Distributed Arithmetic for Taylor-Series Expansion

Xaybandith Hemthavy

J. Wei, S. Katayama, A. Kuwana,

H. Kobayashi, K. Kubo

Gunma University

Oyama National College of Technology

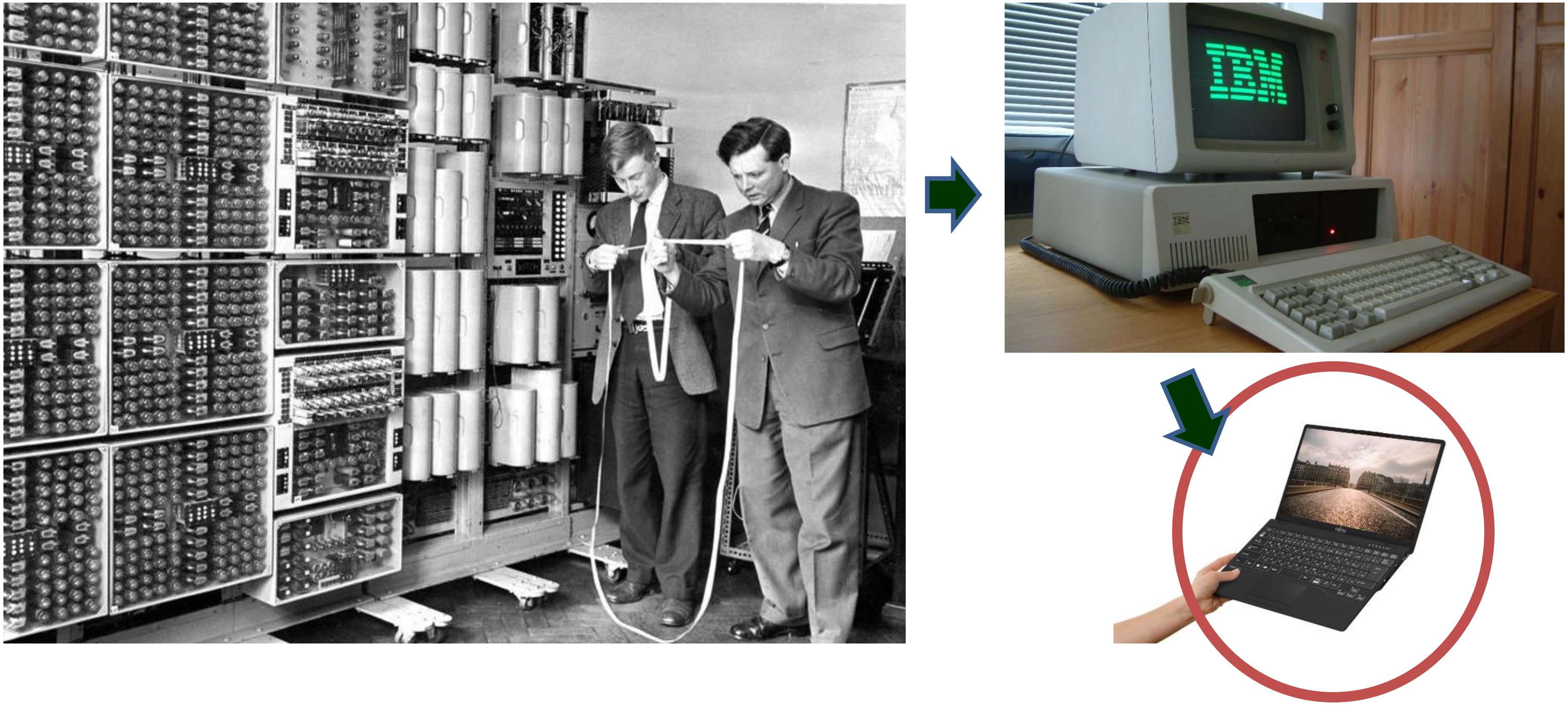
Contents

Contents

Changes in Electronic Devices

PC, Phone…Electronic Devices

Smaller



Research Objective

Find algorithm that can

Make small circuit to

Calculate Taylor-Series Expansion

$$\boldsymbol{f(x) = f(\alpha) + \frac{f'(\alpha)}{1!}(x-\alpha) + \frac{f''(\alpha)}{2!}(x-\alpha)^2 + \frac{f'''(\alpha)}{3!}(x-\alpha)^3 + \cdots}$$

Contents

- **Outline**
  - Research Objective
  - **Problem and Approach**
- How to do “Distributed Arithmetic”
- Design & Evaluation
- Conclusion

Taylor-Series Expansion and DA (Problem 1)

DA: Distributed Arithmetic

Taylor-Series Expansion

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7 + a_8 x^8 + a_9 x^9$$

$$\Downarrow$$

$$f(x) = a_0 + \underbrace{x^1}_{}\left(a_1 + \underbrace{x^1}_{}\left(a_2 + \underbrace{x^1}_{}\left(a_3 + \underbrace{x^1}_{}\left(a_4 + \underbrace{x^1}_{}\left(a_5 + \underbrace{x^1}_{}\left(a_6 + \underbrace{x^1}_{}\left(a_7 + \underbrace{x^1}_{}\left(a_8 + \underbrace{a_9 x^1}_{}\right)\right)\right)\right)\right)\right)\right)\right)$$

9 Multiplications

$\Downarrow$ Distributed Arithmetic

Use Multiplier → $\underline{x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9}$ for Look Up Table

8 Multiplications

Large number of multiplications

Taylor-Series Expansion and DA (Problem 2)

DA: Distributed Arithmetic

Taylor-Series Expansion

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7 + a_8 x^8 + a_9 x^9$$

⇓ Distributed Arithmetic

$x^0, x^1, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9$ for Look Up Table

Look Up Table

x^0	x^1	x^2	…	x^9	Output Data
0	0	0	…	0	0
1	0	0	…	0	a_0
0	1	0	…	0	a_1
…	…	…	…	…	…
1	1	1	…	1	$a_0 + a_1 + a_2 + \cdots + a_9$

2^{10} ↓ **1024**

10

Large size LUT

Our Approach

$$f(x) = \underbrace{(a_0 + a_2x^2 + a_4x^4 + a_6x^6 + a_8x^8)}_{\text{DA}} + x^1\underbrace{(a_1 + a_3x^2 + a_5x^4 + a_7x^6 + a_9x^8)}_{\text{DA}}$$

Multiplication: x^2, x^4, x^6, x^8, $x^1(a_1 + \cdots + a_9x^8)$

5 times

Look Up Table (Left DA)

x^0	x^2	x^4	x^6	x^8	Output Data
0	0	0	0	0	0
1	0	0	0	0	a_0
0	1	0	0	0	a_2
…	…	…	…	…	…
1	1	1	1	1	$a_0 + a_2 + a_4 + a_6 + a_8$

2^5 ↓ 32

5

Look Up Table (Right DA)

x^0	x^2	x^4	x^6	x^8	Output Data
0	0	0	0	0	0
1	0	0	0	0	a_1
0	1	0	0	0	a_3
…	…	…	…	…	…
1	1	1	1	1	$a_1 + a_3 + a_5 + a_7 + a_9$

Smaller number of multiplications

&

Smaller size memory

How to Reduce Multiplication?

Distributed Arithmetic

Multiplications / Additions

↓

LUT + Adder

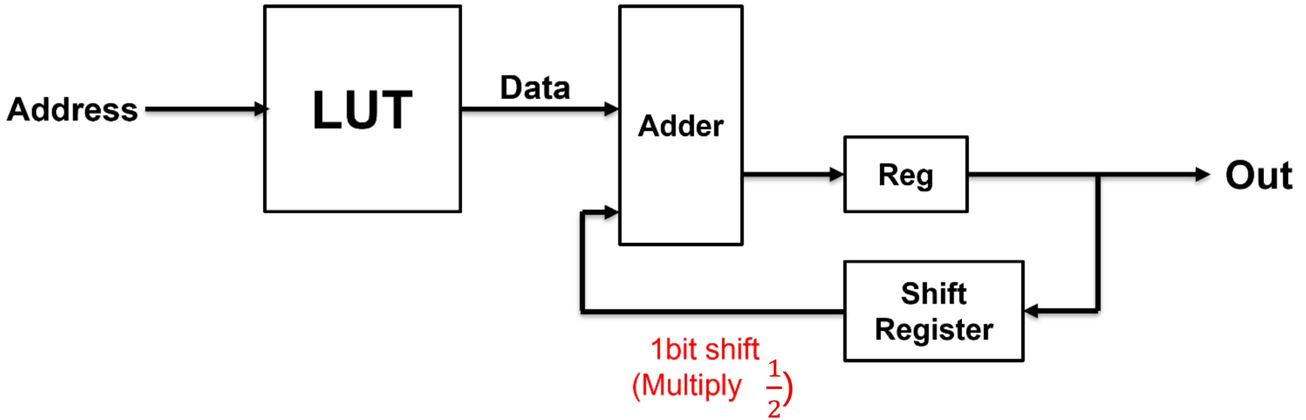
Without Multiplier

Contents

- Outline
  - Research Objective
  - Problem and Approach
- **How to do “Distributed Arithmetic”**
- Design & Evaluation
- Conclusion

Flow of “Distributed Arithmetic”

- Look Up Table (LUT)
- Calculate x^n
- Bit-shift and Addition
- Correct the position of Decimal Point



Look Up Table (LUT)

$$f(x) = a_0 + a_1x + a_2x^2$$

LUT			
x^0	x^1	x^2	**Sum**
0	0	0	0
1	0	0	a_0
0	1	0	a_1
0	0	1	a_2
1	1	0	$a_0 + a_1$
1	0	1	$a_0 + a_2$
0	1	1	$a_1 + a_2$
1	1	1	$a_0 + a_1 + a_2$

Calculate x^n

$$f(x) = a_0 + a_1 x + a_2 x^2$$

If Decimal $x = 1.5$ → Binary $x = 1.1$

x^0	0	1.	0	0
x^1	0	1.	1	0
x^2	1	0.	0	1

x^0	x^1	x^2
0	0	1
1	1	0
0	1	0
0	0	1

Bit-shift and Addition

$$f(x) = a_0 + a_1 x + a_2 x^2$$

	x^0	x^1	x^2
MSB	0	0	1
	1	1	0
	0	1	0
LSB	0	0	1

LUT			
x^0	x^1	x^2	**Sum**
0	0	0	0
1	0	0	a_0
0	1	0	a_1
0	0	1	a_2
1	1	0	$a_0 + a_1$
1	0	1	$a_0 + a_2$
0	1	1	$a_1 + a_2$
1	1	1	$a_0 + a_1 + a_2$

Bit-shift and Addition (Start from LSB)

$$f(x) = a_0 + a_1 x + a_2 x^2$$

	x^0	x^1	x^2
MSB	0	0	1
	1	1	0
	0	1	0
LSB	0	0	1

LUT			
x^0	x^1	x^2	**Sum**
0	0	0	0
1	0	0	a_0
0	1	0	a_1
0	0	1	a_2
1	1	0	$a_0 + a_1$
1	0	1	$a_0 + a_2$
0	1	1	$a_1 + a_2$
1	1	1	$a_0 + a_1 + a_2$

a_2

Bit-shift and Addition (LSB + 1)

$$\boldsymbol{f(x) = a_0 + a_1 x + a_2 x^2}$$

	x^0	x^1	x^2
MSB	0	0	1
	1	1	0
	0	1	0
LSB	0	0	1

LUT			
x^0	x^1	x^2	**Sum**
0	0	0	0
1	0	0	a_0
0	1	0	a_1
0	0	1	a_2
1	1	0	$a_0 + a_1$
1	0	1	$a_0 + a_2$
0	1	1	$a_1 + a_2$
1	1	1	$a_0 + a_1 + a_2$

$$\left(\begin{matrix}\frac{1}{2} \times (a_2) \\ +(a_1)\end{matrix}\right)$$

Bit-shift and Addition (LSB + 2)

$$f(x) = a_0 + a_1 x + a_2 x^2$$

	x^0	x^1	x^2
MSB	0	0	1
	1	1	0
	0	1	0
LSB	0	0	1

LUT			
x^0	x^1	x^2	**Sum**
0	0	0	0
1	0	0	a_0
0	1	0	a_1
0	0	1	a_2
1	1	0	$a_0 + a_1$
1	0	1	$a_0 + a_2$
0	1	1	$a_1 + a_2$
1	1	1	$a_0 + a_1 + a_2$

$$\left(\begin{array}{c}\frac{1}{2}\times\left(\begin{array}{c}\frac{1}{2}\times(a_2)\\ +(a_1)\end{array}\right)\\ +(a_0 + a_1)\end{array}\right)$$

Bit-shift and Addition (End at MSB)

$$f(x) = a_0 + a_1 x + a_2 x^2$$

	x^0	x^1	x^2
MSB	0	0	1
	1	1	0
	0	1	0
LSB	0	0	1

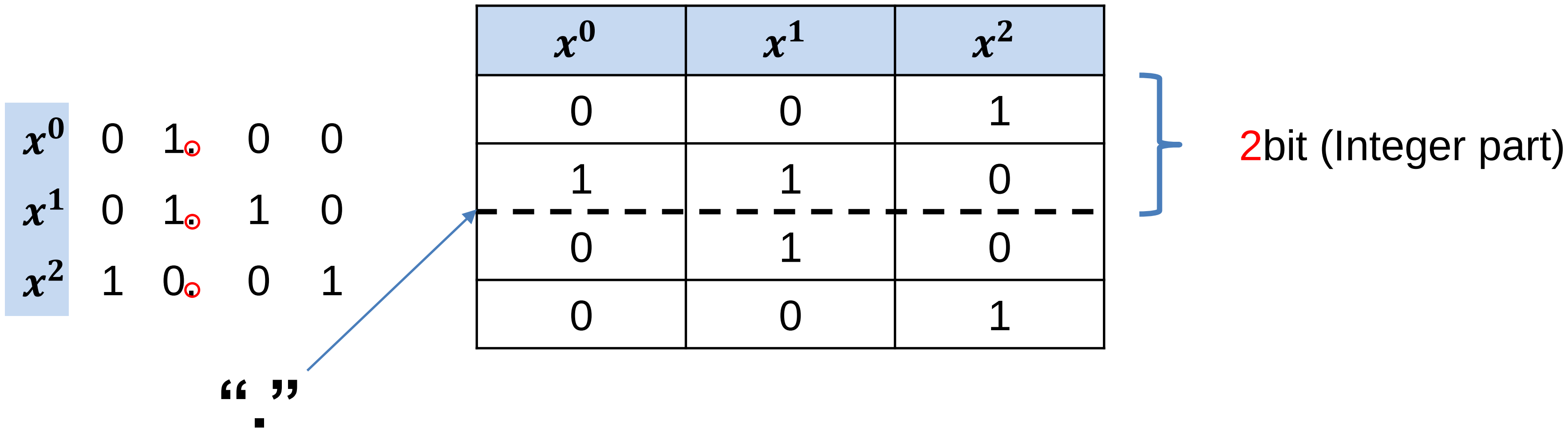
LUT

x^0	x^1	x^2	Sum
0	0	0	0
1	0	0	a_0
0	1	0	a_1
0	0	1	a_2
1	1	0	$a_0 + a_1$
1	0	1	$a_0 + a_2$
0	1	1	$a_1 + a_2$
1	1	1	$a_0 + a_1 + a_2$

$$\left(\begin{array}{c}\frac{1}{2} \times \left(\begin{array}{c}\frac{1}{2} \times \left(\begin{array}{c}\frac{1}{2} \times (a_2) \\ +(a_1)\end{array}\right) \\ +(a_0 + a_1)\end{array}\right) \\ +(a_2)\end{array}\right)$$

Correct the position of Decimal Point

$$f(x) = a_0 + a_1 x + a_2 x^2$$



x^0	0	1.	0	0
x^1	0	1.	1	0
x^2	1	0.	0	1

x^0	x^1	x^2
0	0	1
1	1	0
0	1	0
0	0	1

“.”

2bit (Integer part)

$$f(x) = \left(\frac{1}{2} \times \left(\frac{1}{2} \times \left(\frac{1}{2} \times (a_2) + (a_1) \right) + (a_0 + a_1) \right) + (a_2) \right) \times (2)^{2}$$

Contents

- Outline
  - – Research Objective
  - – Problem and Approach
- How to do “Distributed Arithmetic”
- **Design & Evaluation**
- Conclusion

Calculation Amount “Without DA”

$$f(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 + a_8x^8 + a_9x^9$$

Direct calculation without DA

$$f(x) = a_0 + \underbrace{x^1}\left(a_1 + \underbrace{x^1}\left(a_2 + \underbrace{x^1}\left(a_3 + \underbrace{x^1}\left(a_4 + \underbrace{x^1}\left(a_5 + \underbrace{x^1}\left(a_6 + \underbrace{x^1}\left(a_7 + \underbrace{x^1}\left(a_8 + a_9\underbrace{x^1}\right)\right)\right)\right)\right)\right)\right)\right)$$

9 Multiplications

DA with 10 terms

$$\boldsymbol{f(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 + a_8x^8 + a_9x^9}$$

Multiplication: x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9

8 times

Look Up Table

x^0	x^1	x^2	…	x^9	Output Data
0	0	0	…	0	0
1	0	0	…	0	a_0
0	1	0	…	0	a_1
…	…	…	…	…	…
1	1	1	…	1	$a_0 + a_1 + a_2 + \cdots + a_9$

2^{10} ↓ **1024**

10

DA with 5 terms × 2

$$\boldsymbol{f(x) = \underbrace{\left(a_0 + a_2x^2 + a_4x^4 + a_6x^6 + a_8x^8\right)}_{\text{DA}} + x^1\underbrace{\left(a_1 + a_3x^2 + a_5x^4 + a_7x^6 + a_9x^8\right)}_{\text{DA}}}$$

Multiplication: x^2, x^4, x^6, x^8, $x^1(a_1 + \cdots + a_9x^8)$

5 times

Look Up Table

x^0	x^2	x^4	x^6	x^8	Output Data
0	0	0	0	0	$0,\ 0$
1	0	0	0	0	$a_0,\ a_1$
0	1	0	0	0	$a_2,\ a_3$
…	…	…	…	…	…
1	1	1	1	1	$a_0+a_2+a_4+a_6+a_8,$ $a_1+a_3+a_5+a_7+a_9$

2^5 ↓ **32**

× 2

5

10 terms versus 5 terms × 2

Distributed Arithmetic
with **10 terms**

→

Distributed Arithmetic
with **5 terms × 2**

$\frac{1}{16}$ size of LUT

Almost half number of multiplications

$$f(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 + a_8x^8 + a_9x^9$$

$$f(x) = \underline{(a_0 + a_2x^2 + a_4x^4 + a_6x^6 + a_8x^8)} + x^1\underline{(a_1 + a_3x^2 + a_5x^4 + a_7x^6 + a_9x^8)}$$

Look Up Table

x^0	x^1	x^2	…	x^9	Output
1	0	0	…	0	a_0
0	1	0	…	0	a_1
0	0	1	…	0	a_2
…	…	…	…	…	…
1	1	1	…	1	$a_0 + a_1 + a_2 + \cdots + a_9$

2^{10} ↓ **1024**

10

Multiplication: x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9

8 times

x^0	x^2	x^4	x^6	x^8	Output Data	x^0	x^2	x^4	x^6	x^8	Output Data
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	a_0	1	0	0	0	0	a_1
0	1	0	0	0	a_2	0	1	0	0	0	a_3
…	…	…	…	…	…	…	…	…	…	…	…
1	1	1	1	1	$a_0 + a_2 + a_4 + a_6 + a_8$	1	1	1	1	1	$a_1 + a_3 + a_5 + a_7 + a_9$

2^5 ↓ **32**

5

Multiplication: x^2, x^4, x^6, x^8

5 times

Contents

- Outline
  - Research Objective
  - Problem and Approach
- How to do “Distributed Arithmetic”
- Design & Evaluation
- **Conclusion**

Conclusion

Calculation of Taylor-series expansion

Distributed arithmetic & **Division of terms**

⇒ Reduction of LUT memory size

$1024 \;\Rightarrow\; 32 \times 2$

⇒ Reduction of x^n calculation by Multiplier

9 ⇒ 5

Q & A

- How about dividing the original formula into 3 or 4 parts?
  - As considering 10 terms formula, the number of multiplication is the fewest when it’s divided into 2 parts.

- Are you planning to try this arithmetic in real circuit?
  - I’m not sure but I want to try.

References

[1] J. Wei, A. Kuwana, H. Kobayashi, K. Kubo,
“Divide and Conquer: Floating-Point Exponential Calculation
Based on Taylor-Series Expansion”,
IEEE 14th International Conference on ASIC (Oct. 2021).

[2] R. Bala, S. Aktar,
“Fast Fourier Transformation Realization with Distributed Arithmetic”,
International Journal of Computer Applications,
vol. 102, no. 15 (Sept. 2014).

Finally

Thank you all for listening.
It was a pleasure being here today.