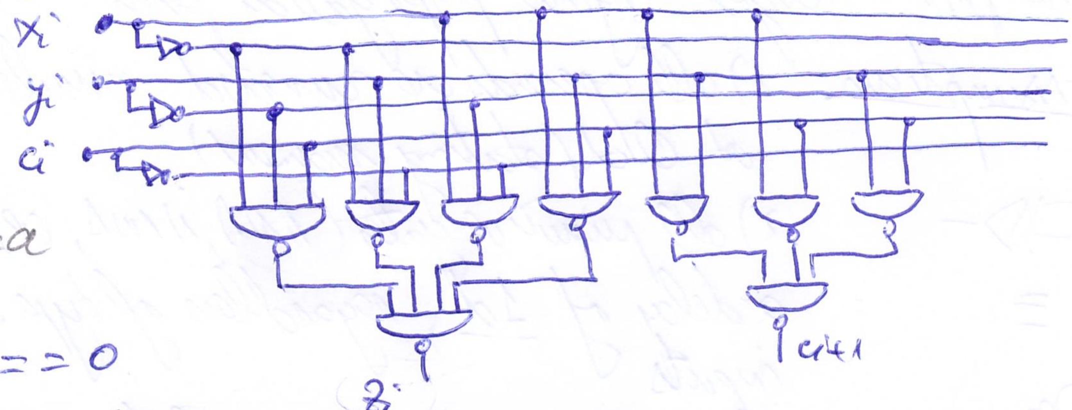
CA C5 (D) NAND gates

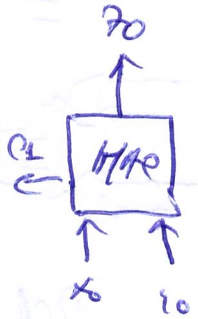


$a \oplus 0 = a$

c_0 can be 0 : $c_0 == 0$

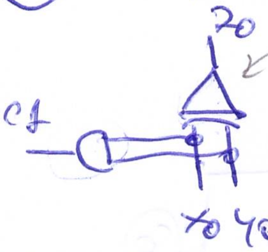
$$z_0 = x_0 \oplus y_0 \cancel{\oplus 0} = x_0 \oplus y_0$$
$$c_1 = x_0 y_0 + \cancel{x_0 \cdot 0} + \cancel{y_0 \cdot 0} = x_0 \cdot y_0$$

} HAC (Half Adder Cell)



HAC's synthesis

(A') EXOR - AND

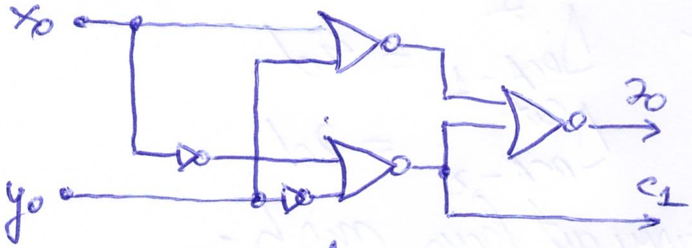


(B') NOR gates $a \oplus b = \bar{a} b + a \bar{b}$

$a \cdot \bar{a} = 0$

$$z_0 = x_0 \oplus y_0 = x_0 \bar{y}_0 + \bar{x}_0 y_0 =$$
$$= x_0(\bar{x}_0 + \bar{y}_0) + y_0(\bar{x}_0 + \bar{y}_0) =$$
$$= \overline{\overline{(x_0 + y_0)(\bar{x}_0 + \bar{y}_0)}} = \overline{\overline{x_0 + y_0} + \overline{\bar{x}_0 + \bar{y}_0}}$$

$$c_1 = \overline{\overline{x_0 \cdot y_0}} = \overline{\bar{x}_0 + \bar{y}_0}$$

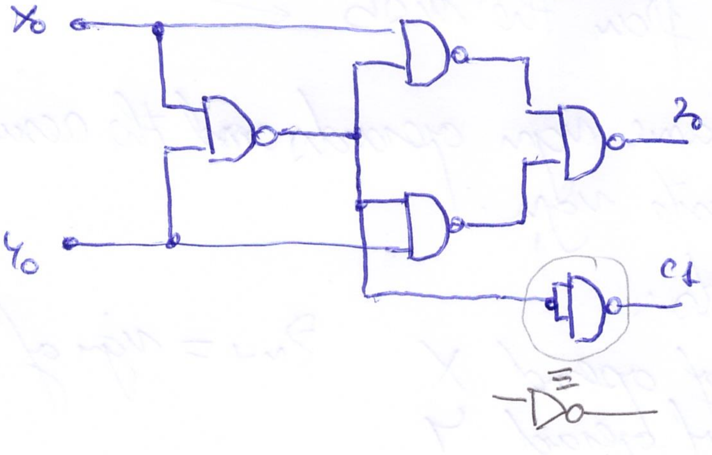


$\bar{x}_0 + \bar{y}_0 = \overline{x_0 \cdot y_0}$

(C) NAND gates

$$z_0 = \overline{\overline{x_0(\bar{x}_0 + \bar{y}_0) + y_0(\bar{x}_0 + \bar{y}_0)}} =$$
$$= \overline{\overline{x_0 \cdot \overline{x_0 \cdot y_0}} \cdot \overline{y_0 \cdot \overline{x_0 \cdot y_0}}}$$

$$c_1 = \overline{\overline{x_0 \cdot y_0}}$$



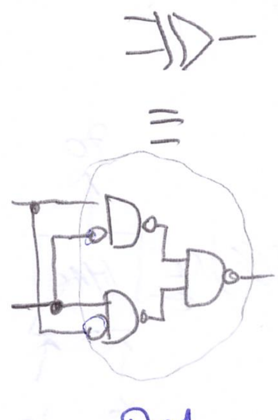
Critical path = longest signal propagation path

<u>Assumptions</u>: 1) all operands are connected simultaneously at 0d (starting moment)

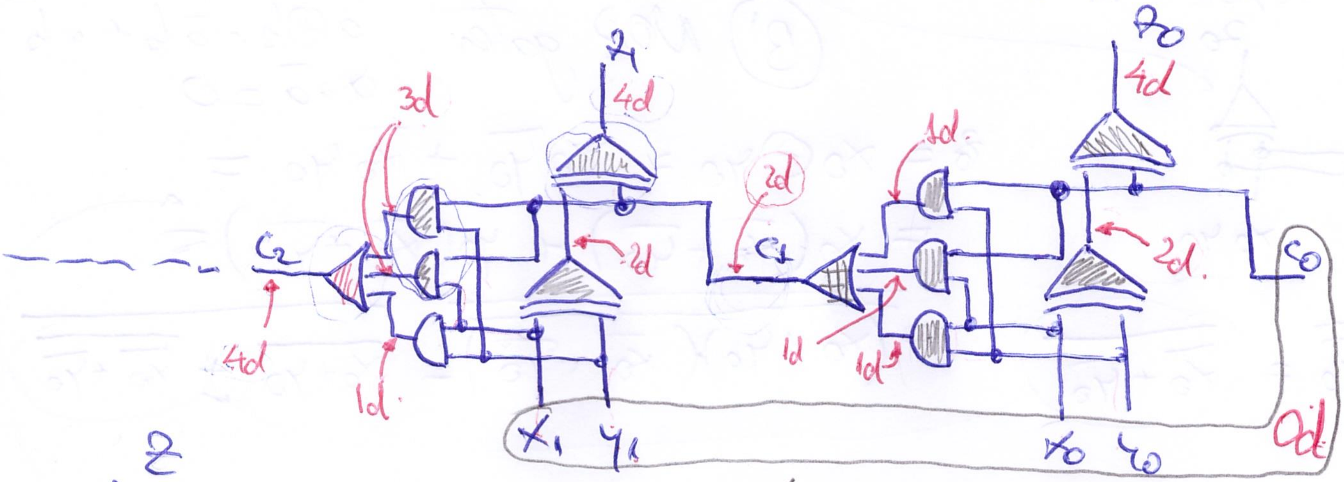
2) all primitive gates (AND, NAND, OR, NOR) have a delay of <u>1d</u>, regardless of type and number of inputs

3) EXOR gates have a latency of <u>2d</u>

4) inverters are delay-free



RCA on n bits



$\Delta^{z}_{RCA-n} = 2nd$

$\Delta^{Cout}_{RCA-n} = 2nd$

exception

$\Delta^{z}_{RCA-1} = 4d.$

$\Delta^{Cout}_{RCA-2} = 2d.$

Addition's special conditions:
- carry out from msb
- zero result
- negative result
- overflow

notation
V = overflow

V:
- unsigned numbers : carry out from the msb
- signed numbers : adding same sign operands and the result has opposite sign!

Consider the sign bits:

x_{n-1} – sign of operand X

y_{n-1} – sign of operand Y

z_{n-1} = sign of result

Inputs			Outputs	
X_{n-1}	Y_{n-1}	C_{n-1}	Z_{n-1}	V
0	0	0	0	0
0+	0+	1	1-	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1-	1-	0	0+	1
1	1	1	1	0

Boolean identities

$$I_1: (A \oplus B) \cdot C = AC \oplus BC$$

$$I_2: A + B = A \oplus B \oplus AB$$

$$I_2': A \oplus B = (A+B) \oplus AB$$

$$\underbrace{\overline{C_{n-1}} = C_{n-1} \oplus 1}$$

$$\cancel{a \oplus a} = 0$$

$$V = \overline{X_{n-1}}\,\overline{Y_{n-1}}\,C_{n-1} + X_{n-1}Y_{n-1}\overline{C_{n-1}} \overset{I_2}{=}$$

$$\overline{X_{n-1}}\,\overline{Y_{n-1}}\,C_{n-1} \oplus X_{n-1}Y_{n-1}\overline{C_{n-1}} \oplus \underbrace{\cancel{\overline{X_{n-1}}\,\overline{Y_{n-1}}\,C_{n-1}X_{n-1}Y_{n-1}\overline{C_{n-1}}}}_{0} =$$

$$\overline{X_{n-1}}\,\overline{Y_{n-1}}\,C_{n-1} \oplus X_{n-1}Y_{n-1}\overline{C_{n-1}} \overset{I_1}{=} \overline{X_{n-1}}\,\overline{Y_{n-1}}\,C_{n-1} \oplus X_{n-1}Y_{n-1}C_{n-1} \oplus X_{n-1}Y_{n-1} \cdot 1 \overset{I_1}{=}$$

$$(\overline{X_{n-1}}\,\overline{Y_{n-1}} \oplus X_{n-1}Y_{n-1})C_{n-1} \oplus X_{n-1}Y_{n-1} \overset{I_2'}{=}$$

$$(\underbrace{\overline{X_{n-1}}\,\overline{Y_{n-1}} + X_{n-1}Y_{n-1}})C_{n-1} \oplus X_{n-1}Y_{n-1} \overset{I}{=}$$

$$\overline{X_{n-1} \oplus Y_{n-1}} = X_{n-1} \oplus Y_{n-1} \oplus 1$$

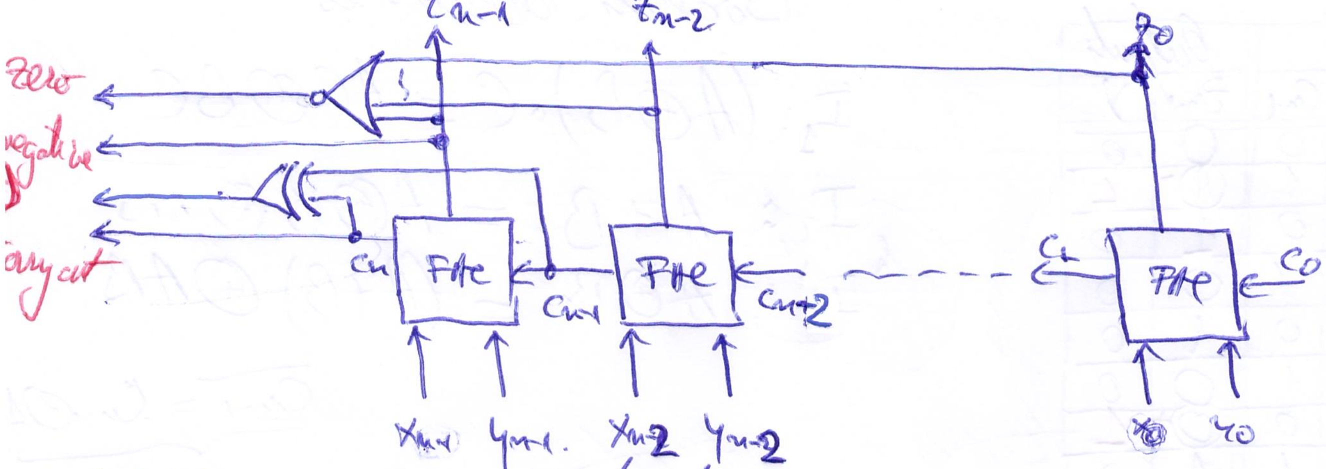
$$(X_{n-1} \oplus Y_{n-1} \oplus 1)\,C_{n-1} \oplus X_{n-1}Y_{n-1} \overset{I_1}{=}$$

$$\underbrace{X_{n-1}C_{n-1} \oplus Y_{n-1}C_{n-1} \oplus X_{n-1}Y_{n-1}} \oplus C_{n-1} \overset{I_2'}{=}$$

$$(\underbrace{X_{n-1}C_{n-1} + Y_{n-1}C_{n-1} + X_{n-1}Y_{n-1}}_{C_n}) \oplus C_{n-1} \Rightarrow$$

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

Adder with special condition generator.
- operands are n bits.



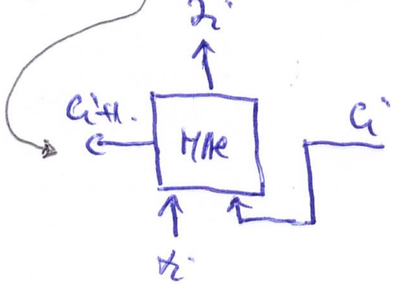
Addition with a constant.
- odd constants
- Y operand is the constant

$X = x_3\, x_2\, x_1\, x_0$
$Y = 1\ 1\ 0\ 0$ (const)
$Z = z_3\ \overline{x_2}\ x_1\ x_0$

Let X, Y – on n bits.

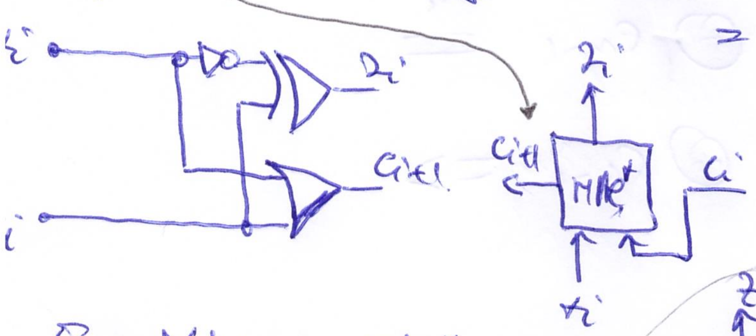
$Y = y_{n-1}\, y_{n-2} \ldots y_1\, y_0$ – constant.

if $y_i == 0$: $x_i + \underbrace{y_i}_{=0} + c_i$ $\begin{cases} z_i = x_i \oplus \cancel{0 \oplus} c_i = x_i \oplus c_i \\ c_{i+1} = \cancel{x_i \cdot 0} + \cancel{0 \cdot c_i} + x_i c_i = x_i \cdot c_i \end{cases}$ MAC



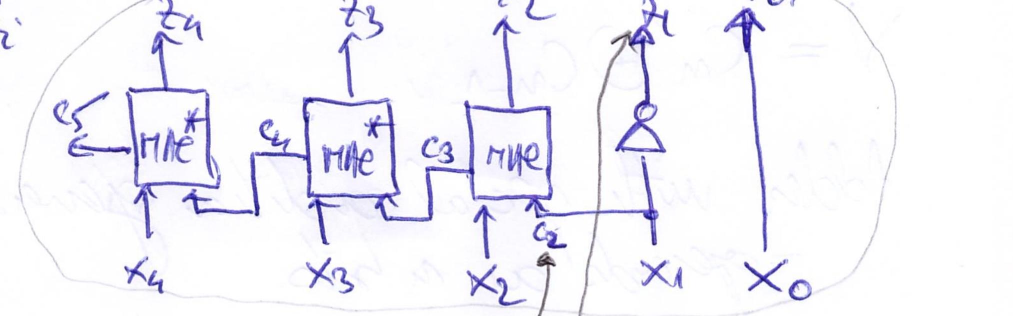
if $y_i == 1$: $x_i + \underbrace{y_i}_{=1} + c_i$ $\begin{cases} z_i = \overbrace{x_i \oplus 1}^{\overline{x_i}} \oplus c_i = \overline{x_i} \oplus c_i \\ c_{i+1} = x_i \cdot 1 + 1 \cdot c_i + x_i c_i = \\ = x_i + c_i + x_i c_i = x_i + c_i \end{cases}$ MAC*

absorbed.



Ex: X, Y – on 5 bits.

$X =$	$x_4\, x_3\, x_2\, x_1$	x_0
$Y =$	$1\ 1\ 0\ 1$	0
	odd const.	
$Z =$	$z_4\, z_3\, z_2\, z_1$	x_0



$z_1: x_1 + 1 + 0$ $\begin{cases} z_1 = x_1 \oplus 1 \oplus 0 = \overline{x_1} \\ c_2 = x_1 \cdot 1 + x_1 \cdot 0 + 1 \cdot 0 = x_1 \end{cases}$

2.2.2. Decimal Adders based on serial carry propagation

- use binary adders for performing BCD addition

① BCD8421 (BCD) addition.
- on tetrads (= sequence of 4 bits)

$X_i + Y_i \begin{cases} z_i \\ c_{i+1} \end{cases}$

X_i, Y_i, z_i – BCD digits.
c_{i+1} = carry for digit $i+1$.
z_i = sum digit.

$X_i = x_3x_2x_1x_0$
$Y_i = y_3y_2y_1y_0$
$z_i = z_3z_2z_1z_0$

if $X_i + Y_i < 10 \begin{cases} z_i = X_i + Y_i \\ c_{i+1} = 0 \end{cases}$

if $X_i + Y_i \geq 10 \begin{cases} z_i = X_i + Y_i - 10 \\ c_{i+1} = 1 \end{cases}$

* z_i needs some correction

carry to next digit

$$\begin{array}{r} 7+ \\ 8 \\ \hline 15 \\ 10 \\ \hline 5 \end{array}$$

← unit's figure

– sum digit

$X_i + Y_i$ = add 2 4-bit binary numbers
⇒ 5 bit result

$X_i + Y_i = c^* z_3^* z_2^* z_1^* z_0^*$

$X_i + Y_i \geq 10 \begin{cases} 10 \leq c^* z_3^* z_2^* z_1^* z_0^* < 16 \; (C_1) \\ c^* z_3^* z_2^* z_1^* z_0^* \geq 16 \; (C_2) \end{cases}$ → logic OR

$2^4 = 16$

1: $10 \leq c^* z_3^* z_2^* z_1^* z_0^* < 16 \begin{cases} c^* = 0 \\ z_3^* z_2^* z_1^* z_0^* \in [10; 15] \end{cases}$ → AND

$z_3^*z_2^* \backslash z_1^*z_0^*$	00	01	11	10
00				
01				
11	1	1	1	1
10			1	1

$10 \leq z_3^* z_2^* z_1^* z_0^* \leq 15$

$\equiv z_3^* \cdot z_2^* + z_3^* \cdot z_1^*$

$(C_1): \overline{c^*} (z_3^* \cdot z_2^* + z_3^* \cdot z_1^*)$

c_2: $c^* z_3^* z_2^* z_1^* z_0^* \geq 16 \Rightarrow \underbrace{c^* = 1}$

absorbed.

$$X_i + Y_i \geq 10 \equiv (C1) \underset{OR}{+} (C2) = \overline{c^*}(z_3^* z_2^* + z_3^* z_1^*) + c^*$$

$$X_i + Y_i \geq 10 \equiv \boxed{c^* + z_3^* z_2^* + z_3^* z_1^*}$$

Subtracting 10 from $X_i + Y_i$

if $X_i + Y_i \geq 10$ $\quad z_i = X_i + Y_i - 10$

z_i = on 4 bits.

$$\boxed{a \bmod b = (a+b) \bmod b.}$$

$$\Rightarrow z_i = (X_i + Y_i - 10) \bmod 2^4$$

$$= (X_i + Y_i - 10 + 16) \bmod 2^4.$$

$$= (X_i + Y_i + 6) \bmod 2^4.$$

$$\begin{array}{r} 15 - \\ 10 \end{array} = \begin{array}{r} 1111 - \\ 1010 \\ \hline \underline{0101} \\ 5 \end{array} \qquad \begin{array}{r} 1111 + \\ 0110 \\ \hline \cancel{1}\underbrace{0101}_{5} \end{array} \qquad \begin{array}{r} 15 + \\ 6 \end{array}$$

Subtract 10 on 4 bits $\equiv$ Add 6 on 4 bits

$$\underset{(X_i + Y_i \geq 10)}{1} \begin{cases} z_i = \begin{array}{r} z_3 z_2 z_1 z_0 + \\ 0110 \end{array} \text{carry} \\ c_{i+1} = \underline{1} \end{cases}$$

z_i's calculation

depends on: $c^* + z_3^* \cdot z_2^* + z_3^* \cdot z_1^*$