FPGA Implementation of Custom Floating-Point Logarithm and Division

Nelson Campos[1], Slava Chesnokov[3], Eran Edirisinghe[1] and Alexis Lluis[2]

[1]Loughborough University, [2]ARM Ltd, [3]Imaging-CV Ltd

Introduction

In this paper[1] we present dedicated FPGA architectures for implementing the logarithm and division operations in floating-point arithmetic. A numeric analysis is provided to find the optimum bit-width of each coefficient of the polynomial approximations used to compute the transcendental functions such as division and logarithm, but it is shown that the method can be expanded to implement any other function that can be approximated to a polynomial.

To summarize, the main contributions of this paper are:

- propose floating-point hardware architectures for mathematical division and logarithm modules;
- parameterizable modules with the mantissa and exponent widths as inputs defined by the user;
- apply techniques of precision analysis and bit-width optimization using differential evolution in polynomial approximations to reduce the resource usage of the division and logarithm hardware modules.

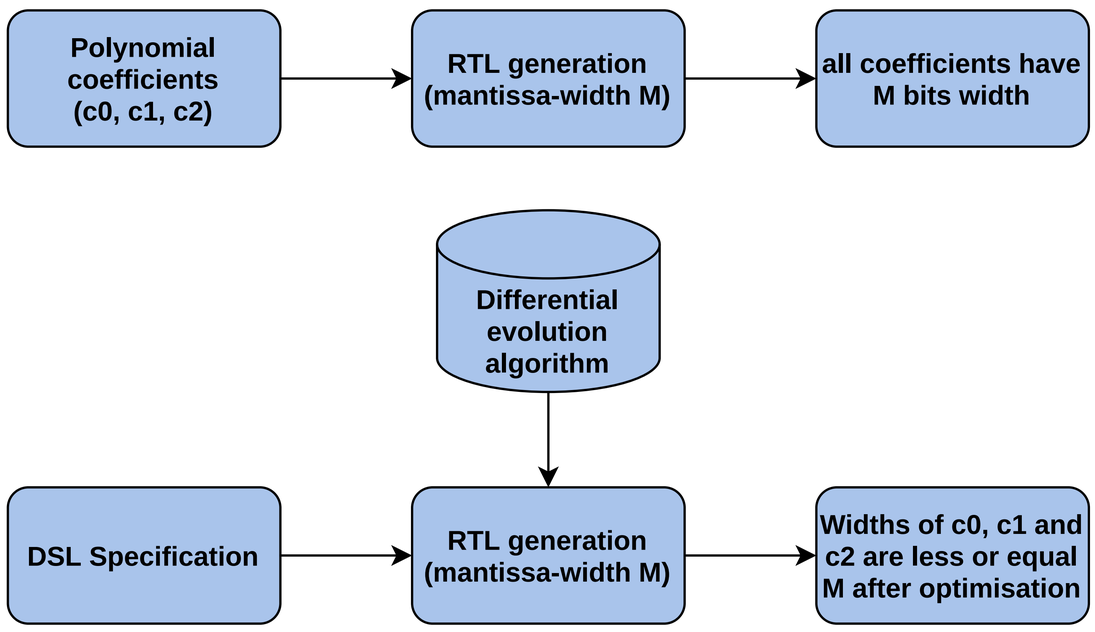
Results and discussions

The polynomial approximation for division, when the mantissa width is 23 bits and the desired accuracy is 2^{-16}, the bit-width optimization showed a reduction of 10.3093% and 20% in the number of LUTs and DSP blocks, respectively. Similarly, for the same mantissa bit-width and accuracy, in the logarithm approximation the LUTs increased by 41.5584% and the DSP blocks reduced 20%. The trends observed for both logarithm and division approximations showed a smaller reduction in terms of LUTs and DSP blocks when the mantissa is 10 bits wide. The designs of the floating-point division and logarithm have a latency of 6 and 7 clock cycles, respectively, and throughput of one operation per clock cycle. The logarithm and divider in floating-point half-precision (mantissa M=10-bit and exponent E=5-bit) and in single-precision (M=23-bit and E=8-bit) modules were also synthesized in Vivado. The target used is the Artix 7 FPGA at a clock frequency of 100MHz and the designs are described in SystemVerilog.

Proposed Methods

Table 1:Coefficients of the polynomial approximation for$log_2(1+m)$ and $\frac{1}{m+1}$, where $m \in [0,1)$

Range	c_0	c_1	c_2
$1.0 \leq 1.m < 1.25$	-0.573	1.4288	0.0003
$1.25 \leq 1.m < 1.5$	-0.3829	1.3381	0.0115
$1.5 \leq 1.m < 1.75$	-0.2739	1.2312	0.0379
$1.75 \leq 1.m < 2.0$	-0.2056	1.1299	0.0756



Floating-point format	Function	Architecture	LUTs	FFs	DSP-blocks	BRAM
half-precision	Div	Proposed UFB	43	0	3	0.0
		Proposed MFB	41	0	3	0.0
		LogiCORE Div	253	431	0	0.0
		FloPoCo Div	498	212	0	0.0
	Log	Proposed UFB	211	49	2	0.0
		Proposed MFB	210	49	2	0.0
		LogiCORE Log	274	469	2	0.0
		FloPoCo Log	411	377	1	0.5
single-precision	Div	Proposed UFB	137	0	7	0.0
		Proposed MFB	127	0	6	0.0
		LogiCORE Div	809	1487	0	0.0
		FloPoCo Div	1617	624	0	0.0
	Log	Proposed UFB	534	65	5	0.0
		Proposed MFB	555	65	4	0.0
		LogiCORE Log	721	1191	4	0.0
		FloPoCo Log	681	842	3	2.0

Figure 1:Floating-point divider and logarithm FPGA architectures

Hardware architectures

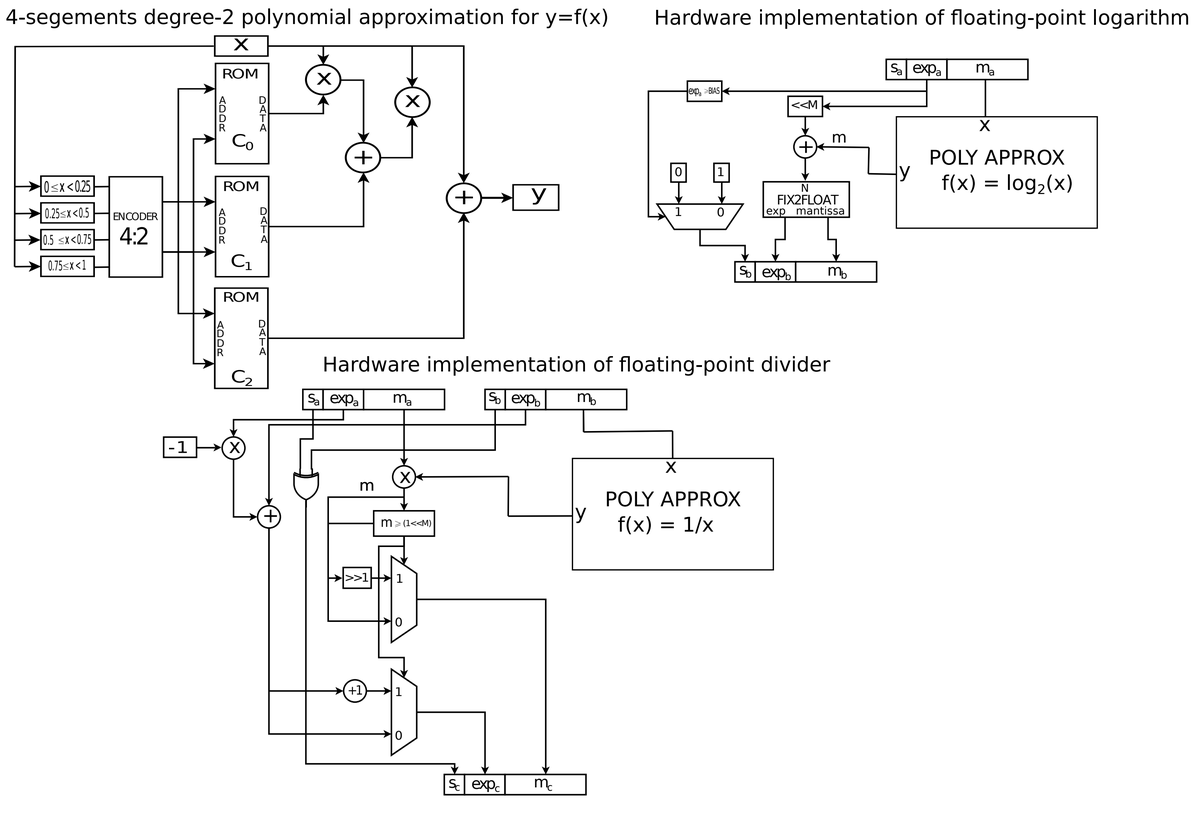


Figure 2:Floating-point divider and logarithm FPGA architectures

Table 2:Coefficients of the polynomial approximation for
$log_2(1+m)$ and $\frac{1}{m+1}$, where $m \in [0,1)$

Range	c_0	c_1	c_2
$1.0 \leq 1.m < 1.25$	-0.573	1.4288	0.0003
$1.25 \leq 1.m < 1.5$	-0.3829	1.3381	0.0115
$1.5 \leq 1.m < 1.75$	-0.2739	1.2312	0.0379
$1.75 \leq 1.m < 2.0$	-0.2056	1.1299	0.0756

Conclusions

In this paper we have presented dedicated FPGA architectures for logarithm and division mathematical operations in floating-point arithmetic. For both modules, the mantissa and exponent are flexible parameters that can be defined by the user. The bit-width optimization of both methods is dependent of the desired accuracy defined in the design stage and is accomplished using differential evolution with the Python library *mystic*. For both modules, the bit-width optimization showed a reduction of LUTs, flip-flops and DSP blocks, leading to significant resource savings compared to existing IP cores.

References

[1] Nelson Campos, Slava Chesnokov, Eran Edirisinghe, and Alexis Lluis. Fpga implementation of custom floating-point logarithm and division. In *Applied Reconfigurable Computing. Architectures, Tools, and Applications*, pages 295–304, Cham, 2021. Springer International Publishing.

[2] IS Committee et al. 754–2008 ieee standard for floating-point arithmetic. *IEEE Computer Society Std*, 2008, 2008.

Acknowledgements

Supported by CDT-EI at Loughborough University and ARM Holdings PLC.

Contact Information

- Email: N.C.S.Campos@lboro.ac.uk
- Email: E.A.Edirisinghe@lboro.ac.uk
- Email: slava@chesnokov.org
- Email: alexis.lluis@arm.com