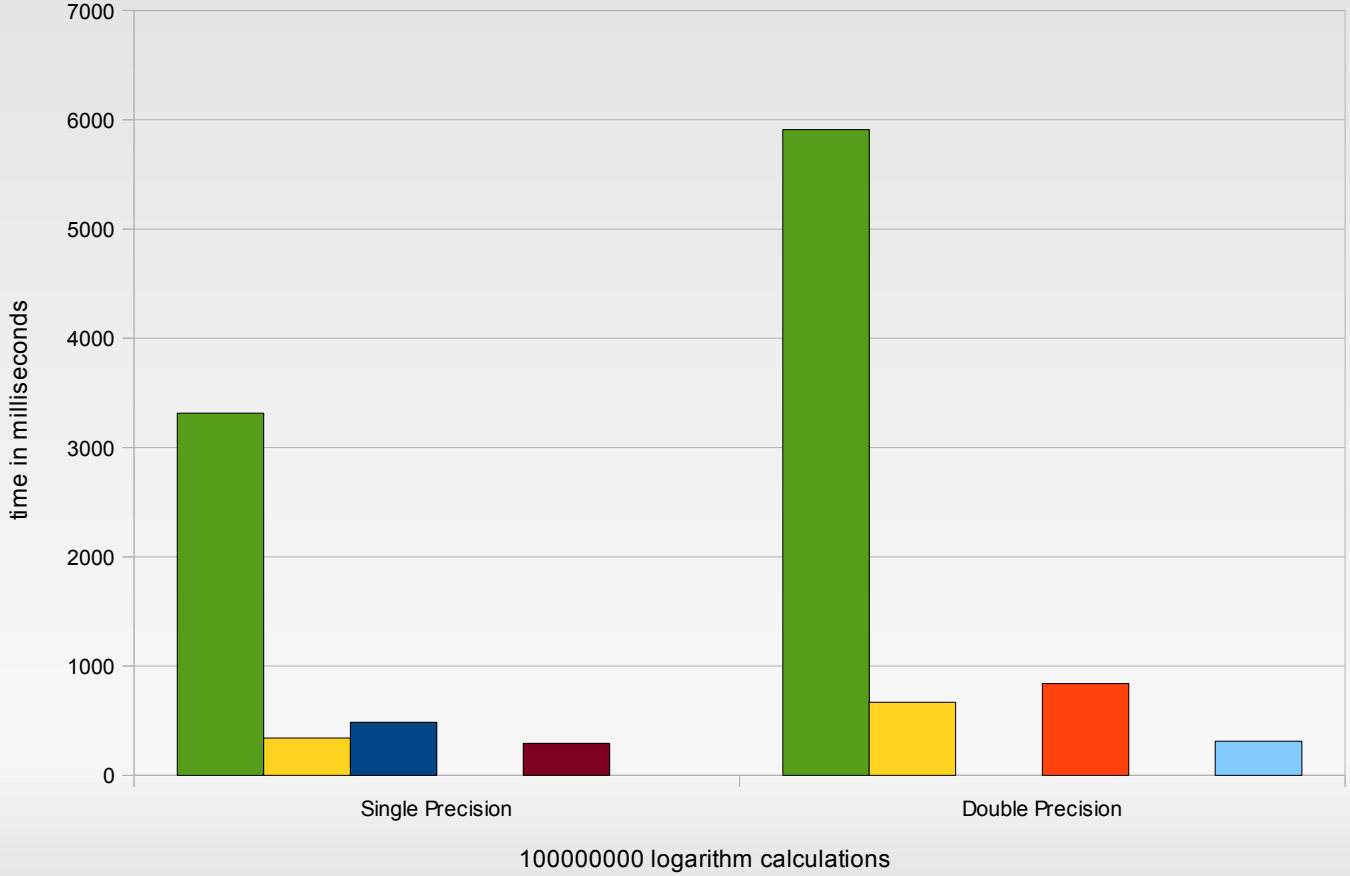
	FPLog	LAU
Clock Latency	20	22
Max Frequency	239.6	320.6

PERFORMANCE EVALUATION



Performance:

LAU vs SP/DP-ICSILog vs GNU Log vs MKL Log



Intel Core2 DUO T9600 @ 2.8GHz

6MB L2 Cache

SP-LAU VS
- GNU-LOG : 11X
- MKL-LOG : 1.6X

DP-LAU VS
- GNU-LOG: 18X
- MKL-LOG: 2.5X

CONCLUSION and FUTURE WORK

AVAILABILITY

DP-ICSILog C Implementation and SP/DP LAU FPGA core for Virtex4 and Virtex5 FPGAs

http://wwwkrammer.in.tum.de/exelixis/nikos/ipcores.html

Or

OpenCores.org: Project name: fp_log

http://www.opencores.org/project,fp_log

CONCLUSION and FUTURE WORK

RELATED PROJECTS

Implementation of a UDP/IP core for Virtex 5 FPGAs (optimized for PC-FPGA communication)

http://wwwkrammer.in.tum.de/exelixis/nikos/ipcores.html

Or

OpenCores.org: Project name: udp_ip__core

http://www.opencores.org/project,udp_ip__core

FUTURE WORK

Implementation of a resource-efficient exponential function

Integration of the LOG and EXP cores into the general Phylogenetic Architecture