Computer Electronics

Lecture 18: Multiplier Circuits – Part 1

Lecture outline

- Parallel unsigned multiplier
- Parallel carry-save multiplier
- Signed multiplier
  - Subtraction review
  - Parallel signed multiplier

Unigned matrix multiplier: algorithm

				a3	a2	a1	a0
			x	b3	b2	b1	b0
				a3b0	a2b0	a1b0	a0b0
			a3b1	a2b1	a1b1	a0b1	
		a3b2	a2b2	a1b2	a0b2		
+	a3b3	a2b3	a1b3	a0b3			
p7	p6	p5	p4	p3	p2	p1	p0

multiplicand

multiplier

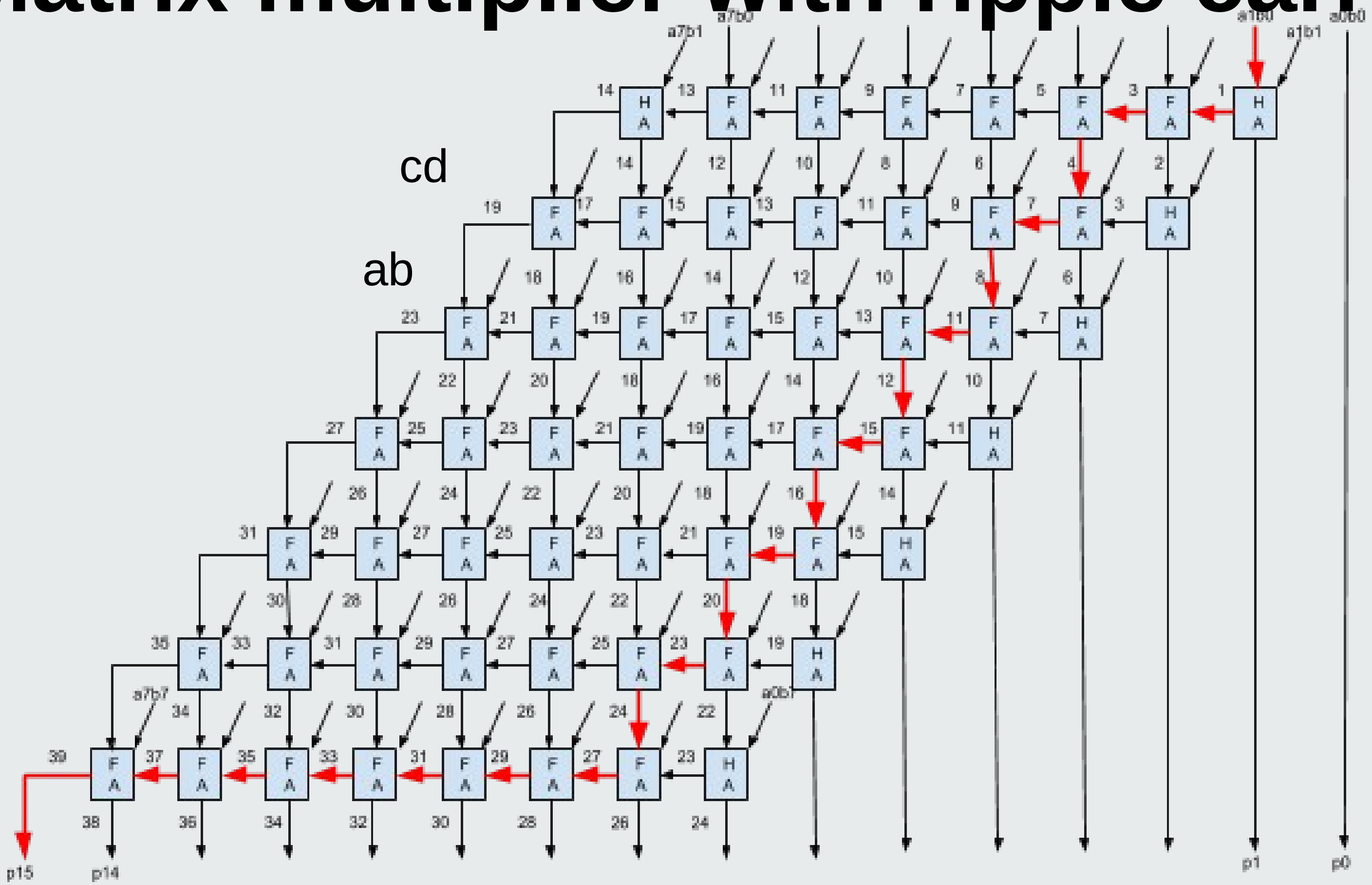
partial product

product

Matrix multiplier with ripple carry

cd

ab

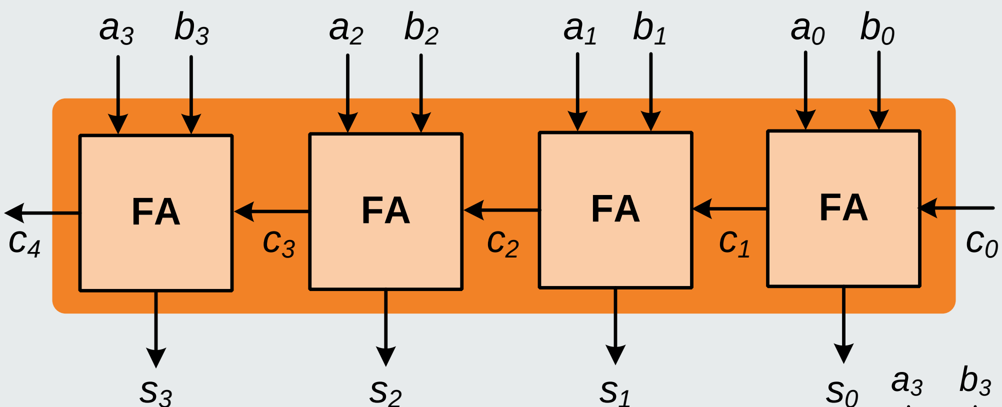


Pipelined MM w/ RCA

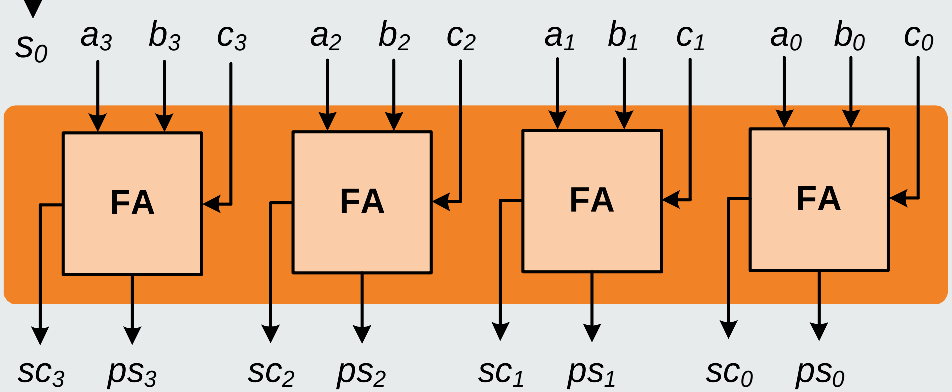
- Pipeline stages after each partial product addition
- What is the maximum clock frequency?

<u>Carry-Save</u> Adder Principle

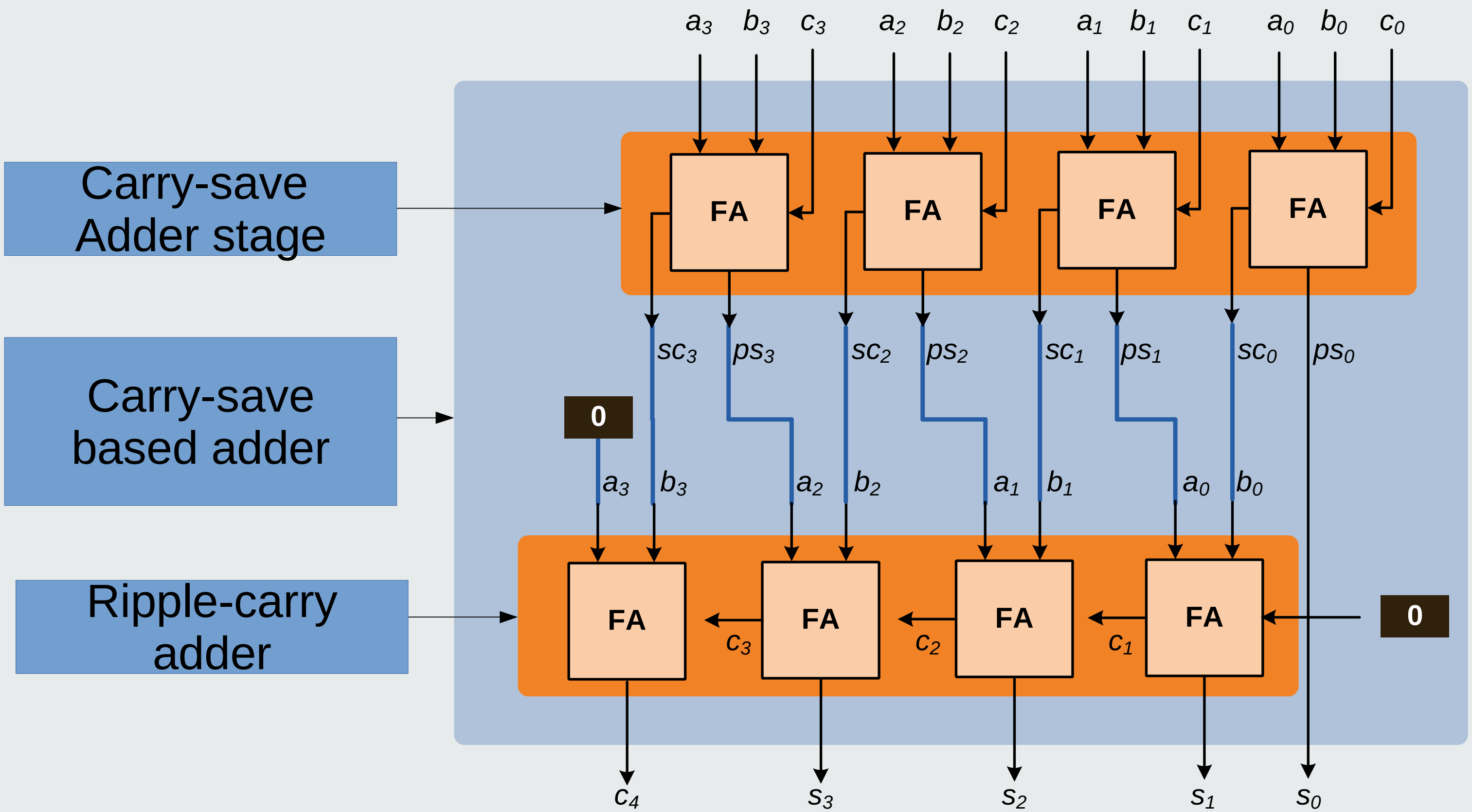
Ripple-carry adder



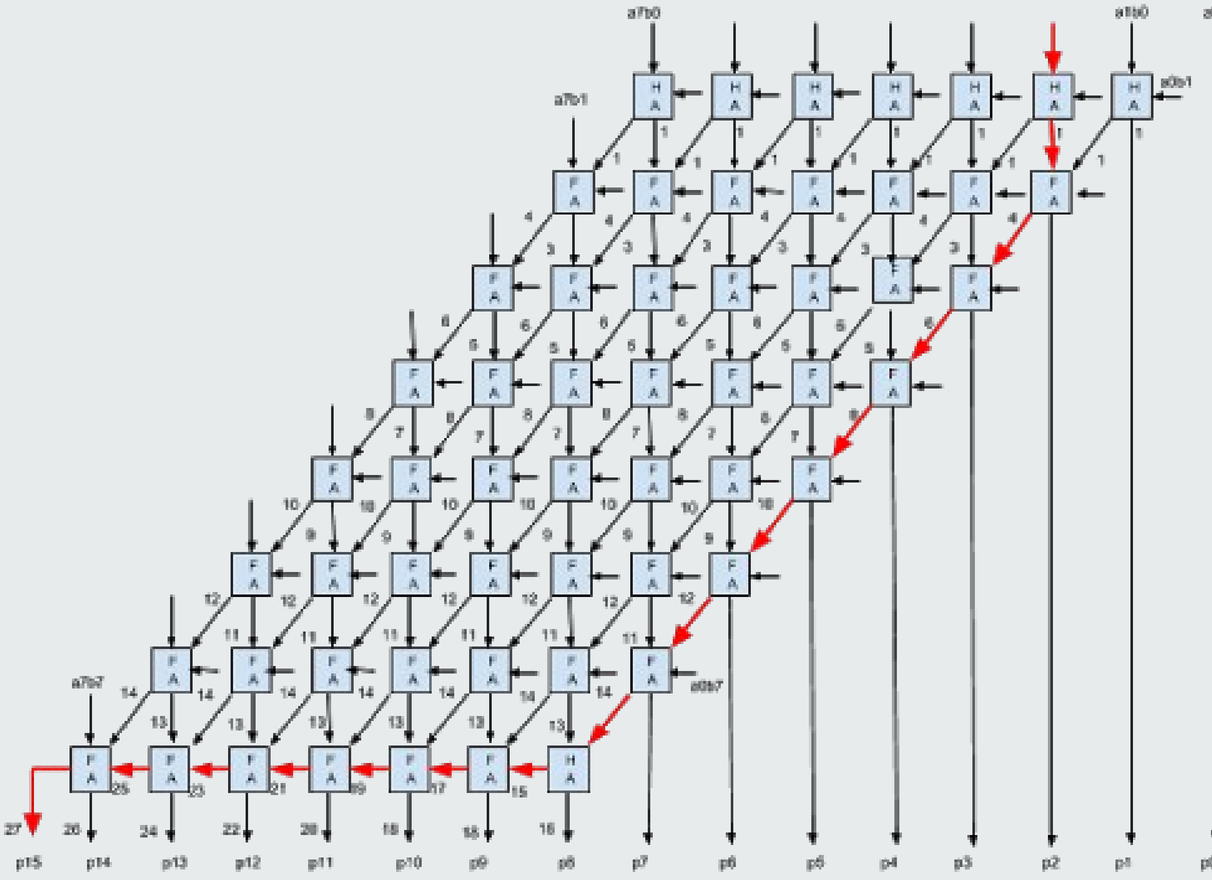
Carry-save Adder stage



3-input adder using carry-save stage



Matrix multiplier with carry save



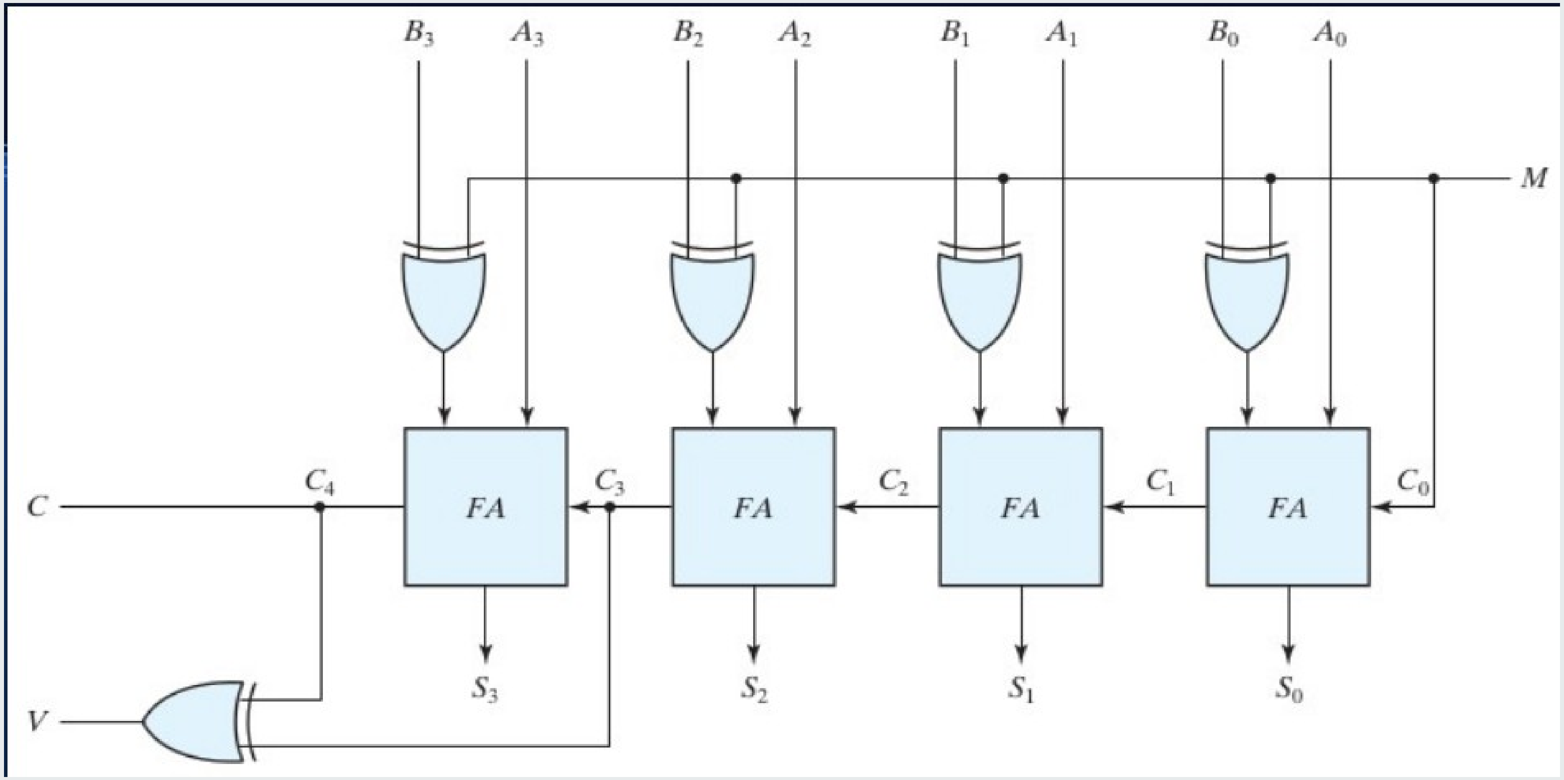
Pipelined MM w/ CSaA

- Pipeline stages after each partial product addition
- What is the maximum clock frequency?
  - What is the stage responsible for it?
- Solution for increasing the frequency
  - Work at the CSaA clock frequency
  - Use N-1 clock cycles to compute the last stage

Subtraction: fundamentals

- Adder circuits work equally well with unsigned and signed numbers
- A-B = A + (-B)
  - To subtract simply add the symmetric
  - Symmetric is 2’s complement
  - Easiest way to compute 2’s complement in hardware
    - Complement and add 1

Subtraction: circuit



Sign extension

- Sometimes we need to increase the number of bits for representation in different signed integer format
- Sign of the number should be preserved
- Q1: represent signed number **1**000 (-8_{10}) with 8 bits
- A1: **1111****1**000 (sign extension in red)
- Q2: represent signed number 0111 (7_{10}) with 8 bits
- A2: **0000****0**111(sign extension in red)

Logical/Arithmetic right shift

- Shifting a number **left** by *p* positions is the same as multiplying it by 2^p
- Shifting a **signed** number **right** by *p* positions is the same as dividing it by 2^p
- **Sign must me preserved: arithmetic right shift**
- 1000 (-8_{10}) shift right by 2 (divide by 4)
  - Logical right shift **00**10 (2_{10}): **wrong**!
  - Arithmetic right shift **11**10 (-2_{10}): **correct**!

Signed multiplication: fundamentals

- A = a3 a2 a1 a0, B = b3 b2 b1 b0
- In decimal notation

$B = -b3*2^3 + b2*2^2 + b1*2^1 + b0*2^0$

$A*B = A * [-b3*2^3 + b2*2^2 + b1*2^1 + b0*2^0]$

- Algorithm:
  - Add least significant partial products, subtract most significant partial product

1

Signed multiplication: circuit