Approximate Multiplication and Squaring Circuits with Parallel Correction

Patricio Bulić[1], Jason Newman[2]

[1]Faculty of Computer and Information Science
University of Ljubljana

[2]Faculty of Electrical Engineering
University of Suburbia

University of Ljubljana
Faculty of Computer and Information Science

Agenda

- Motivation
- Approximate Multiplier and Squarer
- Hardware Implementation
- Error Analysis
- Conclusions

Introduction

- Implementation bla bla:
  - bla bla bla
  - higher performance at smaller power consumption

Introduction

- Implementation bla bla:
  - bla bla bla
  - higher performance at smaller power consumption
- Some beautiful chips:
  - noise and learning ability of analogue and digital designs
  - ASIC and FPGA (field programmable gate array)

Introduction

- Implementation bla bla:
  - bla bla bla
  - higher performance at smaller power consumption
- Some beautiful chips:
  - noise and learning ability of analogue and digital designs
  - ASIC and FPGA (field programmable gate array)
- Signal processing applications incorporate complex algorithms with many multiplications
  - Multiplication is area, power and time consuming operation
  - bla bla bla

Approximate Multiplier

- An approximate multiplier, introduced by Babic et al. (2010):
  - Reduced usage of logic resources: one adder and a shifter
  - Reduced power consumption
- The product of the two numbers, N_1 and N_2

$$N_1 \cdot N_2 \quad = \quad (2^{k_1} + N_1^{(1)}) \cdot (2^{k_2} + N_2^{(1)})$$

Approximate Multiplier

- An approximate multiplier, introduced by Babic et al. (2010):
  - Reduced usage of logic resources: one adder and a shifter
  - Reduced power consumption
- The product of the two numbers, N_1 and N_2

$$\begin{aligned} N_1 \cdot N_2 &= (2^{k_1} + N_1^{(1)}) \cdot (2^{k_2} + N_2^{(1)}) \\ &= 2^{k_1+k_2} \end{aligned}$$

1. shift left the leading "1" from the first number by k_2 places

Approximate Multiplier

- An approximate multiplier, introduced by Babic et al. (2010):
  - Reduced usage of logic resources: one adder and a shifter
  - Reduced power consumption
- The product of the two numbers, N_1 and N_2

$$
\begin{aligned}
N_1 \cdot N_2 &= (2^{k_1} + N_1^{(1)}) \cdot (2^{k_2} + N_2^{(1)}) \\
&= 2^{k_1+k_2} + N_1^{(1)} \cdot 2^{k_2}
\end{aligned}
$$

1. shift left the leading "1" from the first number by k_2 places
2. shift left the first remainder by k_2 places

Approximate Multiplier

- An approximate multiplier, introduced by Babic et al. (2010):
  - Reduced usage of logic resources: one adder and a shifter
  - Reduced power consumption
- The product of the two numbers, N_1 and N_2

$$
\begin{aligned}
N_1 \cdot N_2 &= (2^{k_1} + N_1^{(1)}) \cdot (2^{k_2} + N_2^{(1)}) \\
&= 2^{k_1+k_2} + N_1^{(1)} \cdot 2^{k_2} + N_2^{(1)} \cdot 2^{k_1}
\end{aligned}
$$

1. shift left the leading "1" from the first number by k_2 places
2. shift left the first remainder by k_2 places
3. shift left the second remainder by k_1 places

Approximate Multiplier

- An approximate multiplier, introduced by Babic et al. (2010):
  - Reduced usage of logic resources: one adder and a shifter
  - Reduced power consumption
- The product of the two numbers, N_1 and N_2

$$\begin{aligned} N_1 \cdot N_2 &= (2^{k_1} + N_1^{(1)}) \cdot (2^{k_2} + N_2^{(1)}) \\ &= 2^{k_1+k_2} + N_1^{(1)} \cdot 2^{k_2} + N_2^{(1)} \cdot 2^{k_1} + N_1^{(1)} \cdot N_2^{(1)} \end{aligned}$$

1. shift left the leading "1" from the first number by k_2 places
2. shift left the first remainder by k_2 places
3. shift left the second remainder by k_1 places
4. multiply the two remainders

Approximate Multiplier

- An approximate multiplier, introduced by Babic et al. (2010):
  - Reduced usage of logic resources: one adder and a shifter
  - Reduced power consumption
- The product of the two numbers, N_1 and N_2

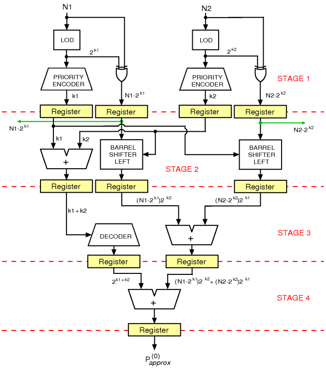
$$\begin{aligned} N_1 \cdot N_2 &= (2^{k_1} + N_1^{(1)}) \cdot (2^{k_2} + N_2^{(1)}) \\ &= 2^{k_1+k_2} + N_1^{(1)} \cdot 2^{k_2} + N_2^{(1)} \cdot 2^{k_1} + N_1^{(1)} \cdot N_2^{(1)} \end{aligned}$$

1. shift left the leading "1" from the first number by k_2 places
2. shift left the first remainder by k_2 places
3. shift left the second remainder by k_1 places
4. multiply the two remainders

Hardware Implementation

- A Basic block implements the approximate product

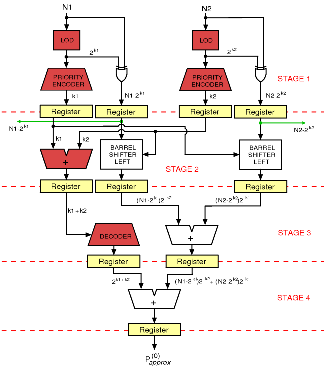
  $(N_1 \cdot N_2)_{approx} =$



Hardware Implementation

- A Basic block implements the approximate product

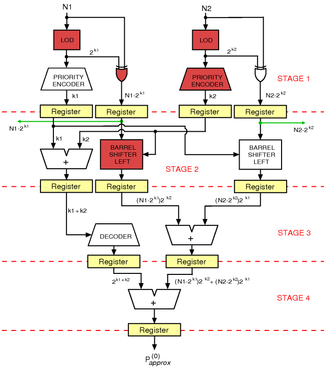
$$(N_1 \cdot N_2)_{approx} = 2^{k_1+k_2}$$



Hardware Implementation

- A Basic block implements the approximate product

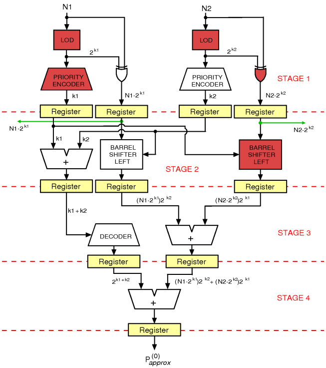
$$(N_1 \cdot N_2)_{approx} = 2^{k_1+k_2} N_1^{(1)} \cdot 2^{k_2}$$



Hardware Implementation

- A Basic block implements the approximate product

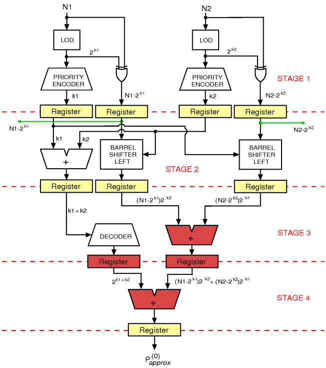
$$(N_1 \cdot N_2)_{approx} = 2^{k_1+k_2}$$
$$N_1^{(1)} \cdot 2^{k_2}$$
$$N_2^{(1)} \cdot 2^{k_1}$$



Hardware Implementation

- A Basic block implements the approximate product

$$
\begin{aligned}
(N_1 \cdot N_2)_{approx} &= 2^{k_1+k_2} \\
&+ N_1^{(1)} \cdot 2^{k_2} \\
&+ N_2^{(1)} \cdot 2^{k_1}
\end{aligned}
$$



Error Analysis

Theorem

The probability of an error in the circuit is directly proportional to the trouble it can cause.

Error Analysis

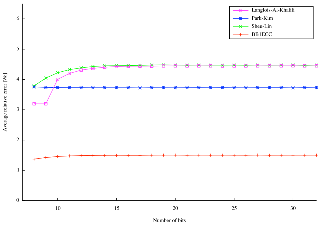
Theorem

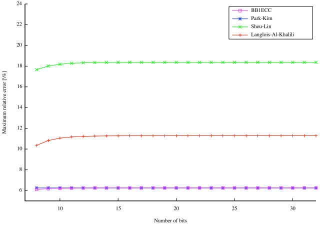
The probability of an error in the circuit is directly proportional to the trouble it can cause.

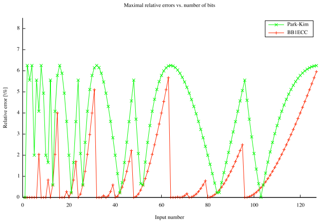
Proof.

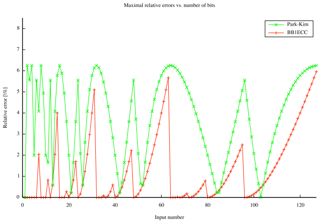
The proof is straightforward. □

Errors


Langlois-Al-Khalili
Park-Kim
Sheu-Lin
BB1ECC
Average relative error [%]
Number of bits


BB1ECC
Park-Kim
Sheu-Lin
Langlois-Al-Khalili
Maximum relative error [%]
Number of bits


Maximal relative errors vs. number of bits
Park-Kim
BB1ECC
Relative error [%]
Input number


Maximal relative errors vs. number of bits
Park-Kim
BB1ECC
Relative error [%]
Input number

Conclusions

- The proposed approach improves the average and maximum relative errors compared to the existing square approximations.

Conclusions

- The proposed approach improves the average and maximum relative errors compared to the existing square approximations.
- Error analysis has shown that an error in the circuit is directly proportional to the trouble it can cause.

Conclusions

- The proposed approach improves the average and maximum relative errors compared to the existing square approximations.
- Error analysis has shown that an error in the circuit is directly proportional to the trouble it can cause.
- We can calculate the correction terms in parallel.