Subsystems Design

- Adder
- Counters
- Multipliers
- Memories
- PLAs
- Finite state machines

Adders

- Combinational adder
- Dynamic combinational adder
- Transmission gate adder
- Carry lookahead adder
- Manchester carry adder
- Binary lookahead adder
- Carry select adder

Combinational Adder

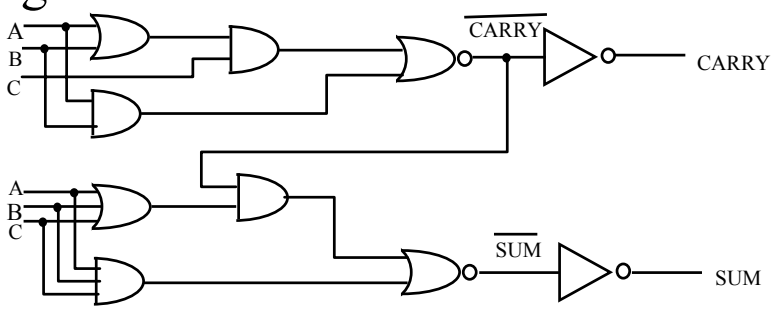
- Adder truth table

C	A	B	A•B(G)	A+B(P)	A⊕B	SUM	CARRY
0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	0
0	1	0	0	1	1	1	0
0	1	1	1	1	0	0	1
1	0	0	0	0	0	1	0
1	0	1	0	1	1	0	1
1	1	0	0	1	1	0	1
1	1	1	1	1	0	1	1

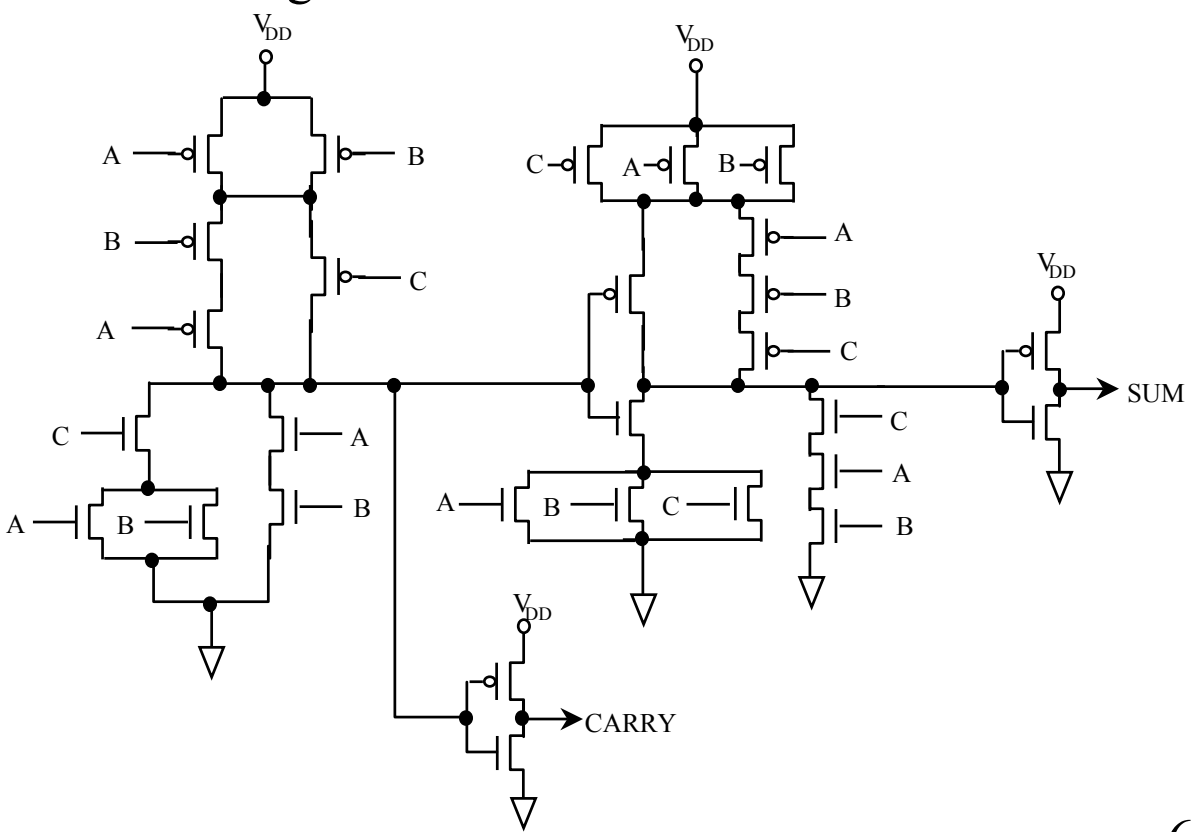
$$SUM=ABC+\overline{ABC}+\overline{ABC}+\bar{A}\bar{B}\bar{C}$$

$$CARRY=AB+AC+BC=AB+C(A+B)=G+CP$$

- Logic gate

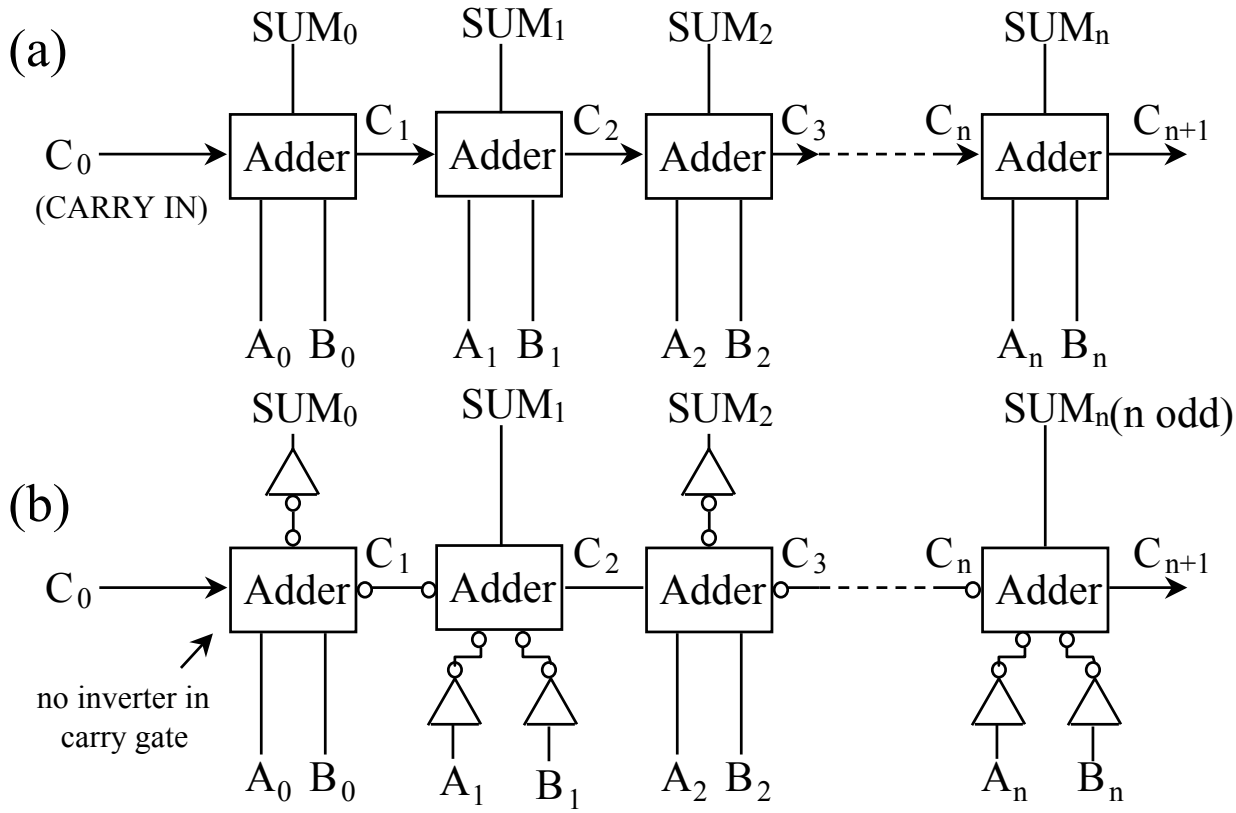


- Circuit Diagram



N-Bit Combinational Adder

- Ripple carry adder



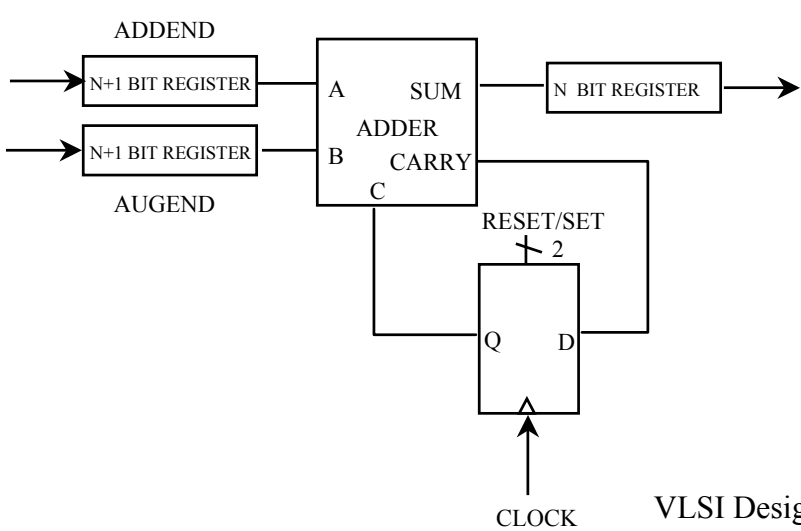
(a): long carry delay ($C_0 \longrightarrow C_{n+1}$)

(b): using inverter logic to shorten

carry delay ($C_0 \longrightarrow C_{n+1}$)

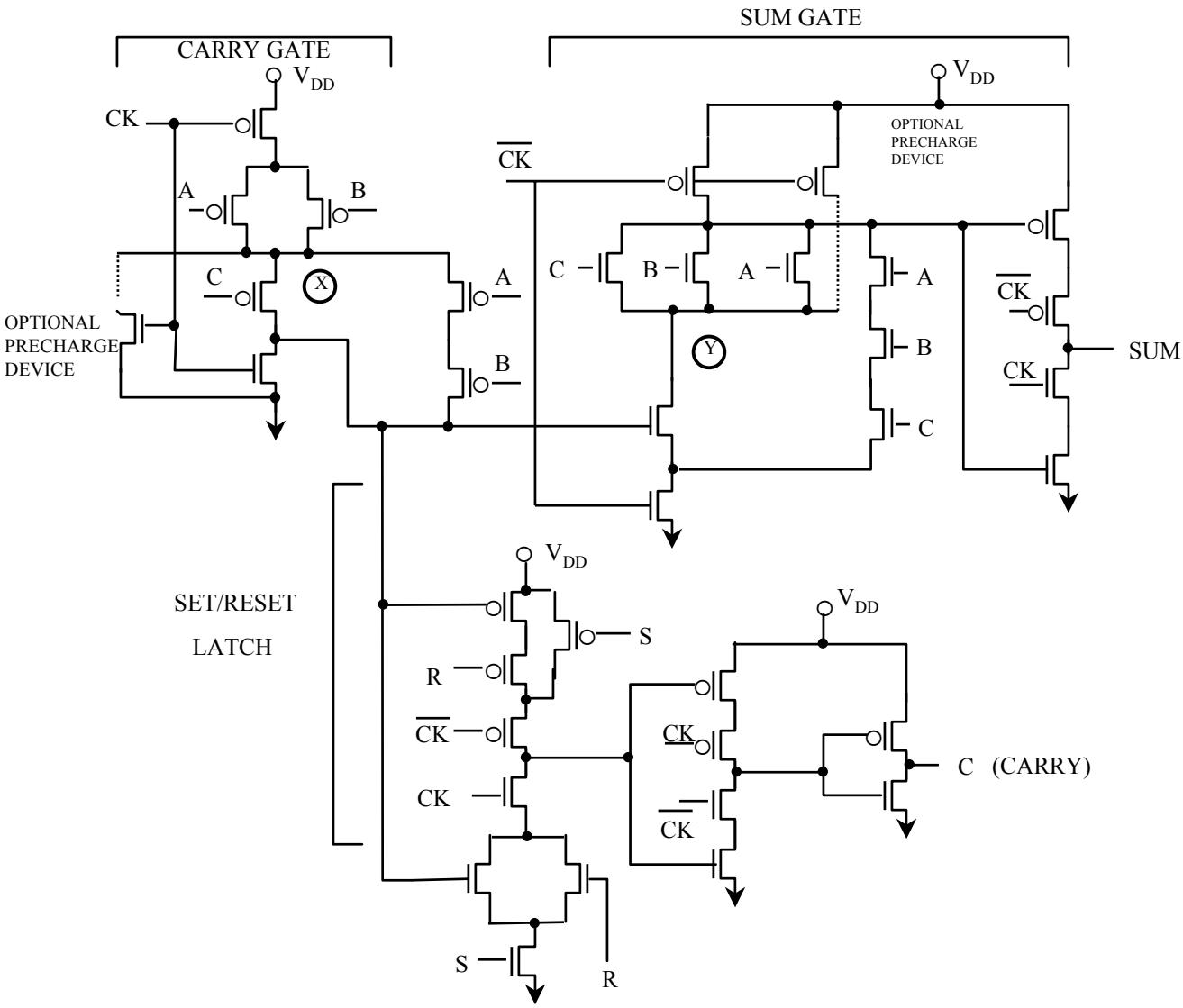
(The inverter at the carry output is omitted)

- Serial adder
  - low cost
  - low speed

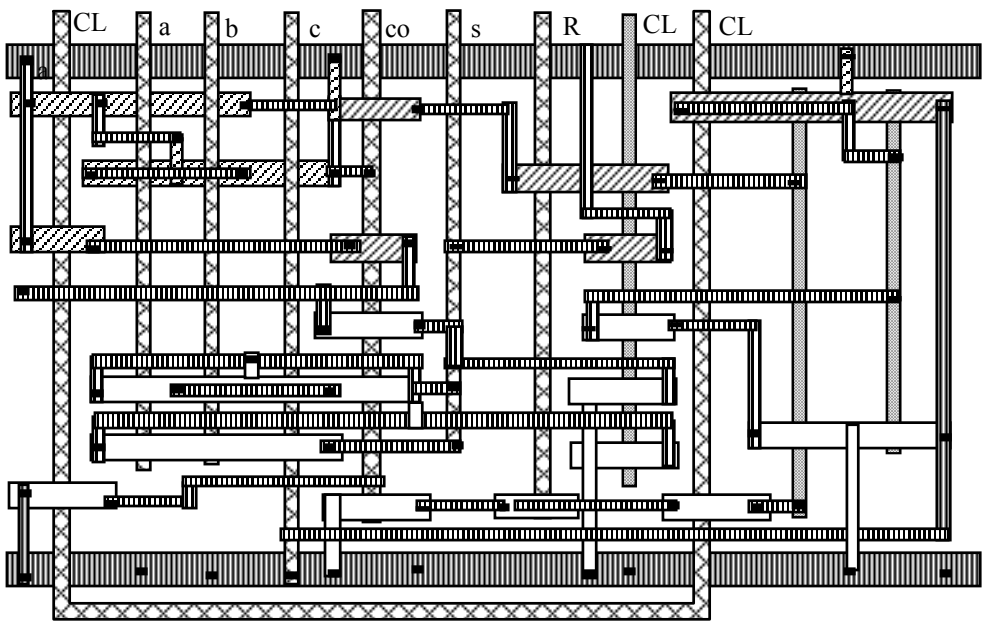


Dynamic Combinational Adder

- Gates are implemented using dynamic circuits

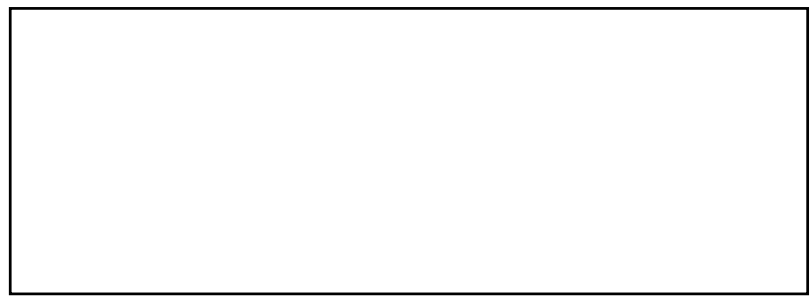


- Layout



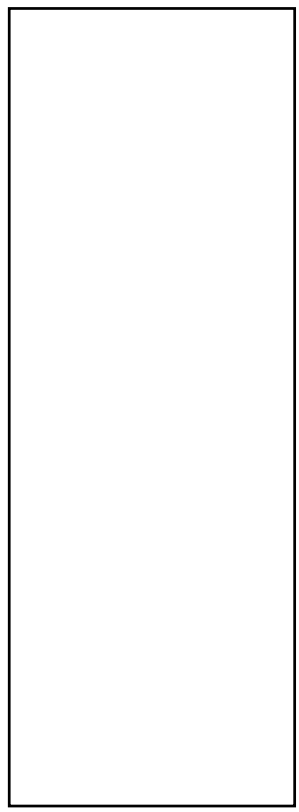
Combinational Adder Layout

- (Symbolic layout)：3 styles

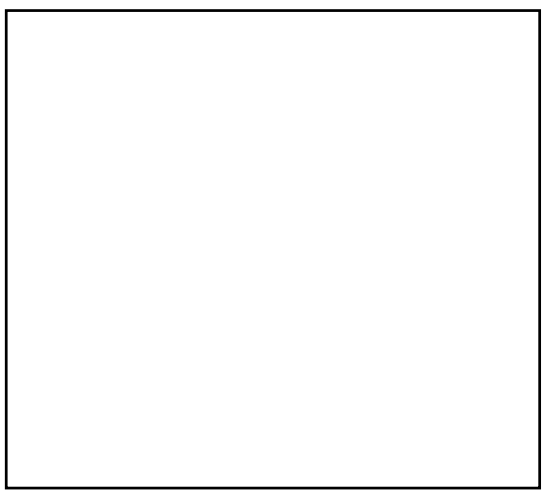


(a) minimum height

(b) minimum width

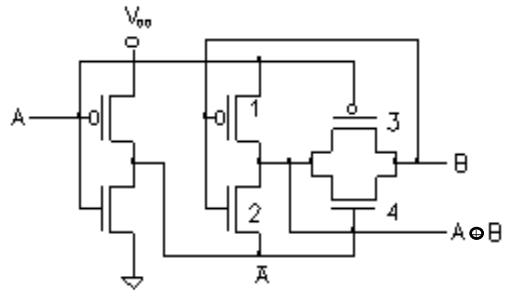


(c) Compromise between(a)&(b)

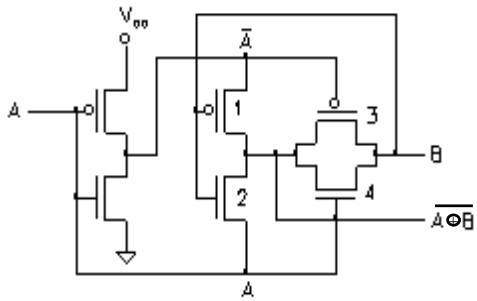


Transmission Gate (TG) Adder

- TG exclusive-or gate



- TG exclusive-nor gate

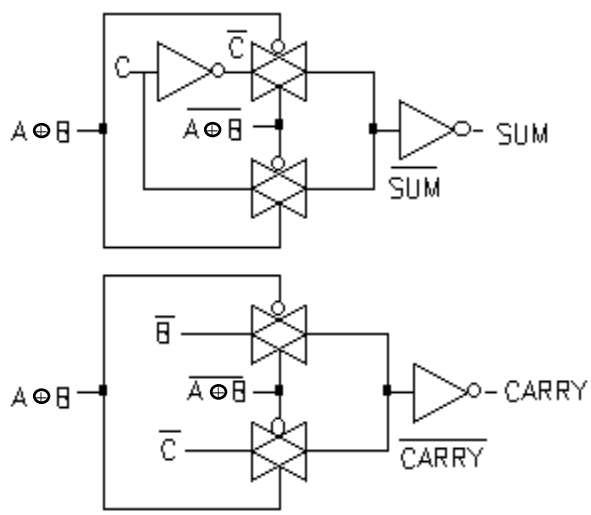


- TG Adder

$$SUM = ABC + A\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}\,\overline{B}C$$
$$= (AB + \overline{A}\,\overline{B})C + (A\overline{B} + \overline{A}B)C$$
$$= (\overline{A\oplus B})C + (A\oplus B)\overline{C}$$

$$\overline{SUM} = \overline{A}BC + A\overline{B}C + AB\overline{C} + \overline{A}\,\overline{B}\,\overline{C}$$
$$= (\overline{A}B + A\overline{B})C + (AB + \overline{A}\,\overline{B})\overline{C}$$
$$= (A\oplus B)C + (\overline{A\oplus B})\overline{C}$$

$$\overline{CARRY} = \overline{A}\,\overline{B}C + A\overline{B}C + \overline{A}B\overline{C} + \overline{A}\,\overline{B}\,\overline{C}$$
$$= \overline{A}\,\overline{B} + (A\oplus B)\overline{C}$$
$$= (\overline{A\oplus B})\overline{B} + (A\oplus B)\overline{C}$$



Carry Lookhead Adder

- Fast compared with ripple counter
- High cost

$$C_i = A_iB_i + (A_i + B_i)C_{i-1} = G_i + P_iC_{i-1}$$

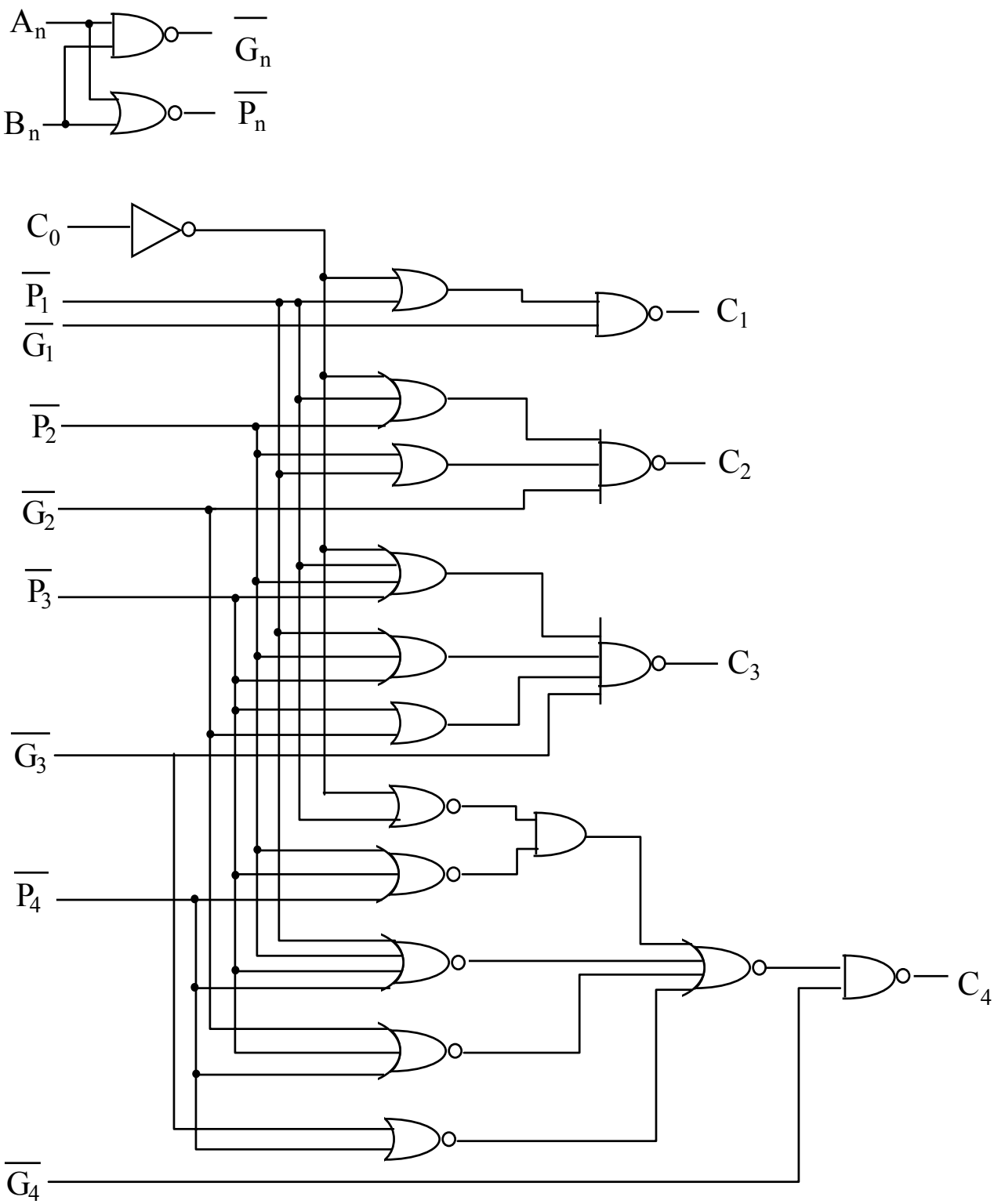
where $G_i = A_i \cdot B_i$ generate signal

$P_i = A_i + B_i$ propagate signal

$$\begin{aligned} C_i &= A_iB_i + (A_i + B_i)C_{i-1} \\ &= G_i + P_iC_{i-1} \\ &= G_i + P_i(C_{i-1} + P_{i-1}C_{i-2}) \\ &\vdots \\ &= G_i + P_iC_{i-1} + P_iP_{i-1}C_{i-2} + P_iP_{i-1}P_{i-2}C_{i-3} + \ldots \end{aligned}$$

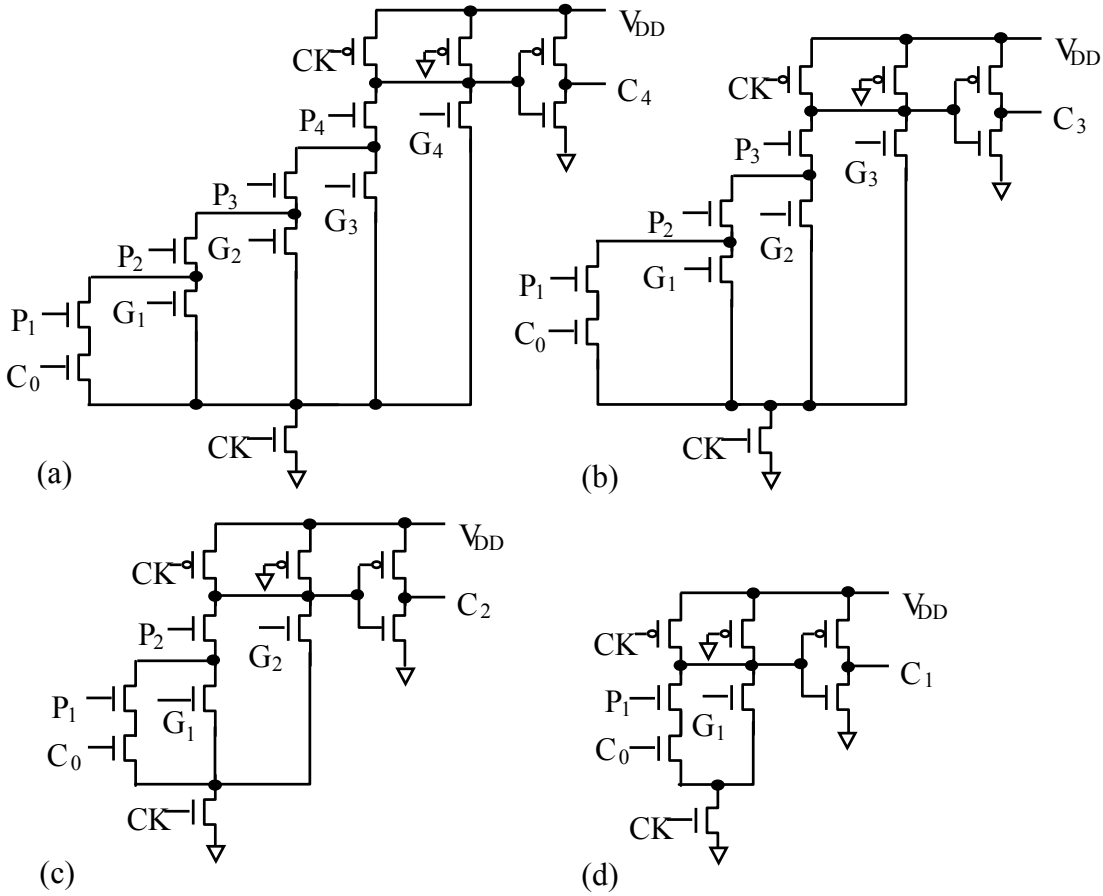
$$\Rightarrow \begin{cases} C_1 = G_1 + P_1C_0 \\ C_2 = G_2 + P_2G_2 + P_2P_1C_0 \\ C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1C_0 \\ C_4 = G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1 + P_4P_3P_2P_1C_0 \end{cases}$$

4-bit Full Carry Lookhead Adder

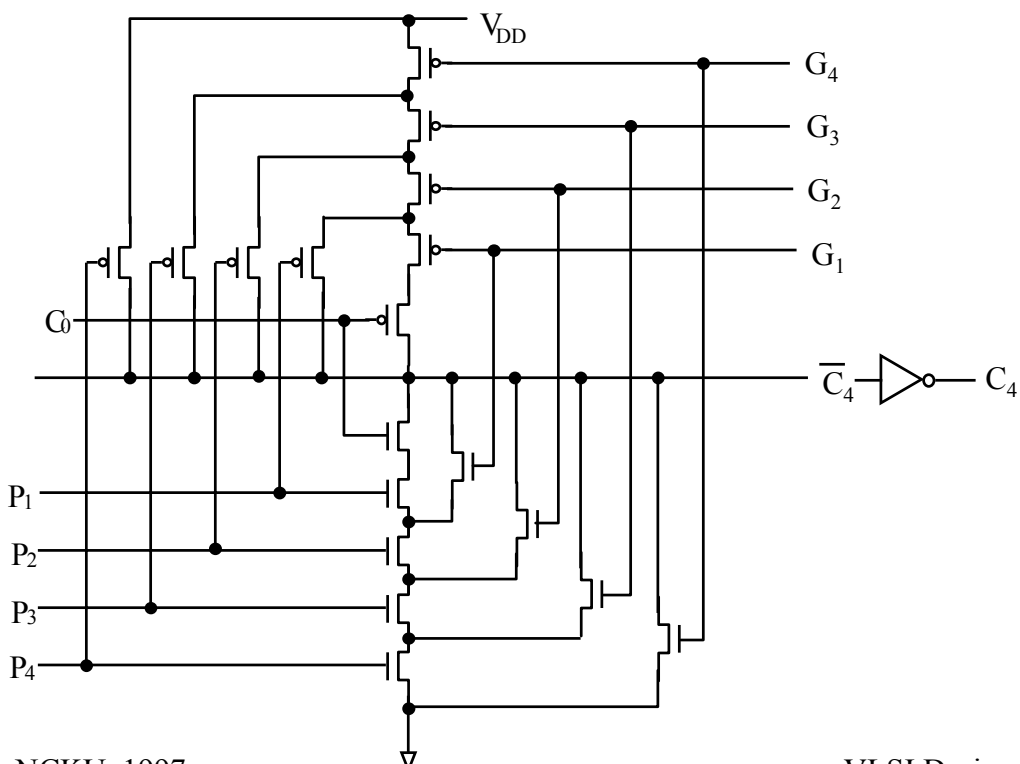


Carry Lookhead Adder

- Domino type

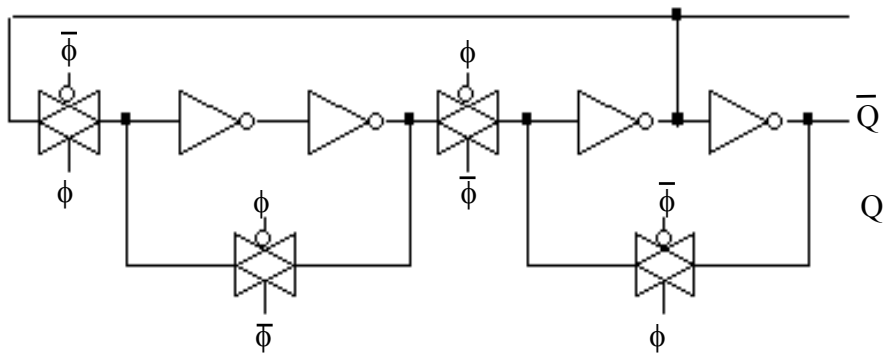


- Static type

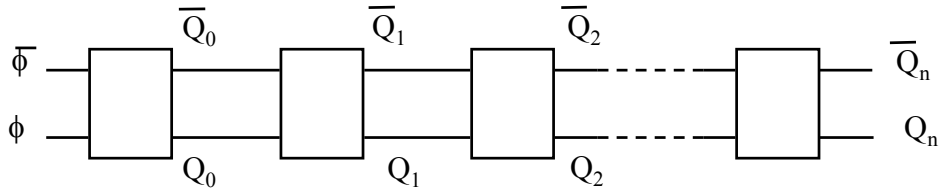


Counter

- Asynchronous counter
  - — Slow: clock is provided by the previous stage
    => long counter chain delay

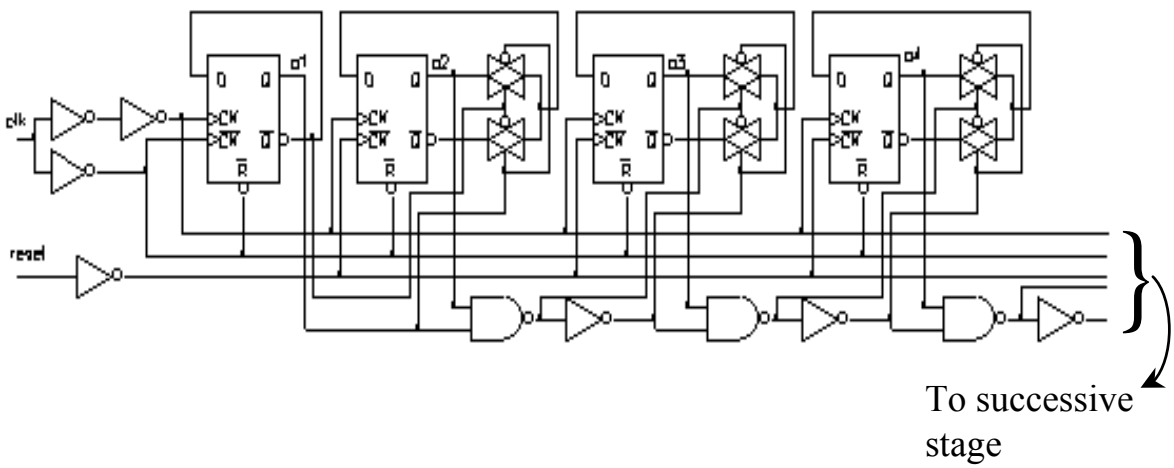


(a)



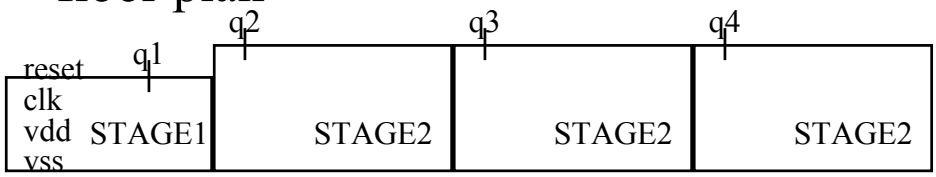
(b)

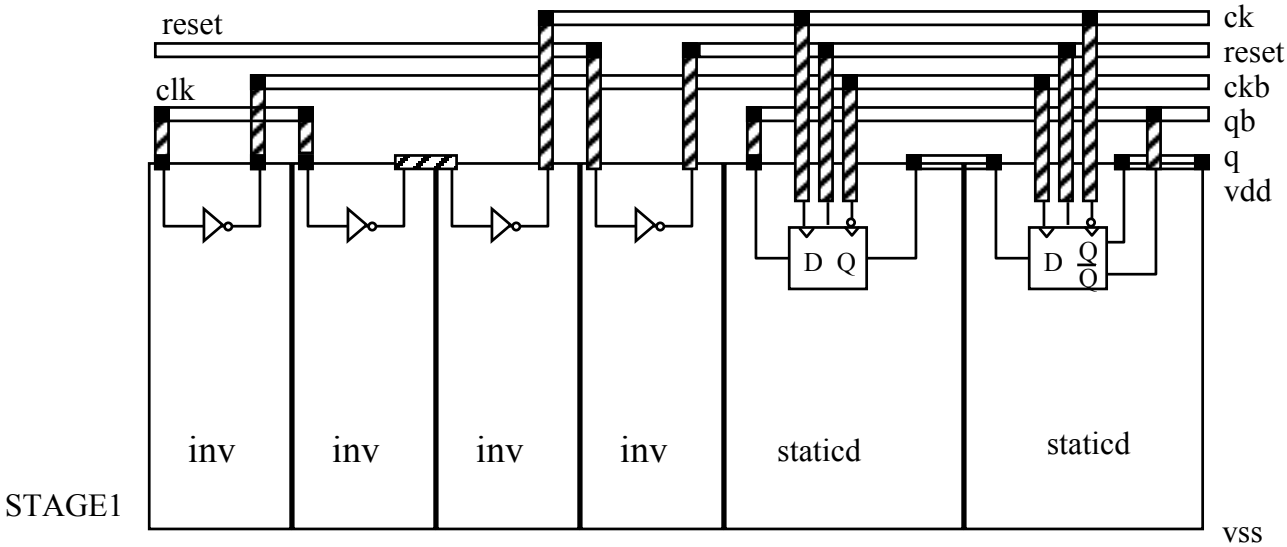
- synchronous counter
  - — Fast: Each stage is clocked simultaneously and the output change in a synchronous manner.

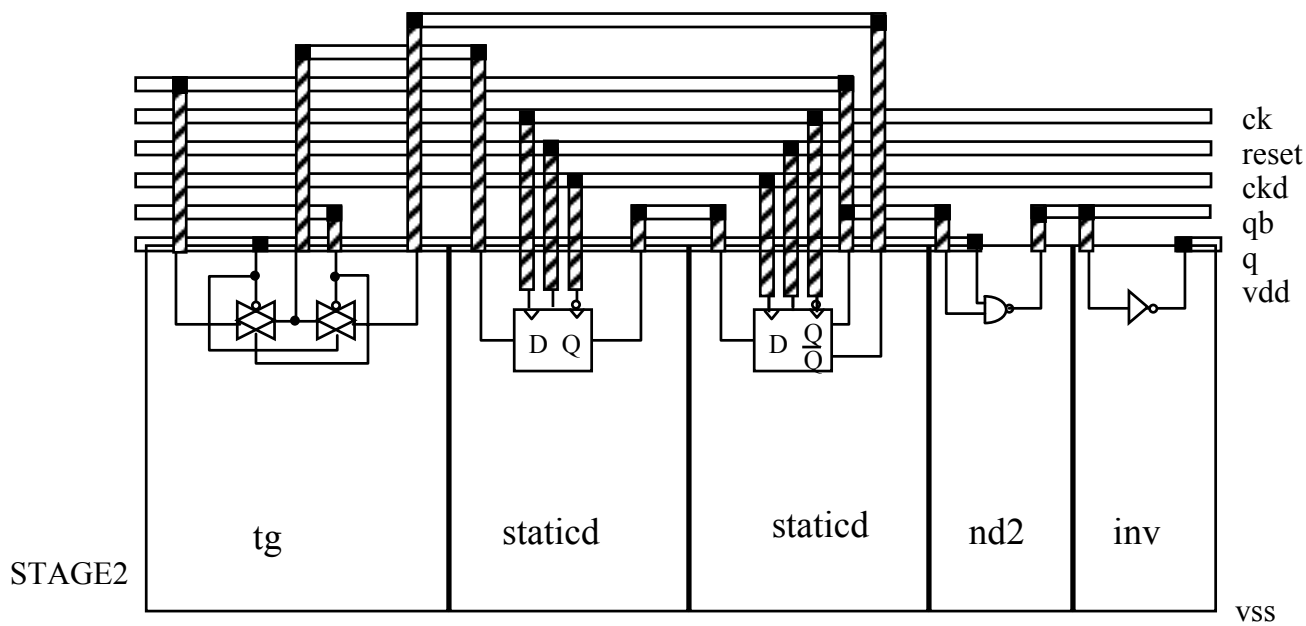


Floor plan using standard cells

- Synchronous counter
  - floor plan





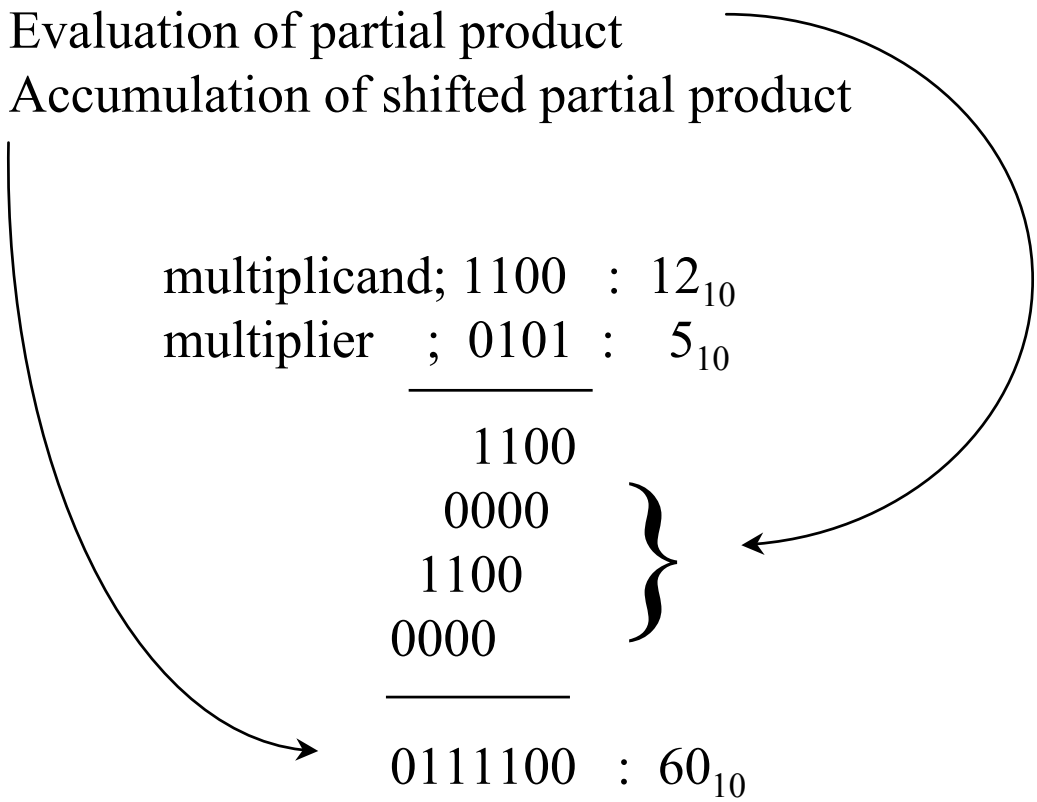


Multiplier

- Applications
  - signal processing
  - correlation
  - convolution
  - filtering
  - frequency analysis
  - CPU

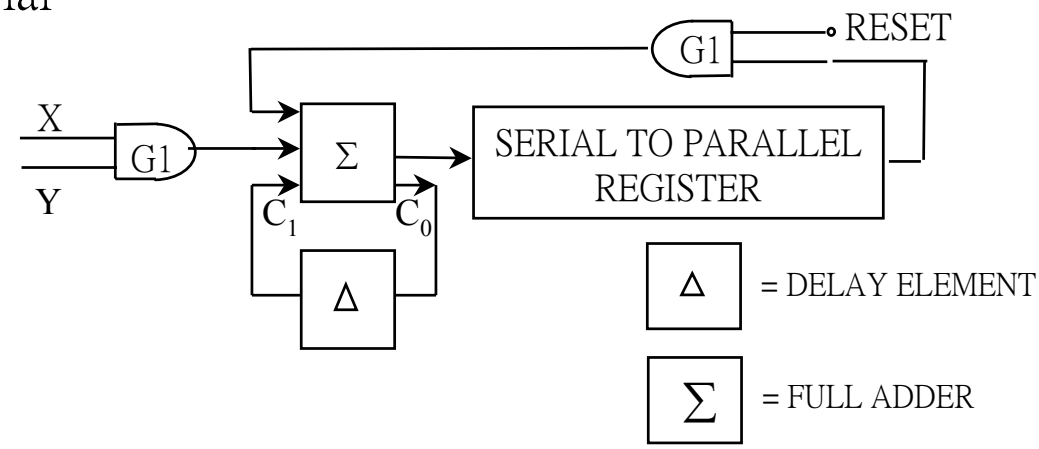
- Three types
  - Serial
  - Serial/Parallel
  - Parallel

- Operations
  - Evaluation of partial product
  - Accumulation of shifted partial product

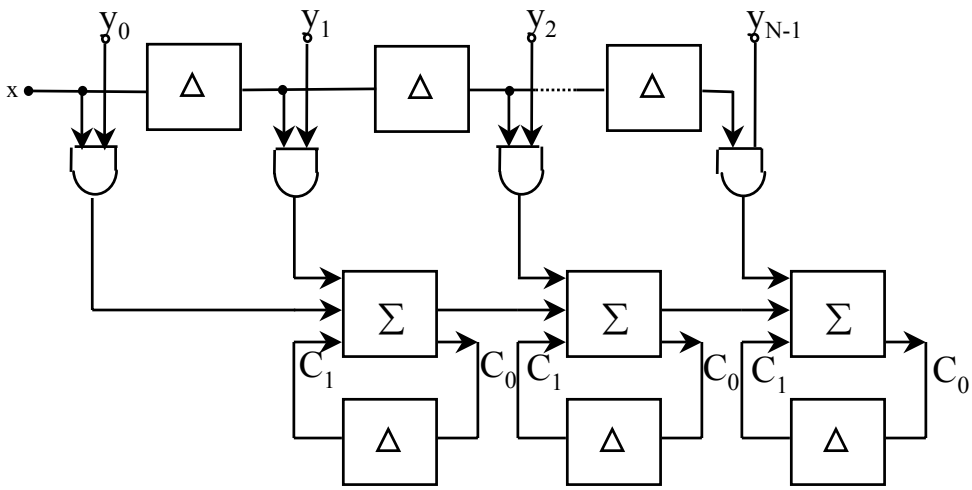


Multipliers

- Serial



- Serial / Parallel



$X = x_{M-1} \cdots x_2 x_1 x_0$

$Y = y_{M-1} \cdots y_2 y_1 y_0$

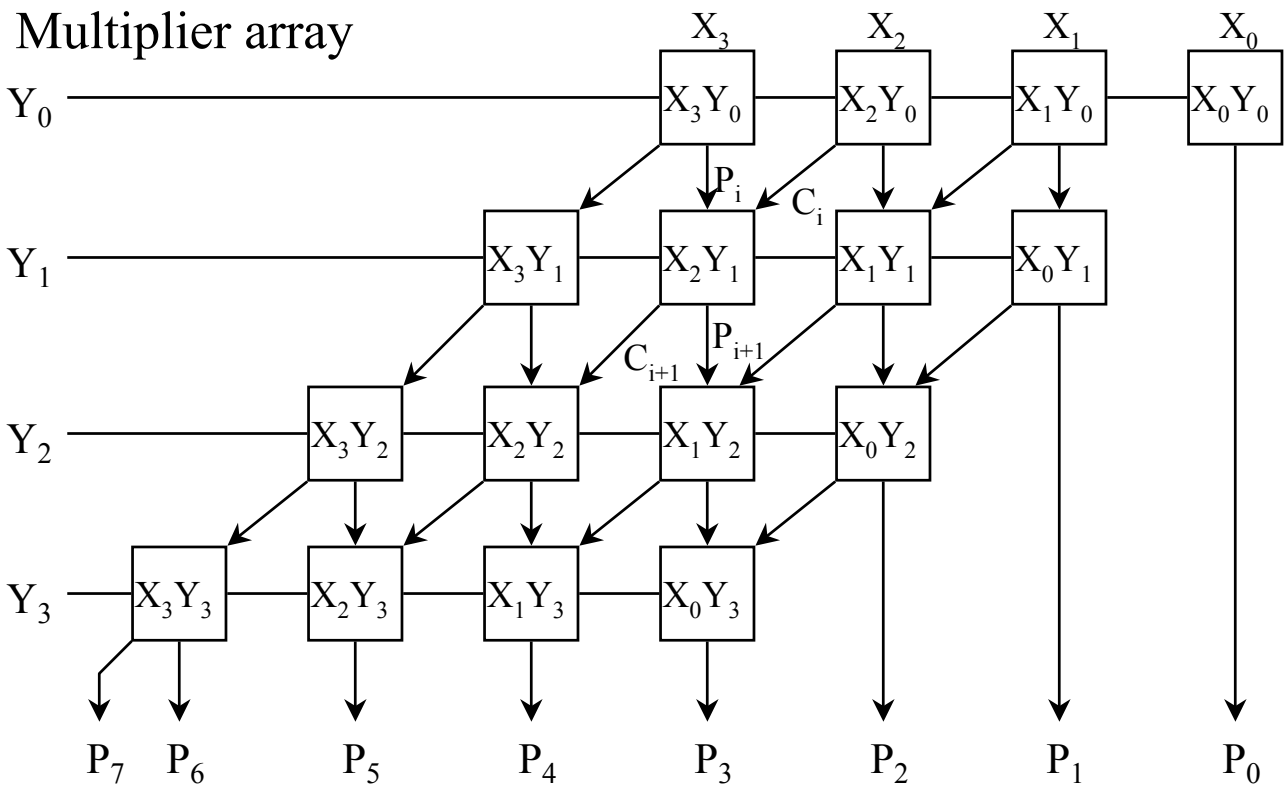
Σ = FULL ADDER Δ = DELAY ELEMENT

Parallel Multiplier

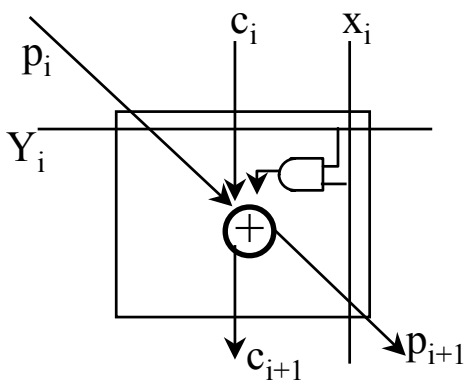
- TABLE 8.2 4-bit multiplier partial products

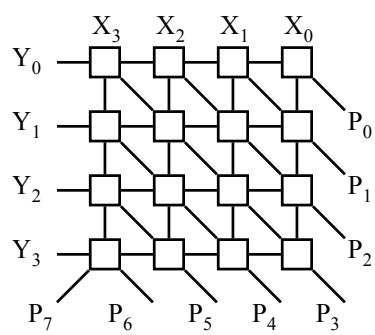
				X_3	X_2	X_1	X_0	Multiplicand
				Y_3	Y_2	Y_1	Y_0	Multiplier
				X_3Y_0	X_2Y_0	X_1Y_0	X_0Y_0	
			X_3Y_1	X_2Y_1	X_1Y_1	X_0Y_1		
		X_3Y_2	X_2Y_2	X_1Y_2	X_0Y_2			
	X_3Y_3	X_2Y_3	X_1Y_3	X_0Y_3				
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0	Product

- Multiplier array



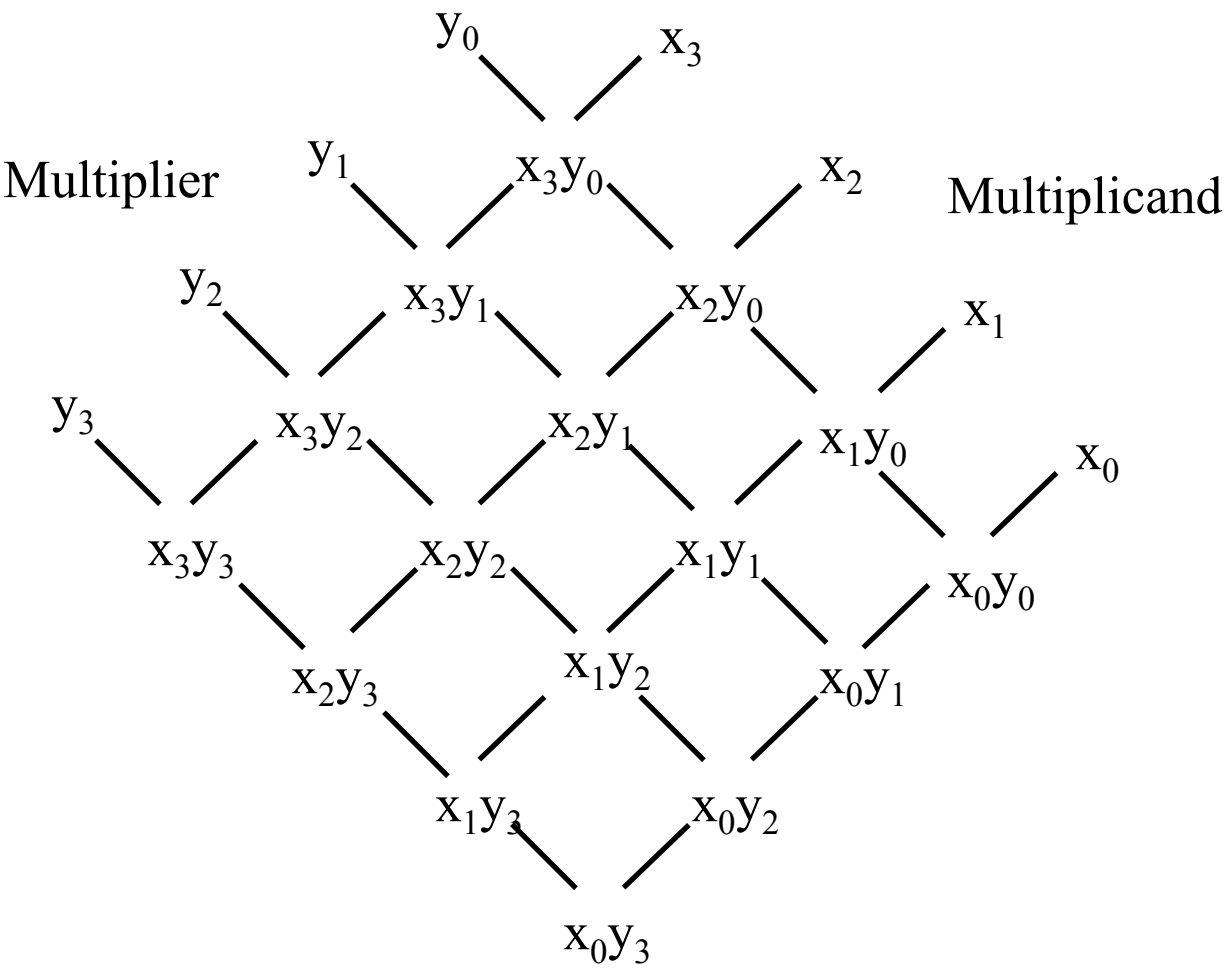
- Parallel multiplier cell



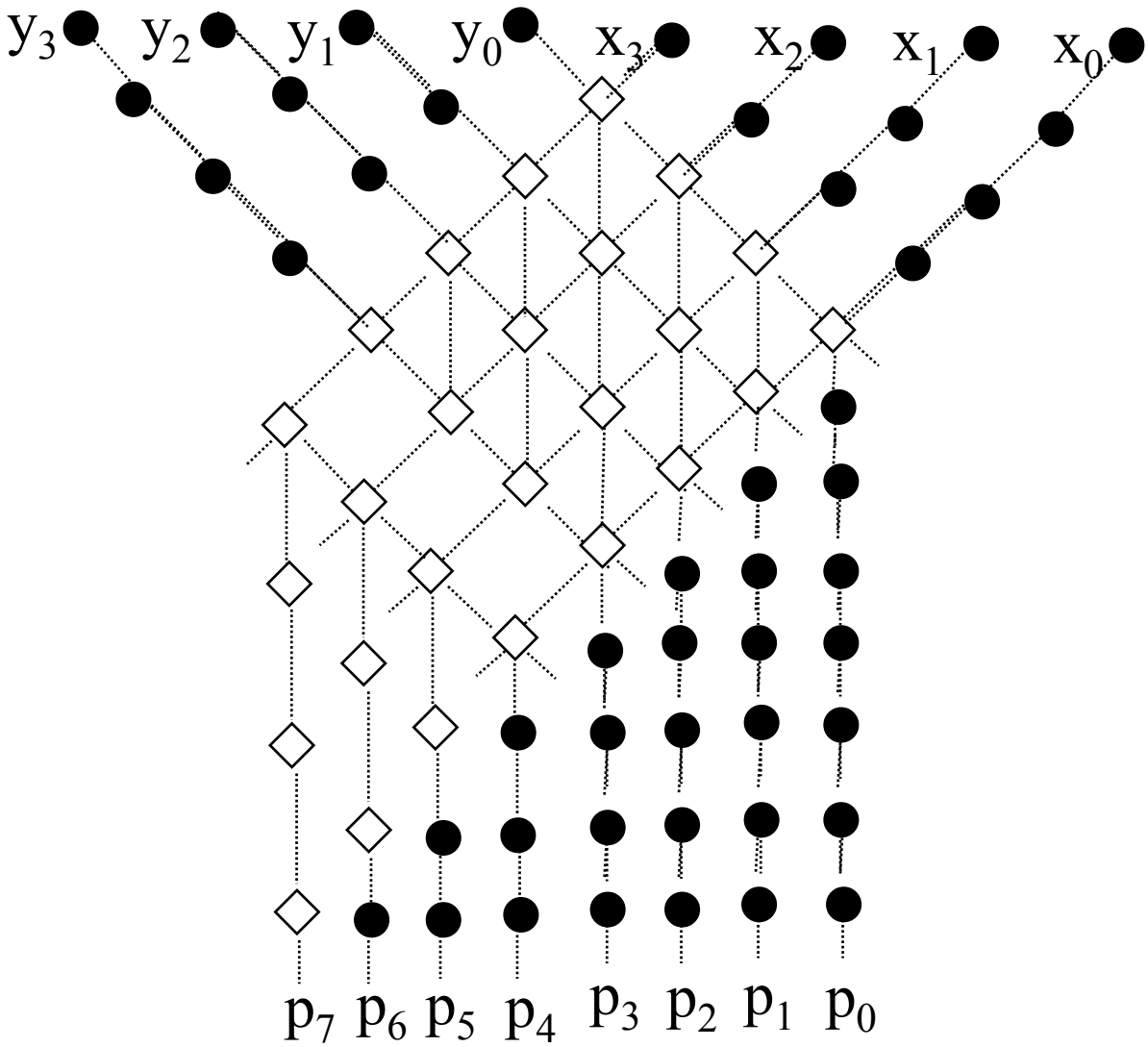


- Speed : parallel > serial/parallel > serial
- Cost : parallel > serial/parallel > serial

Systolic Array Multiplier



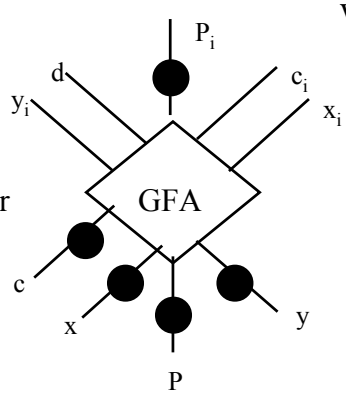
Multiplier structure



Basic cell

● = Latch

GFA = Gated full adder



where p_i= partial product sum in

p = partial product sum out

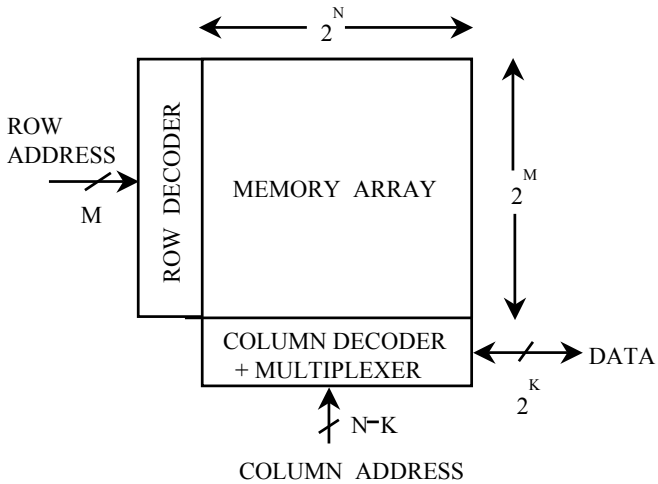
c_i = carry in

c = carry out

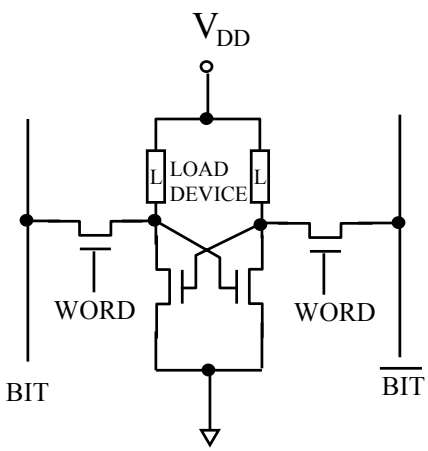
d = line required for two's complement operation

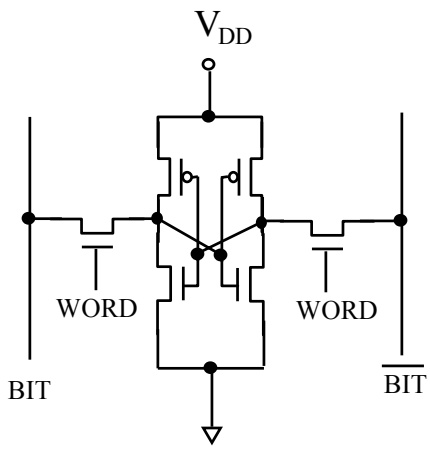
Memory

- RAM (Read & Write) < SRAM, DRAM
  ROM (Read only)
- RAM structure

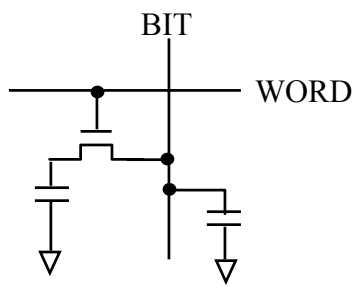


- SRAM (Static random access memory)
  - 4T cell
  - 6T cell



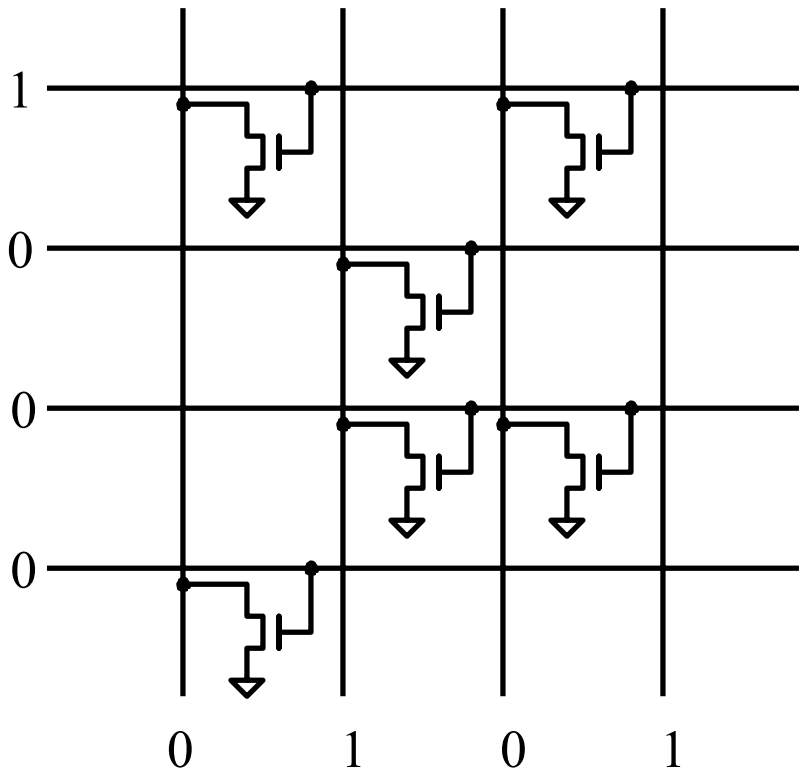


- DRAM (Dynamic random access memory)
  - Slower than SRAM
  - Less expensive than SRAM
  - ☝ one transistor cell

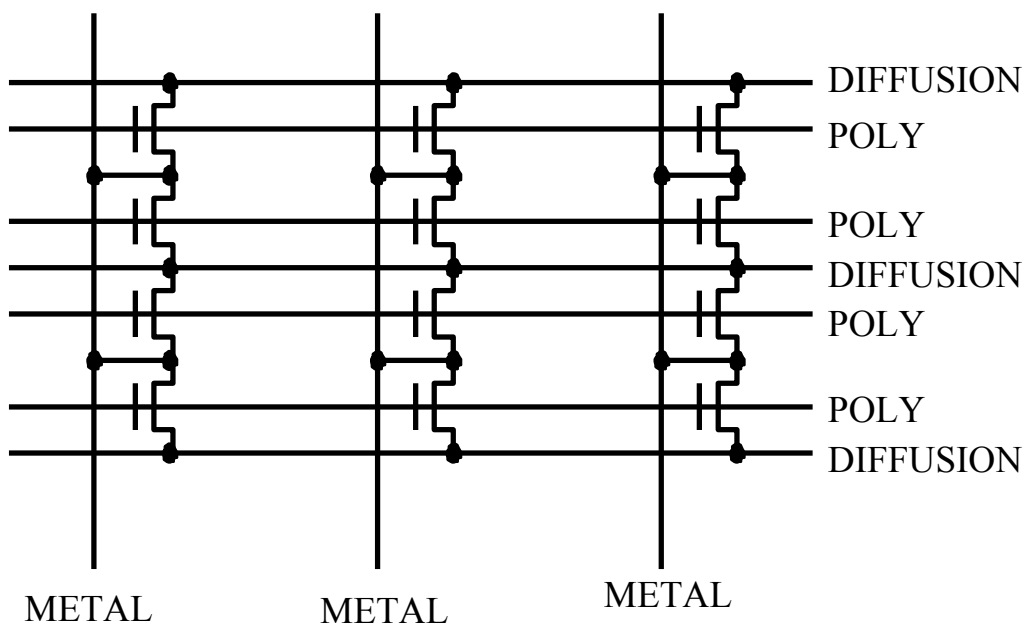


ROM

- Read only memory
  - ROM array



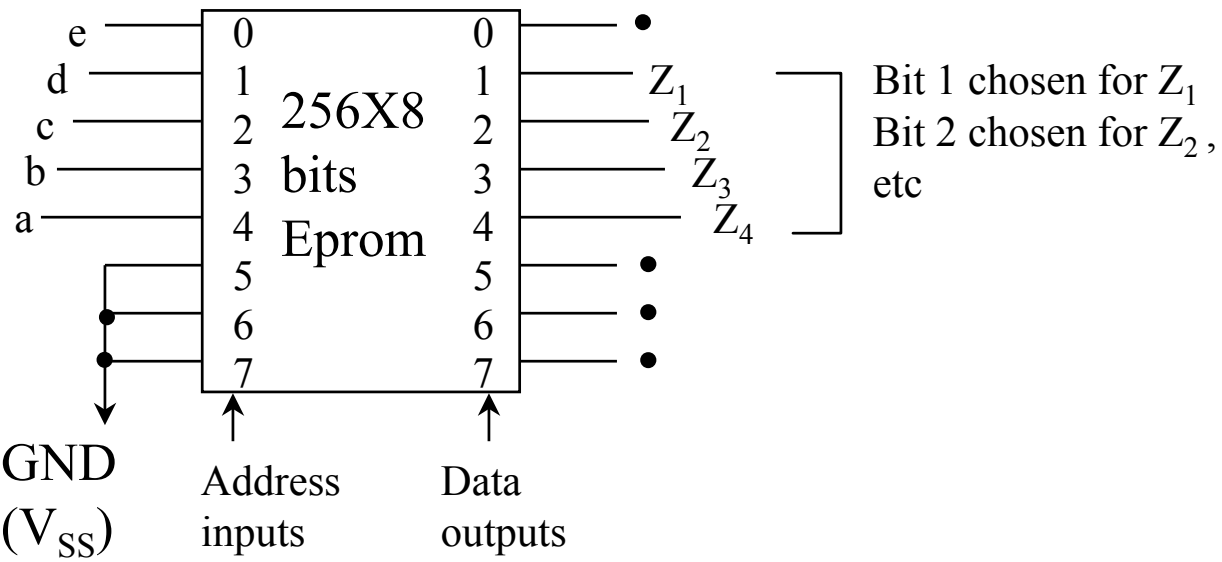
  - Mask layer assignment



Realization of Sum of Product

- ROM
  - more space
    2^8 X 8

Input variables Output functions



Output functions

$Z_1 = a\bar{b}\bar{d}e + \bar{a}\bar{b}c\bar{d}\bar{e} + bc + de = \Sigma\, m(0,3,7,11,12,13,14,15,17,19,21,23,27,28,29,30,31)$

$Z_2 = a\bar{c}\bar{e} = \Sigma\, m(1,3,9,11)$

$Z_3 = bc + de + \bar{c}\bar{d}\bar{e} + bd = \Sigma\, m(0,3,7,8,10,11,12,13,14,15,16,19,23,24,26,27,28,29,30,31)$

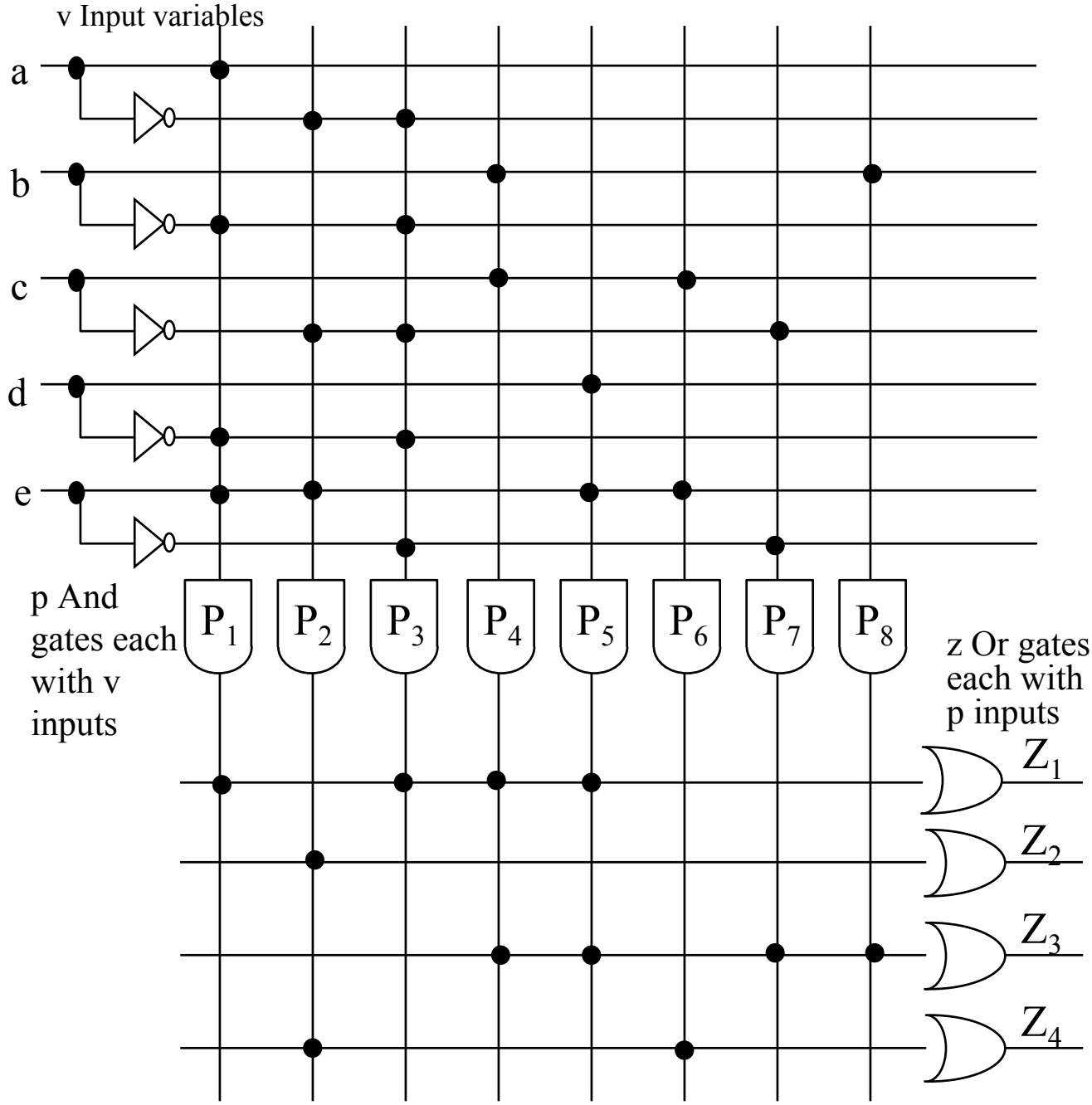
$Z_4 = \bar{a}\bar{c}e + ce = \Sigma\, m(1,3,5,7,9,11,13,15,21,23,29,31)$

Location/address

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Bit 1	1	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	0	1	1	1	1	1
Bit 2	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit 3	1	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1
Bit 4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

Realization of Sum of Product(Cont.)

- PLA(programmable logic array)
  - general solution $V \times P \times Z \Rightarrow 2V \times P + P \times Z$
  - example: $5 \times 8 \times 4 \Rightarrow 10 \times 8 + 8 \times 4$



5X8X4 PLA shown symbolically and programmed for:

$Z_1 = p_1 + p_3 + p_4 + p_5$ $\therefore$ $Z_1 = a\bar{b}de + \bar{a}b\bar{c}d\bar{e} + bc + de$

$Z_2 = p_2$ $\therefore$ $Z_2 = \bar{a}\bar{c}e$

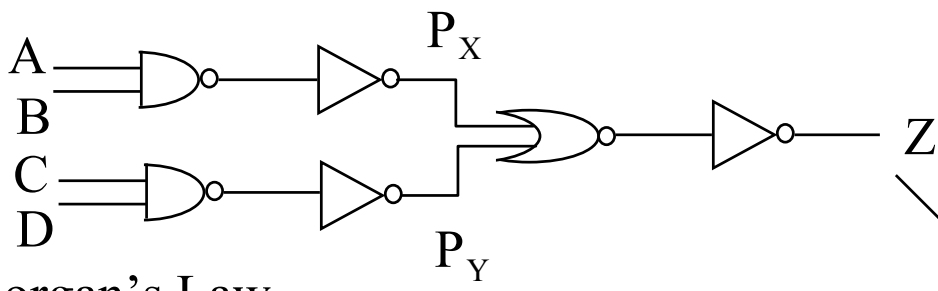
$Z_3 = p_4 + p_5 + p_7 + p_8$ $\therefore$ $Z_3 = bc + de + \bar{c}d\bar{e} + bd$

$Z_4 = p_2 + p_6$ $\therefore$ $Z_4 = \bar{a}\bar{c}e + ce$

PLA Floor Plan

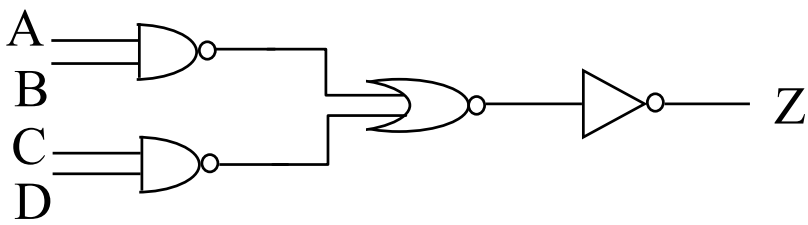
- Sum of product

$$Z=P_X+P_Y=AB+CD;\ P_X=AB,\ P_Y=CD$$

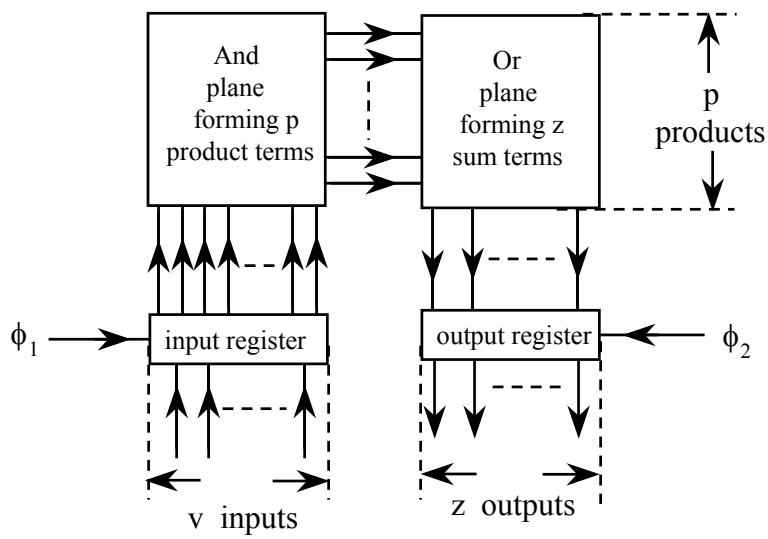


— Demorgan's Law

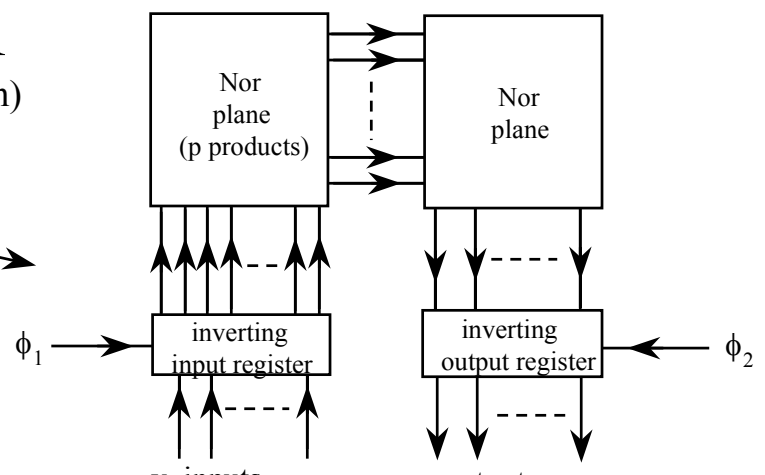
$$Z=AB+CD=\overline{\overline{A}+\overline{B}+\overline{C}+\overline{D}}$$



- AND–OR



- NOR— NOR (General Solution)



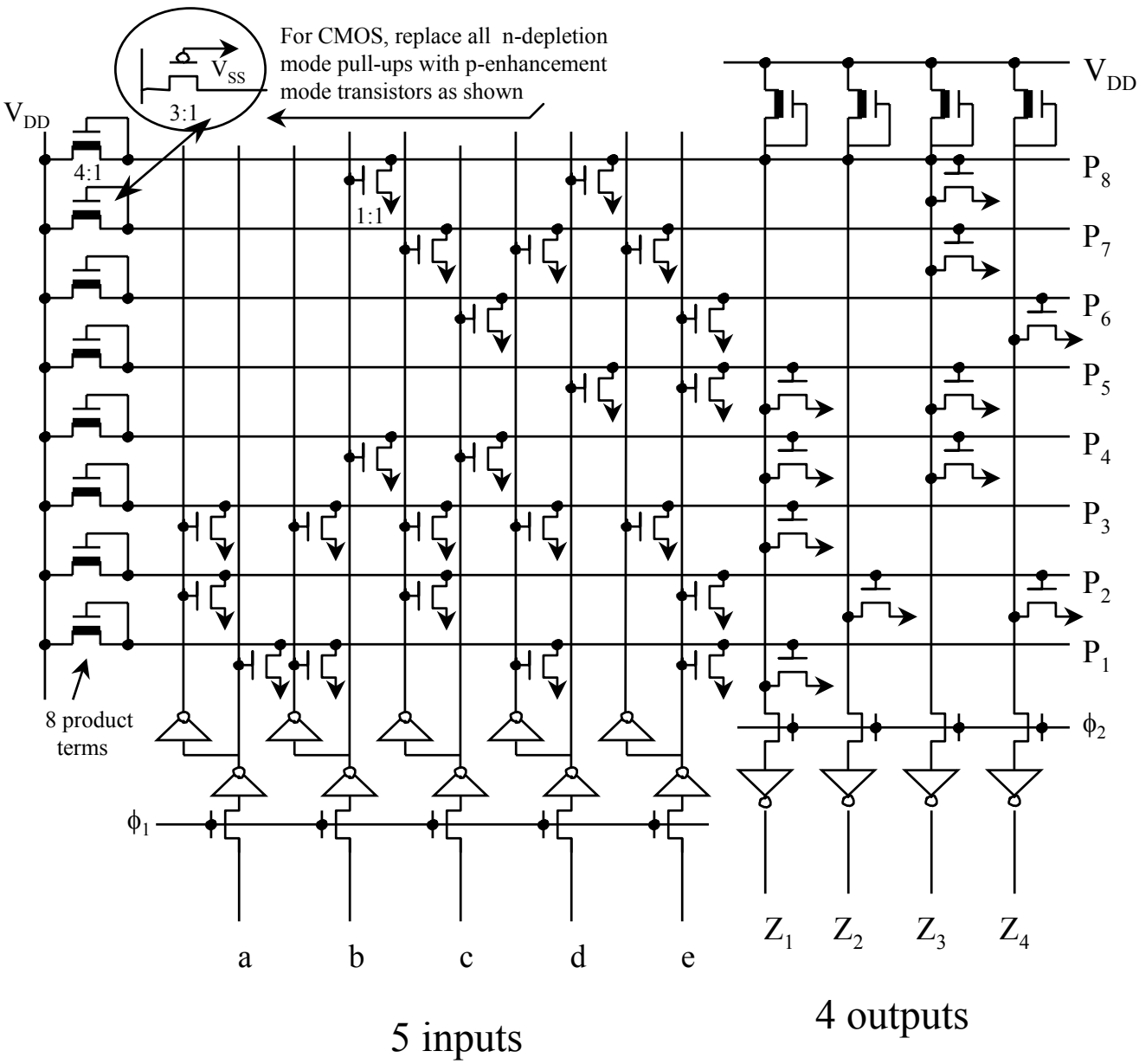
Circuit Diagram of PLA

- Cost depends on dimension $V \times P \times Z$

  V： of input
  P： of product term
  Z： # of output

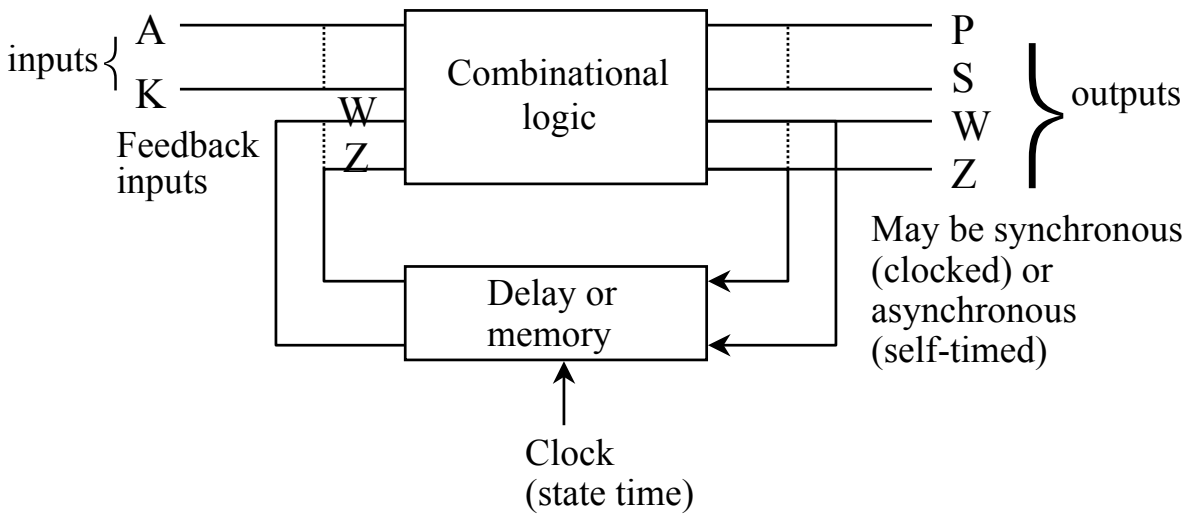
Dimesion $P\{\square_{2V} + P\{\square_{Z} = 2PV + PZ$

- Example: $5 \times 8 \times 4$



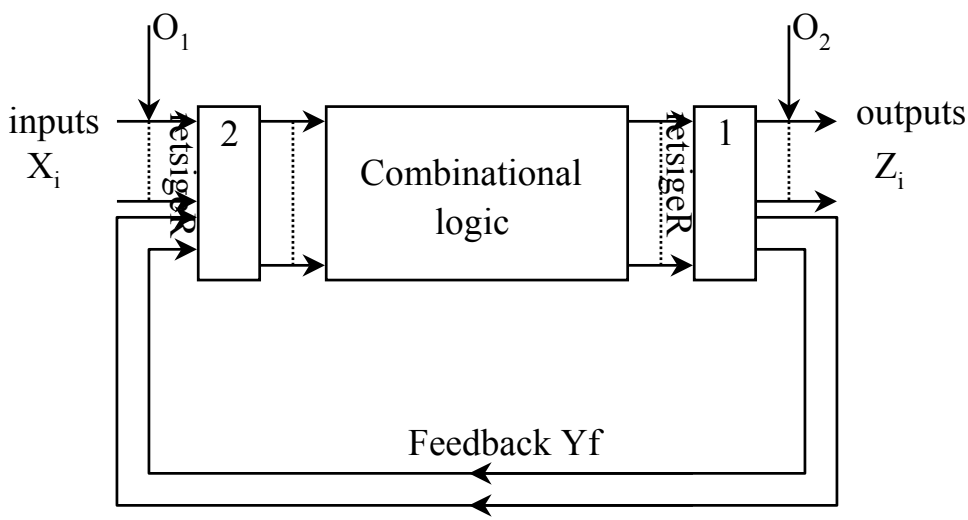
Finite State Machine (FSM)

- Seguential circuit
  - General form :
    combinational logic could be implemented
    by ROM, PLA, and other circuits



  - PLA with feedback

    (General approach)



Feedback in a register transfer path implementing
a finite state machine (synchronous)

FSM Design Example

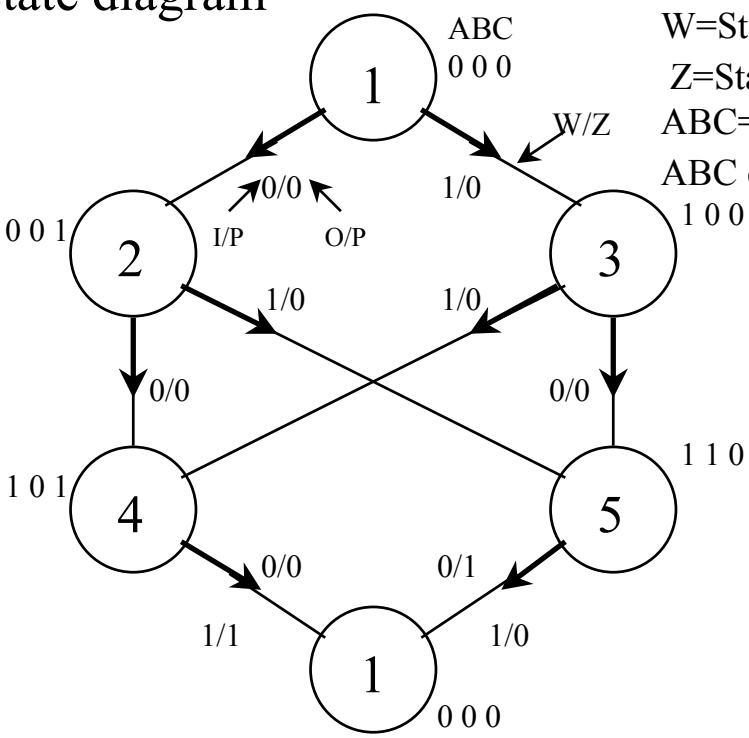
- Specification

We are to design a circuit, the input to which are serial binary digits clocked by ϕ_2 and presented at an input W. The circuit is to produce an output Z=1 when the circuit has detected an odd number of 1s in groups of three bits arriving at input W. Otherwise, Z is to be 0.Groups of three bits arriving at W do not overlap, that is, bits 1, 2, and 3 from the first group, bits 4, 5, and 6 the second, bits 7, 8, and 9 the third group of three, etc. Z is to be clocked by ϕ_2 and W can be clocked into the PLA by ϕ_1 since it is available following the preceding ϕ_2.

- Design procedure

state diagram ⟶ state transition table
⟶ minimization of number of product term ⟶ circuit ⟶ layout

- state diagram



W=State of I/P clocked in by ϕ_1
Z=State of O/P after ϕ_2
ABC=State vector
ABC clocked in by ϕ_1

- State transition table

State number	Inputs					Outputs			
	W	State vector A_p	B_p	C_p	And term	Next vector A_p	B_p	C_p	Z
✸	0	0	0	0	R_1	0	0	1	0
	1	0	0	0	R_2	1	0	0	0
✸	0	0	0	1	R_3	1	0	1	0
	1	0	0	1	R_4	1	1	0	0
✸	0	1	0	0	R_5	1	1	0	0
	1	1	0	0	R_6	1	0	1	0
✵	0	1	0	1	R_7	0	0	0	0
	1	1	0	1	R_8	0	0	0	1
⌖	0	1	1	0	R_9	0	0	0	1
	1	1	1	0	R_{10}	0	0	0	0

FSM Design Example (Cont.)

- Sum of product

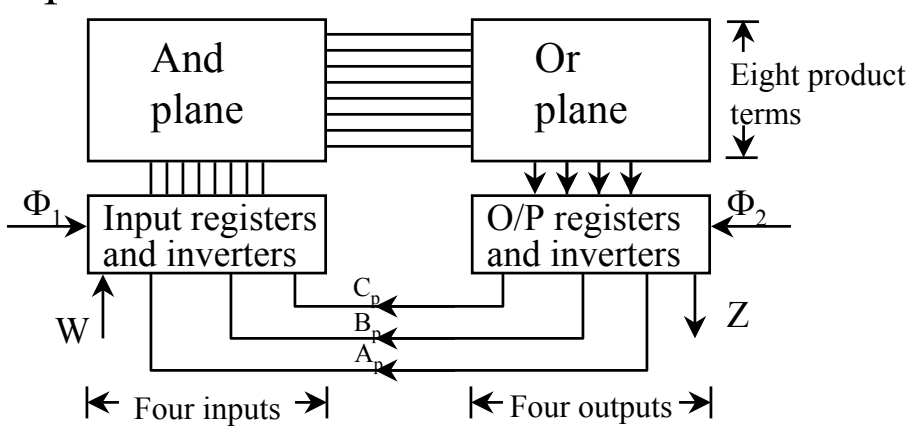
$A_n = R_2 + R_3 + R_4 + R_5 + R_6$
$B_n = R_4 + R_5$
$C_n = R_1 + R_3 + R_6$
$Z = R_8 + R_9$

Product term	*Minterm Form*	*Simplified form*
R_1	$\overline{W}A\overline{B}\overline{C}$	$\overline{W}A\overline{C}$
R_2	$W\overline{A}\overline{B}\overline{C}$	$W\overline{A}\overline{C}$
R_3	$W\overline{A}B\overline{C}$	$W\overline{A}C$
R_4	$WAB\overline{C}$	$WA\overline{C}$
R_5	$\overline{W}AB\overline{C}$	$WAB\overline{C}$
R_6	$WAB\overline{C}$	$WAB\overline{C}$
R_7	$\overline{W}AB\overline{C}$	$\overline{W}AC$
R_8	$WAB\overline{C}$	WAC
R_9	$\overline{W}AB\overline{C}$	$\overline{W}B$
R_{10}	$WAB\overline{C}$	WB

And plane (*Nor* plane)

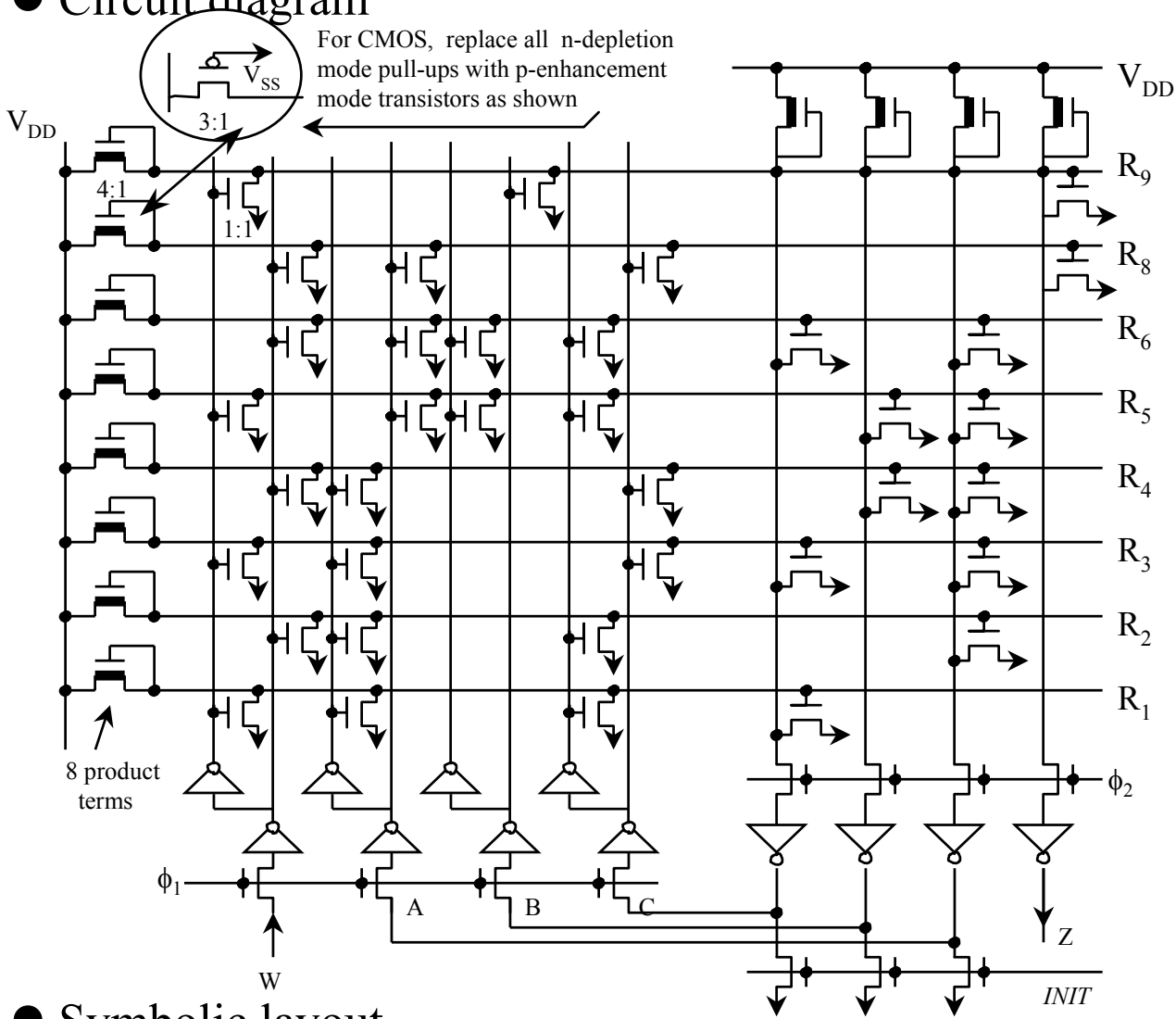
$\overline{R}_1 = W+A+C$ that is $R_1 = \overline{W}\overline{A}\overline{C}$
$\overline{R}_2 = \overline{W}+A+C$ -etc.-
$\overline{R}_3 = W+A+\overline{C}$ -
$\overline{R}_4 = \overline{W}+A+\overline{C}$ -
$\overline{R}_5 = W+\overline{A}+B+C$ -
$\overline{R}_6 = \overline{W}+\overline{A}+B+C$ -
$\overline{R}_8 = \overline{W}+\overline{A}+\overline{C}$ -
$\overline{R}_9 = W+\overline{B}$ $R_9 = \overline{\overline{W}B}$

- minimization of number of product term (Homework)
- Floor plan

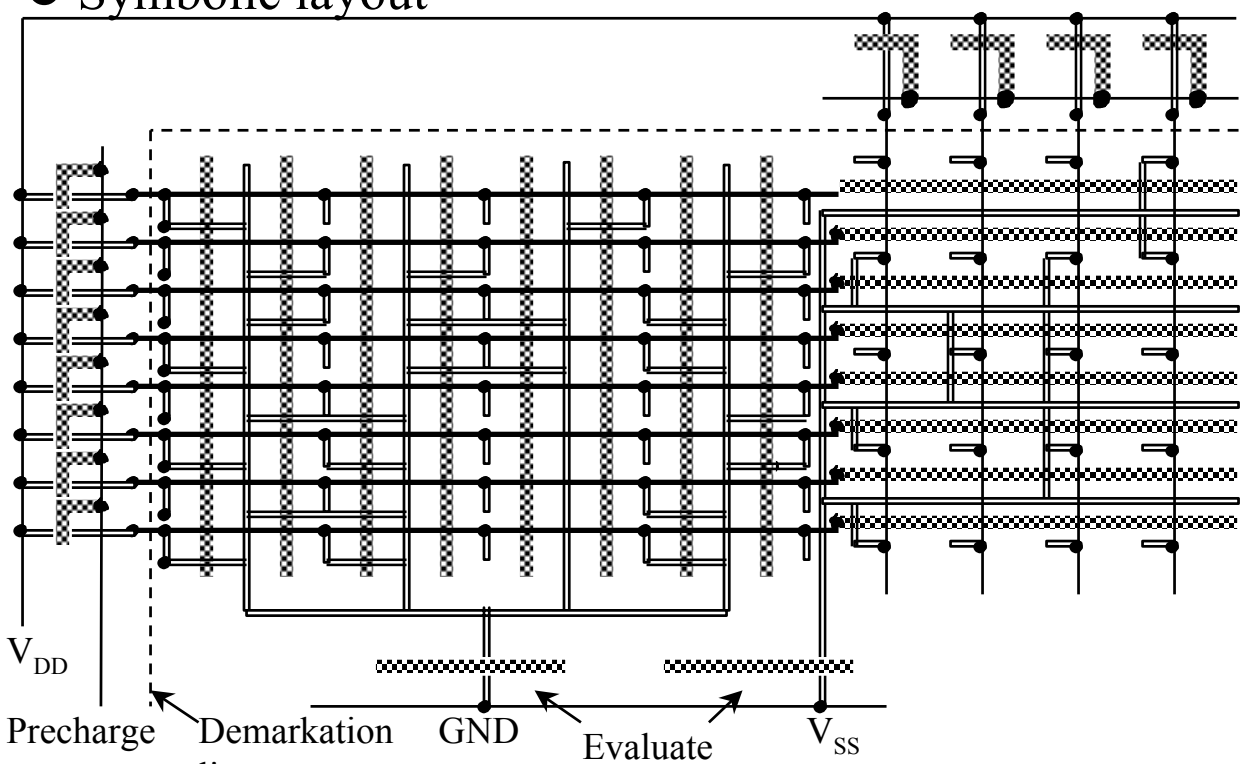


FSM Design Example (Cont.)

- Circuit diagram

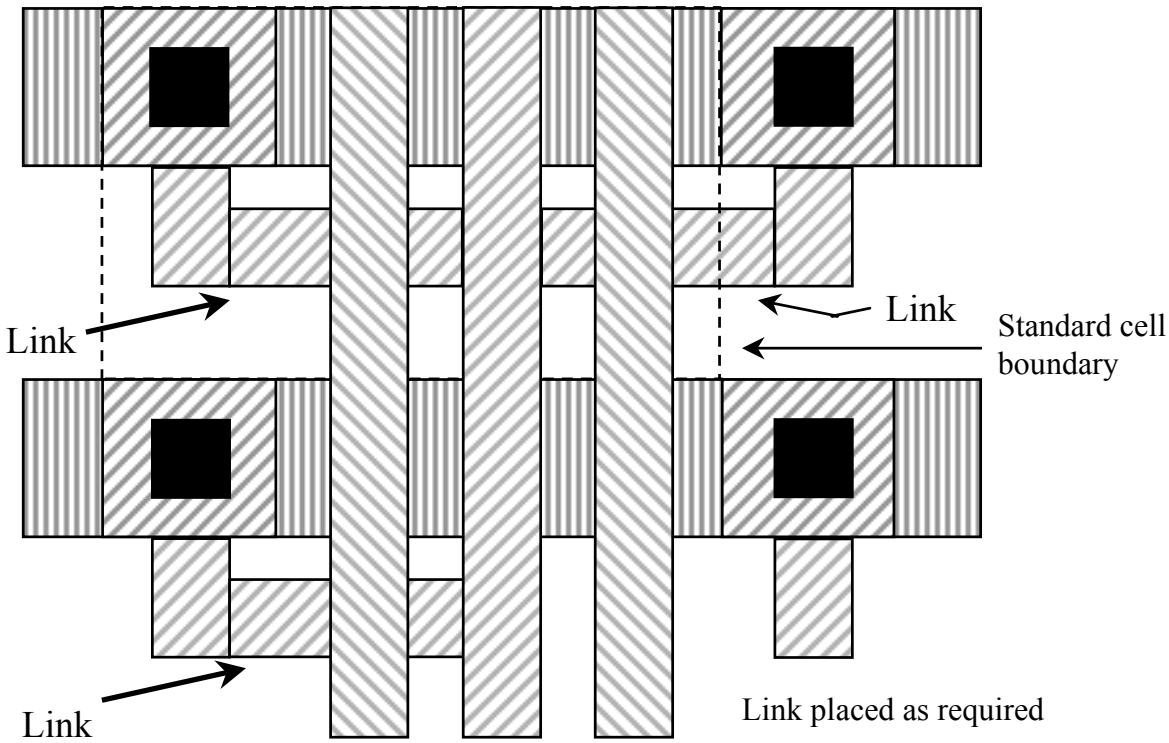


- Symbolic layout



FSM Design Example(Cont.)

- FSM layout cell



- Standard cell approach