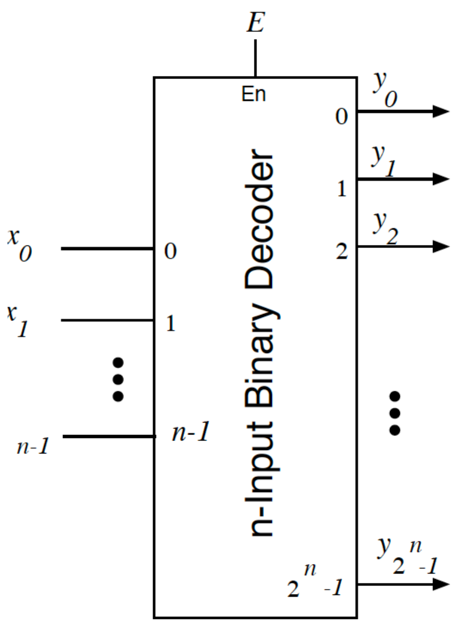
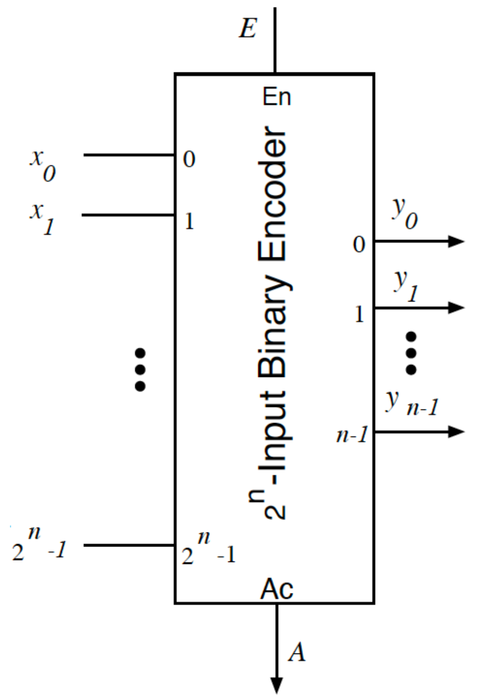
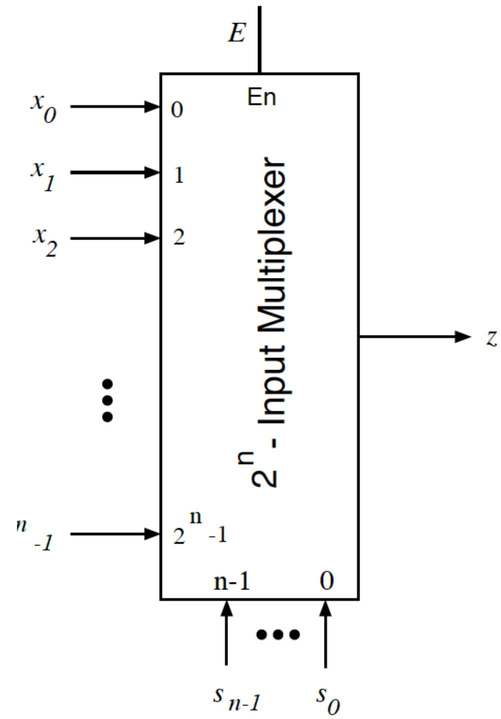
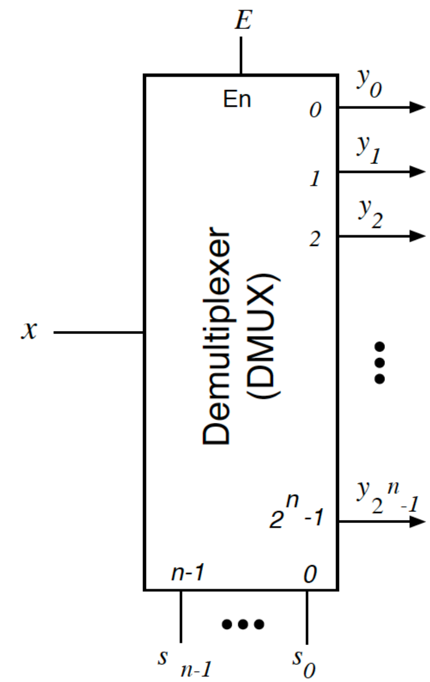
CS M51A
Logic Design of Digital Systems
Winter 2021

Some slides borrowed and modified from:
M.D. Ercegovac, T. Lang and J. Moreno, Introduction to Digital Systems.

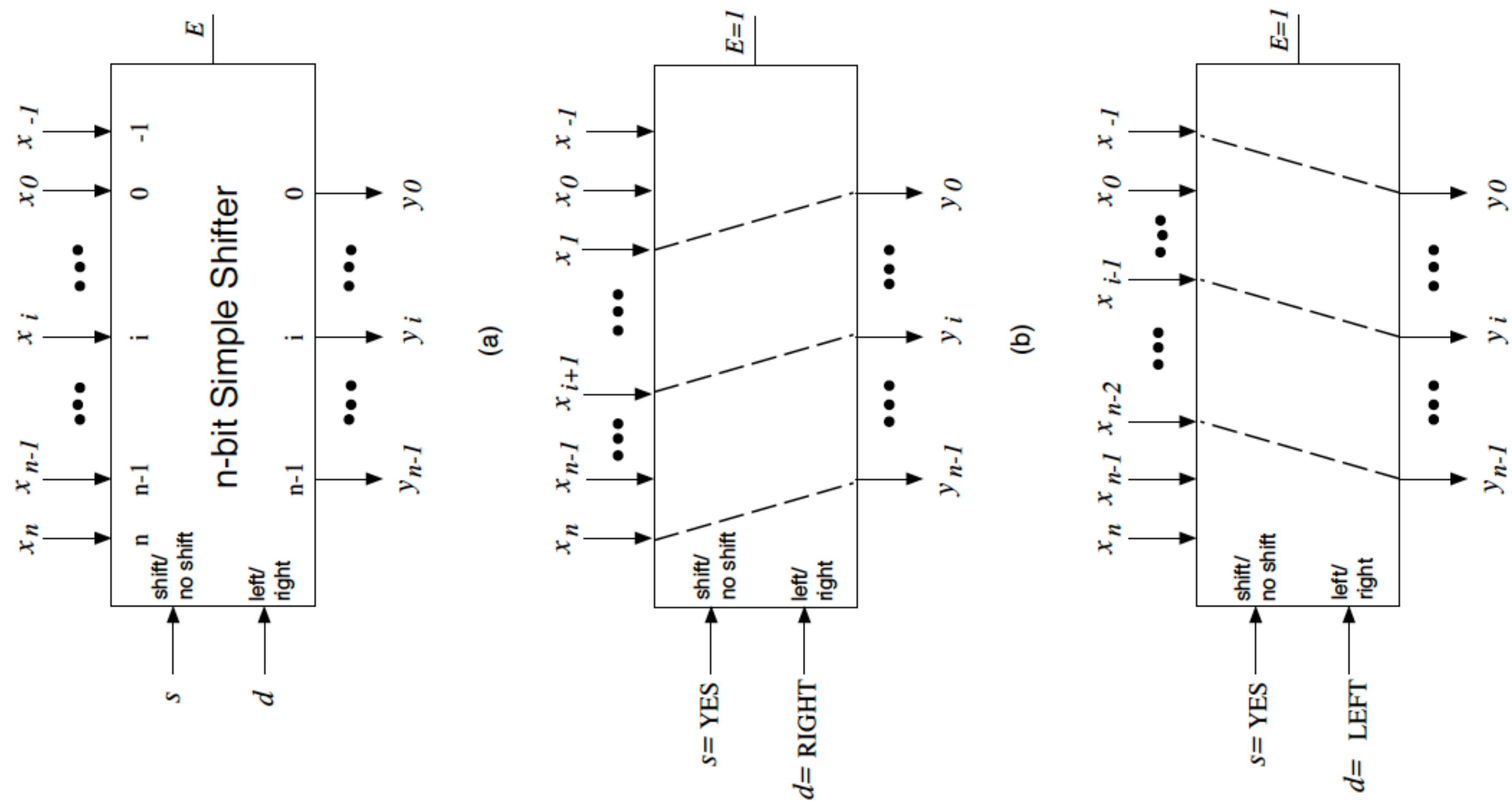

E
En
n-Input Binary Decoder
x_0
x_1
n-1
0
1
n-1
y_0
y_1
y_2
0
1
2
2^n-1
y_{2^n-1}


E
En
2^n-Input Binary Encoder
x_0
x_1
2^n-1
0
1
2^n-1
y_0
y_1
y_{n-1}
0
1
n-1
Ac
A


E
En
2^n - Input Multiplexer
x_0
x_1
x_2
^{n}-1
0
1
2
2^n-1
n-1
0
s_{n-1}
s_0
z


E
En
Demultiplexer
(DMUX)
x
y_0
y_1
y_2
y_{2^n-1}
0
1
2
2^n-1
n-1
0
s_{n-1}
s_0

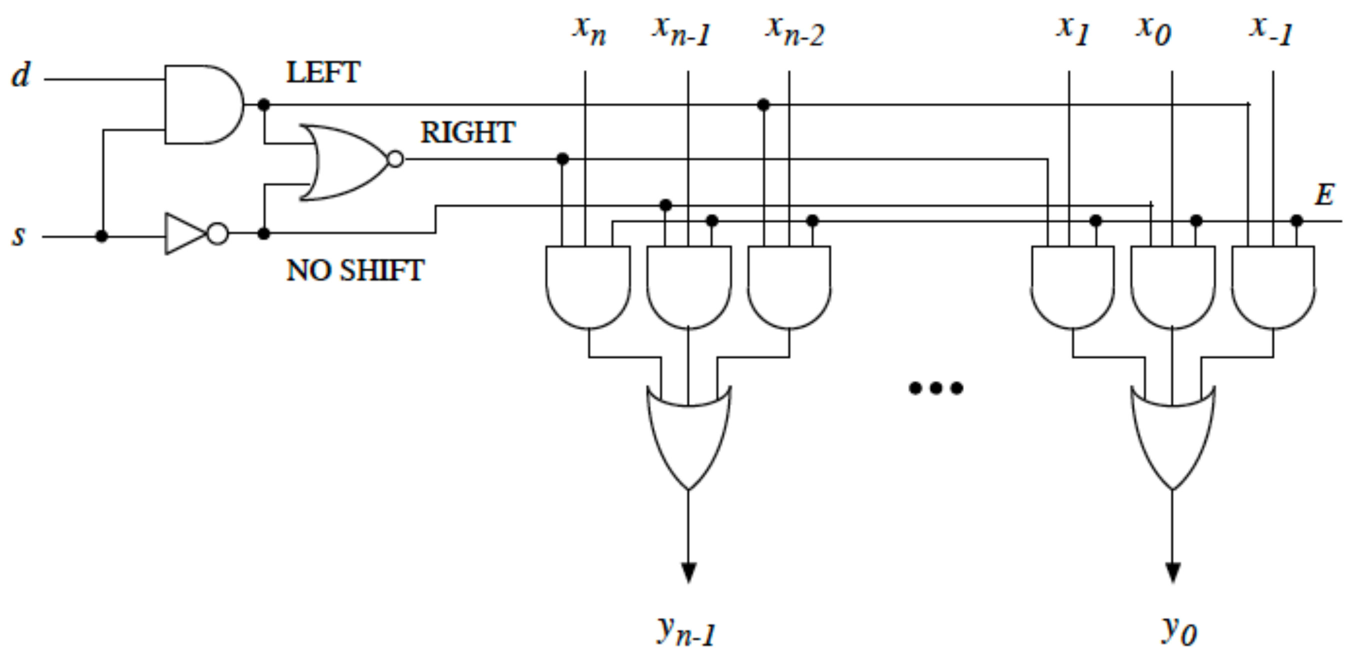
SIMPLE SHIFTER



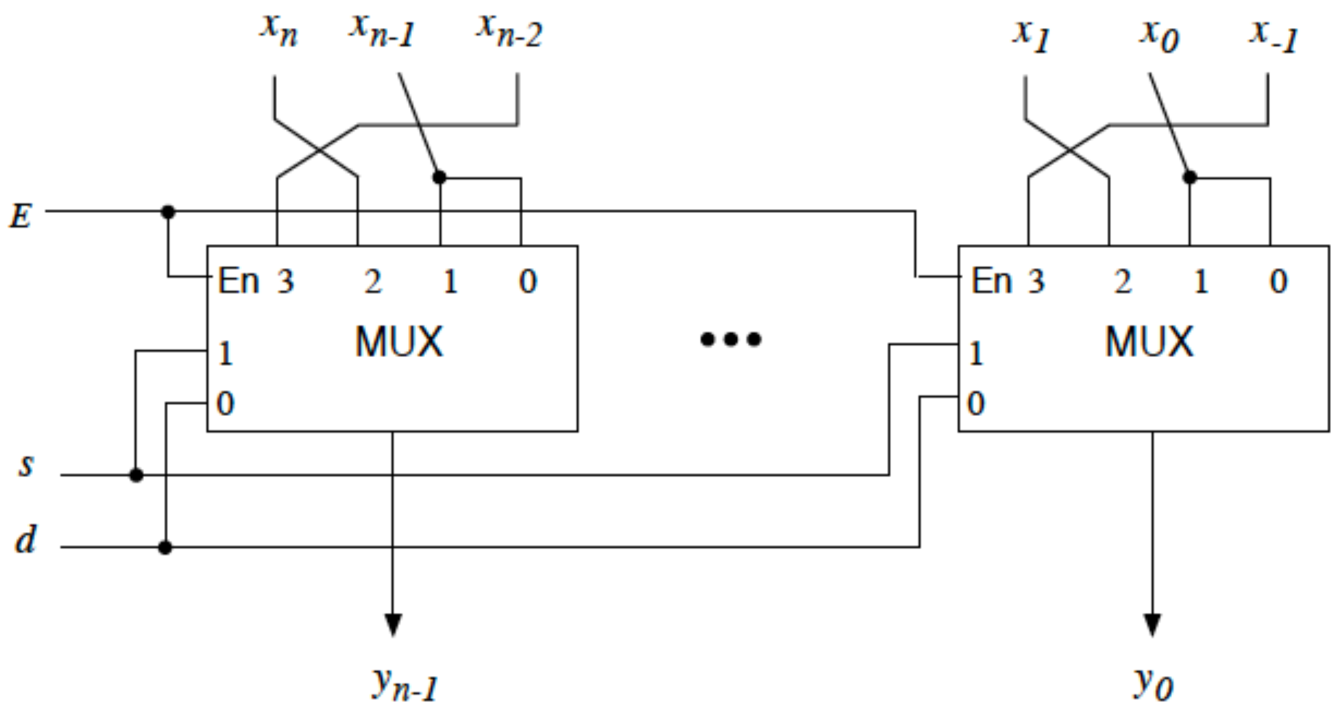
EXAMPLE: 4-INPUT SHIFTER

	Control		Data					
	s	d	x_4	x_3	x_2	x_1	x_0	x_{-1}
			1	0	0	1	1	0
No shift	NO	–		0	0	1	1	
Right shift	YES	RIGHT		1	0	0	1	
Left shift	YES	LEFT		0	1	1	0	
				y_3	y_2	y_1	y_0	

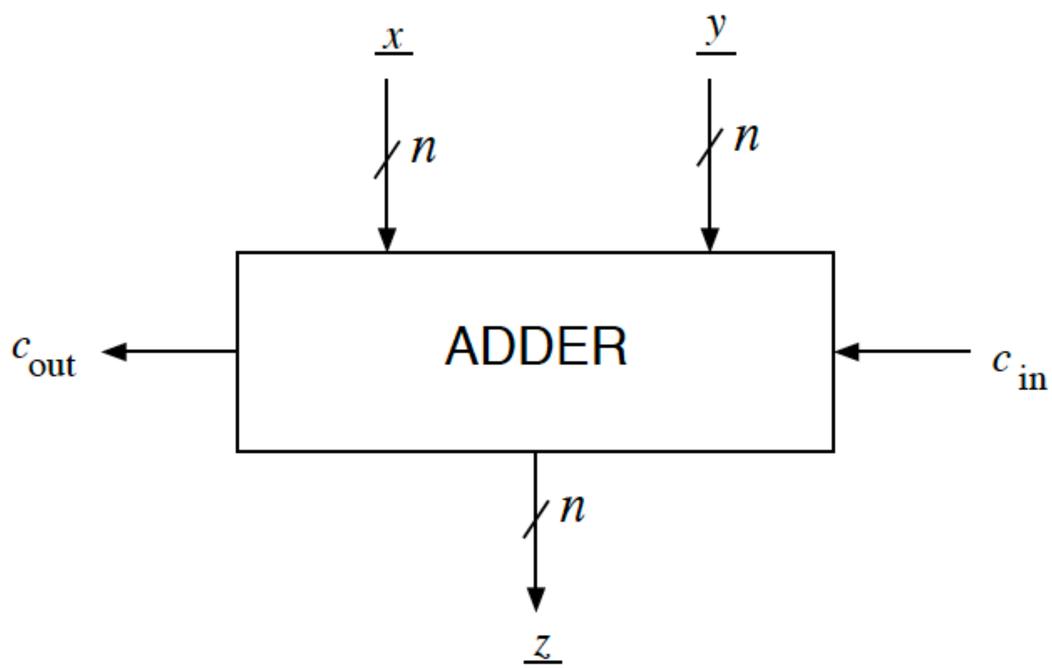
Shifter Implementation



Shifter Implementation

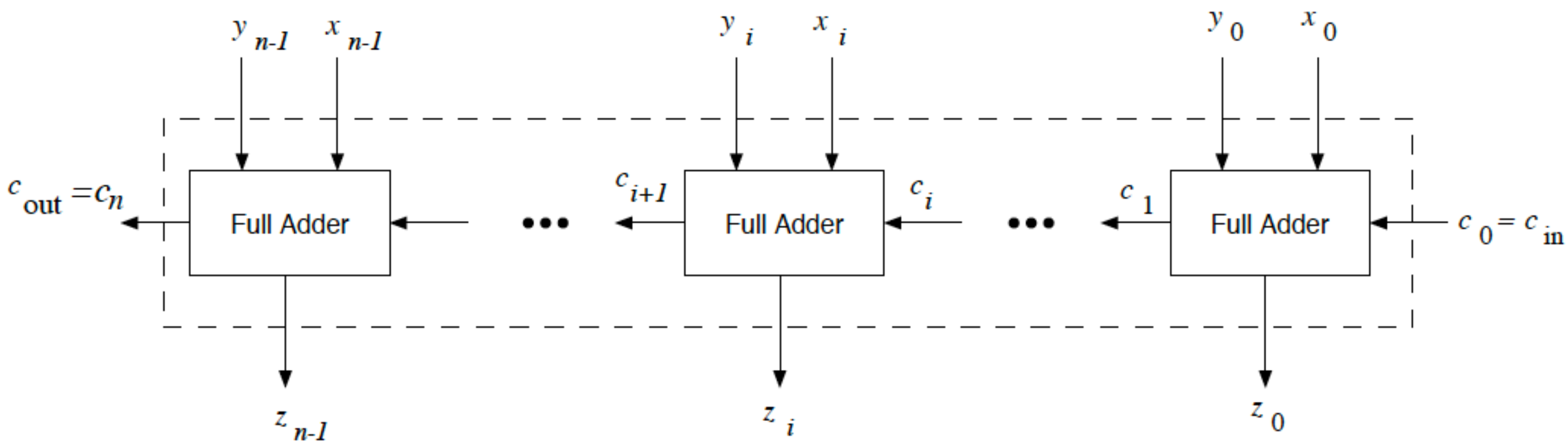


ADDER MODULES FOR POSITIVE INTEGERS



$$x + y + c_{\text{in}} = 2^n c_{\text{out}} + z$$

CARRY-RIPPLE ADDER IMPLEMENTATION



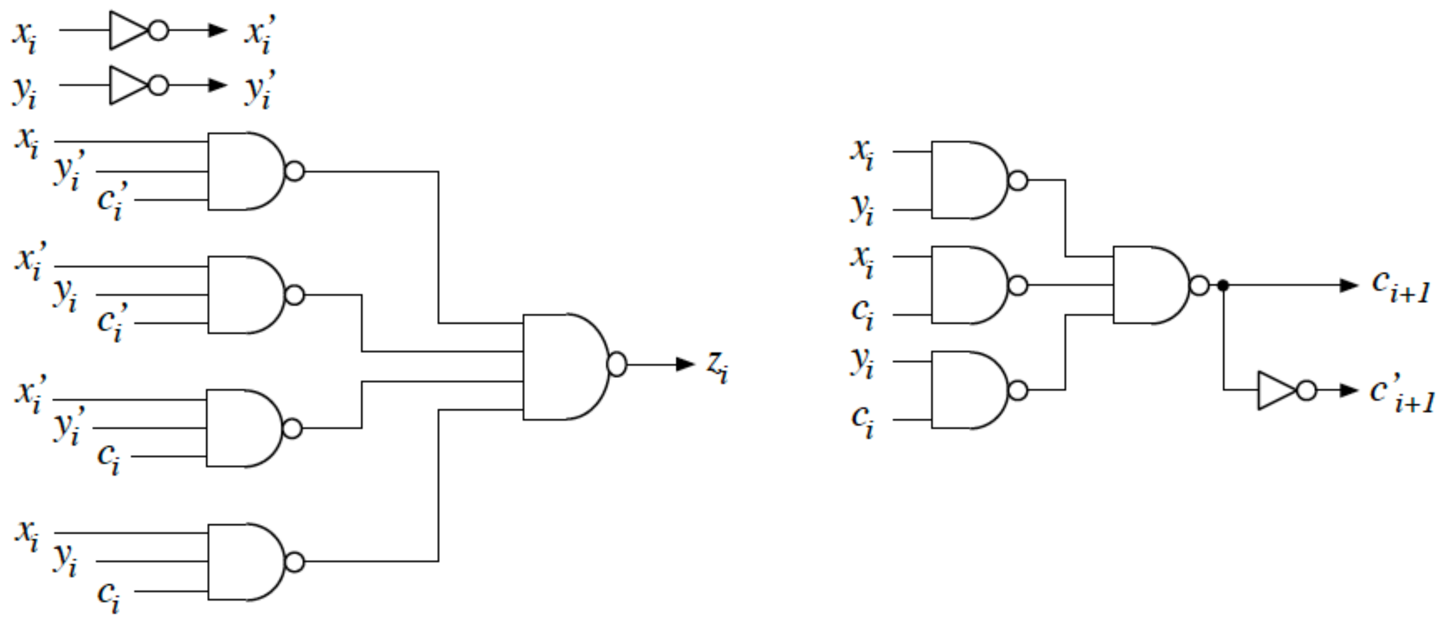
FULL-ADDER IMPLEMENTATION

x_i	y_i	c_i	c_{i+1}	z_i
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

$$z_i =$$

$$c_{i+1} =$$

FULL ADDER TWO-LEVEL IMPLEMENTATION



ALTERNATIVE IMPLEMENTATION

- SUM IS 1 WHEN NUMBER OF 1'S IN INPUTS (including the carry-in) IS ODD:

$$z_i = x_i \oplus y_i \oplus c_i$$

- CARRY-OUT IS 1 WHEN $(x_i + y_i = 2)$ or $(x_i + y_i = 1$ and $c_i = 1)$:

$$c_{i+1} = x_i y_i + (x_i \oplus y_i) c_i$$

x_i	y_i	c_i	c_{i+1}	z_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

ALTERNATIVE IMPLEMENTATION

$$z_i = x_i \oplus y_i \oplus c_i$$

$$c_{i+1} = x_i y_i + (x_i \oplus y_i) c_i$$

- INTERMEDIATE VARIABLES

$$\begin{array}{rl} \text{PROPAGATE} & p_i = x_i \oplus y_i \\ \text{GENERATE} & g_i = x_i \cdot y_i \end{array}$$

- HALF-ADDER

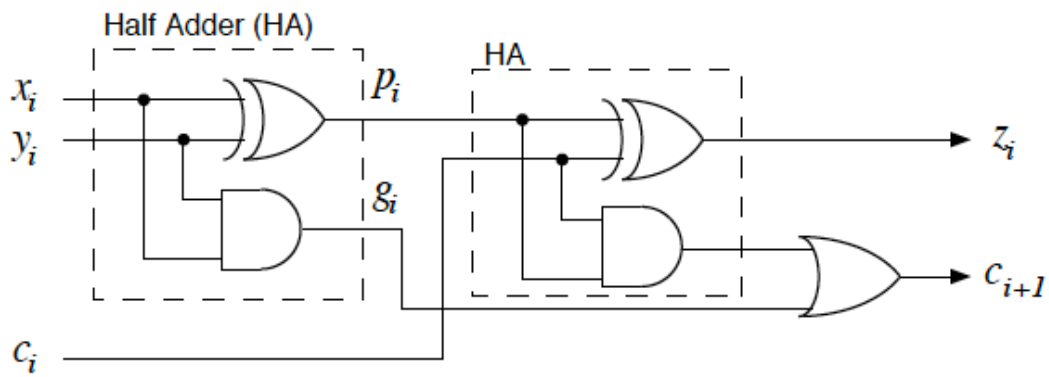
x_i	y_i	g_i	p_i
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

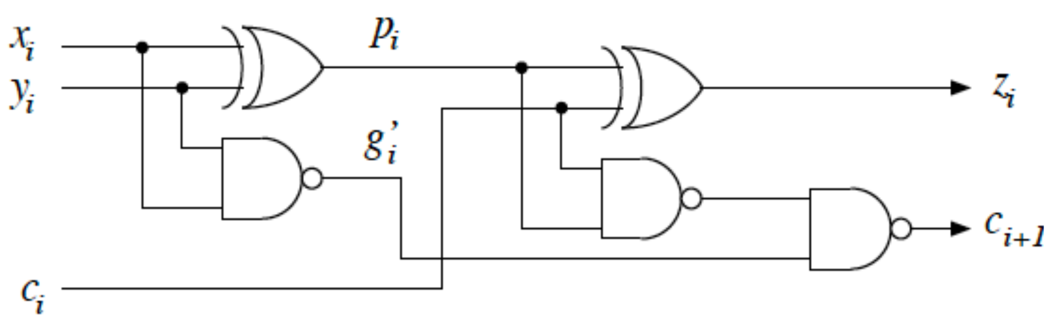
IMPLEMENTATION WITH HALF-ADDERS

EXPRESSIONS IN TERMS OF $p_i's$, $g_i's$ and $c_i' s$

$$z_i = p_i \oplus c_i$$

$$c_{i+1} = g_i + p_i \cdot c_i$$

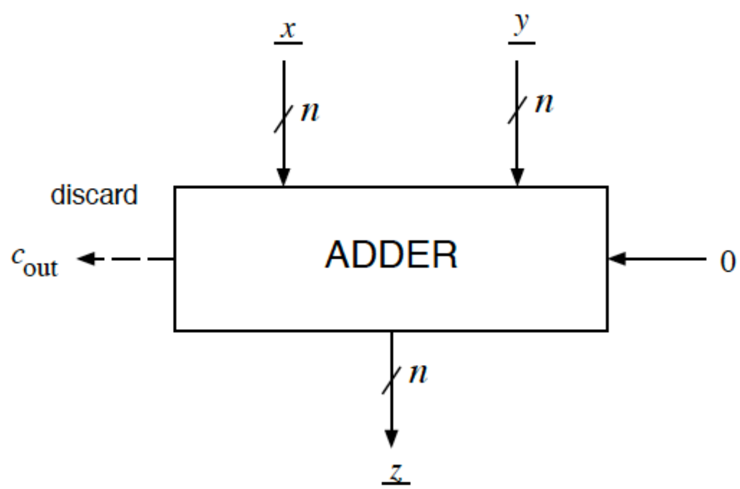




2's Complement Addition

- 2'S COMPLEMENT ADDITION:
  RESULT CORRESPONDS TO OUTPUT OF ADDER, DISCARDING THE CARRY-OUT

$$\underline{z} = ADD(\underline{x}, \underline{y}, 0)$$



SUBTRACTION IN TWO'S COMPLEMENT SYSTEM

- THE CORRESPONDING DESCRIPTION

$$\underline{z} = ADD_R(\underline{x}, \underline{y}', 1)$$

EXAMPLE:

$$
\begin{array}{cc|cr}
\underline{x} & & & 01100000 \\
\underline{y} & 00110001 & \underline{y}' & 11001110 \\
 & & & 1 \\
\underline{z} & & & 00101111
\end{array}
$$

2's Complement Operations

OPERATION	2's COMPLEMENT SYSTEM
$z = x + y$	$\underline{z} = ADD(\underline{x}, \underline{y}, 0)$
$z = -x$	$\underline{z} = ADD(\underline{x'}, 0, 1)$
$z = x - y$	$\underline{z} = ADD(\underline{x}, \underline{y'}, 1)$

TWO'S COMPLEMENT ARITHMETIC UNIT

INPUTS:

$\underline{x} = (x_{n-1}, \ldots, x_0), \quad x_j \in \{0, 1\}$

$\underline{y} = (y_{n-1}, \ldots, y_0), \quad y_j \in \{0, 1\}$

$c_{\text{in}} \in \{0, 1\}$

$F = (f_2, f_1, f_0)$

OUTPUTS:

$\underline{z} = (z_{n-1}, \ldots, z_0), \quad z_j \in \{0, 1\}$

$c_{\text{out}}, sgn, zero, ovf \in \{0, 1\}$

FUNCTIONS:

F	Operation		
001	ADD	add	$z = x + y$
011	SUB	subtract	$z = x - y$
101	ADDC	add with carry	$z = x + y + c_{in}$
110	CS	change sign	$z = -x$
010	INC	increment	$z = x + 1$

$sgn = 1$ **if** $z < 0$, 0 **otherwise** (the sign)

$zero = 1$ **if** $z = 0$, 0 **otherwise**

$ovf = 1$ **if** z overflows, 0 **otherwise**

TWO'S COMPLEMENT ARITHMETIC UNIT