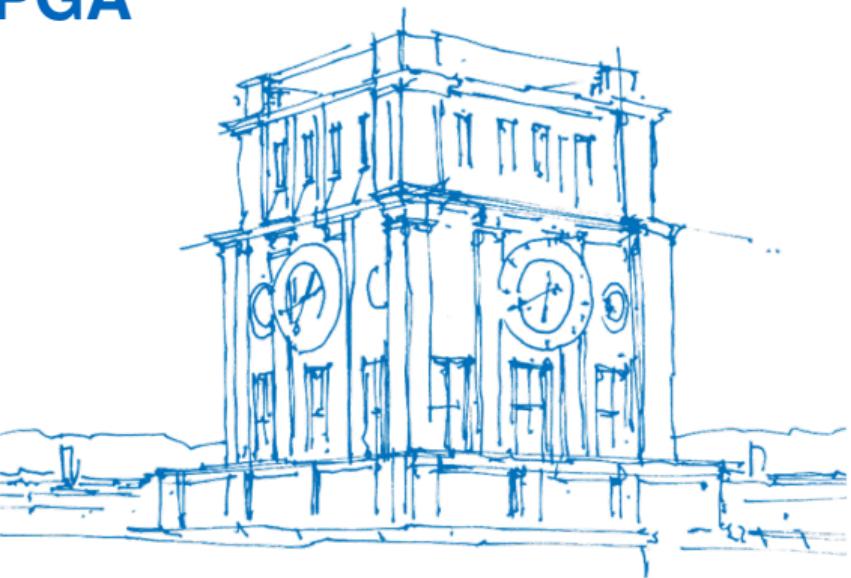


Investigation and Implementation of Approximate Dividers for FPGA

Kiran Krishnadas Bhandarkar

Chair for Integrated Systems
TUM School of Computation, Information and Technology
Technical University of Munich

February 24th, 2023



TUM Uhrenturm

Outline

- 1 Introduction**
- 2 Approximate Divider Approaches
- 3 Design Evaluation
- 4 Experimental Results
- 5 Conclusion and Outlook
- 6 References

Motivation

- Modern computing systems need to deliver high performance with resource and energy constraints.
- Approximate computing paradigm saves energy and resources by leveraging the principle that standard applications are tolerant to some errors.
- Unlike adders and multipliers, the approximate division operation is less explored as it is a resource-intensive operation.

Applications of Approximate Dividers

- Image processing applications (smoothing, sharpening, compression).
- In machine learning applications to reduce delay and save energy.
- To achieve higher speeds and power efficiency in digital processing applications.

Evaluation Metrics

- **Error Rate (ER):** Error rate indicates the probability that an incorrect result is generated.
- **Error Distance (ED):** ED is the arithmetic difference between an exact output (M) and an inexact output (M') as given by $ED = M' - M$.
- **Mean Relative Error Distance (MRED):** The mean of the relative difference to correct output.
- **Normalized Mean Error Distance (NMED):** Normalization of mean error distance using the largest exact output.
- **Circuit Characteristic Metrics:** Power, Area, Critical Path Delay.

Outline

- 1 Introduction
- 2 Approximate Divider Approaches
- 3 Design Evaluation
- 4 Experimental Results
- 5 Conclusion and Outlook
- 6 References

Approximate Divider Designs Normalized Summary

Approximate Divider	Ref.	Type	Area (μm^2)	Power (mW)	Path Delay (ns)
Logarithmic Exponent Approximate Divider (LEAD)	[3]	Fixed	5482.04	0.73	9.91
Divider Based on Piecewise Constant Approximation	[4]	Float	212.04	0.32	3.82
Approximate Restoring Dividers Using Inexact Cells	[5]	Fixed	304.35	0.19	4.05
Scalable Accuracy Approx. Divider with Error Compensation (SAADI-EC)	[6]	Fixed	1973	0.36	2.12
A Multistage Approximation Floating Point Dividers (FPAD)	[7]	Float	749.07	0.03	0.25
Truncation-based Approximate Divider (TruncApp)	[8]	Fixed	1483	0.80	1.05
High Speed Rounding based Approximate Divider (SEERAD)	[9]	Fixed	1343	0.64	0.61
Approximate Divider based on piecewise linear approximation	[10]	Fixed	1829	0.86	1.25
Configurable Approximate Divider for Energy Efficiency (CADE)	[11]	Float	NA	NA	0.51

Outline

- 1** Introduction
- 2** Approximate Divider Approaches
- 3** Design Evaluation
- 4** Experimental Results
- 5** Conclusion and Outlook
- 6** References

Restoring Array Based Approximate Divider I

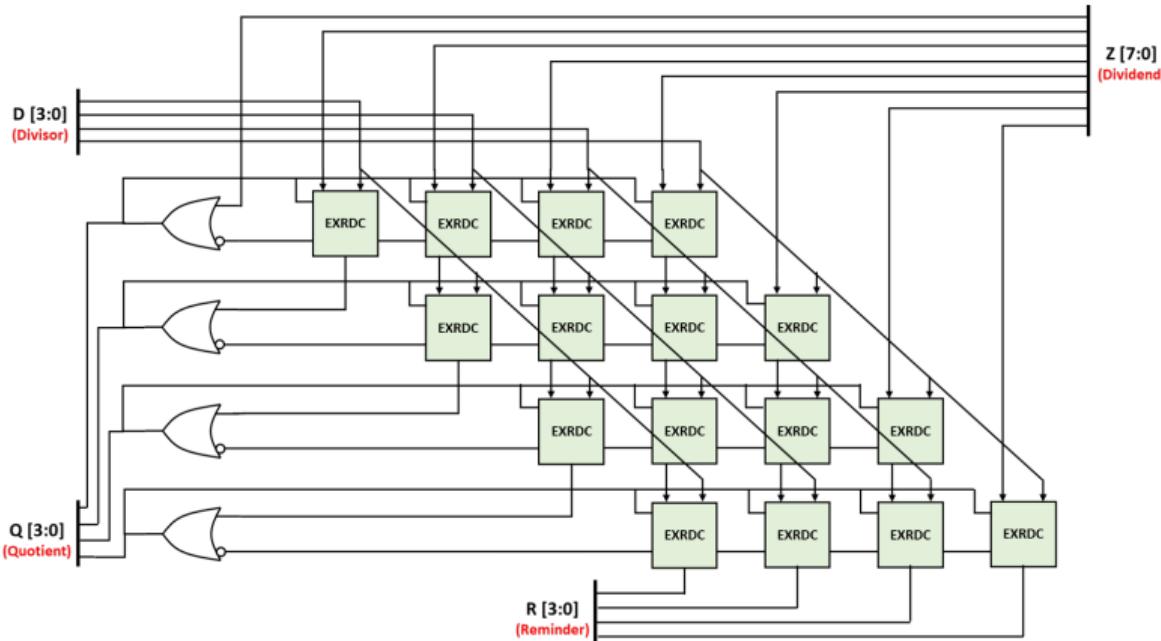


Figure 1 Architecture of 8/4 exact array divider [10]

Restoring Array Based Approximate Divider II

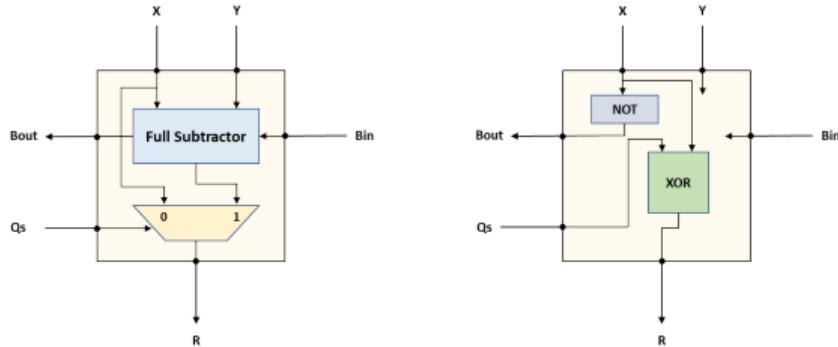


Figure 2 Exact restoring divider cell (EXRDC) (left) and approximate divider cell (AXRDC)(right)

- The approximation/tuning parameter p refers to the magnitude of the approximation in the design.

Linear Piecewise Approximate Divider I

For two unsigned numbers, A and B, the exact division quotient (Q) is given by:

$$Q = \frac{F_A \times 2^k_A}{F_B \times 2^k_B} = F_A \times \left(\frac{1}{F_B} \right) \times 2^{k_A - k_B} = F_A \times X_B \times 2^{k_A - k_B}$$

The X_B term is approximated using linear piecewise approximation technique. Therefore, approximate Q is given by:

$$Q = (1 + f_A) \times (\alpha \times f_b + \beta) \times 2^{k_A - k_B}$$

The α and β values are to be computed and stored in a LUT.

Linear Piecewise Approximate Divider II

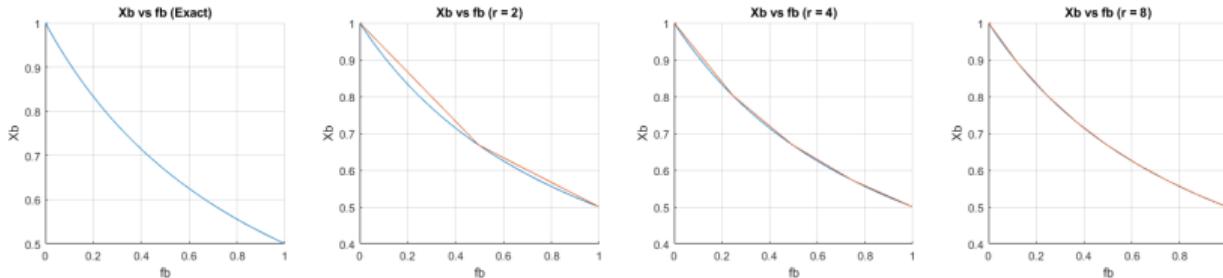


Figure 3 X_b vs f_b curves for various tuning factors (r)

f_b	$r = 2$		f_b	$r = 4$		f_b	$r = 8$	
	α	β		α	β		α	β
0.000, 0.496	-0.668	1	0.000, 0.246	-0.802	1	0.000, 0.121	-0.892	1
0.496, 0.996	-0.334	0.834	0.246, 0.496	-0.536	0.934	0.121, 0.246	-0.715	0.978
			0.496, 0.746	-0.382	0.858	0.246, 0.371	-0.585	0.946
			0.746, 0.996	-0.286	0.786	0.371, 0.496	-0.487	0.910
						0.496, 0.621	-0.412	0.873
						0.621, 0.746	-0.353	0.836
						0.746, 0.871	-0.306	0.801
						0.871, 0.996	-0.267	0.767

Performance Comparison I

Table 1 MATLAB Simulation Summary - Array Based Inexact Divider

(p)	ER (%)	MRED	NMED	ED Max	Area (μm^2)	Power (mW)	Path Delay (ns)
4 (EX:54, IEX:10)	5.07	$0.02e^{-02}$	0.0073	7			
6 (EX:41, IEX:21)	23.40	$0.10e^{-02}$	0.0083	31	304.35	0.19	4.05
8 (EX:28, IEX:36)	66.24	$0.45e^{-02}$	0.0091	127			

Performance Comparison II

Table 2 MATLAB Simulation Summary - Linear Piecewise Approximate Divider

r	ER (%)	MRED	NMED	ED Max	Area (μm^2)	Power (mW)	Path Delay (ns)
2	77.80	0.0309	0.0468	85			
4	70.50	0.0230	0.0349	85	1829	0.86	1.25
8	68.10	0.0228	0.0345	85			

- Inference:** The array divider is preferred since it has superior circuit (area and power) and error characteristics.

Outline

- 1** Introduction
- 2** Approximate Divider Approaches
- 3** Design Evaluation
- 4** Experimental Results
- 5** Conclusion and Outlook
- 6** References

HDL Simulation Results I

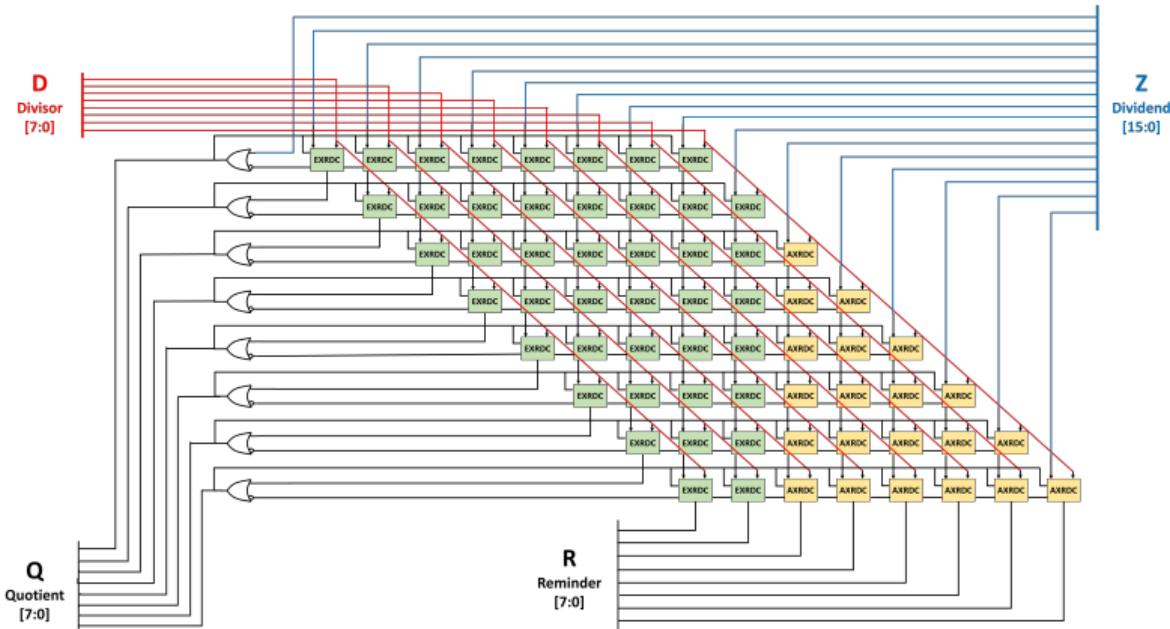


Figure 4 Architecture of 16/8 approximate non-pipelined array divider with $p=6$

HDL Simulation Results II

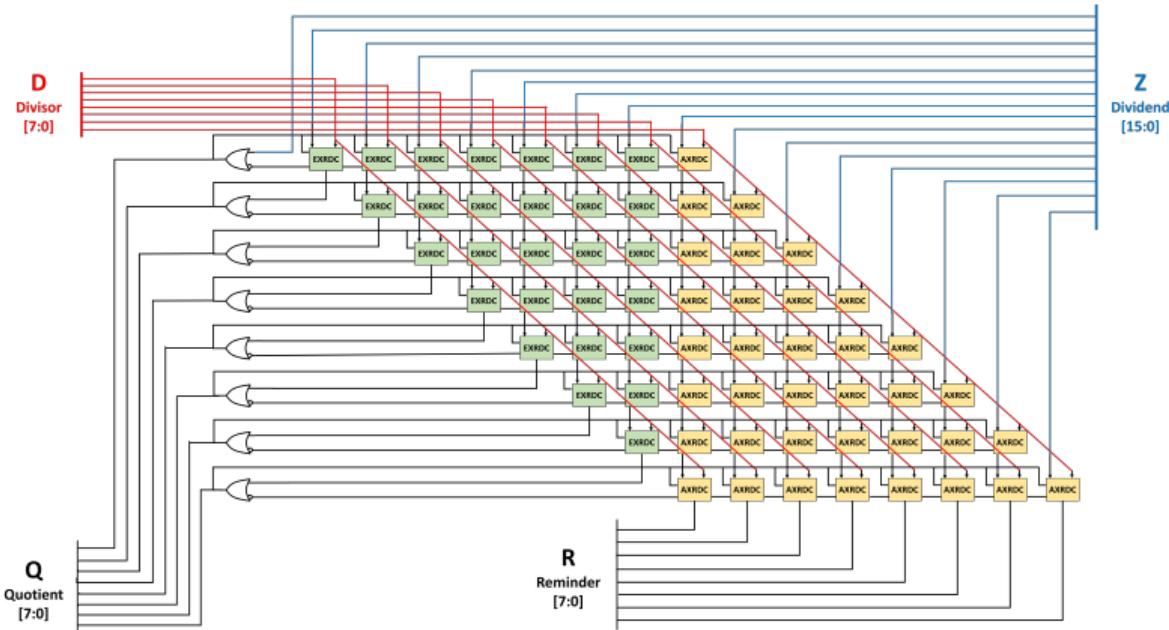


Figure 5 Architecture of 16/8 approximate non-pipelined array divider with $p=8$

HDL Simulation Results III

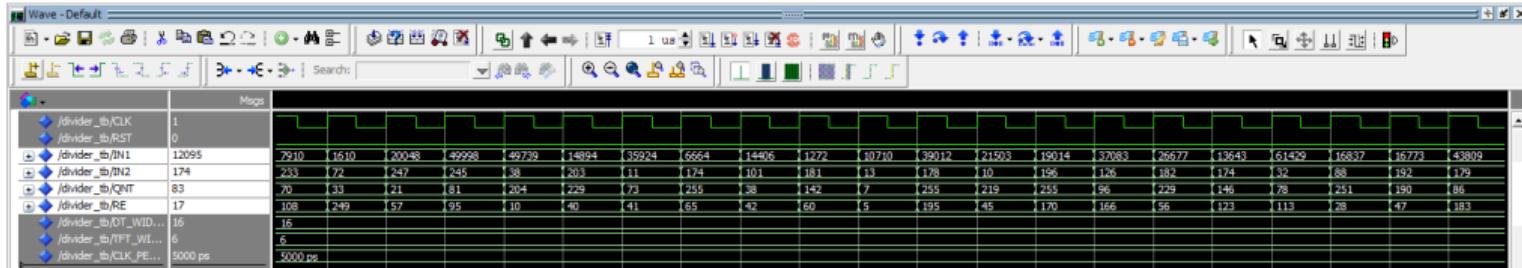


Figure 6 Simulation result of 16/8 approximate non-pipelined array divider with $p=6$

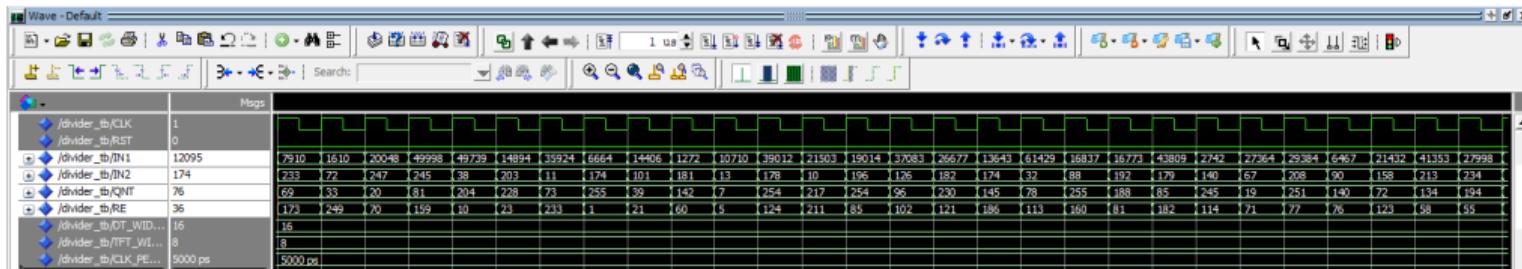


Figure 7 Simulation result of 16/8 approximate non-pipelined array divider with $p=8$

HDL Simulation Results IV

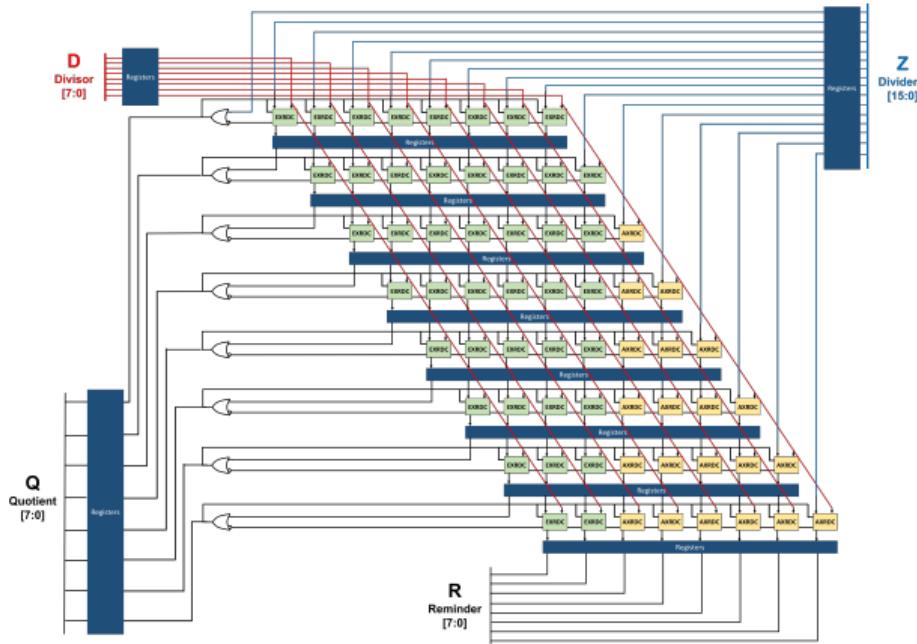


Figure 8 Architecture of 16/8 approximate pipelined array divider with $p=6$

HDL Simulation Results V

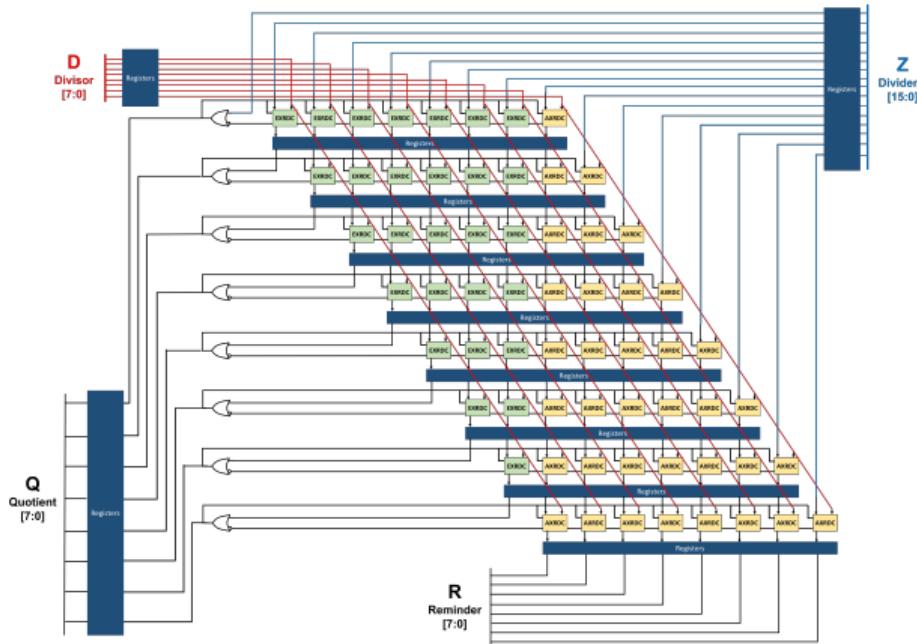


Figure 9 Architecture of 16/8 approximate pipelined array divider with $p=8$

HDL Simulation Results VI

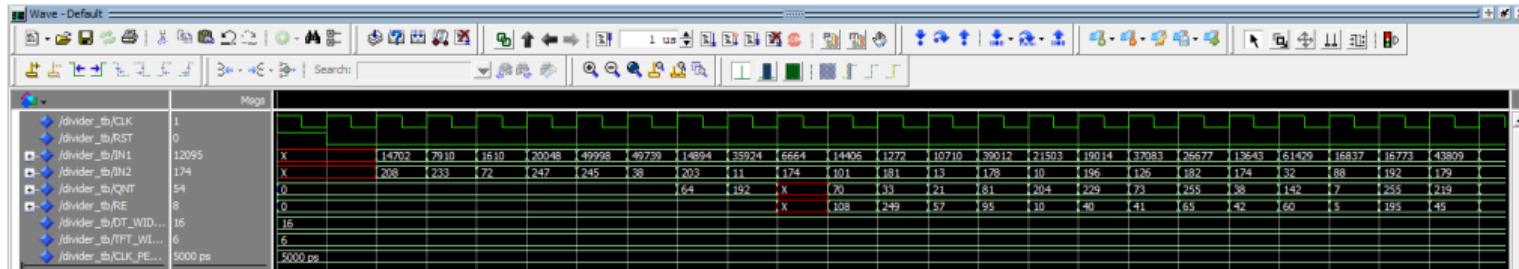


Figure 10 Simulation result of 16/8 approximate pipelined array divider with $p=6$

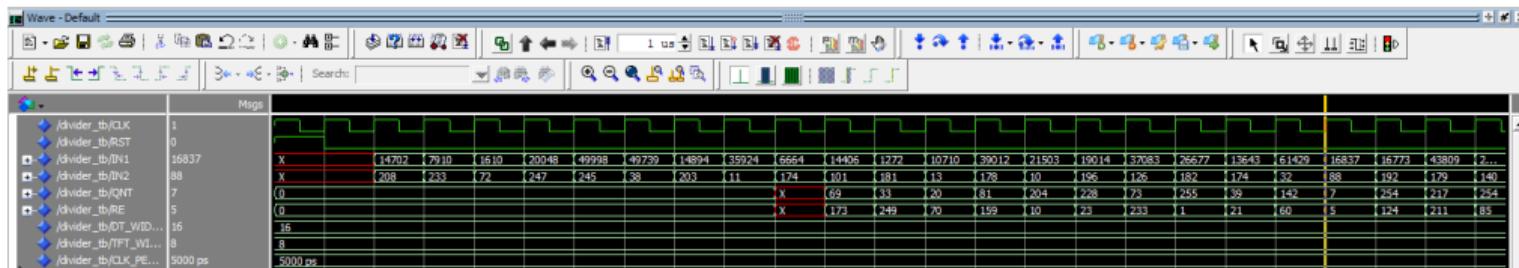


Figure 11 Simulation result of 16/8 approximate pipelined array divider with $p=8$

HDL Synthesis Results I

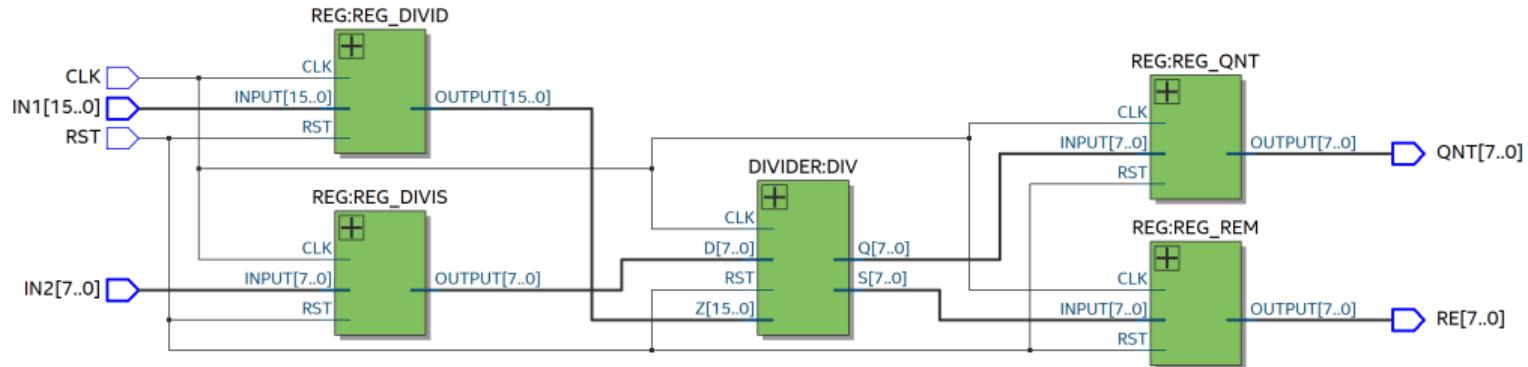


Figure 12 RTL View of synthesised design (16/8 divider) (Top View)

HDL Synthesis Results II

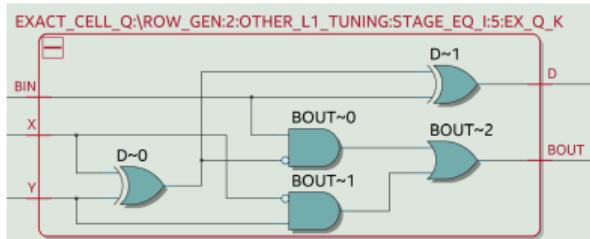
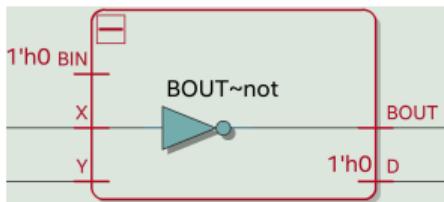
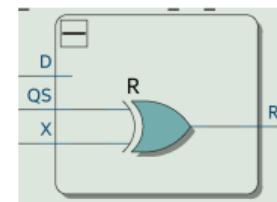


Figure 13 RTL View of Exact Divider Cell (EXRDC)



(a) Inexact Quotient Cell (Q)



(b) Inexact Reminder Cell (R)

Figure 14 RTL View of Inexact Divider Cell (AXRDC)

HDL Synthesis Results III

Table 3 Summary of Resource Utilisation and Clock Performance of 16/8 Approximate Array Divider

Tuning Factor (p)		0 (Exact)	4	6	8
Error Rate (%)			5.07	23.40	66.24
Without Pipeline	Freq. Max (MHz)		31.92	38.02	42.75
	Resources	ALUTs	163	141	112
		Regs	40	40	40
Resource Savings (%)			10.83	25.13	44.34
Path Delay Reduction (%)			19.12	33.93	82.15
With Pipeline	Freq. Max (MHz)		136.63	135.24	159.77
	Resources	ALUTs	112	114	107
		Regs	224	214	203
Resource Savings (%)			2.38	7.74	15.18
Path Delay Reduction (%)			-1.01	16.94	41.30

HDL Synthesis Results IV

Table 4 Summary of Resource Utilisation and Clock Performance of 32/16 Approximate Array Divider

Tuning Factor (p)			0 (Exact)	12	15	18
Without Pipeline	Freq. Max (MHz)		13.43	15.71	19.86	22.05
	Resources		ALUTs 498	427	350	255
	Resource Savings (%)		80	80	80	80
	Path Delay Reduction (%)			12.28	25.61	42.04
With Pipeline	Freq. Max (MHz)		92.66	98.08	103.68	78.87
	Resources		ALUTs 460	394	360	332
	Resource Savings (%)		Regs 832	754	712	667
	Path Delay Reduction (%)			11.15	17.03	22.68
				5.85	11.90	-14.48

Outline

- 1** Introduction
- 2** Approximate Divider Approaches
- 3** Design Evaluation
- 4** Experimental Results
- 5** Conclusion and Outlook
- 6** References

Conclusion

- The state-of-the-art approximate versions of basic arithmetic circuits such as dividers will reduce development time and effort significantly.
- Nonpipelined array-based inexact dividers are suitable for resource constrained, low speed applications (resource savings $\approx 30\%$).
- Pipelined dividers are best suited for high-speed, resource rich applications ($F_{max} > 150$ MHz).
- The increase in the size of the inputs adversely impacts the performance of an array-based divider. This can be attributed to the larger critical path of the design.
- Overcoming this increase in the critical path and achieving higher speeds for larger array-based dividers can be a part of future research.

Outline

- 1** Introduction
- 2** Approximate Divider Approaches
- 3** Design Evaluation
- 4** Experimental Results
- 5** Conclusion and Outlook
- 6** References

References I

-  Jie Han and Michael Orshansky. Approximate computing: An emerging paradigm for energy-efficient design. In 2013 18th IEEE European Test Symposium (ETS), pages 1–6, 2013.
-  H. Jiang, F. J. H. Santiago, H. Mo, L. Liu and J. Han, "Approximate Arithmetic Circuits: A Survey, Characterization, and Recent Applications," in Proceedings of the IEEE, vol. 108, no. 12, pp. 2108-2135, Dec. 2020.
-  Omkar G. Ratnaparkhi and Madhav Rao. 2022. LEAD: Logarithmic Exponent Approximate Divider For Image Quantization Application. In Proceedings of the Great Lakes Symposium on VLSI 2022 (GLSVLSI '22). Association for Computing Machinery, New York, NY, USA, 437–442.
-  Y. Wu et al., "An Energy-Efficient Approximate Divider Based on Logarithmic Conversion and Piecewise Constant Approximation," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 7, pp. 2655-2668, July 2022.
-  E. Adams, S. Venkatachalam and S. -B. Ko, "Approximate Restoring Dividers Using Inexact Cells and Estimation From Partial Remainders," in IEEE Transactions on Computers, vol. 69, no. 4, pp. 468-474, 1 April 2020.
-  J. Melchert, S. Behroozi, J. Li and Y. Kim, "SAADI-EC: A Quality-Configurable Approximate Divider for Energy Efficiency," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 11, pp. 2680-2692, Nov. 2019.

References II

-  C. K. Jha, K. Prasad, V. K. Srivastava and J. Mekie, "FPAD: A Multistage Approximation Methodology for Designing Floating Point Approximate Dividers," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 2020.
-  S. Vahdat, M. Kamal, A. Afzali-Kusha, M. Pedram and Z. Navabi, "TruncApp: A truncation-based approximate divider for energy efficient DSP applications," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, Switzerland, 2017, pp. 1635-1638.
-  R. Zendegani, M. Kamal, A. Fayyazi, A. Afzali-Kusha, S. Safari and M. Pedram, "SEERAD: A high speed yet energy-efficient rounding-based approximate divider," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 2016, pp. 1481-1484.
-  Marzieh Vaeztourshizi, Mehdi Kamal, Ali Afzali-Kusha, and Massoud Pedram. 2018. An Energy-Efficient, Yet Highly-Accurate, Approximate Non-Iterative Divider. In Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '18). Association for Computing Machinery, New York, NY, USA, Article 14, 1–6.
-  M. Imani, R. Garcia, A. Huang and T. Rosing, "CADE: Configurable Approximate Divider for Energy Efficiency," 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 2019, pp. 586-589.

Thank You. Any Questions?