ALU HW5 (Bonus): Truncated Multipliers

Outlines

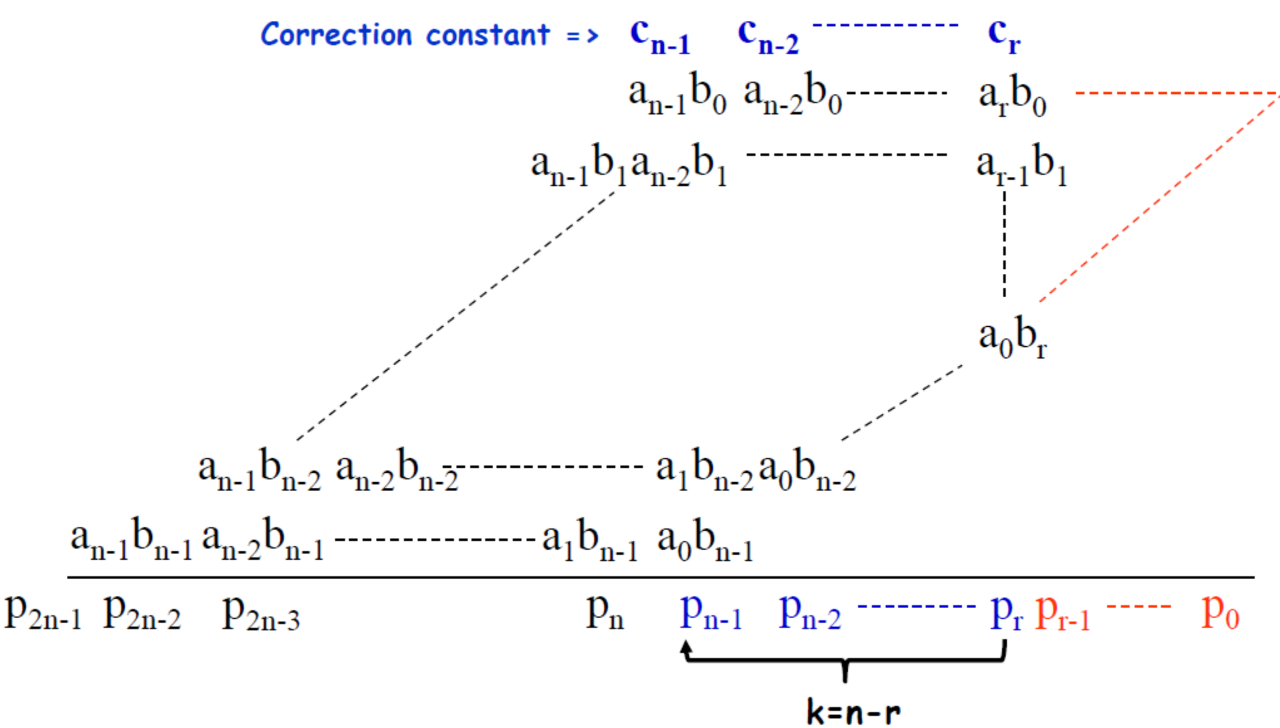
- truncated unsigned n x n multiplier (1 ulp=2^{n-1}) with n=8 and n=16, and output n-bit most significant half part
  - Verilog unsigned operator * (operator)
    - ✓ true output results
  - ppb row addition without any truncation (row)
  - constant-correction truncation with accumulation of partial product rows (trunc_const_row)
  - unsigned array multiplier without truncation (array)
  - variable-correction truncated array multiplier (trunc_var_array)

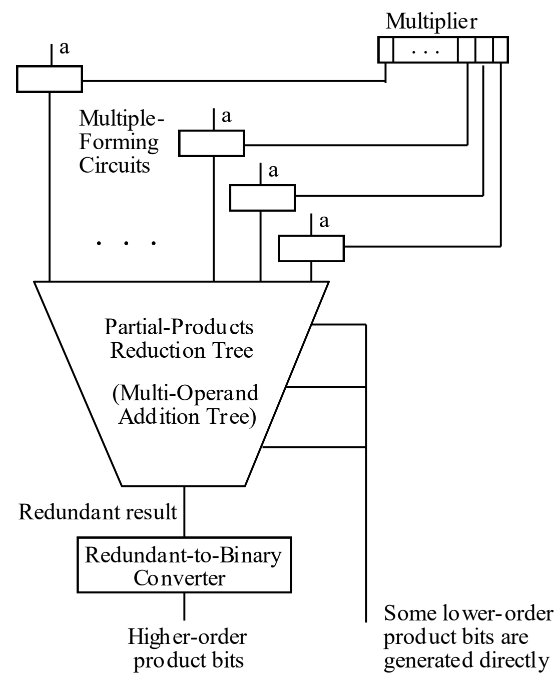
Truncated Multipliers

- Design truncated multipliers using two different correction methods
  - constant correction
  - variable correction
- two implementations
  - constant-correction with row-addition (trunc_const_row)
    - ✓ constant correction for CSA partial product reduction
  - variable-correction with array (trunc_var_array)
    - ✓ variable correction for array multiplier

Constant Correction with Row-Addition (trunc_const_row)

- accumulation of all rows, including the constant





- $r = n\text{-}k$ least significant columns are eliminated, a correction constant is added, and the product is truncated to n bits
- $k=n\text{-}r$ is the number of reserved columns for PP compression

Slide 118

Variable Correction with Row-Addition (trunc_var_row)

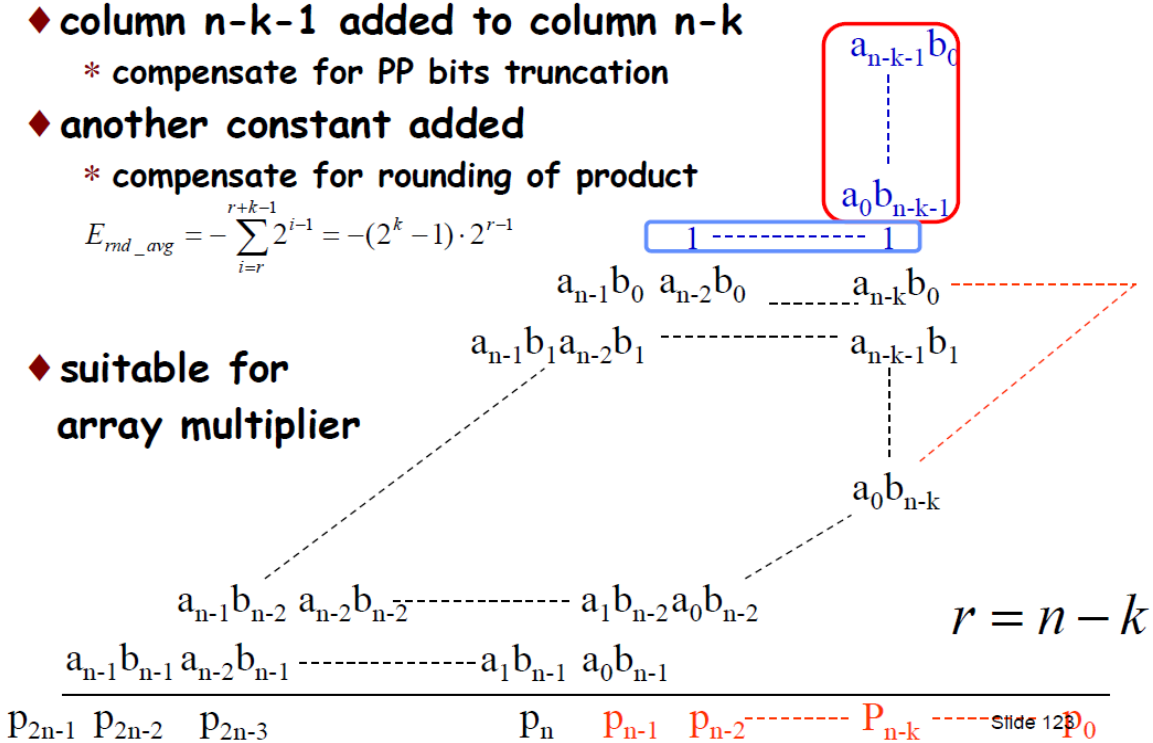
- correction variables and constant are fed into the boundary cells of the array multiplier

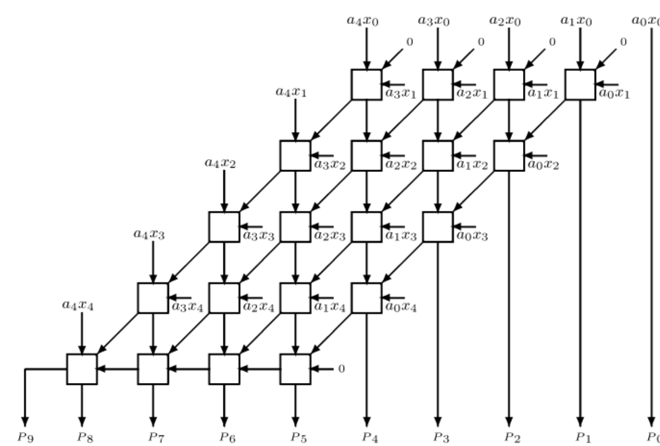
Variable Correction

- **column n-k-1 added to column n-k**
  - * compensate for PP bits truncation
- **another constant added**
  - * compensate for rounding of product

$$E_{rnd\_avg} = -\sum_{i=r}^{r+k-1} 2^{i-1} = -(2^k - 1)\cdot 2^{r-1}$$

- **suitable for array multiplier**

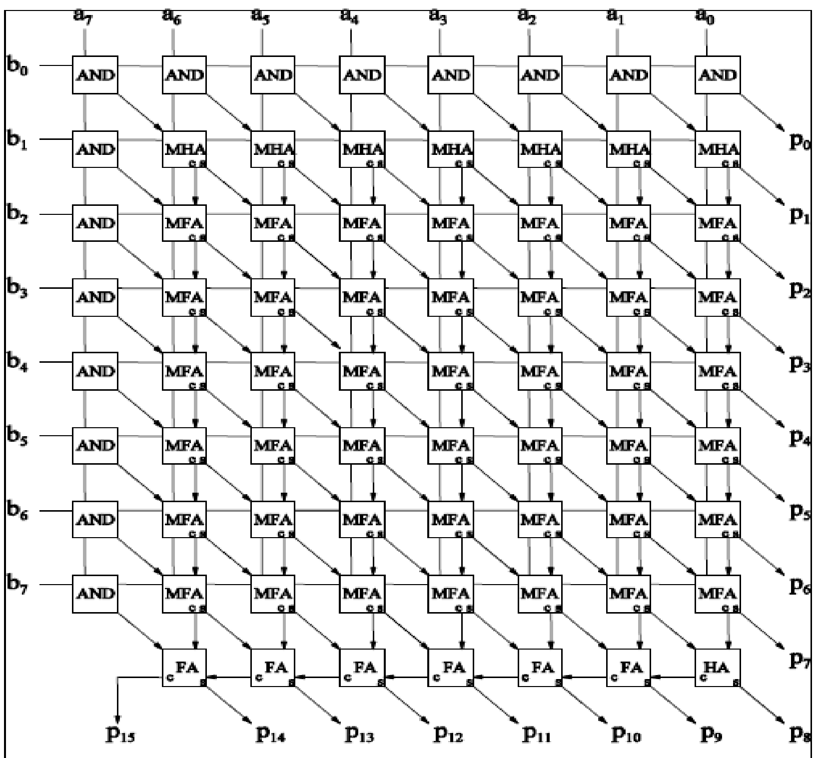




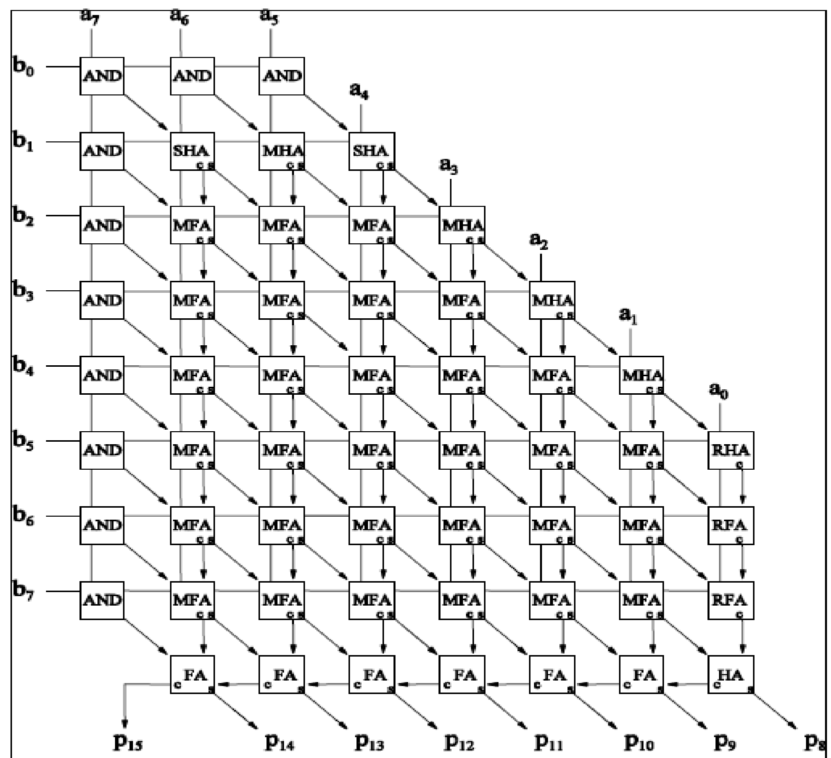
Variable Correction with Array (trunc_var_array)

M. J. Schule, J. E. Stine, and J. G. Jansen, "Reduced power dissipation through truncated multiplication," in IEEE Alessandro Volta Memorial International Workshop on Power Design, pp. 61-69, Mar. 1999.

Truncated Array Multipliers (variable-correction)



8 by 8 Array Multiplier



8 by 8 Truncated Array Multiplier (k = 3, t = 5)

MFA (modified FA): AND+FA

RFA (reduced FA): FA producing carry out only

MHA (modified HA): AND+HA

SHA (specialized HA): MFA with one input set to 1

Accuracy (n, k, r)

- required k for max absolution error < 1ulp = 2^n-1
- e.g., for n=8, k=3 (constant) or 2 (variable)
- e.g. for n=16, k=4 (constant) or 3 (variable)

k	Constant Correction	Variable Correction
0	n = 1	1 <= n <= 3
1	2 <= n <= 3	4 <= n <= 7
2	4 <= n <= 6	8 <= n <= 14
3	7 <= n <= 11	15 <= n <= 27
4	12 <= n <= 20	28 <= n <= 52
5	21 <= n <= 37	53 <= n <= 101
6	38 <= n <= 70	102 <= n <= 198

$E_{total} = -(2^k + r - 2) \cdot 2^r - 1$

$E_{tot\_max} = -(2^k + r - 2) \cdot 2^r - 1$

Summarized Comparison Table

- one table for each of 8x8 and 16x16

signed mxn		Operator (signed)			row			trunc_const_row			array			trunc_var_array		
		area	mid	delay	area	mid	delay	area	mid	delay	area	mid	delay	area	mid	delay
area (um2)	CL(total)															
time (ns)	delay															
power (uW)	dynamic															
	leakage															
	total															