Example 1

Do the following operations on numbers A and B using binary arithmetic.

a) A + B   b) A - B

c) A * B   d) A / B

A = 100110, B = 1101

b.2

(+)

```
     1 1
a)  100110   ->  38 (10
 +    1101       13 (10
    ------      -------
    110011       51 (10

1+2+16+32=51
```

```
                10 -> 2

b)    100110        38
    -  11101        13
      ------        --
      011001        25

1+8+16=25  ✓
```

```
c)         100110
       x     1101
       ----------
          1 1100110
          100110
    +   100110
       ----------
        111101110

3=01
```

```
d)  100110 |1101
     1101  |10      -> 2         {0,1}
     ----
    001100 -> 12

    38 |13
    26 |2
    --
    12
```

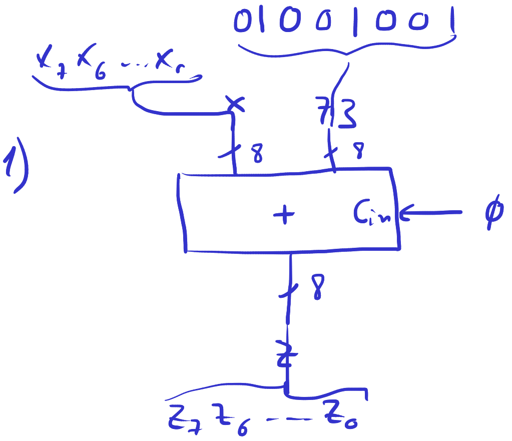
Example 2

Input signal x and output signal z are 8-bits wide. Design circuits to perform the following operations considering two alternatives: 1) using magnitude adders, 2) using basic adder blocks (FA and HA).

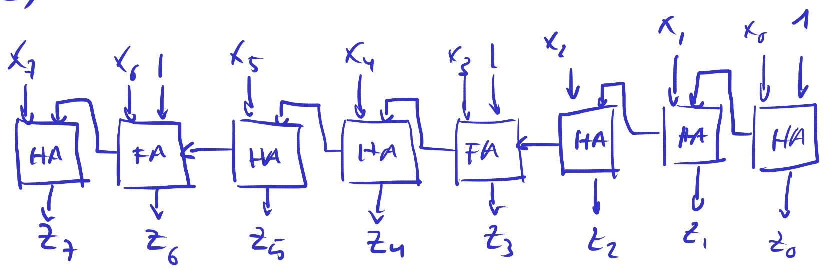
a) z = x + 73 b) z = 2 * x c) z = 5 * x

a)

1)

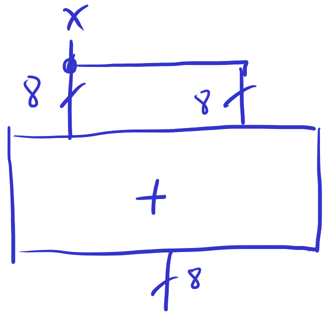


2)



b)

1)



$z = (x + x = 2x) \bmod 256$

Eg. $x = 1101 \rightarrow 2x = 11010$

2)

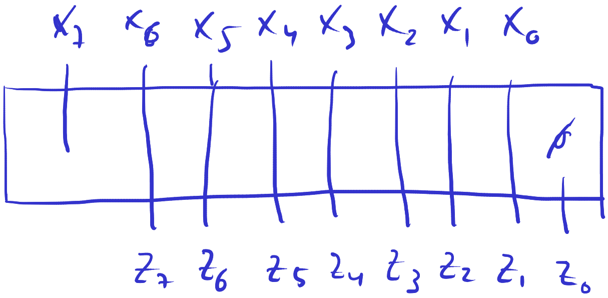
$x = x_7 2^7 + x_6 2^6 + \dots + x_1 2 + x_0$

$2x = x_7 2^8 + x_6 2^7 + \dots + x_1 2^2 + x_0 \cdot 2$

$\parallel$

$z = z_7 2^7 + \dots + z_2 2^2 + z_1 2 + z_0$

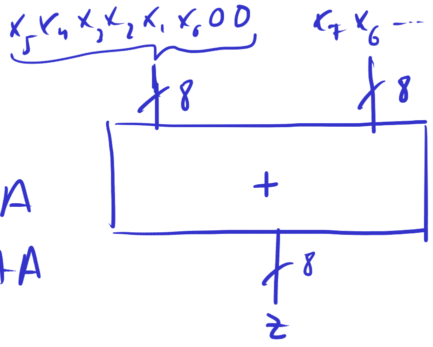
$\times 2$



c) $z = 5x = \underset{\hookrightarrow 2^2}{4}x + x = x + x + x + x + x$

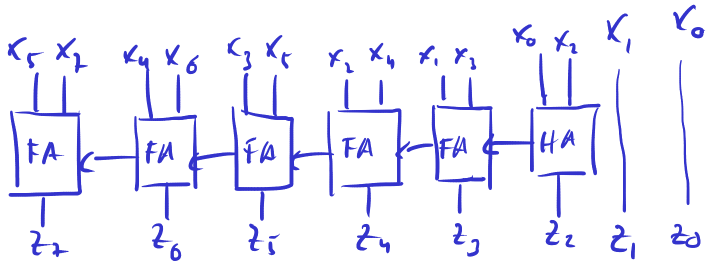
$$\begin{array}{rcccccccc} & x_5 & x_4 & x_3 & x_2 & x_1 & x_0 & 0 & 0 \\ + & x_7 & x_6 & x_5 & x_4 & x_3 & x_2 & x_1 & x_0 \\ \hline & z_7 & & & \dots & z_3 & z_2 & z_1 & z_0 \end{array}$$

1)



7xFA
1xHA

2)



Example 3

Represent the following number in S-M, OC and TC notation with 8 bits.

a) 32, b) -13, c) 115, d) -140, e) 128, f) -128

-128	64	32	16	8	4	2	1
1	1	1	1	0	0	1	1

$\boxed{S\text{-}M}$

a) $32_{10} = 100000_2$

32_{10} 00100000_{SM}

c) $115_{10} = 1110011_2$

01110011_{SM}

e) $128_{10} = 10000000_2$

↓

Not possible.

b) $13_{10} = 1101_2$

10001101_{SM}

d) $2^7 - 1 = 127$ → Not possible.

$140_{10} = \underbrace{10001100}_{8\text{ bits.}}{}_2$ ↑

f) Not possible.

$\boxed{OC}$

↓

8 bits.

1st bit sign.

↑

a) 00100000_{OC} ↑

c) 01110011_{OC}

e) $128_{10} = \cancel{1}0000000$

Not posible

b) -13 ; $13_{10} = 00001101_{OC}$

$-13_{10} = 11110010_{OC}$

d) $\cancel{0}1110011$?

Not posible.

f) $\cancel{0}1111111$? -128

Not posible

$\boxed{TC}$

a) 00100000_{TC}

c) $115_{10} = 01110011_{TC}$

e) $128 = \cancel{1}0000000$

Not posible

f) $128 = 10000000$) TC

$-128 = 10000000_{TC}$

b) $13_{10} = 00001101$) TC ←

$-13_{10} = 11110011_{TC}$

d) $+140$ | 10001100) TC

-140 $\cancel{0}1110100$

Not pos.

$TC_n(x) = OC_n(x) + 1$

Example 4

Obtain the minimum number of bits to represent the following number in S-M, OC and TC notation and represent them:

a) 32, b) -13, c) 115, d) -140, e) 128, f) -128

a) $32_{(16} = 100000_{(2}$

S-M: 0100000

OC: 0100000

TC: 0100000

b) $-13_{(10} \Rightarrow 13_{(10} = 1101$

S-M: 11101

OC: $01101 \xrightarrow{OC} 10010_{(OC}$

TC: $01101 \xrightarrow{TC} 10011_{TC}$

c) $115_{(10} = 1110011_{2}$

SM: 01110011

OC:

TC:

d) $-140_{(10} \Rightarrow 140_{(10} = 10001000_{(2}$

SM: 110001000

OC: $010001000 \xrightarrow{OC} 101110111_{OC}$

TC: $\xrightarrow{TC} 101111000_{TC}$

e) $128_{(10} = 10000000_{(10}$

SM: 010000000

OC:

TC:

f) $-128_{(10} \Rightarrow 128_{(10} = 10000000_{(2}$

SM: 110000000

OC: $010000000 \xrightarrow{OC} 1|01111111$

TC:

$1|10000000_{TC}$

$13_{(10} = \cancel{0}\cancel{0}01101$ SM OC TC

$-13 \xrightarrow{OC} \cancel{1}\cancel{1}|10010$

$\xrightarrow{TC} \cancel{1}\cancel{1}|\underbrace{10011}$

$TC_8 \quad -2^7 \le x \le 2^7 - 1$

$\quad\quad\quad -128 \quad\quad\quad +127$

Example 5

Calculate the decimal value of the following words when interpreted in S-M, OC and TC representation with 8 bits.

a) 01001100, b) 11110000

a)

01001100 → SM: $+(4+8+64) = +56$

→ OC: $+56$

→ TC: $+56$

b)

11110000 → SM: $-(16+32+64) = -112$

→ OC: -15

OC → 00001111 = 15

→ TC: -16

TC → 00010000

$\boxed{\text{TC}}$

-128	64	32	16	8	4	2	1	
0	1	0	0	1	1	0	0	$= 64+8+4 = \underline{\underline{56}}$
1	1	1	1	0	0	0	0	$= -128+64+32+16 = \underline{\underline{-16}}$

Example 6

Represent the following number in TCR with 8 bits using a weights table:

a) 32, b) -13, c) 115, d) -140, e) 128, f) -128

	-128	64	32	16	8	4	2	1	
a) 32	0	0	1	0	0	0	0	0	→ 0 0 1 0 0 0 0 0
b) -13	1	1	1	1	0	0	1	1	→ 1 1 1 1 0 0 1 1
c) 115	0	1	1	1	0	0	1	1	→ 0 1 1 1 0 0 1 1
d) -140	1								→ Not possible.
e) 128	~~0~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	→ Not possible.
f) -128	1	0	0	0	0	0	0	0	→ 1 0 0 0 0 0 0 0
		64	96	112	120	124	126	127	

Example 7

Do the following operations in binary form with numbers in two's complement representation by first extending the TC representations to 8 bits. Check the results repeating the operations in decimal.

a) 011110 + 10100, b) 01101111 + 0100000, c) 11111010 + 100001, d) 11111001 + 11001

$-128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$

a)
$$\begin{array}{rr} {}^{1\,1\,1} & \\ 00011110 & 30 \\ 11110100 & -12 \\ \hline 00010010 = & 18 \end{array}$$

$V = C_n \oplus C_{n-1} = 0$

$0 \ldots 01100$

b)
$$\begin{array}{rr} {}^{0\,1\,1} & \\ 01101111 & 111 \\ 00100000 & 32 \\ \hline 10001111 \neq & 143 \end{array}$$

$-2^{8-1} = -128 \leq x \leq 127 = 2^{8-1} - 1$

c)
$$\begin{array}{rr} {}^{1\,1\,1} & \\ 11111010 & -6 \\ 11100001 & -31 \\ \hline 11011011 = & -37 \end{array}$$

$0 \ldots 0110 = 6$

$0 \ldots 011111 = 31$

$00100101 = 37$

d)
$$\begin{array}{rr} {}^{1\,1\,1} & \\ 11111001 & -7 \\ 11111001 & -7 \\ \hline 11110010 = & -14 \end{array}$$

$0 \ldots 0111 = 7$

$0 \ldots 01110 = 14$

$\times 2 \quad \cancel{1}\,\underbrace{11110010} = -14$

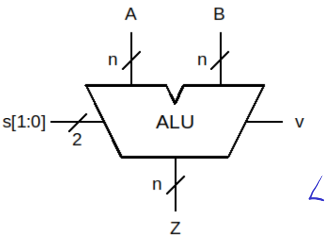
$0 \ldots 01110 = 14$

Example 8

Design an n bit ALU according to the operation table and figure below. Arithmetic operations use two's complement representation. The ALU has an overflow (v) output.

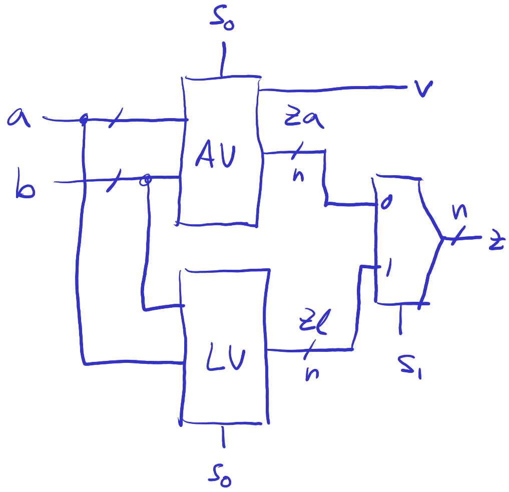
a) Base the design in a magnitude adder with an overflow output.

b) Write a Verilog description.



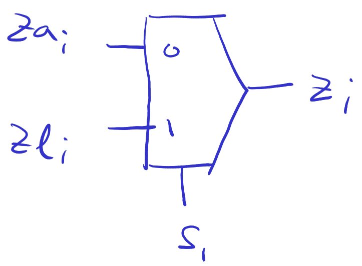
s_1s_0	z
00	a + 2
01	a - b
10	NOT a
11	NOT b

LU



Typical stage

$i = 0, \dots, n-1$



Notation

a → A

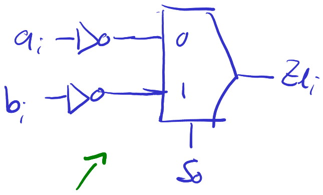
number

word that represents the number

LU

S_0	Zl	ZL	Zl_i
0	NOTa	$\bar{A}$	$\bar{a}_i$
1	NOT b	$\bar{B}$	$\bar{b}_i$

$i = 0, \dots, n-1$

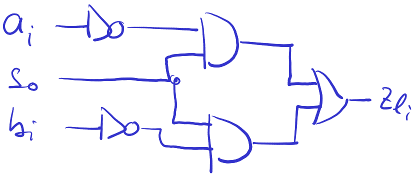


With MUX

$S_0 \backslash a_ib_i$	00	01	11	10
0	1	1	0	0
1	1	0	0	1

$Zl_i = \bar{a}_i \bar{S}_0 + \bar{b}_i S_0$

With gates



AU

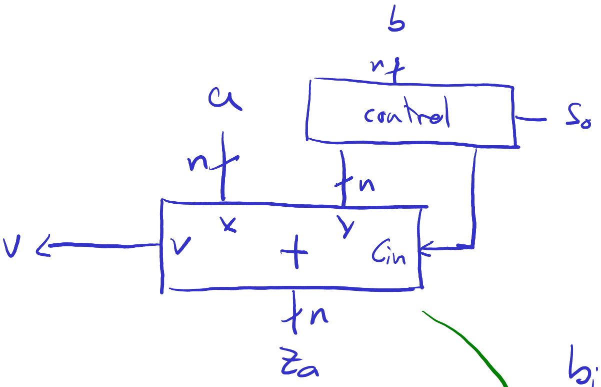
$2 = 000 \dots 010$

S_0	Za	X	Y	Cin	Y_i ($i \neq 1$)	y_1 ($i = 1$)
0	a+2	A	2 1	0 1	0	1
1	a-b	A	$\bar{B}$	1	$\bar{b}_i$	$\bar{b}_1$

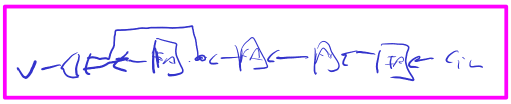
$a + (-b)$

$b \rightarrow B$

$-b \rightarrow TC(B) = \bar{B} + 1$



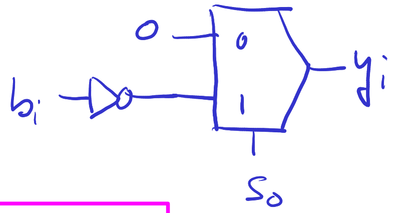
$Za = x + y + Cin$



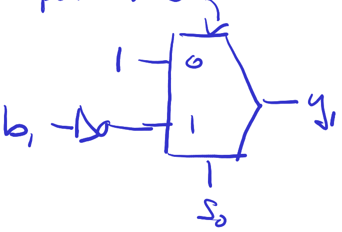
MUX

typical

$i \neq 1$



particular



$C_{in} = S_0$

S_0 —— C_{in}

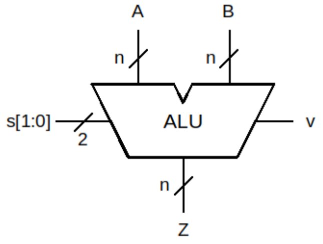
Example 9

Design an n bit ALU according to the operation table and figure below. Arithmetic operations use two's complement representation. The ALU has an overflow (v) output.

a) Base the design in a magnitude adder with an overflow output.

b) Write a Verilog description.

(NOTE: be careful when calculating the overflow)



s_1s_0	z
00	a + 2b
01	a - 4
10	NOT a
11	a XOR b

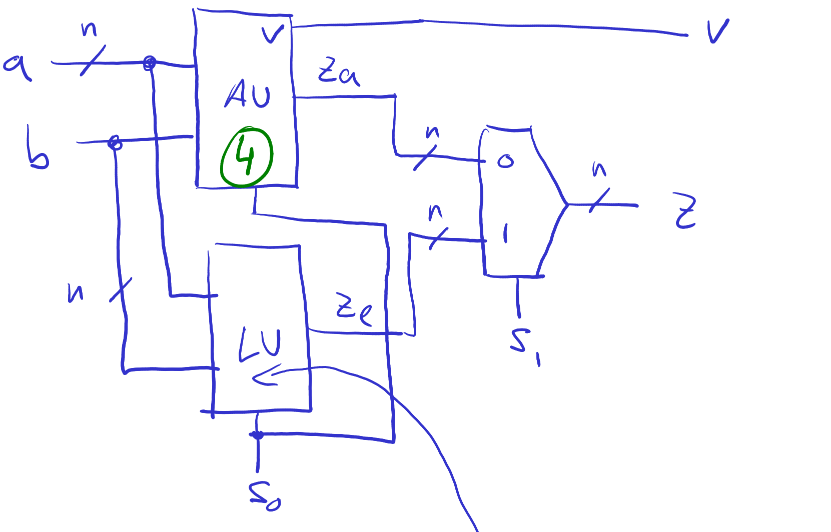
1) Partitioning → arithmetic unit
→ logic unit

2) Logic unit

3) Arithmetic unit

1) $s_1 = 0 \Rightarrow$ aritmetic

$s_1 = 1 \Rightarrow$ logic



2)

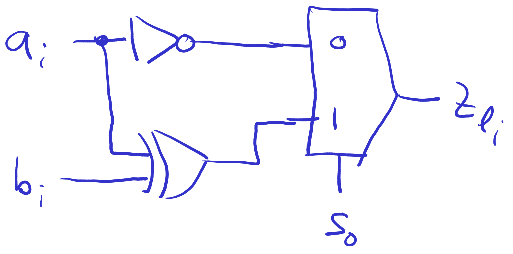
S_0	z_ℓ	z_{ℓ_i}
0	NOT a	$\overline{a_i}$
1	a XOR b	a_i XOR b_i

n

$i = 0 \ldots n-1$

typical stage

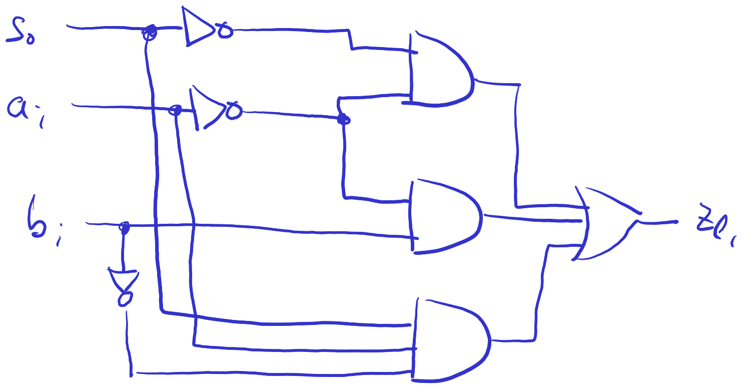
I) MUX ×n



II) Logic gates

S_0 \ a_ib_i	00	01	11	10
0	1	1	0	0
1	0	1	0	1

$$z_{\ell_i} = \overline{s_0}\,\overline{a_i} + \overline{a_i}\,b_i + s_0 a_i \overline{b_i}$$



S_0	Z_a	Y	y_i	y_0	y_1	~~C_{in}~~
0	$a+2b$	$2B$	b_{i-1}	0^*	b_0	~~ϕ~~
1	$a-4$	$TCR(-4)$	1	0^*	0^*	~~ϕ~~

Notation

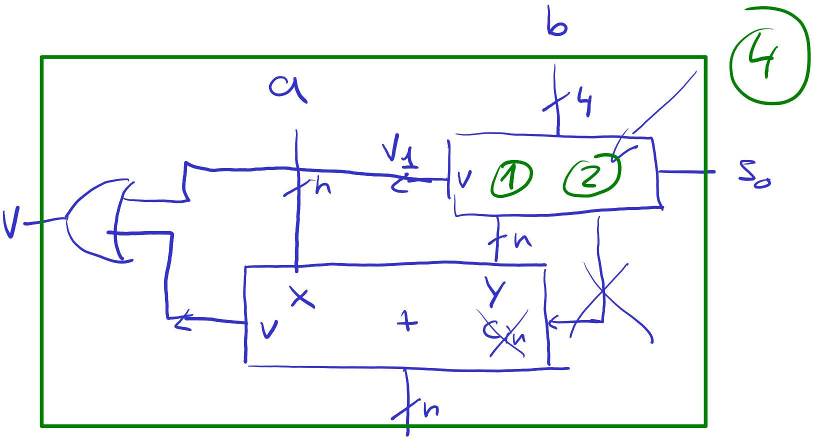
$a \longleftrightarrow A$

value repres.

$X = A$

$a-4 = a + (-4)$

4bits. -3 $1101 = 13$



$Z_a = A + Y + C_{in}$

$y = 2b$

$Y = 2B$

$B = b_{n-1} b_{n-2} \cdots b_1 b_0$

$Y = 2B = b_{n-1} b_{n-2} \cdots b_0 \phi$

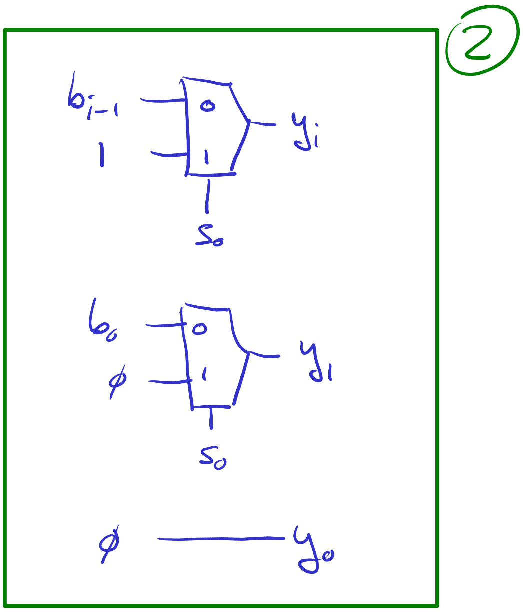
$y_{n-1} \cdots y_1 y_0$

$a-4 \rightarrow A + (-4)_{TC}$ $4 = 0\ldots0100 \rightarrow TCR(-4) = 1\ldots11100$

$\rightarrow$ ~~$\underbrace{A + (-5)}_{Y} + \underbrace{1}_{Cin}$~~

I) MUX

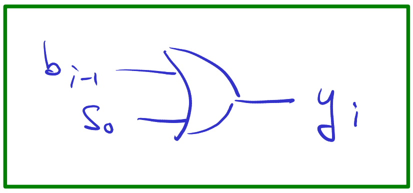
Typ.



II) Logic gates

$S_0 \backslash b_{i-1}$	0	1
0	0	1
1	1	1

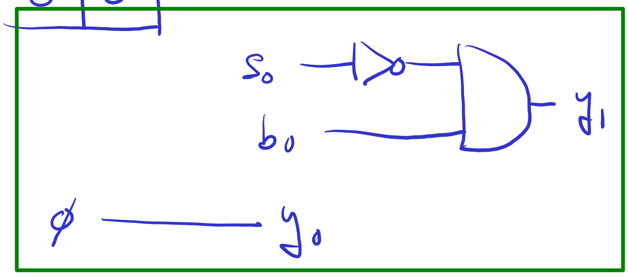
$y_i = b_{i-1} + S_0$



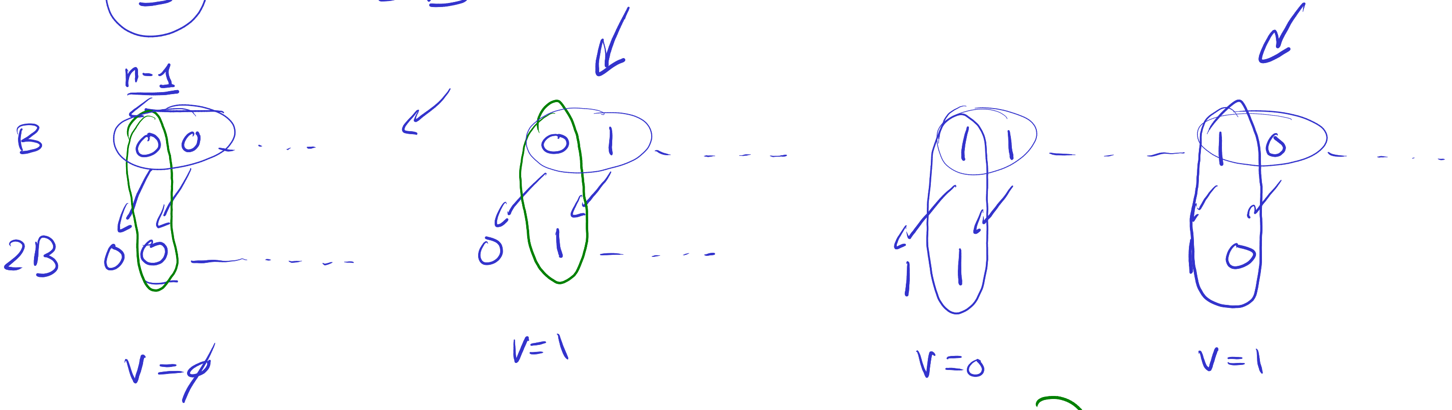
$S_0 \backslash b_0$	0	1
0	0	1
1	0	0

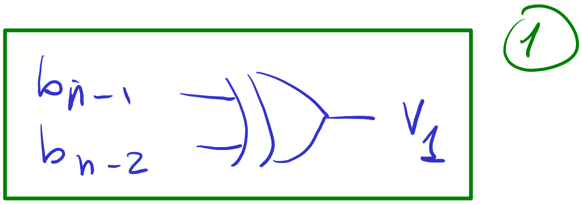
$y_1 = \bar{S_0} b_0$

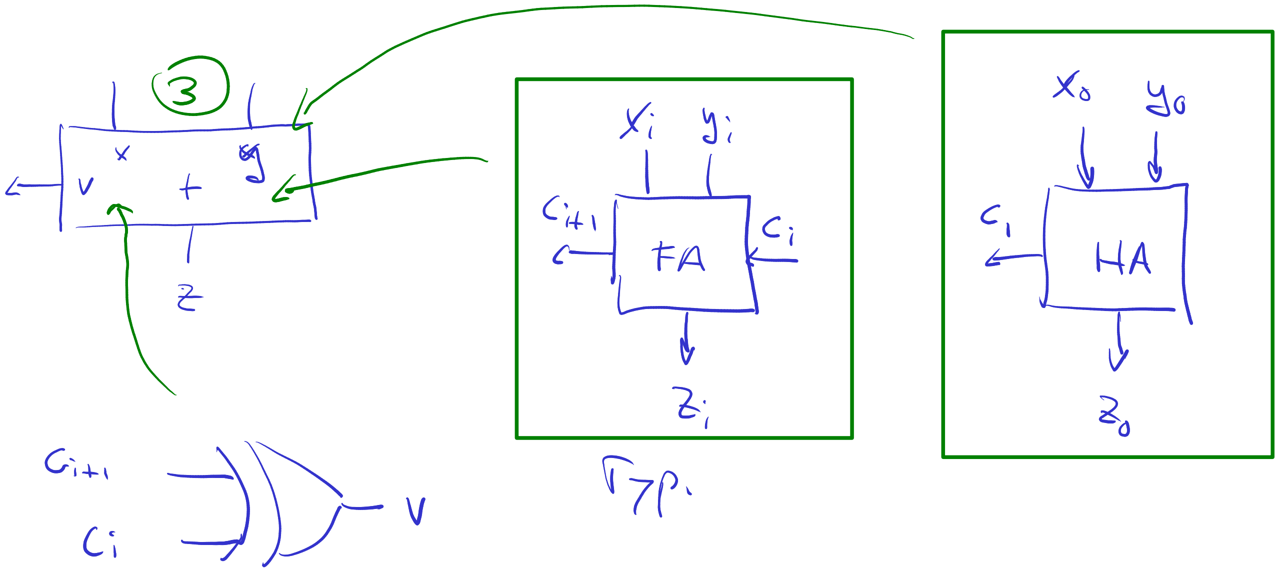
2



B − 2B


n-1
B
0 0
2B
0 0
V=ϕ
0 1
0 1
V=1
1 1
1 1
V=0
1 0
1 0
V=1


1
b_{n-1}
b_{n-2}
V_1


3
x
y
V
+
z
x_i
y_i
C_{i+1}
C_i
FA
z_i
x_0
y_0
C_1
HA
z_0
C_{i+1}
C_i
V
Τ7Ρ.