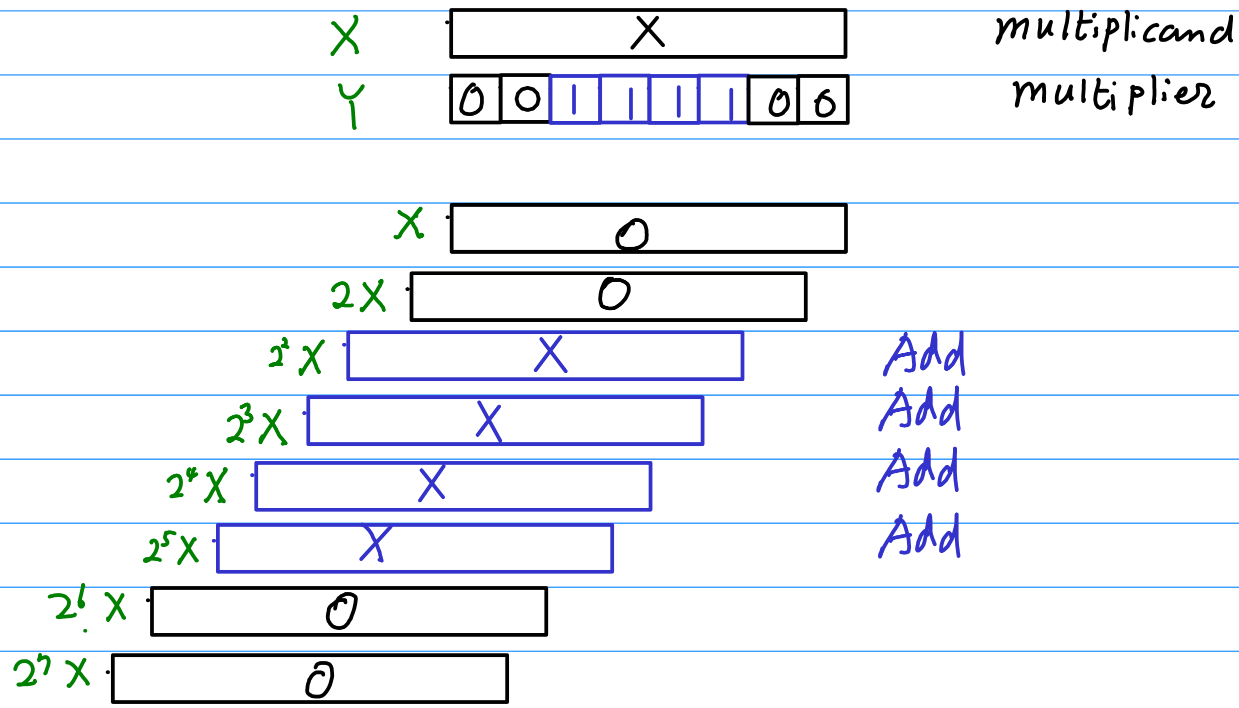
Booth Encoding

20170118

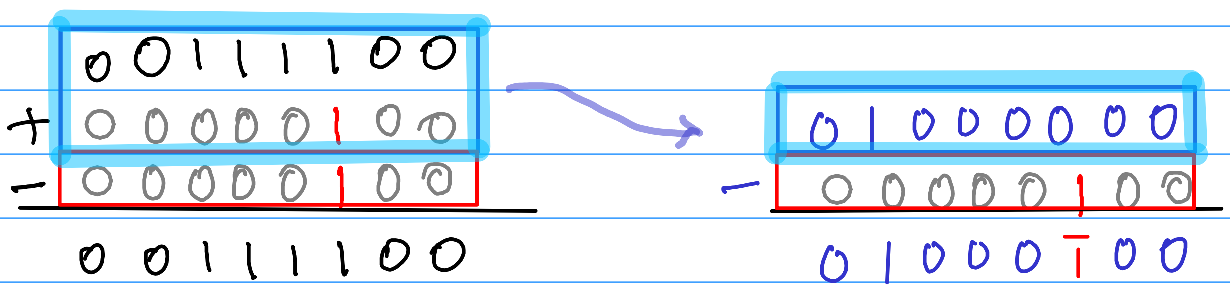
Based on

MJ Flynn's EE 486 Lecture 7 : Integer Multiplication, Stanford University

of Partial Products to be added

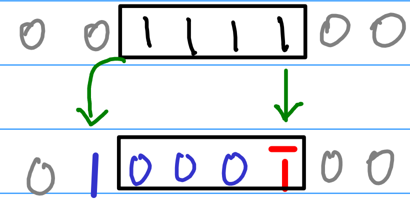


Booth Encoding



+1 −1 = 0

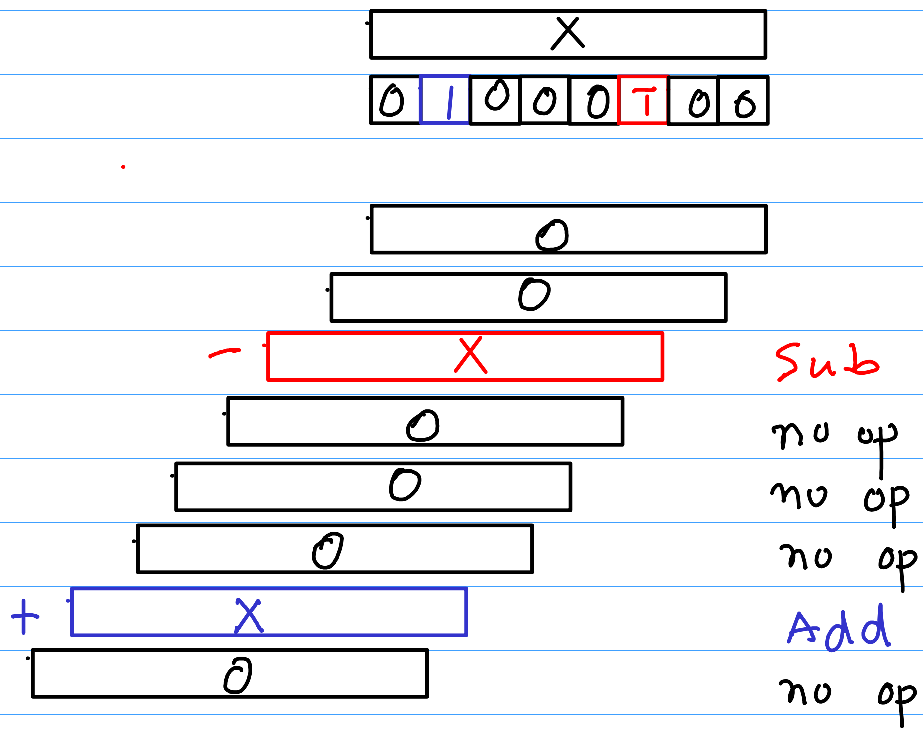
1-run



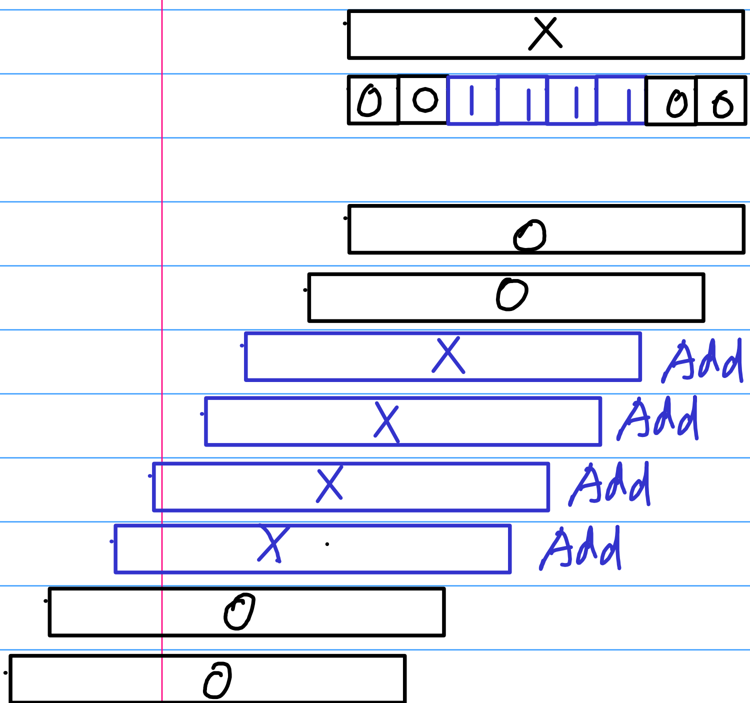
isolated 1 isolated $\bar{1}$

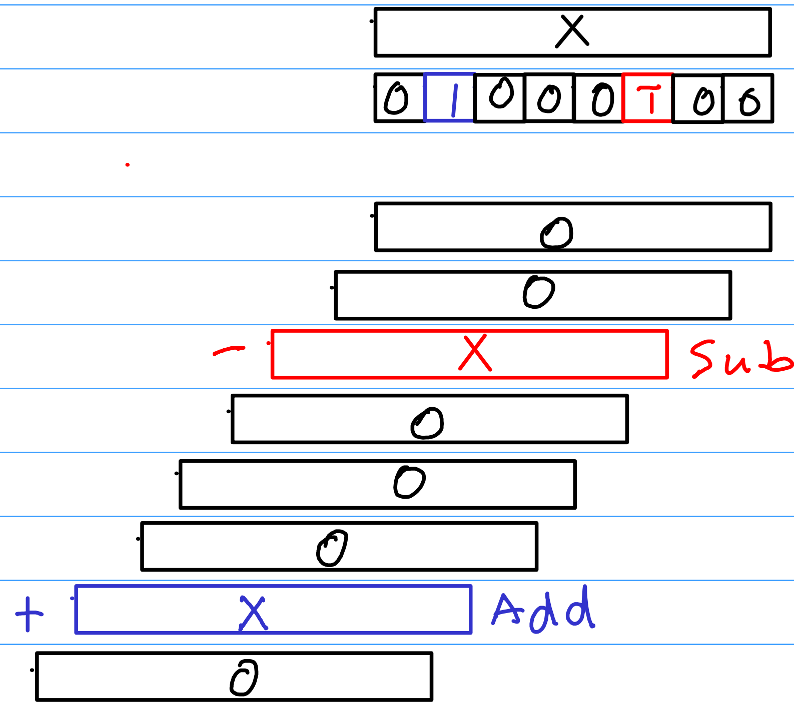
of Partial Products to be added

Original Booth Encoded



- Best case ✓
- Average case ✓
- Worst case





worst case

n add

=

worst case

n add/sub

booth encoding :

the worst case

the number of partial products $\Rightarrow$ still n

not much gain

reduces the number of partial products to be added

delay reduction

works well for serial multiplication

Variable latency

ⓑ worst case : alternating 1's and $\bar{1}$'s

$1\ \bar{1}\ 1\ \bar{1}\ 1\ \bar{1}\ 1\ \bar{1}$

ⓒ Booth encoding does <u>not</u> significantly improve the <u>worst case</u>

$\Rightarrow$ Use <u>**modified**</u> booth encoding

Modified Booth 2

2-bit encoding + 1-bit overlapping

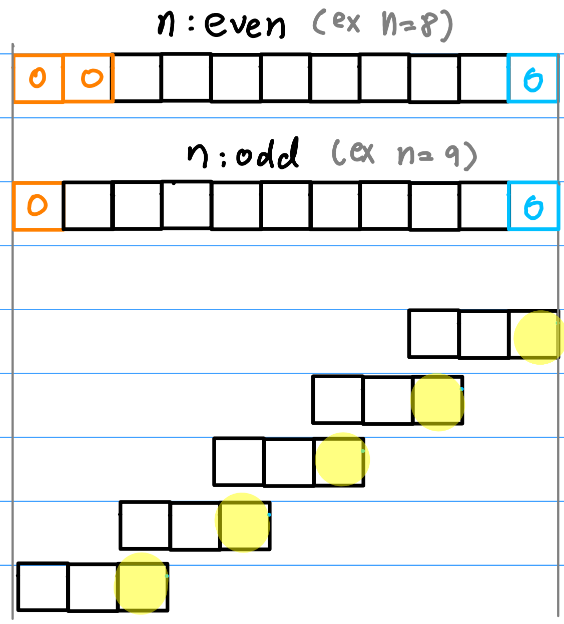
worst case # of p.p's = $\frac{N}{2}+1$

modified Booth 3

3-bit encoding + 1-bit overlapping

worst case # of p.p's = $\frac{N}{3}+1$

Modified Booth 2 (unsigned case)



overlapped 3-bit partitions

$\frac{8}{2} + 1$

$= 5$ partial products

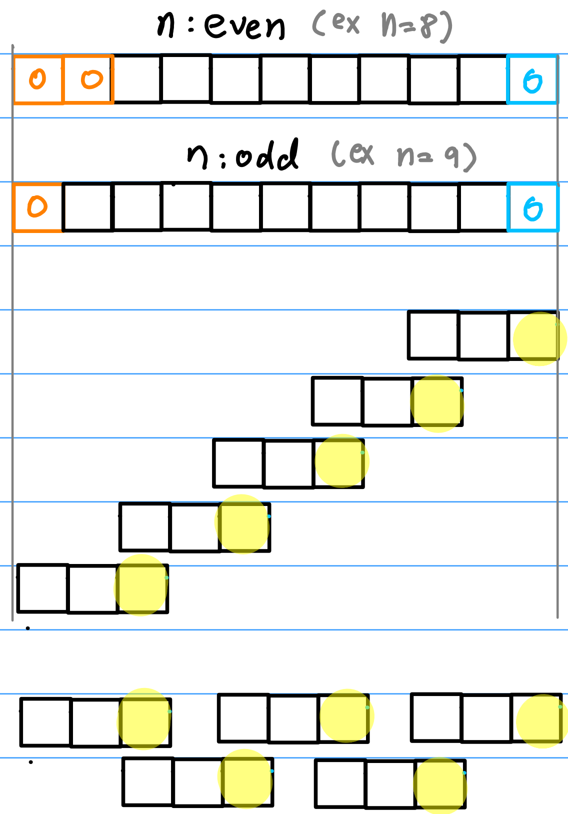
2-bit encoding

2-bit encoding	bits	2^1 2^0	scale factor	
all zero's	0 0 0	0 0 0	$0 \cdot 2 + 0 = +0$	$+0$
end of 1's	0 0 1	0 1 0	$0 \cdot 2 + 1 = +1$	$+X$
isolated 1	0 1 0	1 $\bar{1}$ 0	$1 \cdot 2 + \bar{1} = +1$	$+X$
end of 1's	0 1 1	1 0 0	$1 \cdot 2 + 0 = +2$	$+2X$
start of 1's	1 0 0	$\bar{1}$ 0 0	$\bar{1} \cdot 2 + 0 = -2$	$-2X$
isolated 0	1 0 1	$\bar{1}$ 1 0	$\bar{1} \cdot 2 + 1 = -1$	$-X$
start of 1's	1 1 0	0 $\bar{1}$ 0	$0 \cdot 2 + \bar{1} = -1$	$-X$
all 1's	1 1 1	0 0 0	$0 \cdot 2 + 0 = 0$	$+0$

Scale factor $\{0, \pm 1, \pm 2\}$

$\frac{8}{2}+1$

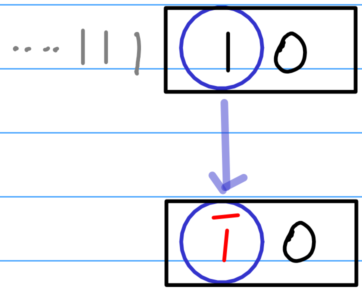
$=5$ partial products



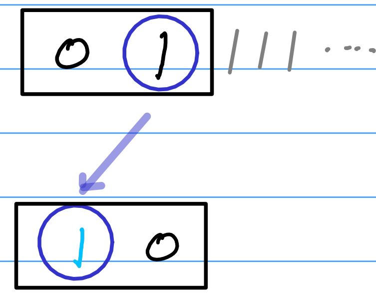
overlapped 3-bit partitions

0-run & 1-run

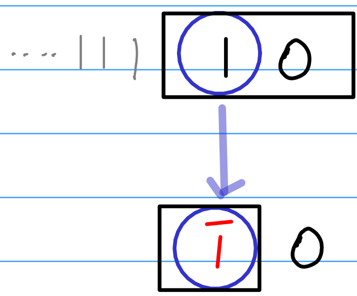
start of a possible 1-run



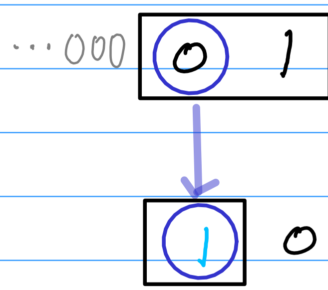
end of a possible 1-run



start of a possible 1's → **start of 1-run**

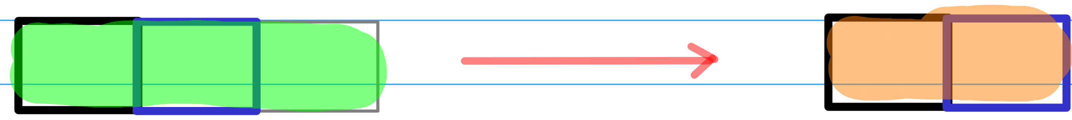


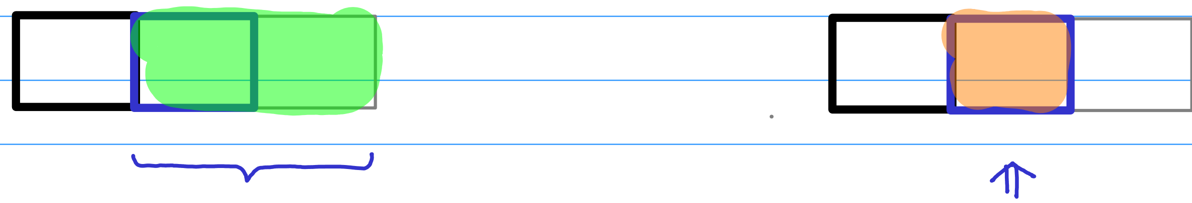
end of a possible 1's → **start of 0-run**



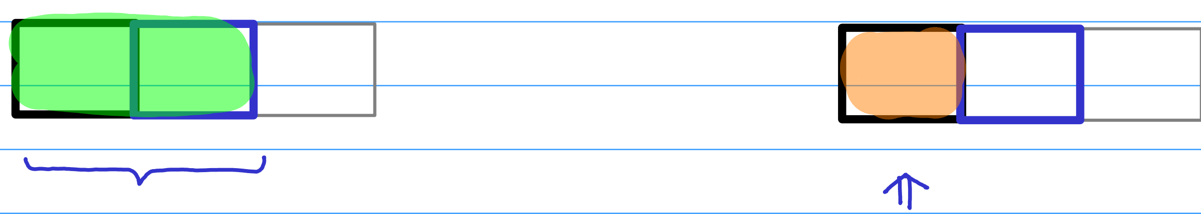
To encode a 1-bit, we need 2-bit info.

Needs 3-bits for encoding a 2-bit



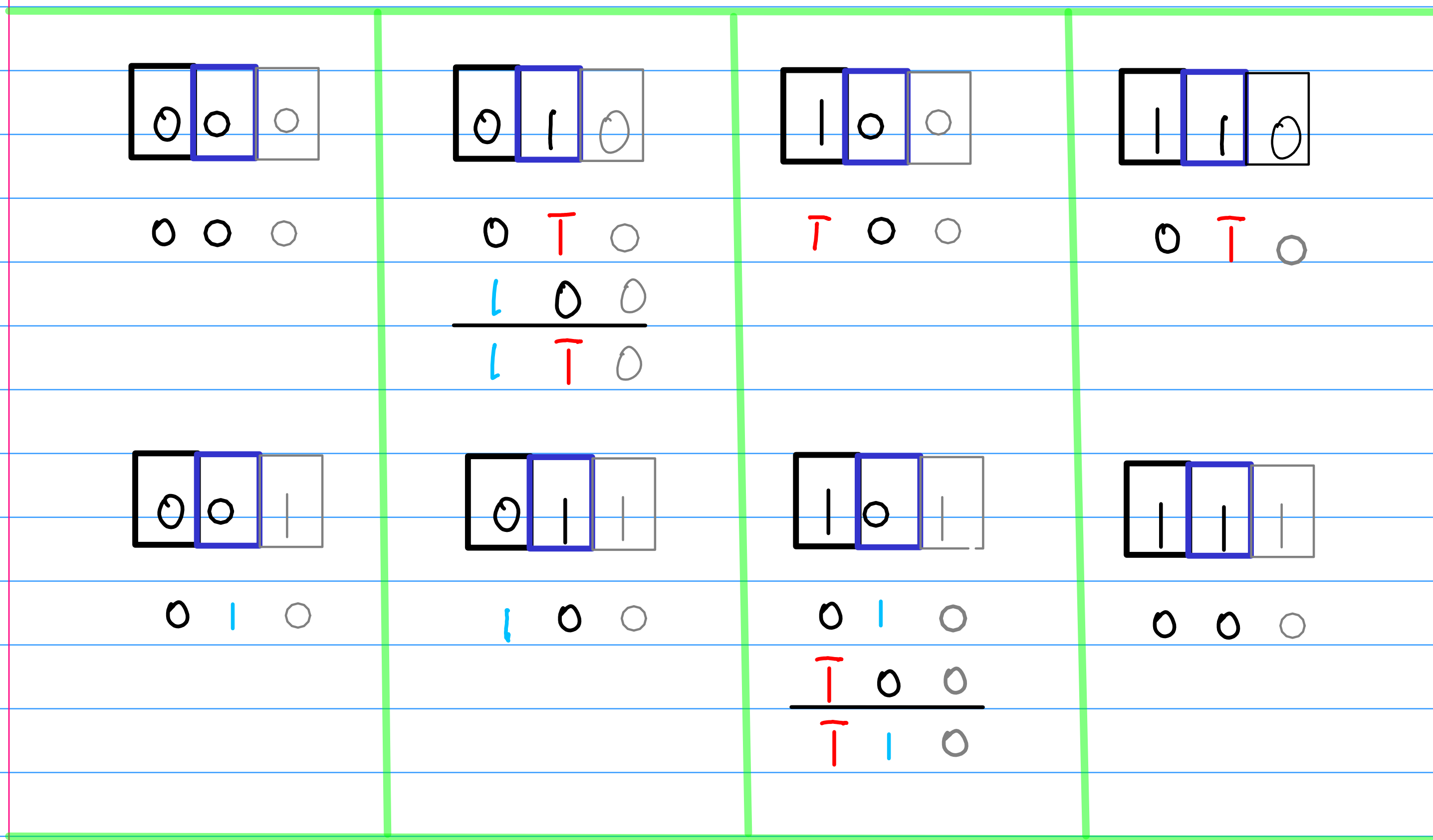


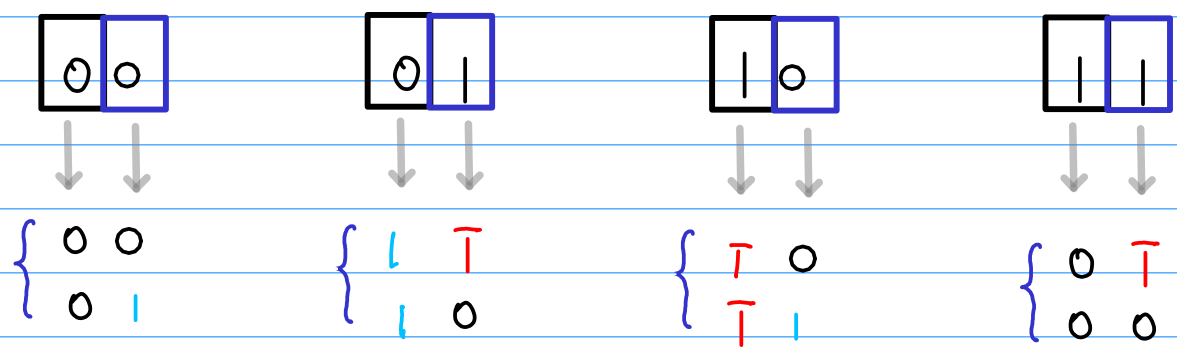
0	0	0- run	0
0	1	0-run starts	1
1	0	1-run starts	$\bar{1}$
1	1	1-run	0

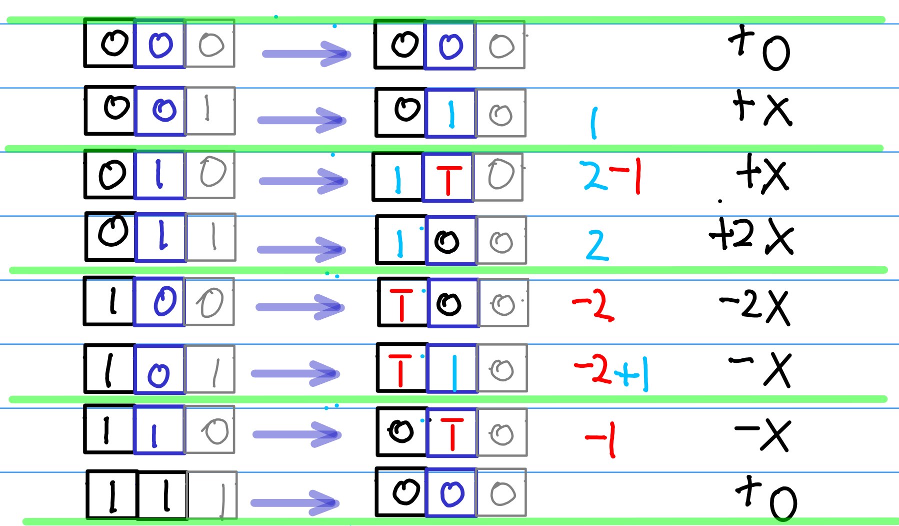


0	0	0- run	0
0	1	0-run starts	1
1	0	1-run starts	$\bar{1}$
1	1	1-run	0

Encoding 2-bit : 2 choices







Scale Factor for 2-bit encoding

all zero's	→ middle of 0-run	0 0 0	→ 0 0 0	+0
end of 1's	→ Start of 0-run at (j)	0 0 1	→ 0 1 0	+X
isolated 1	→ isolated 1 at (j)	0 1 0	→ 1 $\bar{1}$ 0	+X
end of 1's	→ start of 0-run at (j+1)	0 1 1	→ 1 0 0	+2X
start of 1's	→ start of 1-run at (j+1)	1 0 0	→ $\bar{1}$ 0 0	−2X
isolated 0	→ isolated 0 at (j)	1 0 1	→ $\bar{1}$ 1 0	−X
start of 1's	→ start of 1-run at (j)	1 1 0	→ 0 $\bar{1}$ 0	−X
all 1's	→ middle of 1-run	1 1 1	→ 0 0 0	+0

X = | X | multiplicand

× Y = | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | multiplier

↓

modified booth encode 2
3

↓

scale factor

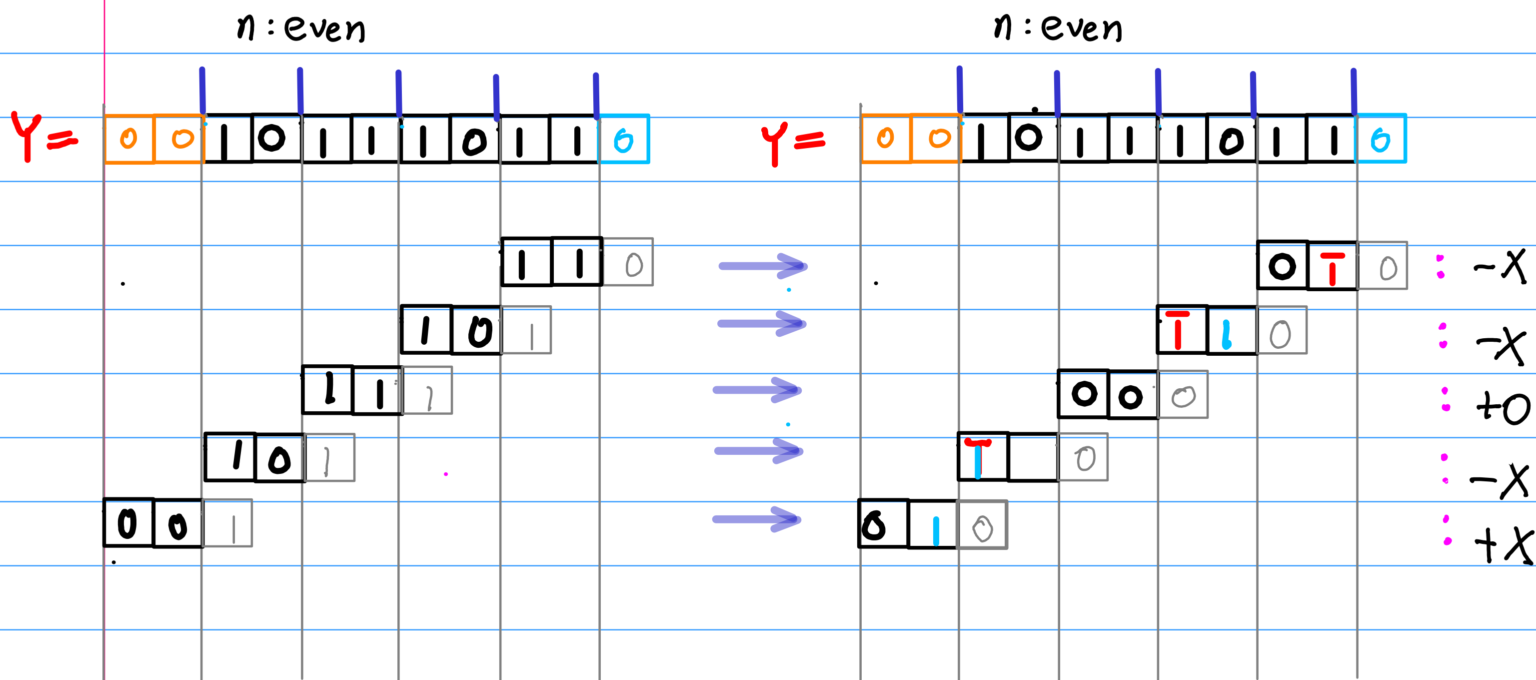
↓

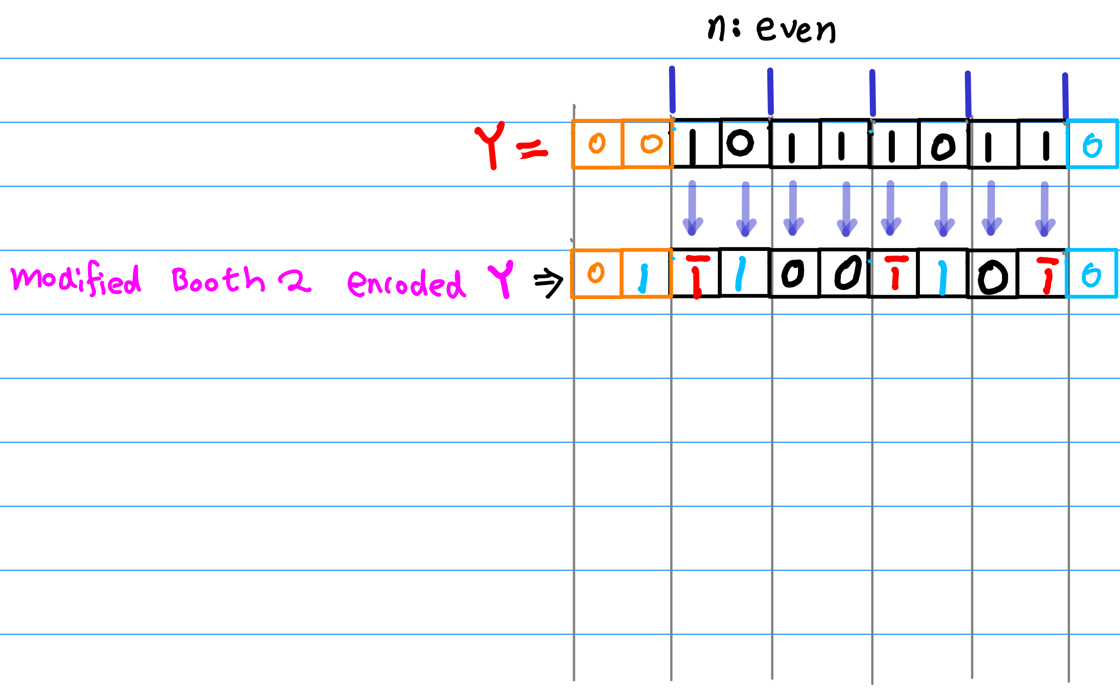
shift multiplicand = X

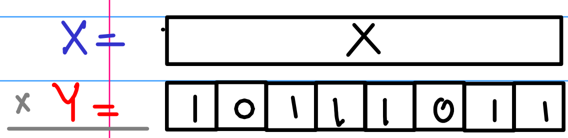
↓

reduced number of partial products for multiplication

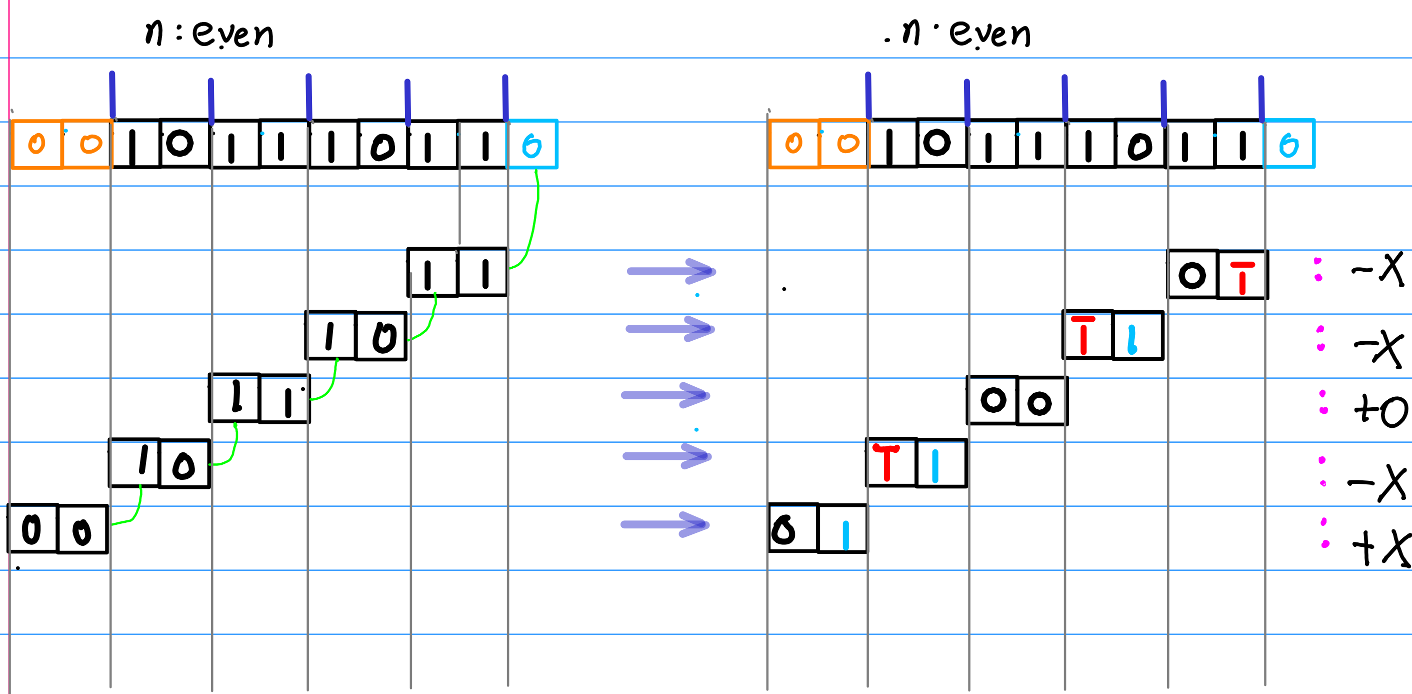
Multiplier Encoding







Basically, $\boxed{\text{2-bit}}$ encoding scheme



context encoding

increasing overlapping bit → no need

0 0 0 0	middle of 0-run	0 0 0 0
0 0 0 1	middle of 0-run	0 0 0 0
0 0 1 0	start of 0-run at (j)	0 1 0 0
0 0 1 1	start of 0-run at (j)	0 1 0 0
0 1 0 0	isolated 1 at (j)	1 1 0 0
0 1 0 1	isolated 1 at (j)	1 1 0 0
0 1 1 0	start of 0-run at (j+1)	1 0 0 0
0 1 1 1	start of 0-run at (j+1)	1 0 0 0
1 0 0 0	start of 1-run at (j+1)	1 0 0 0
1 0 0 1	start of 1-run at (j+1)	1 0 0 0
1 0 1 0	isolated 0 at (j)	1 1 0 0
1 0 1 1	isolated 0 at (j)	1 1 0 0
1 1 0 0	start of 1-run at (j)	0 1 0 0
1 1 0 1	start of 1-run at (j)	0 1 0 0
1 1 1 0	middle of 1-run	0 0 0 0
1 1 1 1	middle of 1-run	0 0 0 0

Overlapping bit at both ends → no need

0 0 0 0	middle of 0- run	0 0 0 0
0 0 0 1	Start of 0-run at (j)	0 0 1 0
0 0 1 0	isolated 1 at (j)	0 1 T 0
0 0 1 1	start of 0-run at (j+1)	0 1 0 0
0 1 0 0	start of 1-run at (j+1)	0 T 0 0
0 1 0 1	isolated 0 at (j)	0 T 1 0
0 1 1 0	start of 1-run at (j)	0 0 T 0
0 1 1 1	middle of 1- run	0 0 0 0
1 0 0 0	middle of 0- run	0 0 0 0
1 0 0 1	Start of 0-run at (j)	0 0 1 0
1 0 1 0	isolated 1 at (j)	0 1 T 0
1 0 1 1	start of 0-run at (j+1)	0 1 0 0
1 1 0 0	start of 1-run at (j+1)	0 T 0 0
1 1 0 1	isolated 0 at (j)	0 T 1 0
1 1 1 0	start of 1-run at (j)	0 0 T 0
1 1 1 1	middle of 1- run	0 0 0 0

2-bit context encoding

0 0 0	middle of 0-run	0 0 0
1 1 1	middle of 1-run	0 0 0
0 0 1	Start of 0-run at (j)	0 1 0
0 1 1	start of 0-run at (j+1)	1 0 0
1 1 0	start of 1-run at (j)	0 T 0
1 0 0	start of 1-run at (j+1)	T 0 0
1 0 1	isolated 0 at (j)	T 1 0
0 1 0	isolated 1 at (j)	1 T 0

0 0 0	middle of 0-run	0 0 0
0 0 1	Start of 0-run at (j)	0 1 0
0 1 0	isolated 1 at (j)	1 T 0
0 1 1	start of 0-run at (j+1)	1 0 0
1 0 0	start of 1-run at (j+1)	T 0 0
1 0 1	isolated 0 at (j)	T 1 0
1 1 0	start of 1-run at (j)	0 T 0
1 1 1	middle of 1-run	0 0 0

context encoding

5 Types of Partial Products

0 0 0	→ 0 0 0		+0
0 0 1	→ 0 1 0	1	+X
0 1 0	→ 1 T 0	2−1	+X
0 1 1	→ 1 0 0	2	+2X
1 0 0	→ T 0 0	−2	−2X
1 0 1	→ T 1 0	−2+1	−X
1 1 0	→ 0 T 0	−1	−X
1 1 1	→ 0 0 0		−0

5 type of scaled multiplicand

+0	0
+X	x_i
+2X ←	x_{i-1}
−X	$\overline{x_i}$
−2X ←	$\overline{x_{i-1}}$

2's complement

i-th bit of the scaled partial products

$1 * X \longrightarrow (i)$

$2 * X \longrightarrow (i-1)$

$(+) \longrightarrow \square$

$(-) \longrightarrow \overline{\square}$ 1's complement

5 types of Scaling X (=multiplier)

$+0$	0
$+X$	x_i
$+2X$ ←	x_{i-1}
$-X$	$\bar{x}_i$
$-2X$ ←	$\bar{x}_{i-1}$

i-th bit of the scaled partial products

$+X$: $x_7\ x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0$ — $\boxed{x_i}\ x_{i-1}$

$+2X$: $x_7\ x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0$ (shifted left by one) — $x_i\ \boxed{x_{i-1}}$

$\bar{X}+1 \Leftarrow -X$: $\bar{x}_7\ \bar{x}_6\ \bar{x}_5\ \bar{x}_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$ $+1$ — $\boxed{\bar{x}_i}\ x_{i-1}$

$\overline{2X}+1 \Leftarrow -2X$: $\bar{x}_7\ \bar{x}_6\ \bar{x}_5\ \bar{x}_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$ (shifted left by one) $+1$ — $\bar{x}_i\ \boxed{\bar{x}_{i-1}}$

0 : $0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

* the last partial product (sign-bit p.p.)

$y_1\ y_0\ 0$ → $\pm 2X,\ \pm X,\ 0$

$y_3\ y_2\ y_1$ → $\pm 2X,\ \pm X,\ 0$

$y_5\ y_4\ y_3$ → $\pm 2X,\ \pm X,\ 0$

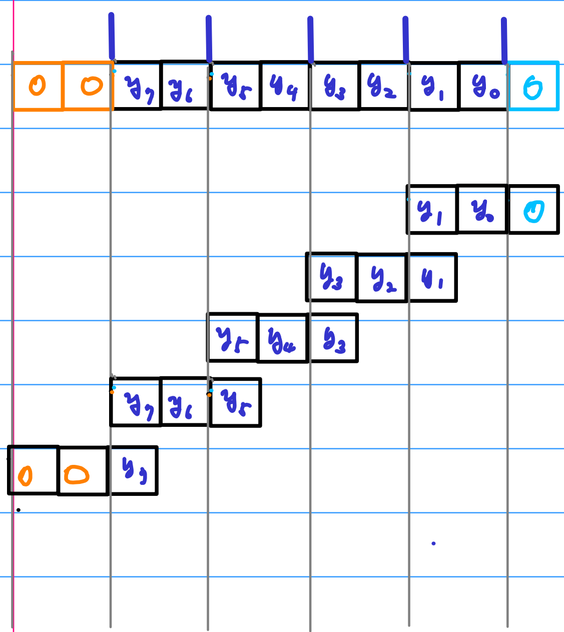
$y_7\ y_6\ y_5$ → $\pm 2X,\ \pm X,\ 0$

000, 001 → $0\ 0\ y_7$ → $+X,\ 0$

2 choices

$y_1, y_3, y_5, y_7 = 1 \Rightarrow$ negative scaling

n : even



0 0 0	$\rightarrow$ 0 0 0	+0
0 0 1	$\rightarrow$ 0 1 0	+X
0 1 0	$\rightarrow$ 1 T 0	+X
0 1 1	$\rightarrow$ 1 0 0	+2X
1 0 0	$\rightarrow$ T 0 0	−2X
1 0 1	$\rightarrow$ T 1 0	−X
1 1 0	$\rightarrow$ 0 T 0	−X
1 1 1	$\rightarrow$ 0 0 0	−0

3-bit partition

$[1][\ ][\ ] \rightarrow \begin{cases} -2X & \overline{2X}+1 \\ -X & \overline{X}+1 \\ -0 & \overline{0}+1 \end{cases}$

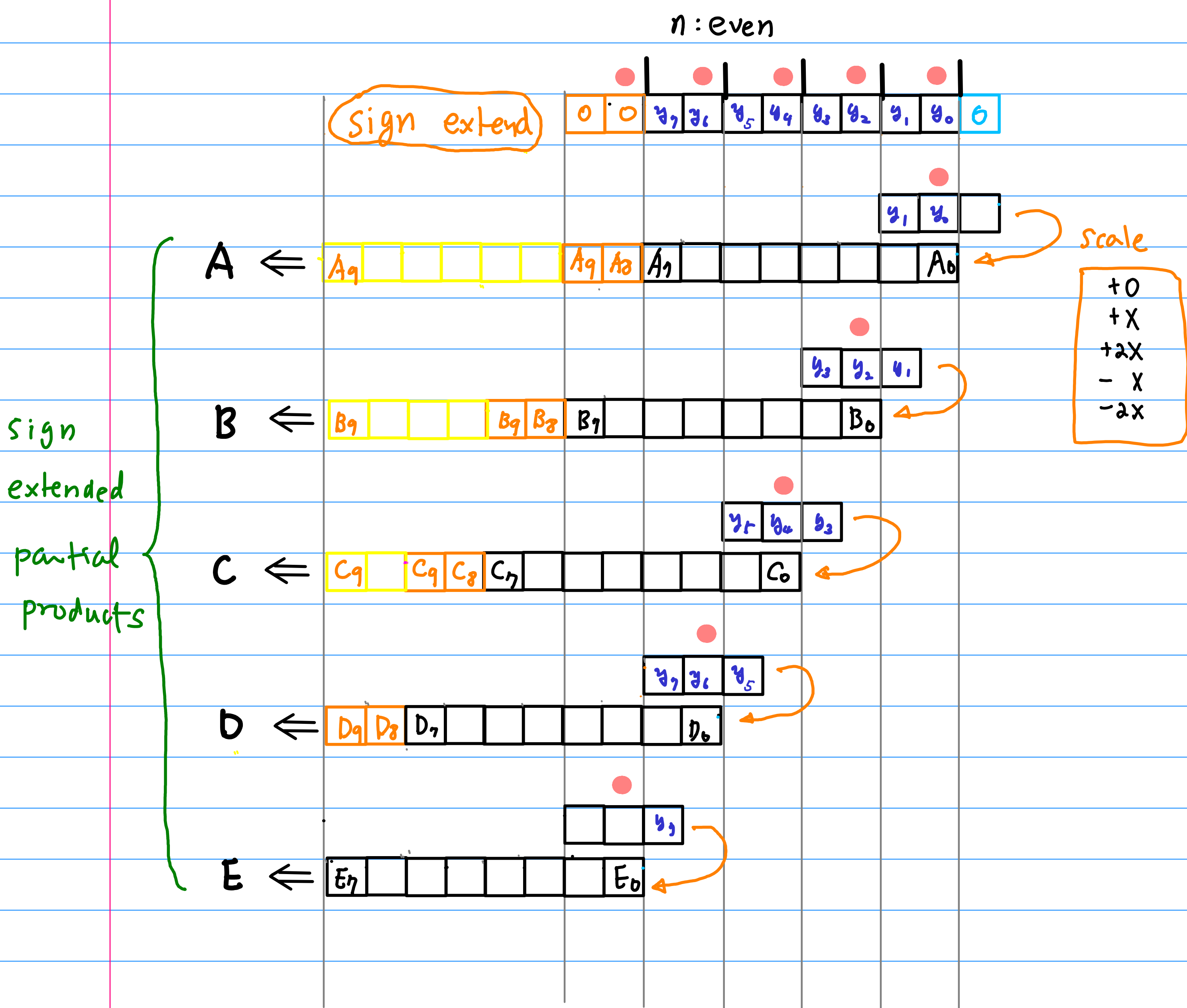
$$\underline{\text{Complement}} \begin{pmatrix} 2X \\ X \\ 0 \end{pmatrix} + 1$$

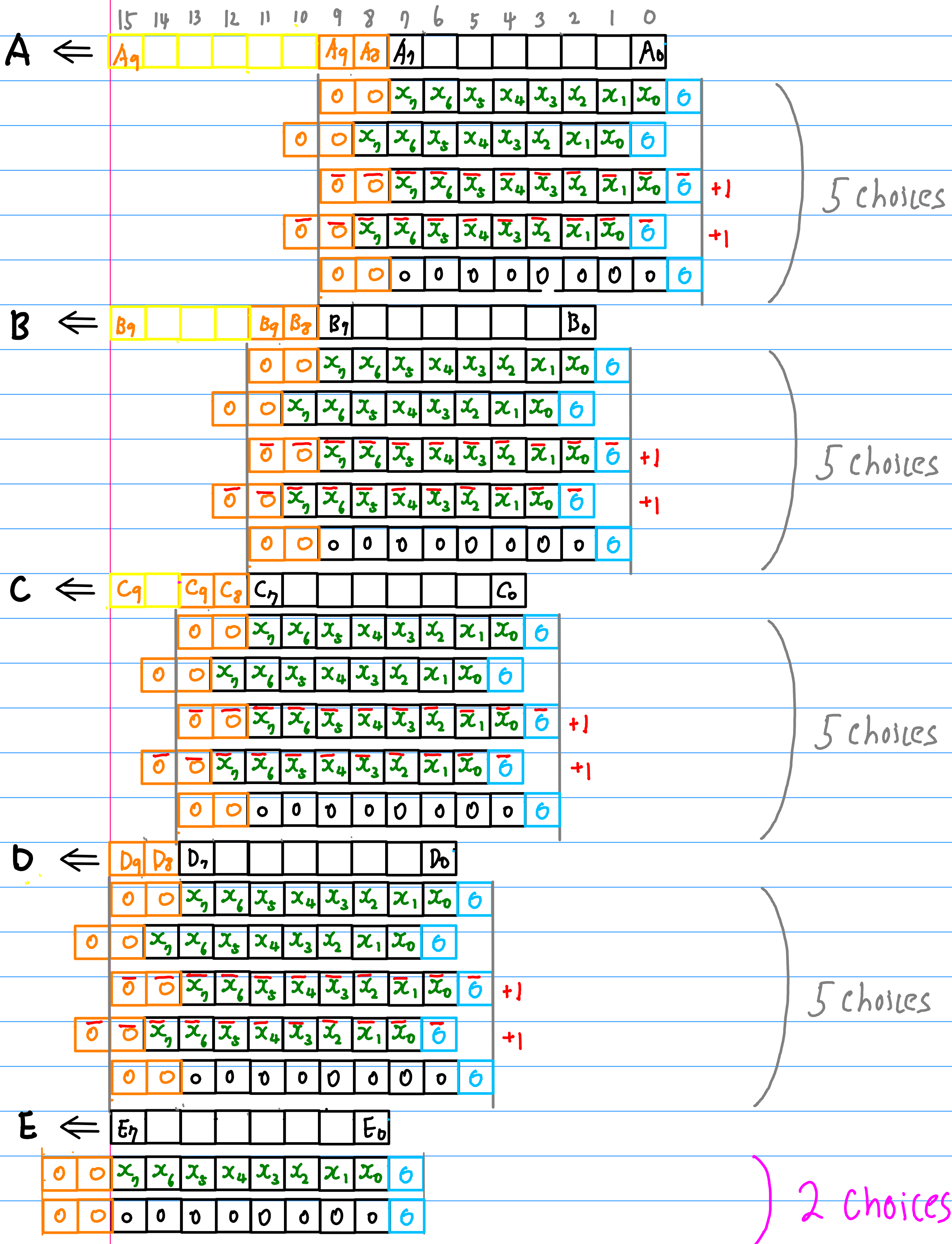
Scaled X's are aligned at y_0, y_2, y_4, y_6, y_8

Sign extended partial product A B C D E

$X =$ | x_7 | x_6 | x_5 | x_4 | x_3 | x_2 | x_1 | x_0 | multiplicand

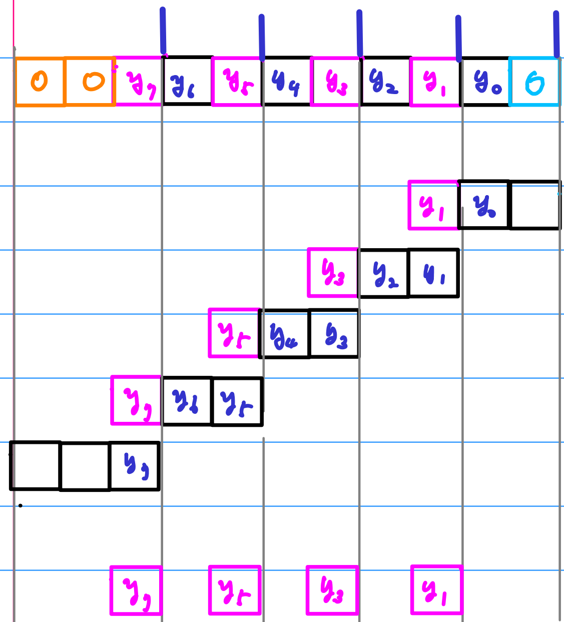
$Y =$ | y_7 | y_6 | y_5 | y_4 | y_3 | y_2 | y_1 | y_0 | multiplier





A
5 choices
B
5 choices
C
5 choices
D
5 choices
E
2 choices

n : even



if $y_1 = 1$, complement partial product + y_1

if $y_3 = 1$, complement partial product + y_3

if $y_5 = 1$, complement partial product + y_5

if $y_7 = 1$, complement partial product + y_7

n : even

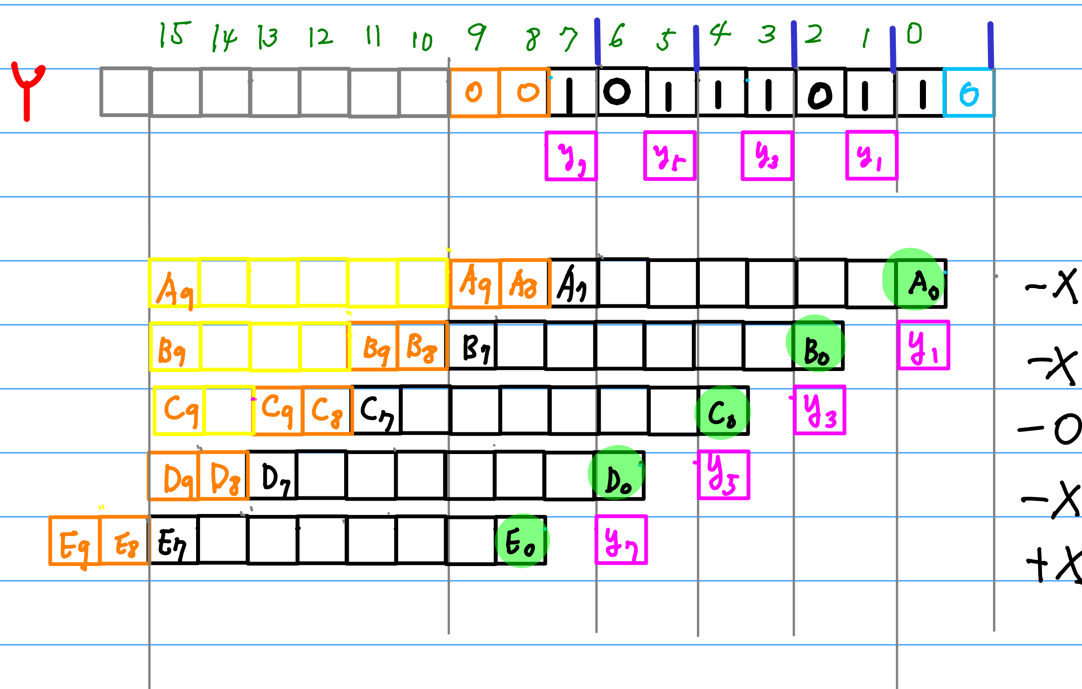
if $y_1 = 1$, compliment A

if $y_3 = 1$, compliment B

if $y_5 = 1$, compliment C

if $y_7 = 1$, compliment D

E



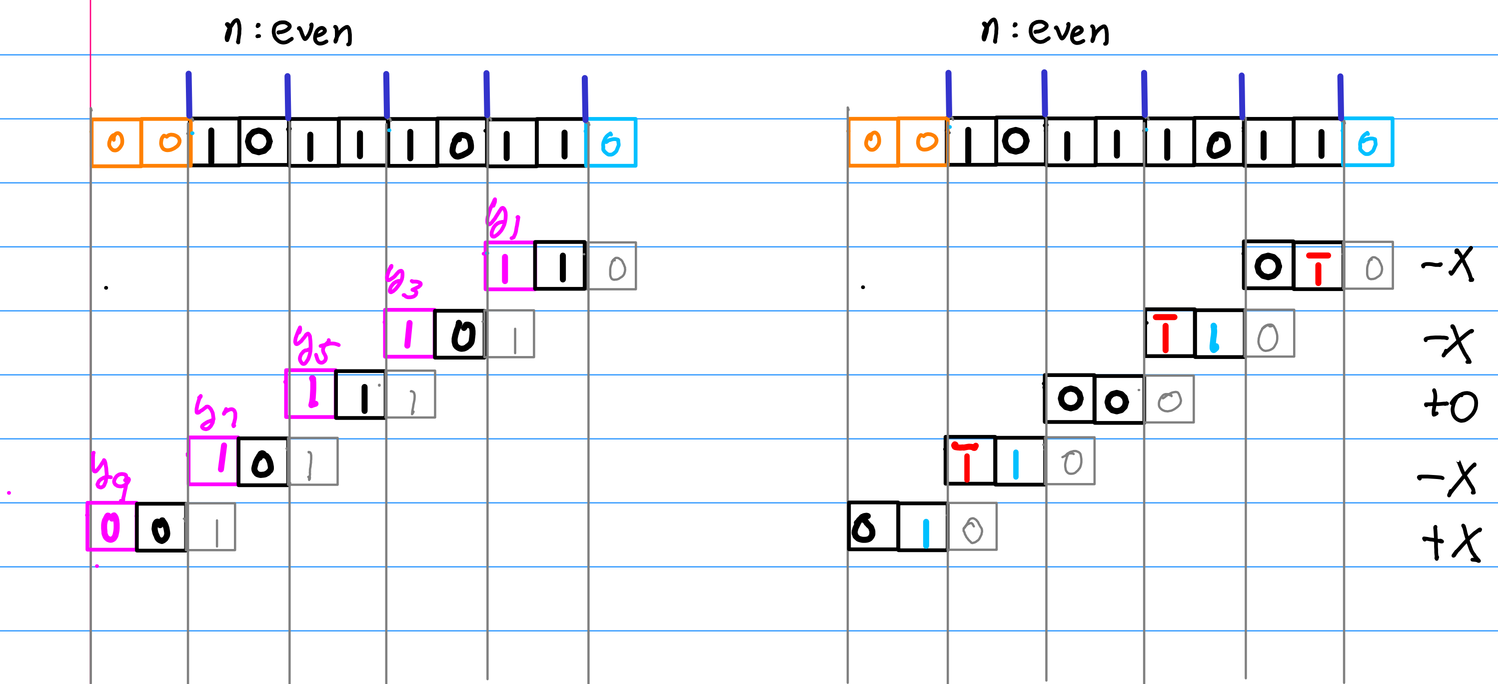
if $y_1 = 1$, add 1 else add 0 ($y_1 = 0$)

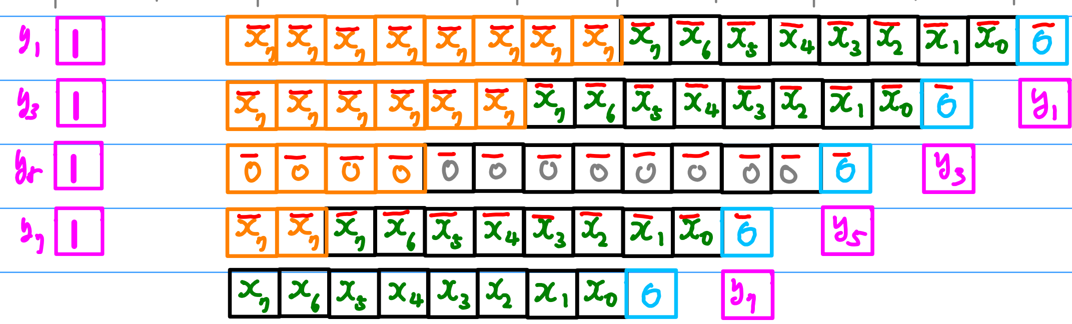
if $y_3 = 1$, add 1 else add 0 ($y_3 = 0$)

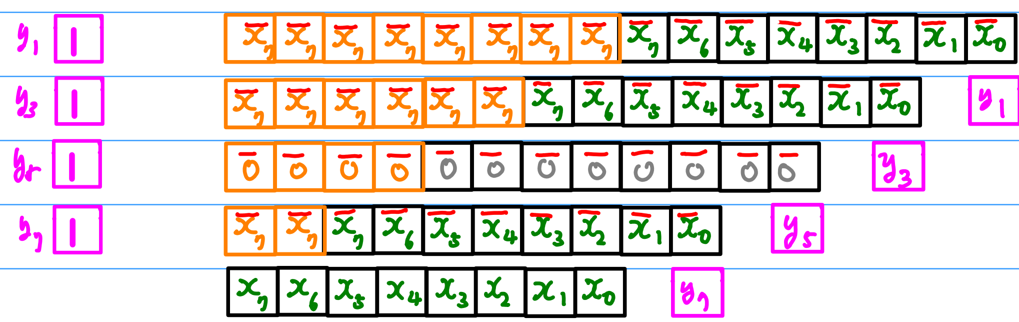
if $y_5 = 1$, add 1 else add 0 ($y_5 = 0$)

if $y_7 = 1$, add 1 else add 0 ($y_7 = 0$)

Handling Negative Scales

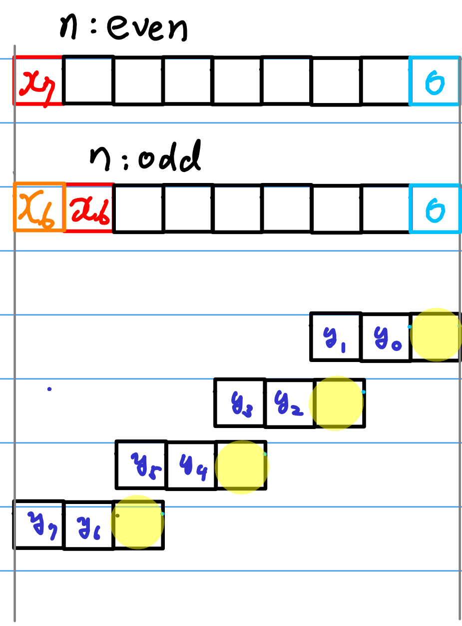






Modified Booth 2 (signed case)

the worst case $\frac{N}{2}$ partial products



multiplicand X

multiplier Y

$\frac{8}{2} = 4$

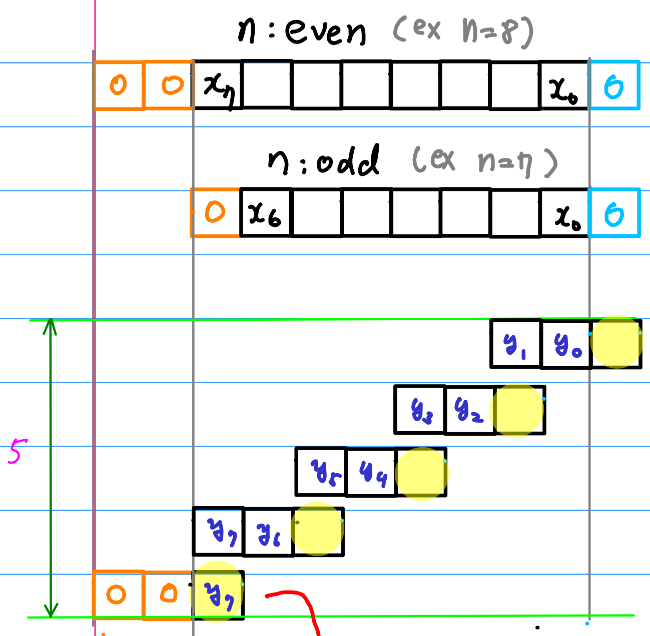
make 4 partial products

x_7, y_n sign bit

Compared to unsigned multiplication

One less partial products

Booth 2 **Unsigned** Case



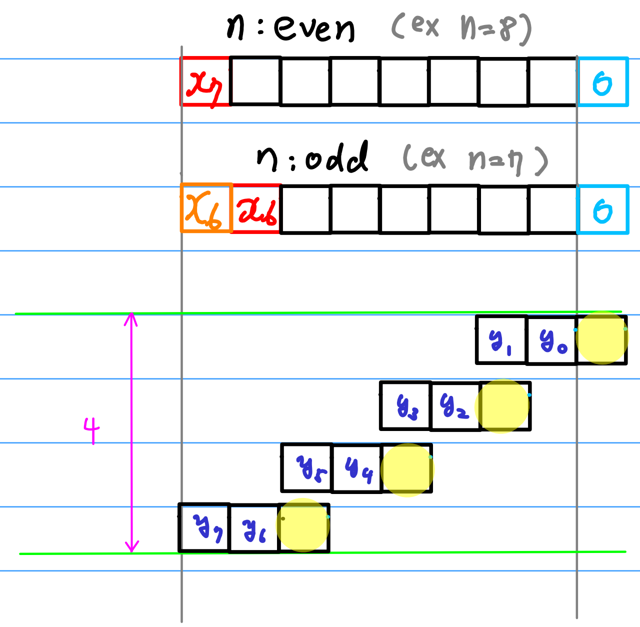
$\frac{8}{2} + 1$

$= 5$

0 0 0	$(y_7 = 0)$	$+0$
0 0 1	$(y_7 = 1)$	$+X$

Necessary to make all positive number (unsigned)

Booth 2 **Signed** case



$\frac{8}{2} = 4$

Booth 2 code :

Signed Digit Number

→ no special treatment for the Sign bit y_7

4-Bit unsigned number & Booth 2 Code

0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	1	$\bar{1}$	2 -1 = 1
2	0	0	0	1	0	0	0	1	$\bar{1}$	0	4 -2 = 2
3	0	0	0	1	1	0	0	1	0	$\bar{1}$	4 -1 = 3
4	0	0	1	0	0	0	1	$\bar{1}$	0	0	8 -4 = 4
5	0	0	1	0	1	0	1	$\bar{1}$	1	$\bar{1}$	8 -4 +2 -1 = 5
6	0	0	1	1	0	0	1	0	$\bar{1}$	0	8 -2 = 6
7	0	0	1	1	1	0	1	0	0	$\bar{1}$	8 -1 = 7
8	0	1	0	0	0	1	$\bar{1}$	0	0	0	16 -8 = 8
9	0	1	0	0	1	1	$\bar{1}$	0	1	$\bar{1}$	16 -8 +2 -1 = 9
10	0	1	0	1	0	1	$\bar{1}$	1	$\bar{1}$	0	16 -8 +4 -2 = 10
11	0	1	0	1	1	1	$\bar{1}$	1	0	$\bar{1}$	16 -8 +4 -1 = 11
12	0	1	1	0	0	1	0	$\bar{1}$	0	0	16 -4 = 12
13	0	1	1	0	1	1	0	$\bar{1}$	1	$\bar{1}$	16 -4 +2 -1 = 13
14	0	1	1	1	0	1	0	0	$\bar{1}$	0	16 -2 = 14
15	0	1	1	1	1	1	0	0	0	$\bar{1}$	16 -1 = 15

Booth 2 code :

Signed Digit Number

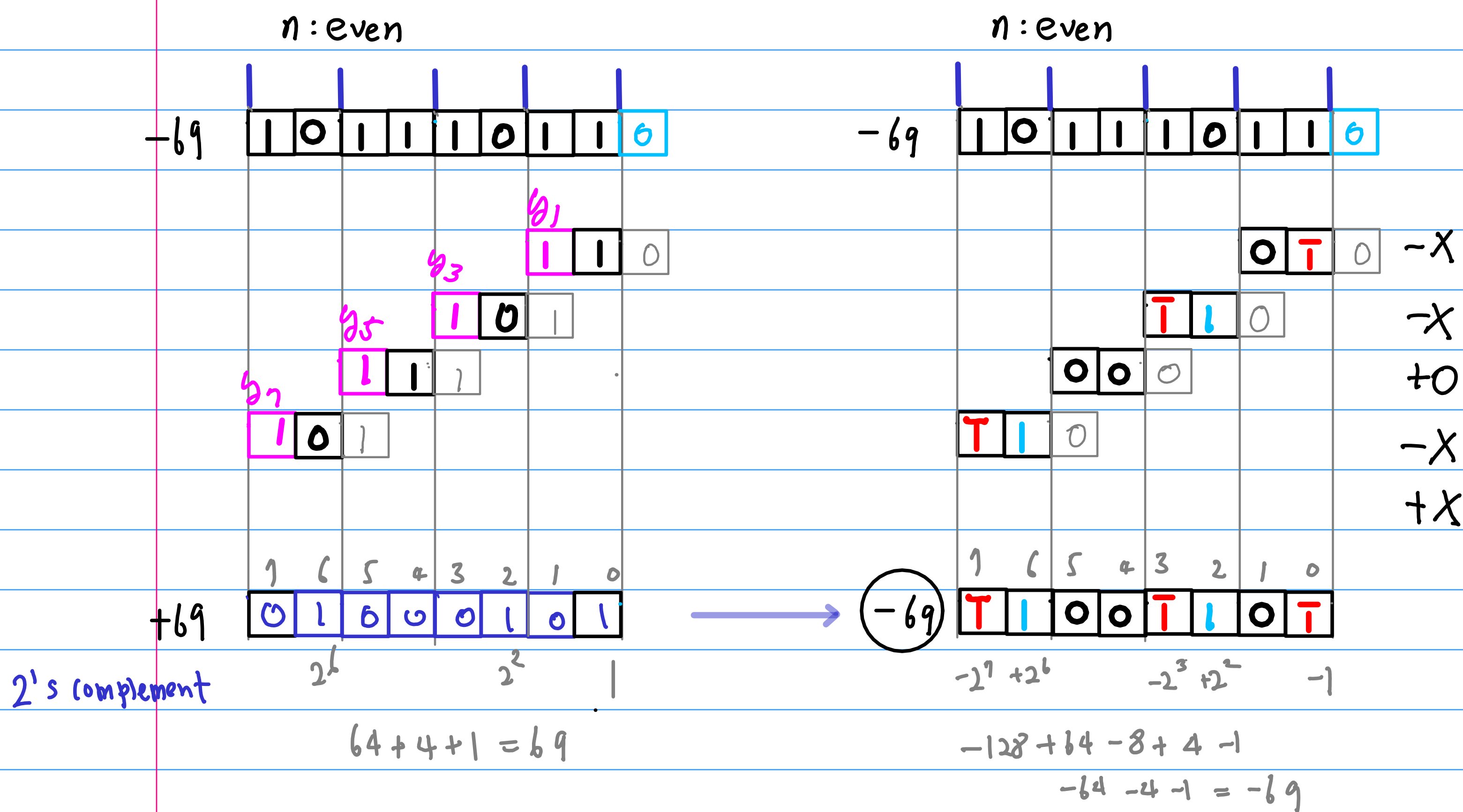
→ Need special treatment

to make all positive number

4-Bit 2's complement number & Booth 2 Code (Signed)

0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1	$\bar{1}$	2 - 1 = 1
2	0	0	1	0	0	1	$\bar{1}$	0	4 - 2 = 2
3	0	0	1	1	0	1	0	$\bar{1}$	4 - 1 = 3
4	0	1	0	0	1	$\bar{1}$	0	0	8 - 4 = 4
5	0	1	0	1	1	$\bar{1}$	1	$\bar{1}$	8 - 4 + 2 - 1 = 5
6	0	1	1	0	1	0	$\bar{1}$	0	8 - 2 = 6
7	0	1	1	1	1	0	0	$\bar{1}$	8 - 1 = 7
-8	1	0	0	0	$\bar{1}$	0	0	0	-8
-7	1	0	0	1	$\bar{1}$	0	1	$\bar{1}$	-8 + 2 - 1 = -7
-6	1	0	1	0	$\bar{1}$	1	$\bar{1}$	0	-8 + 4 - 2 = -6
-5	1	0	1	1	$\bar{1}$	1	0	$\bar{1}$	-8 + 4 - 1 = -5
-4	1	1	0	0	0	$\bar{1}$	0	0	-4
-3	1	1	0	1	0	$\bar{1}$	1	$\bar{1}$	-4 + 2 - 1 = -3
-2	1	1	1	0	0	0	$\bar{1}$	0	-2
-1	1	1	1	1	0	0	0	$\bar{1}$	-1

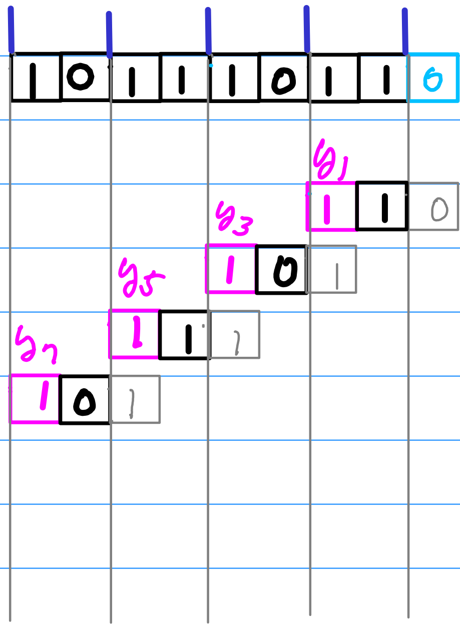
Booth 2 code : Signed Digit Number
→ no special treatment for the Sign bit y_n



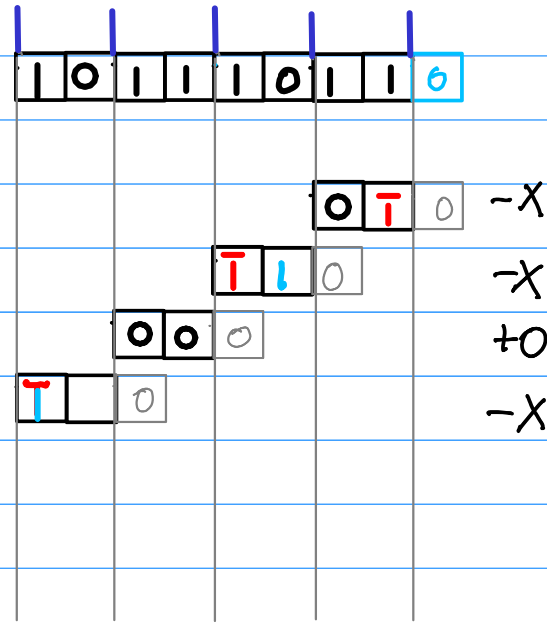
no problem in Booth encoding

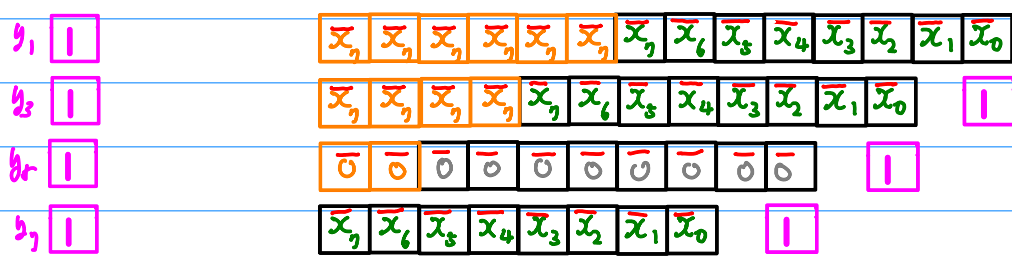
signed numbers!

n : even



n : even

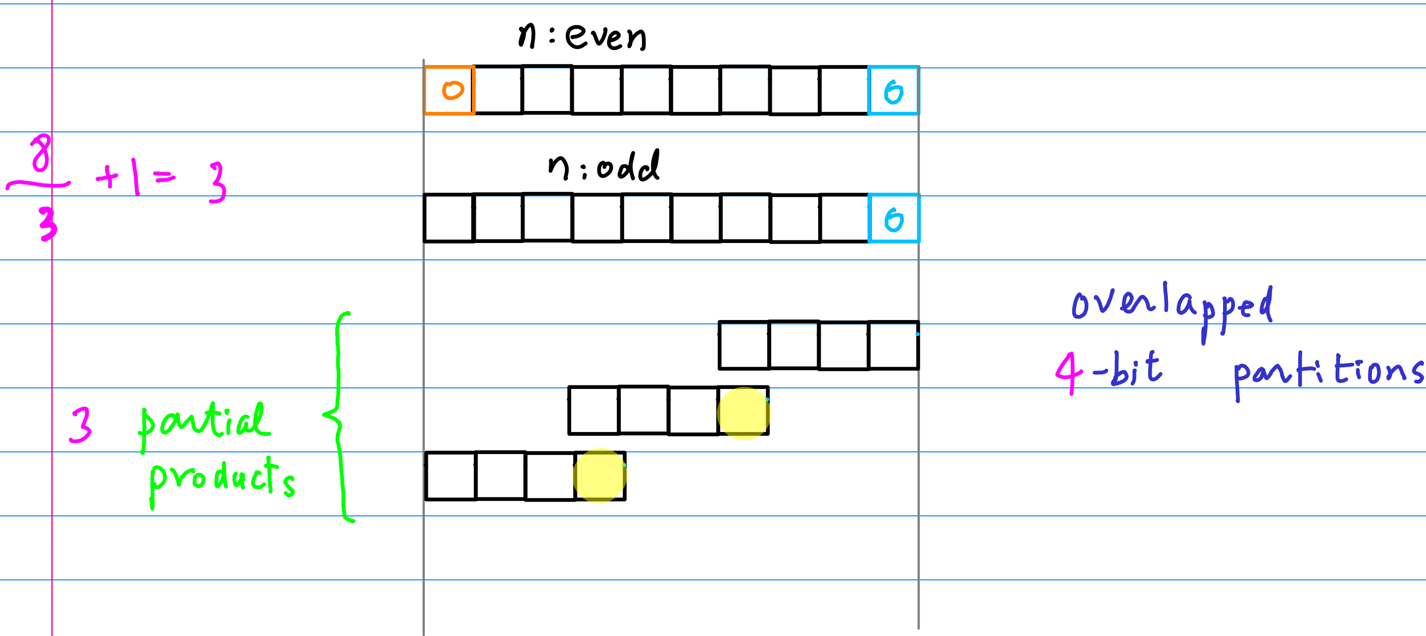




Modified Booth 3 (unsigned)

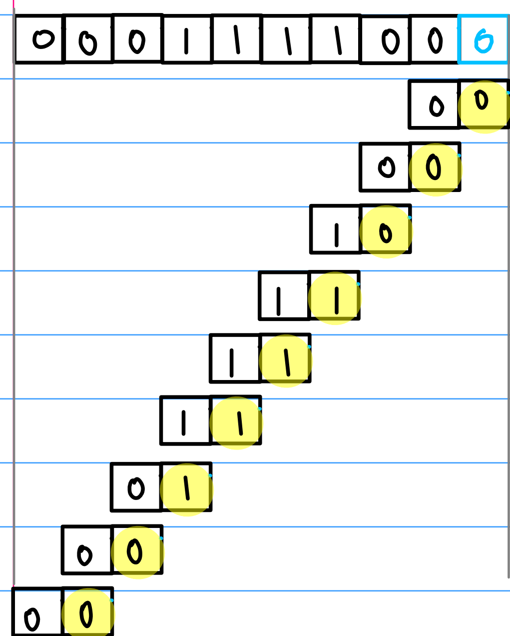
the worst case $\frac{N}{3}+1$ partial products

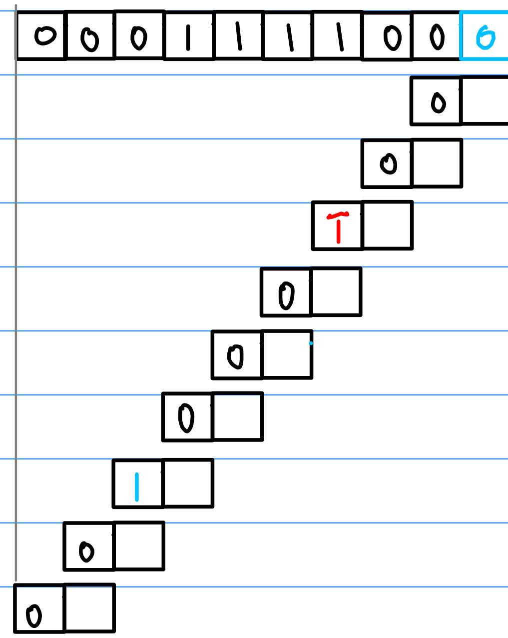
$\frac{8}{3}+1=3$



Scale factor

0 0 0 0	0 0 0 0	+0	1 0 0 0	$\bar{1}$ 0 0 0	−4X
0 0 0 1	0 0 1 0	+X	1 0 0 1	$\bar{1}$ 0 1 0	−3X
0 0 1 0	0 1 $\bar{1}$ 0	+X	1 0 1 0	$\bar{1}$ 1 $\bar{1}$ 0	−3X
0 0 1 1	0 1 0 0	+2X	1 0 1 1	$\bar{1}$ 1 0 0	−2X
0 1 0 0	1 $\bar{1}$ 0 0	+2X	1 1 0 0	0 $\bar{1}$ 0 0	−2X
0 1 0 1	1 $\bar{1}$ 1 0	+3X	1 1 0 1	0 $\bar{1}$ 1 0	− X
0 1 1 0	1 0 $\bar{1}$ 0	+3X	1 1 1 0	0 0 $\bar{1}$ 0	− X
0 1 1 1	1 0 0 0	+4X	1 1 1 1	0 0 0 0	− 0


0 0 0 1 1 1 1 0 0 6
0 0
0 0
1 0
1 1
1 1
1 1
0 1
0 0
0 0


0 0 0 1 1 1 1 0 0 6
0
0
T
0
0
0
1
0
0