
BOOTH ENCODING OF THE "MULTIPLIER" INPUT

Booth Encoding

- Method to reduce the number of partial products
- Booth- n
 - Examines $n+1$ bits of the *multiplier*
 - Encodes n bits
 - $n \times$ reduction in the number of partial products
- But partial products must then be more complex than just 0 or the *multiplicand*

Booth Encoding—Non-Booth

- Simplest partial products
- Look at one bit of *multiplier* (Y_i) at a time

Y_i	Partial product
0	0
1	+x

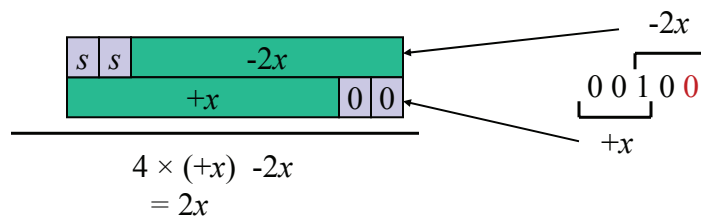
Booth Encoding—Booth-2 or “Modified Booth”

- Examine multiplier bits Y_{i+1} , Y_i , and Y_{i-1}
- Can view *multiplier* as being built of strings of 1's
 - Examine multiplier bits Y_{i+1} , Y_i , and Y_{i-1}
 - Perspective of moving left towards MSB

Y_{i+1}	Y_i	Y_{i-1}	Partial product	Comment
0	0	0	0	no string of 1's
0	0	1	+x	end of string of 1's
0	1	0	+x	a string of 1's
0	1	1	+2x	end of string of 1's
1	0	0	-2x	beginning of string of 1's
1	0	1	-x	-2x + x
1	1	0	-x	beginning of string of 1's
1	1	1	0	center of string of 1's

Booth Encoding—Booth-2 or “Modified Booth”

- Example: *multiplicand* = 0010 = 2
 - Add 0 to right of LSB since first group has no group with which to overlap
 - Examine 3 bits at a time
 - Encode 2 bits at a time
 - Overlap one bit between partial products



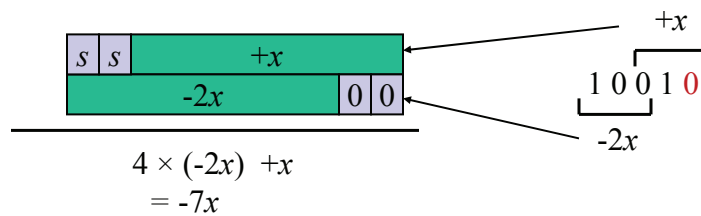
B. Baas, © 2011

EEC 281

69

Booth Encoding—Booth-2 or “Modified Booth”

- Example: *multiplicand* = 1001 = -7
 - Add 0 to right of LSB since first group has no group with which to overlap
 - Examine 3 bits at a time
 - Encode 2 bits at a time
 - Overlap one bit between partial products



B. Baas, © 2011

EEC 281

70

Booth Encoding—Booth-3

Y_{i+2}	Y_{i+1}	Y_i	Y_{i-1}	Partial product
0	0	0	0	0
0	0	0	1	+x
0	0	1	0	+x
0	0	1	1	+2x
0	1	0	0	+2x
0	1	0	1	+3x
0	1	1	0	+3x
0	1	1	1	+4x
1	0	0	0	-4x
1	0	0	1	-3x
1	0	1	0	-3x
1	0	1	1	-2x
1	1	0	0	-2x
1	1	0	1	-x
1	1	1	0	-x
1	1	1	1	0