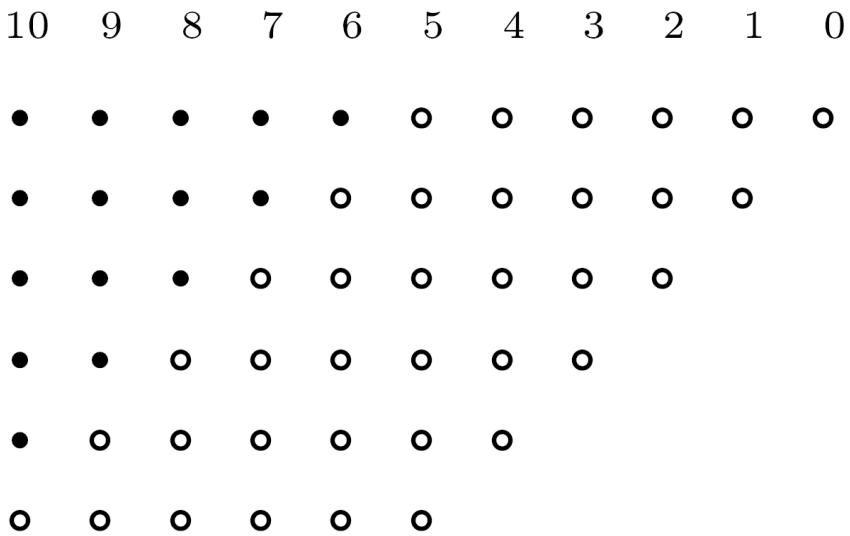
ALU HW4:
Sign-Extension-Bit Reduction

Outlines

- reduction of sign-extension bits for signed multiplier
  - Verilog operator * for unsigned data * (operator_unsigned)
  - Verilog operator for * for signed data (operator_signed)
  - partial product (pp) row addition without optimized sign-extension bits for signed data (row)
  - pp row addition with optimized sign extension bits (row_)opt)
  - modified Baugh-Wooley reduction (row_mbw)
  - addition of radix-4 Booth recoded pp rows without sign-extension bit optimization (row_booth)
  - addition of radix-4 Booth recoded rows with optimized sign-extension bits

partial product reduction

- partial product bit (ppb) matrix contains n rows of partial products, including sign-extension bits
  - e.g., n=6 as shown below
- carry save addition (CSA) can efficiently reduce # of partial product rows into 2 using
  - either row-based CSA (in units of partial product rows), or
  - column-based CSA (in unit of ppb, from rightmost column)
- But for easy coding, provided Verilog codes use for-loop to sum up all rows

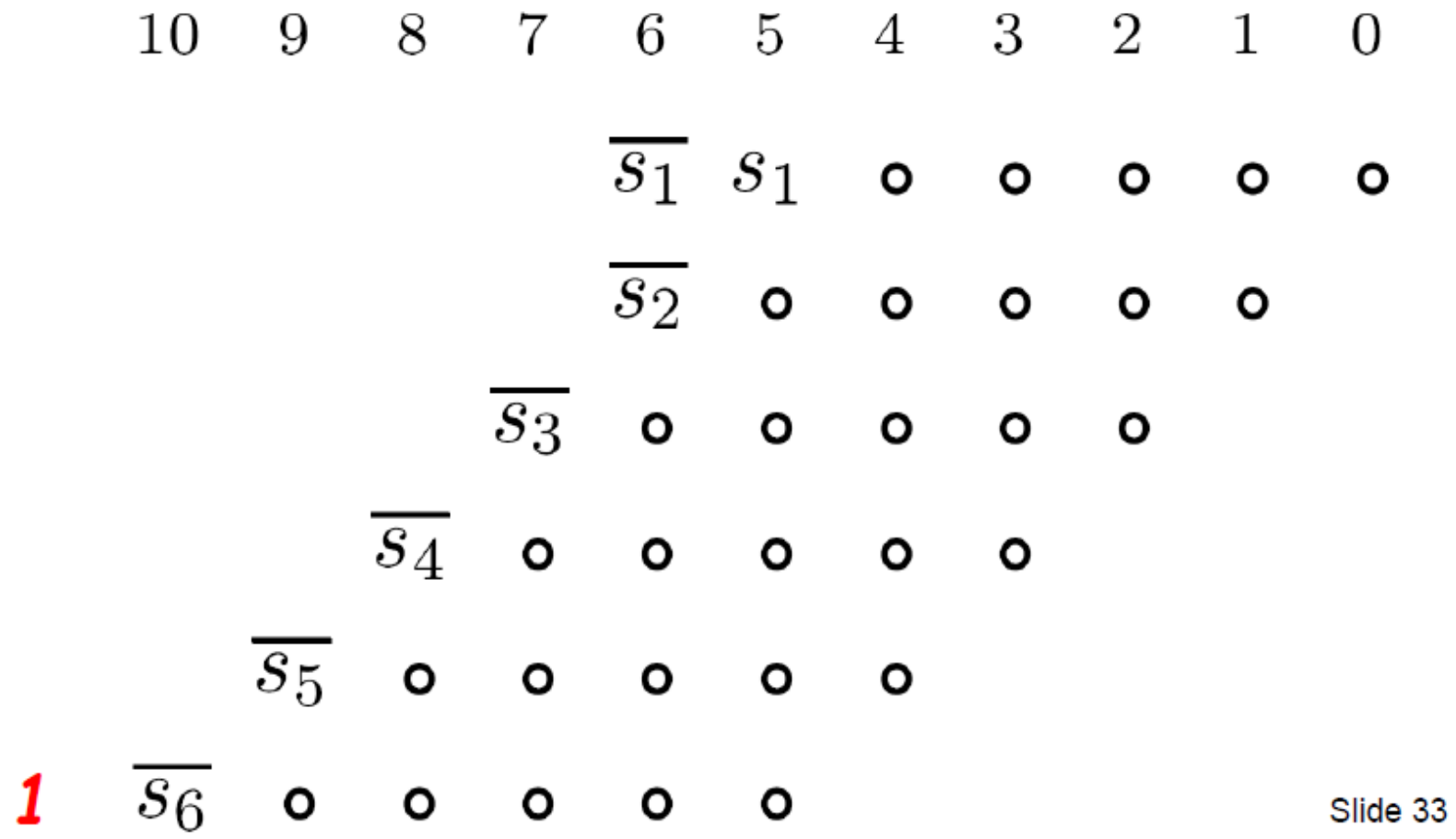


Reduction of Sign-Extension Bits

- modify the partial product generation part in the provided multiplier codes to reduce the sign-extension bits
  - medod-1 (ppb): regular partial-product bit (ppb) array
  - method-2 (mbw): modified Baugh-Wooley
  - method-3 (booth): radix-4 Booth
- Do experiments on three different precisions
  - 8x8
  - 8x16
  - 16x16
- compare with direct * operator for **signed** data

Method-1: partial product row addition (row_opt)

- partial product row accumulation with optimized sign extention bits (row_opt)



Method-2: Modified Baugh-Wooley (row_mbw)

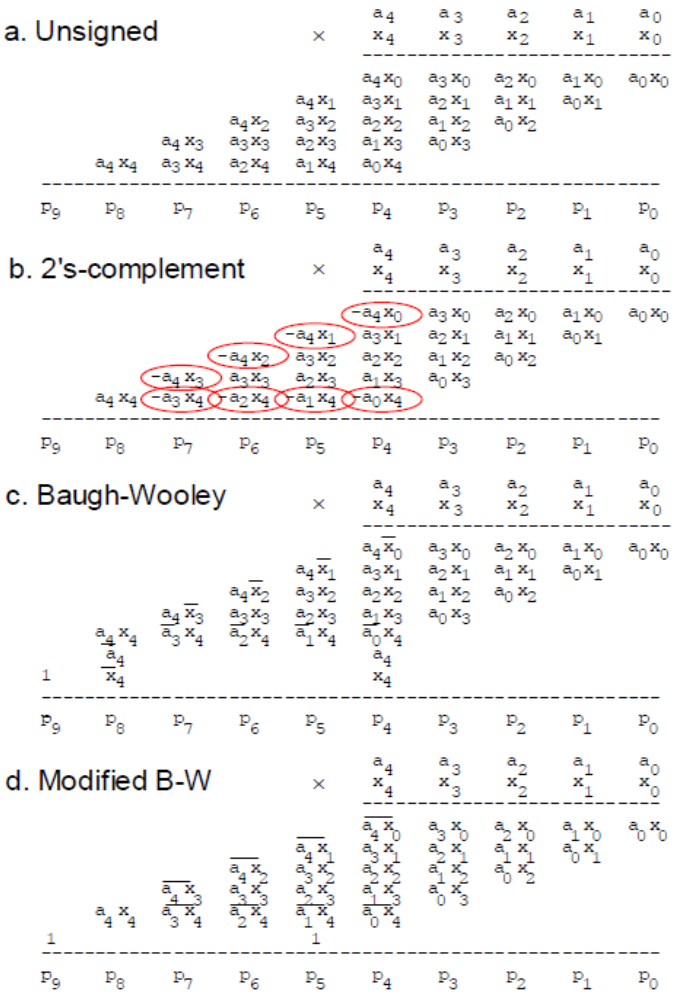
- modified Baugh Wooley for negative partial product bits
- partial product row addition (row_mbw)

Using the Negative-Weight Property of the Sign Bit

Sign extension is a way of converting negatively weighted bits (negabits) to positively weighted bits (posibits) to facilitate reduction, but there are other methods of accomplishing the same without introducing a lot of extra bits

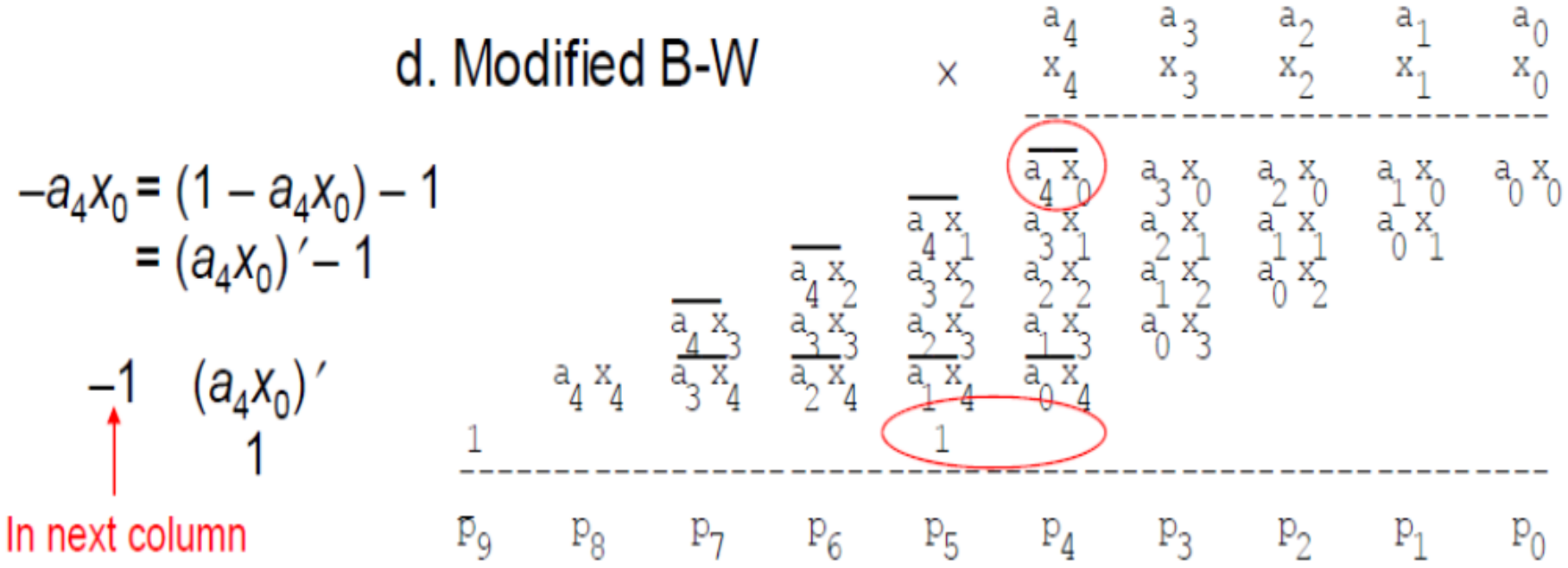
Baugh and Wooley have contributed two such methods

Baugh-Wooley
2's-complement multiplication.



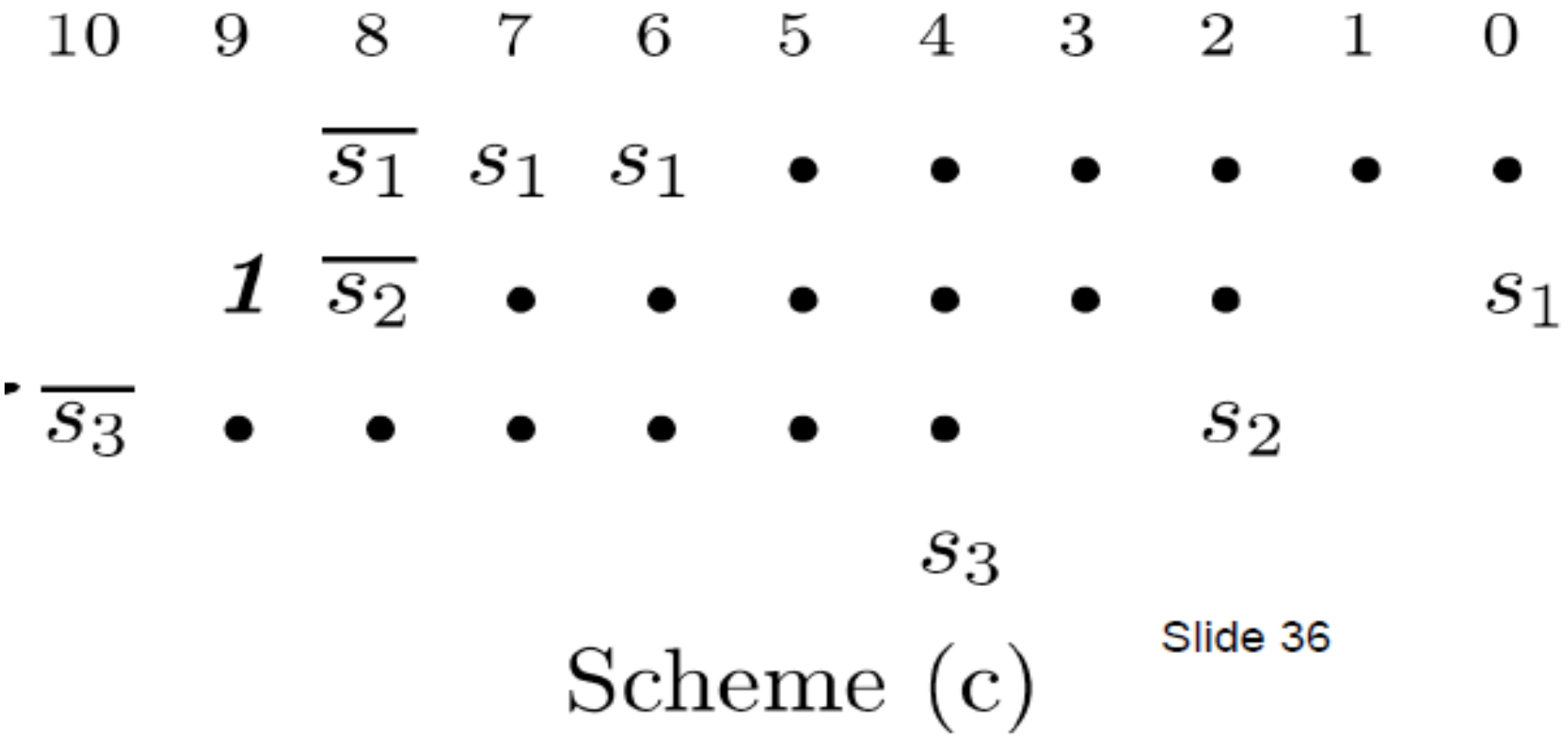
Method-2: Modified Baugh-Wooley (row_mbw)

- 1-s = s’ for s in {0, 1}



Method-3: Radix-4 Boot (row_booth_opt)

- row addition of radix-4 Booth recoded partial products with optimized sign extension bits



Slide 36

Summarized Comparison Table

- operator: using Verilog operator * directly for signed (unsigned) data
- row: partial product row addition without reduction of sign extension bits
- row_opt: partial product row addition with optimized sign-extension bits
- row_booth: radix-4 Booth recoded partial product row addition without reduction of sign extension bits
- row_booth_opt: radix-4 Booth recoded pp row addition with optimized sign-extension bits

signed mxn		operator (unsigned)			row			row_opt			row_mbw			row_booth			row_booth_opt		
		area	mid	delay	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																		