DRUM: A Dynamic Range Unbiased Multiplier for Approximate Applications

Team 6 - Anoushka Vyas - 20171057

Digital VLSI Design - 2021

Overview I

Introduction

Abstract

- The novel approach DRUM focuses on creating a multiplier which has unbiased error distribution, where the errors cancel out on repeated computations instead of accumulating.
- DRUM also uses a smaller multiplier than accurate multiplier thus saving on area and computation power.
- The design is flexible to incorporate any kind of multiplier for the purpose.

Preliminaries

- **Leading One Detector:** Leading One Detectors or LODs determine the location of the most significant one or a leading bit in a given binary.
- **Priority Encoder:** The priority encoders output corresponds to the currently active input which has the highest priority.
- **Multiplexer:** Multiplexer is a device which controls which signal will go on a common transmission line through a switch.
- **Barrel Shifter:** A barrel shifter is a combinational circuit that shifts a data word by a specified number of bits.
- **Wallace-Tree Multiplier:** Wallace-Tree multiplier is an efficient parallel multiplier.

DRUM - Design

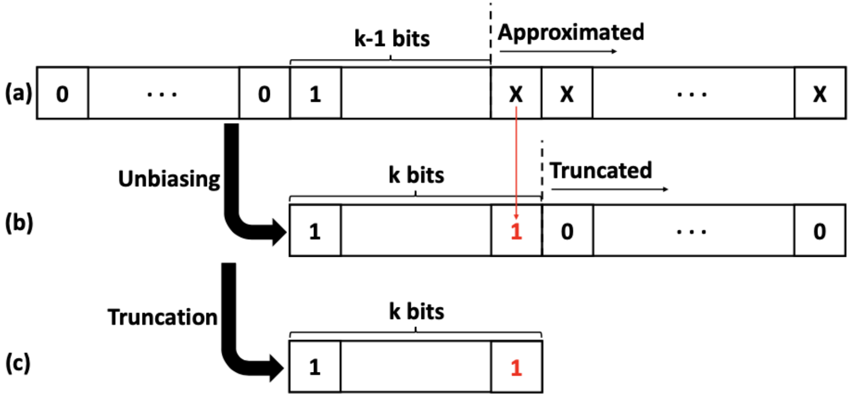

Figure: An example of the approximation process (a) Original number (b) Number after unbiasing (c) Final approximated input

DRUM - Design

(a)						0001	**0111**	0100	1101
	x					0000	0001	**0101**	1010
(b)									101111
	x								101011
(c)							0111	1110	0101
(d)		0000	0000	0001	1111	1001	0100	0000	0000
(e)		0000	0000	0001	1111	0111	1110	0001	0010

Figure: A working example of DRUM ($n = 16$, $k = 6$) (a) The input numbers (b) Approximated inputs (c) Result before shifting (d) Approximate result (e) Accurate result

DRUM - Design

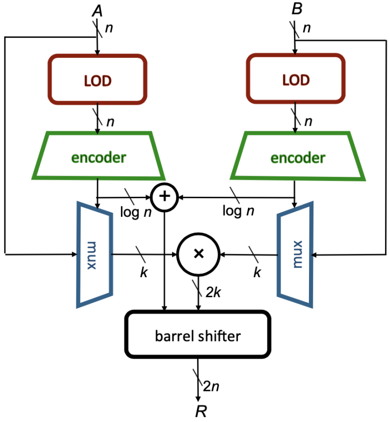

Figure: The simplified schematics for DRUM

DRUM - Error Analysis

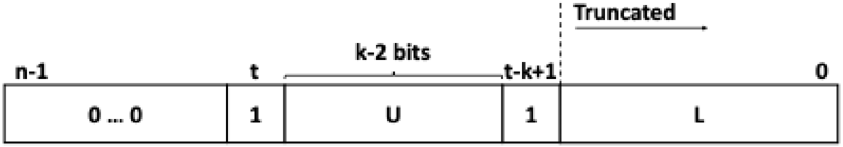

Figure: Operand A as used in error analysis

For operand A, the introduced error relative to the number is given by:

$$Err(A) = \begin{cases} X(U,L,t) = \frac{L}{2^t + (2U+1).2^{t-k+1} + L} & if \quad A[t-k+1] = 1 \\ Y(U,L,t) = \frac{L - 2^{t-k+1}}{2^t + 2U.2^{t-k+1} + L} & if \quad A[t-k+1] = 0 \end{cases} \quad (1)$$

DRUM - Error Analysis

The maximum positive error (MPE) happens when L, and therefore t, is maximized and U is minimized. Thus, the MPE is equal to

$$\text{MPE} = \frac{2^{n-k} - 1}{2^{n-1} + 2^{n-k} - 1}, \tag{2}$$

and the maximum negative error (MNE) happens for minimum values of L, U and t, and therefore it is equal to

$$\text{MNE} = 2^{-k+1}. \tag{3}$$

Using Equation (2) and Equation (3), the maximum truncation error (MTE) is given by Equation (4):

$$\text{MTE} = \max\{\text{MPE}, \text{MNE}\} \tag{4}$$

DRUM - Error Analysis

If the input operands, A and B, are independent, the multiplication error can now be characterized in terms of truncation errors. Therefore, for maximum multiplication error (MME):

$$\mathrm{MME} = \begin{cases} 2\mathrm{MPE} - \mathrm{MPE}^2 & \textit{if} \quad \mathrm{MPE} > \mathrm{MNE} \\ 2\mathrm{MNE} + \mathrm{MNE}^2 & \textit{if} \quad \mathrm{MPE} < \mathrm{MNE} \end{cases} \tag{5}$$

DRUM - Error Analysis

To calculate the expected error, a uniform distribution is assumed on all the operands and it is calculated as:

$$E(Err(A)) = \frac{1}{2^n} \sum_{t=k}^{n-2} \sum_{L=0}^{2^{t-k+1}-1} \sum_{U=0}^{2^{k-2}-1} (X(U, L, t) + Y(U, L, t)) \tag{6}$$

Finally the expected multiplication error is given by

$$E[Err(AB)] = E[Err(A)] + E[Err(B)] + E[Err(A)].E[Err(B)]. \tag{7}$$

Experimental Analysis - Multiplier Results

The DRUM multiplier is run for two set of operands, one is a smaller 16-bit operand set of 11 and 55 and another one is a bigger 16-bit operand set of 3000 and 125.

16- bit numbers	Accurate	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$
11 & 5	55	55	55	55	55	55	55
3000 & 125	375000	337920	365056	379008	372000	374000	375000

Table: Result of DRUM for various operands for different values of k

Experimental Analysis - Changing k

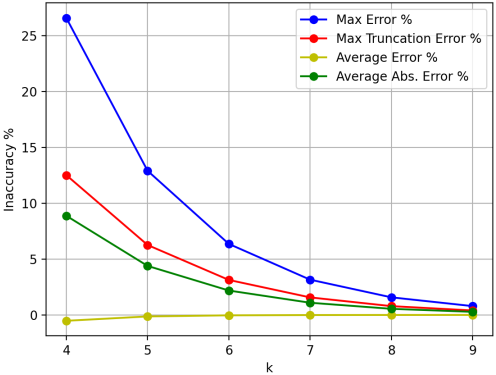

Figure: Errors as given by the equations (4), (5), and (7) for different values of k ($n = 16$)

Experimental Analysis - Changing k

First consider $n = 16$ for a 16-bit multiplier and examine the impact of changing the range as controlled by k.

Errors	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$
Max Error %	26.56	12.89	6.35	3.15	1.57	0.78
Max Truncation Error %	12.50	6.25	3.12	1.56	0.78	0.39
Average Error %	-0.53	-0.14	-0.04	-0.02	-0.01	-0.01
Average Abs. Error %	8.86	4.38	2.18	1.08	0.54	0.27

Table: Accuracy results for different k ($n = 16$)

Experimental Analysis - Changing n

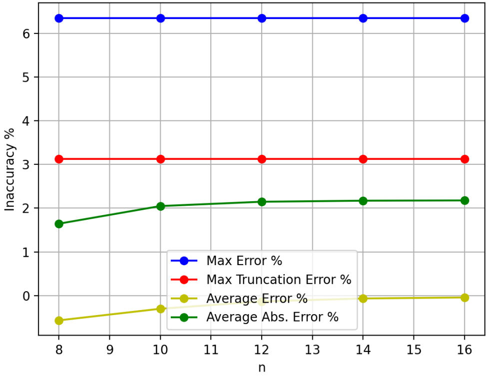

Figure: Errors as given by the equations (4), (5), (7) for different values of n ($k = 6$)

Experimental Analysis - Changing n

Next, the impact of changing the multiplier size n is studied. The table summarizes the accuracy results for the case of $n = 8, 10, 12, 14$ and 16.

Errors	$n = 8$	$n = 10$	$n = 12$	$n = 14$	$n = 16$
Max Error %	6.35	6.35	6.35	6.35	6.35
Max Truncation Error %	3.12	3.12	3.12	3.12	3.12
Average Error %	-0.57	-0.30	-0.13	-0.07	-0.04
Average Abs. Error %	1.64	2.05	2.15	2.17	2.18

Table: Accuracy results for input size n ($k = 6$)

Experimental Analysis - Error Distribution

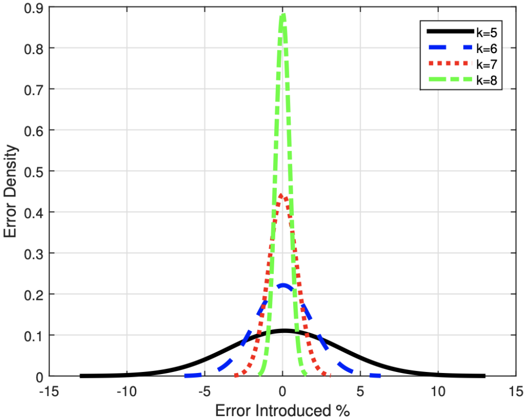

Figure: The fitted Gaussian error distributions for DRUM for different k ($n = 16$)

Experimental Analysis - Power and LUT Analysis

Multiplier	Power(μW)	LUTs
Wallace Tree Multiplier (Signed)	1421	455
Wallace Tree Multiplier (Unsigned)	1326	328
DRUM (Signed)	1307	249
DRUM (Unsigned)	1240	191

Table: Power and LUTs required after synthesising in **Xylinx Vivado** for $n = 6$ and $k = 6$

Applications - Image Filtering

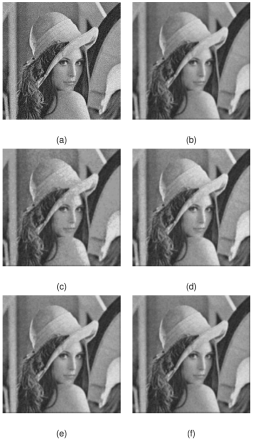

Figure: Gaussian filtering results for different values of k (a) Input image (b) Filtered with accurate multiplier (c) $k = 3$ (d) $k = 4$ (e) $k = 5$ (f) $k = 6$

Applications - JPEG Compression

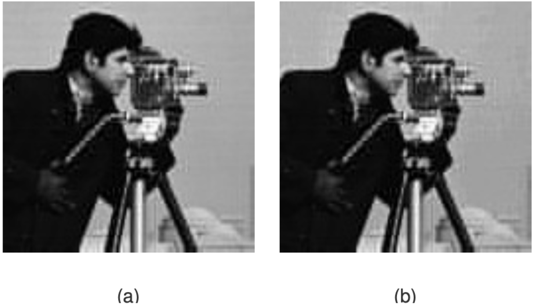

Figure: JPEG compression algorithm (a) Compressed using accurate multiplier (b) Compressed using DRUM

Applications - Perceptron Classifier

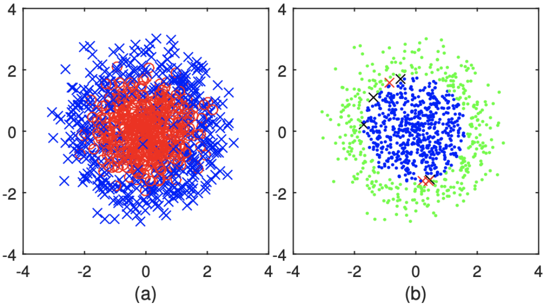

Figure: Perceptron classifier (a) Input dataset with classes -1 (blue), 1 (red) (b) The outputs of accurate and approximate multipliers (dots:matching classification, crosses:mismatch, red:Additional detection, black:False alarm)

Conclusion

Conclusion

- It is observed that with changing k, the error shows an exponential behaviour which is obvious from the fact that a bit change corresponds to a change in power of 2.
- The Gaussian error distribution plot shows that with increasing k there is less error in multiplication.
- Overall, this method is a good exploitation of that fact that some applications are tolerant to small errors, as a result, at the cost of small error, power and area can be optimized.

Thank You