Adder, Shifter, MUX, Parity Generator

Mahit Kumar Paul
Assistant Professor, Dept. of CSE
RUET, Rajshahi-6204

mahit.cse@gmail.com
mahit@cse.ruet.ac.bd

Adder

- Adds two N-bit binary numbers

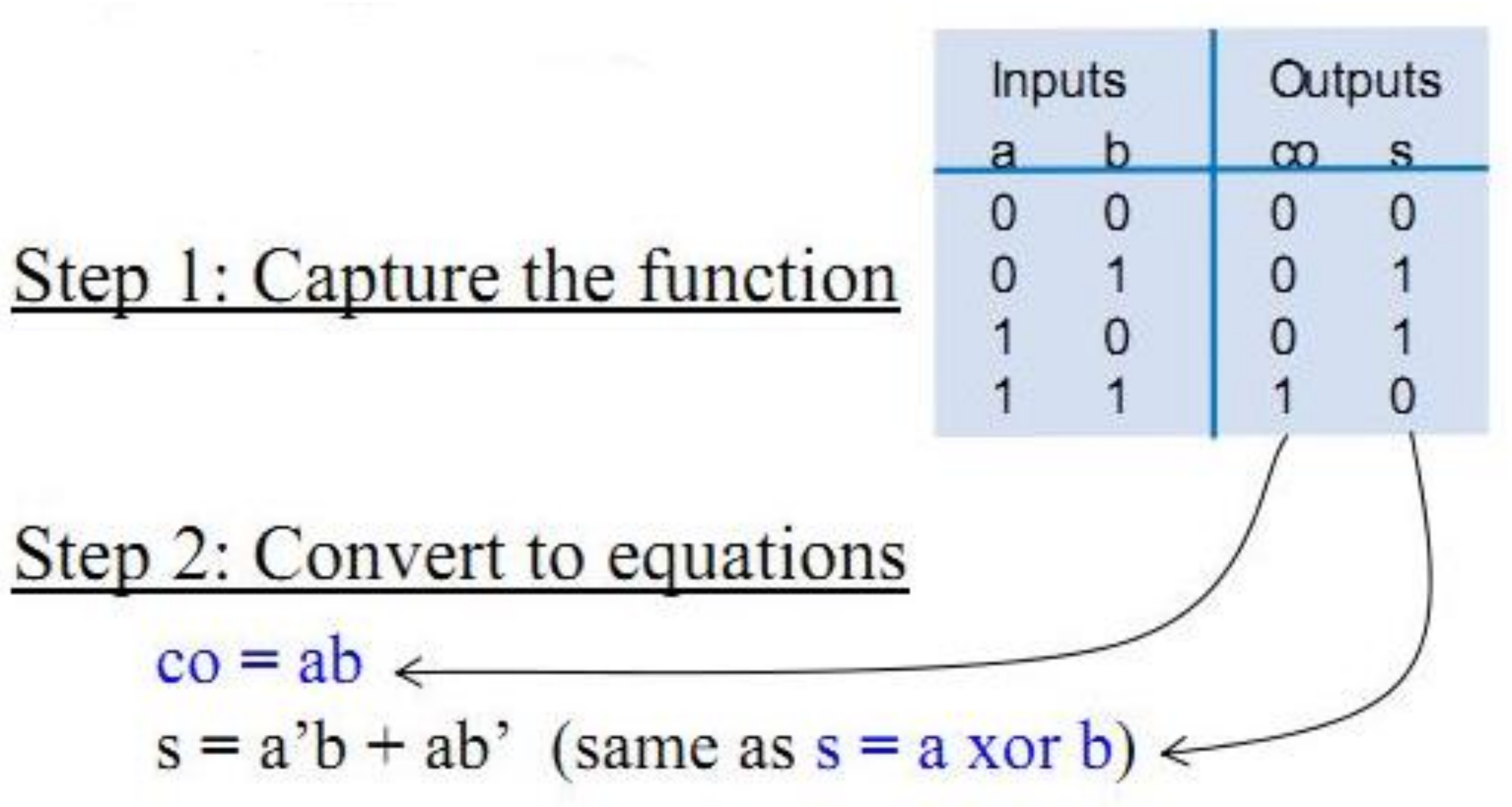
  - 2-bit adder: adds two 2-bit numbers

  - Can give output 3-bit result

    -e.g., 01 + 11 = 100   (1 + 3 = 4)

Half Adder

- Adds **2** bits
- Generates sum and carry



Step 1: Capture the function

Inputs		Outputs	
a	b	co	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

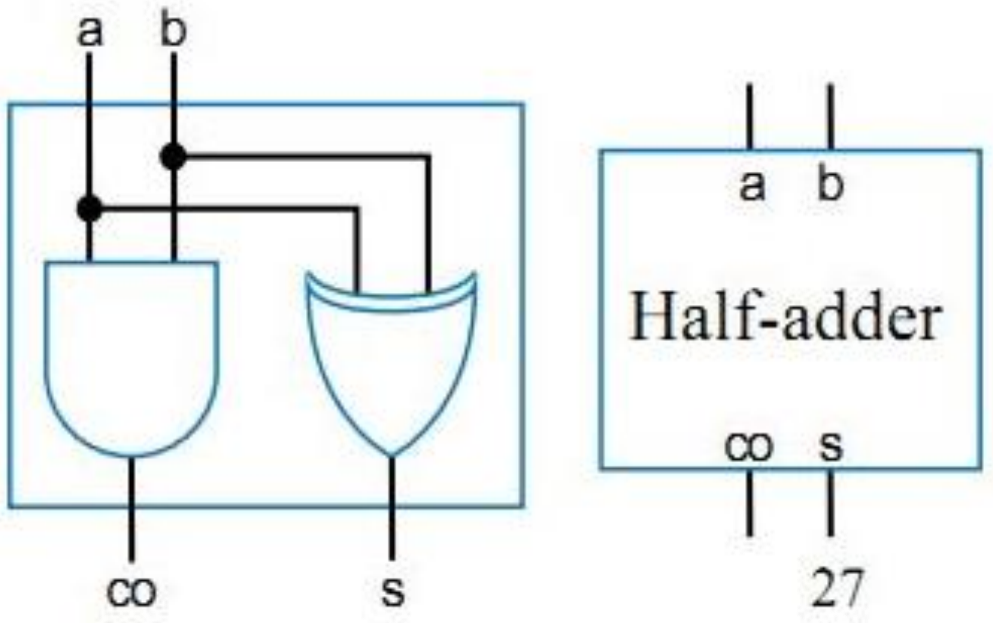
Step 2: Convert to equations

co = ab

s = a'b + ab' (same as s = a xor b)

Half Adder…

Step 3: Create the circuit

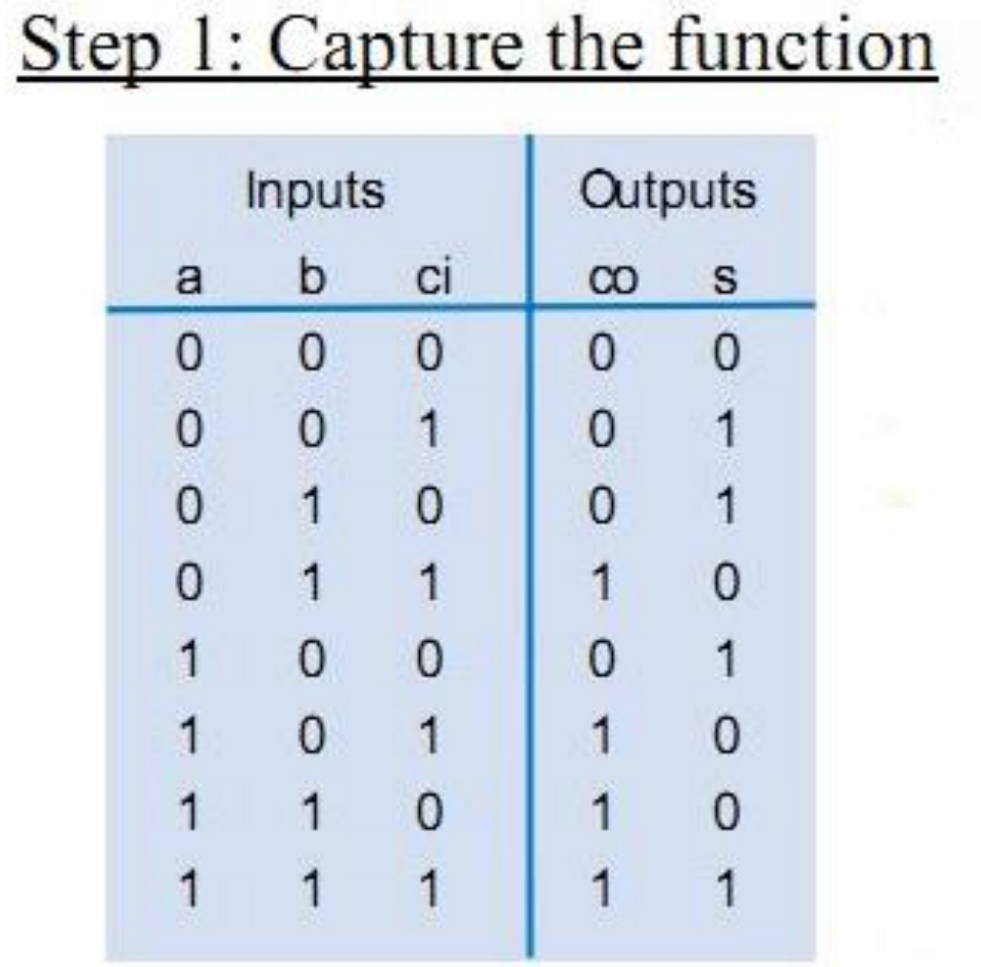


Half Adder Using CMOS Logic

Try Yourself

Full Adder

- Adds **3** bits
- Generates sum and carry



Step 1: Capture the function

Inputs			Outputs	
a	b	ci	co	s
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

Full Adder…

Step 2: Convert to equations

co = a’bc + ab’c + abc’ + abc

co = a’bc +abc +ab’c +abc +abc’ +abc

co = (a’+a)bc + (b’+b)ac + (c’+c)ab

co = bc + ac + ab

$$S = \overline{A}\,\overline{B}\,C + \overline{A}\,B\,\overline{C} + ABC + A\,\overline{B}\,\overline{C}$$

$$S = \overline{A}\,(\overline{B}\,C + B\,\overline{C}) + A(BC + \overline{B}\,\overline{C})$$

$$S = \overline{A}\,(\overline{B}\,C + B\,\overline{C}) + A(\overline{\overline{BC}} + (\overline{B+C}))$$

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

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

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

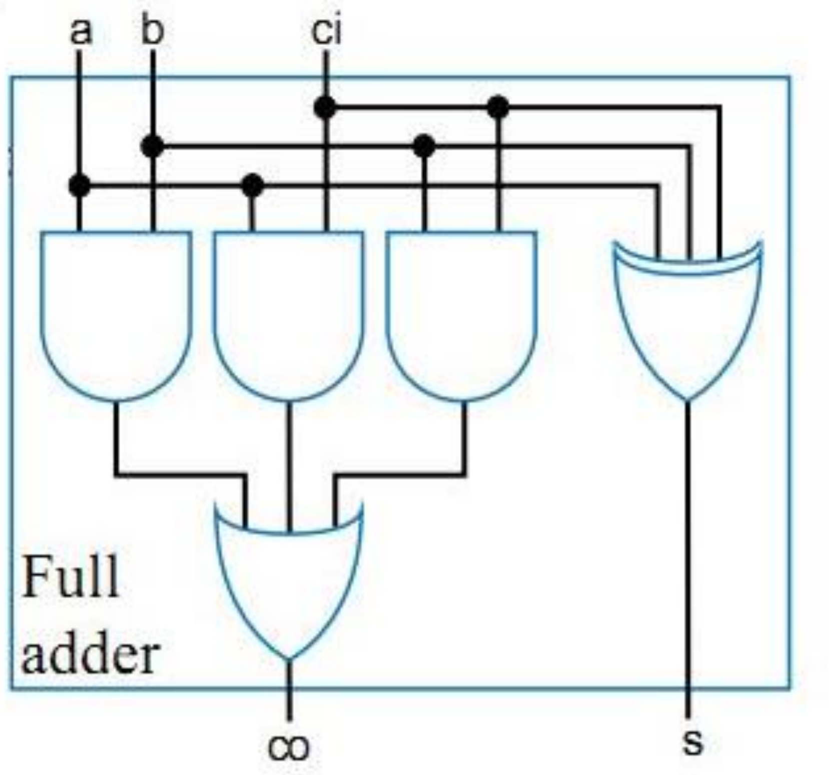
$$S = \overline{A}\,(\overline{B}\,C + B\,\overline{C}) + A(\overline{\overline{B}\,C + B\,\overline{C}})$$

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

$$S = (A \oplus B) \oplus C$$

Full Adder…

Step 3: Create the circuit



Full Adder Using CMOS Logic…

Try Yourself

Shifter

- In Binary Operation, shifting is a bitwise operation that shifts the operand 1 to n-1 places to the right or left.

Original Data	1	0	1	1
After 2 bit shift	1	1	1	0

Shifter…

- Left Shift
- Right Shift
- n-bit shift = original data

1	0	1	1

4-bit Input Data

1	1	0	1

3 bit left shift

1	0	1	1

4 bit left shift

Shifter…

- k-bit right shift = n-k bit left shift

1	0	1	1

4-bit Input Data

1	1	0	1

1 bit right shift

1	1	0	1

3 bit left shift

4x4 Shifter

- Any general purpose n-bit shifter should be able to shift incoming data by up to (n – 1) place in a right-shift or left-shift direction.
- Further specifying that all shifts should be on an end-around basis, so that any bit shifted out at one end of a data word will be shifted in at the other end of the word, then the problem of right shift or left shift is greatly eased.
- The shifter must have:
  - input from a four line parallel data bus
  - four output lines for the shifted data
  - means of transferring input data to output lines with any shift from 0 to 3 bits

Switch

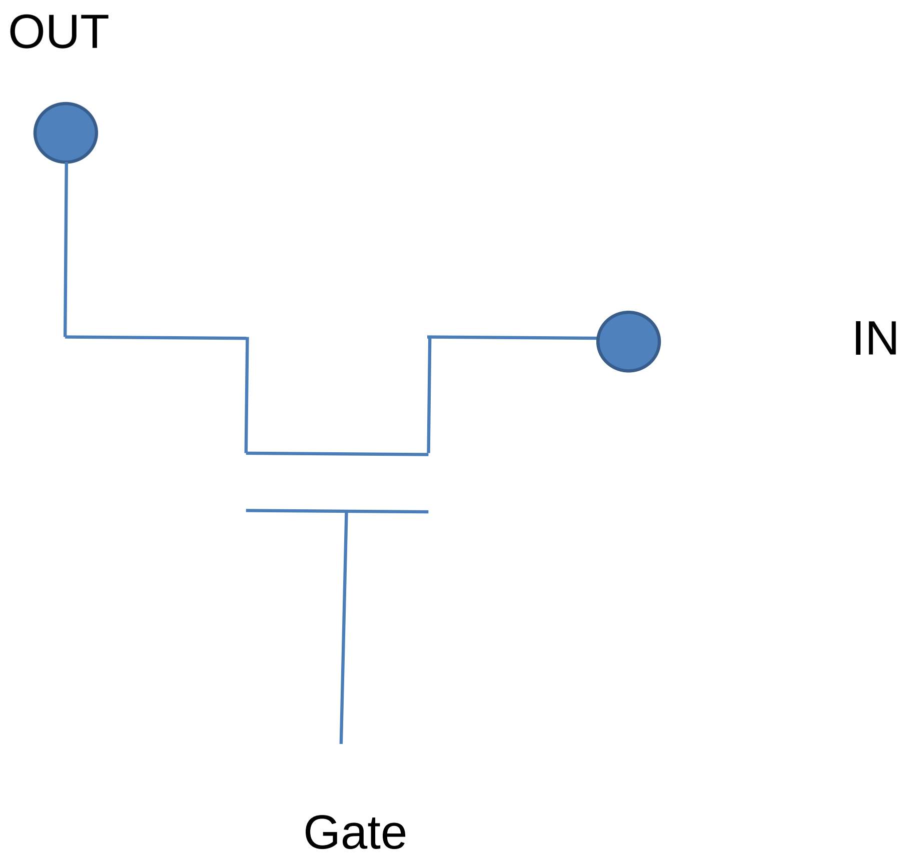


Figure: A switch

4x4 Crossbar Switch

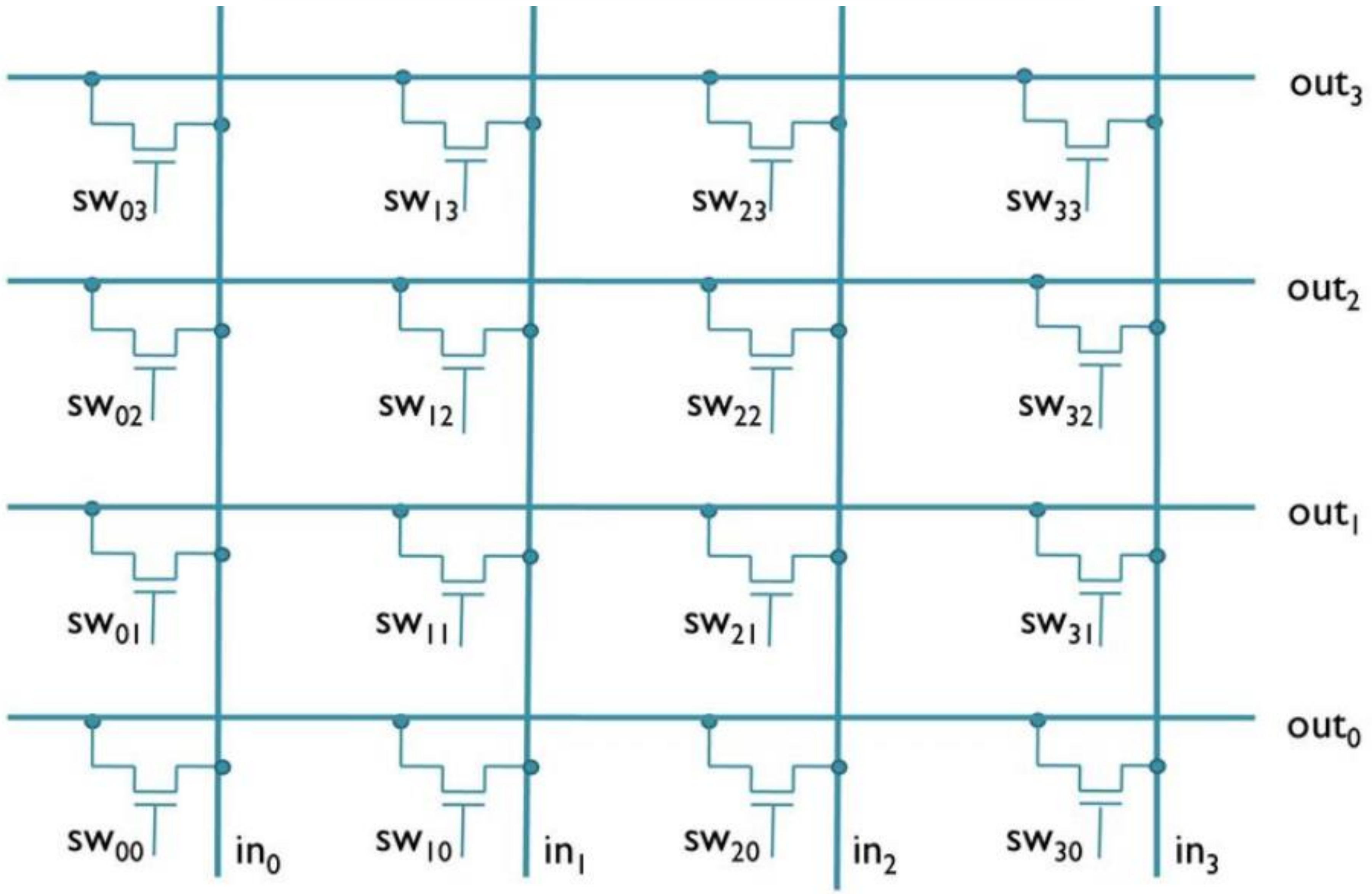
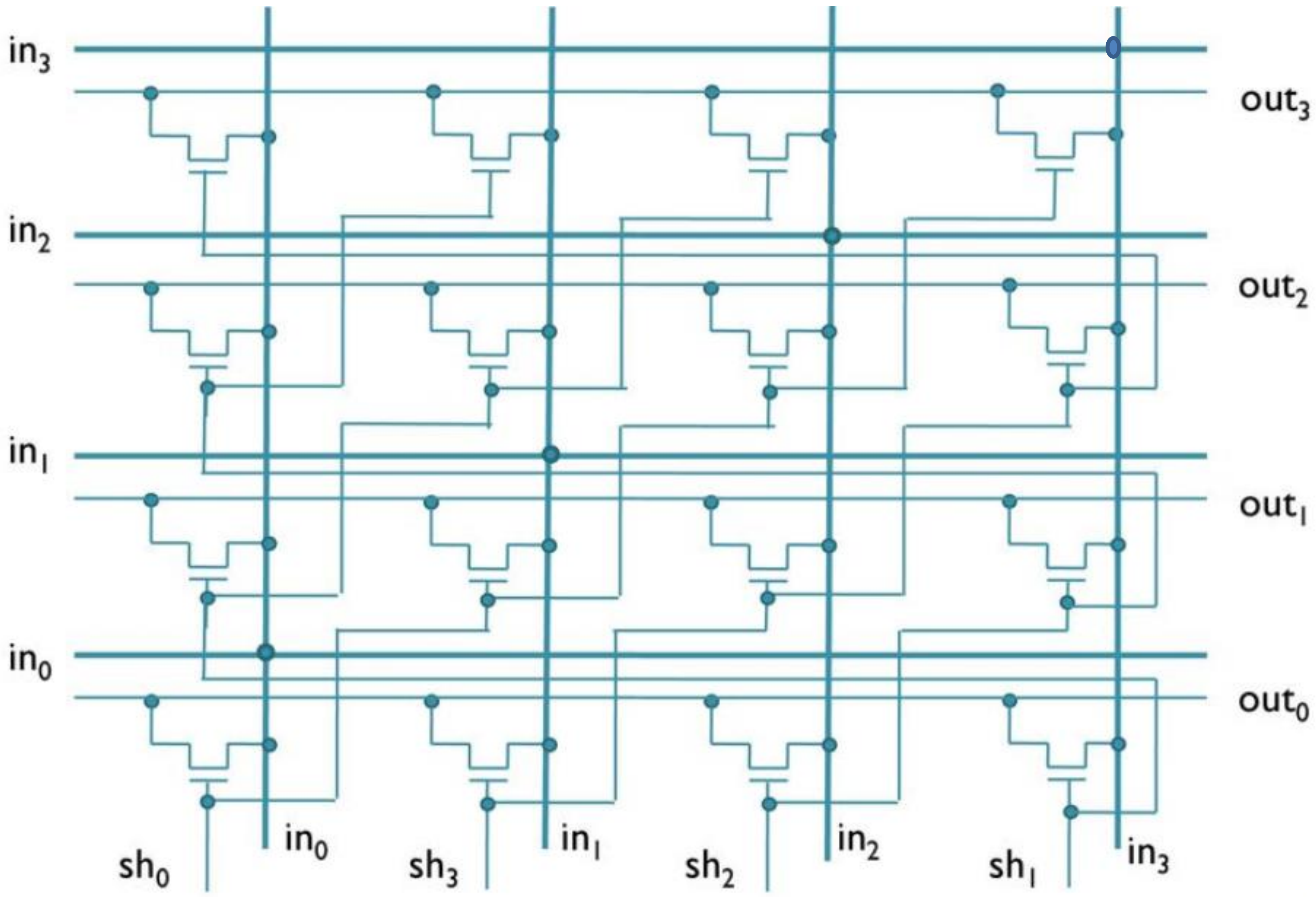
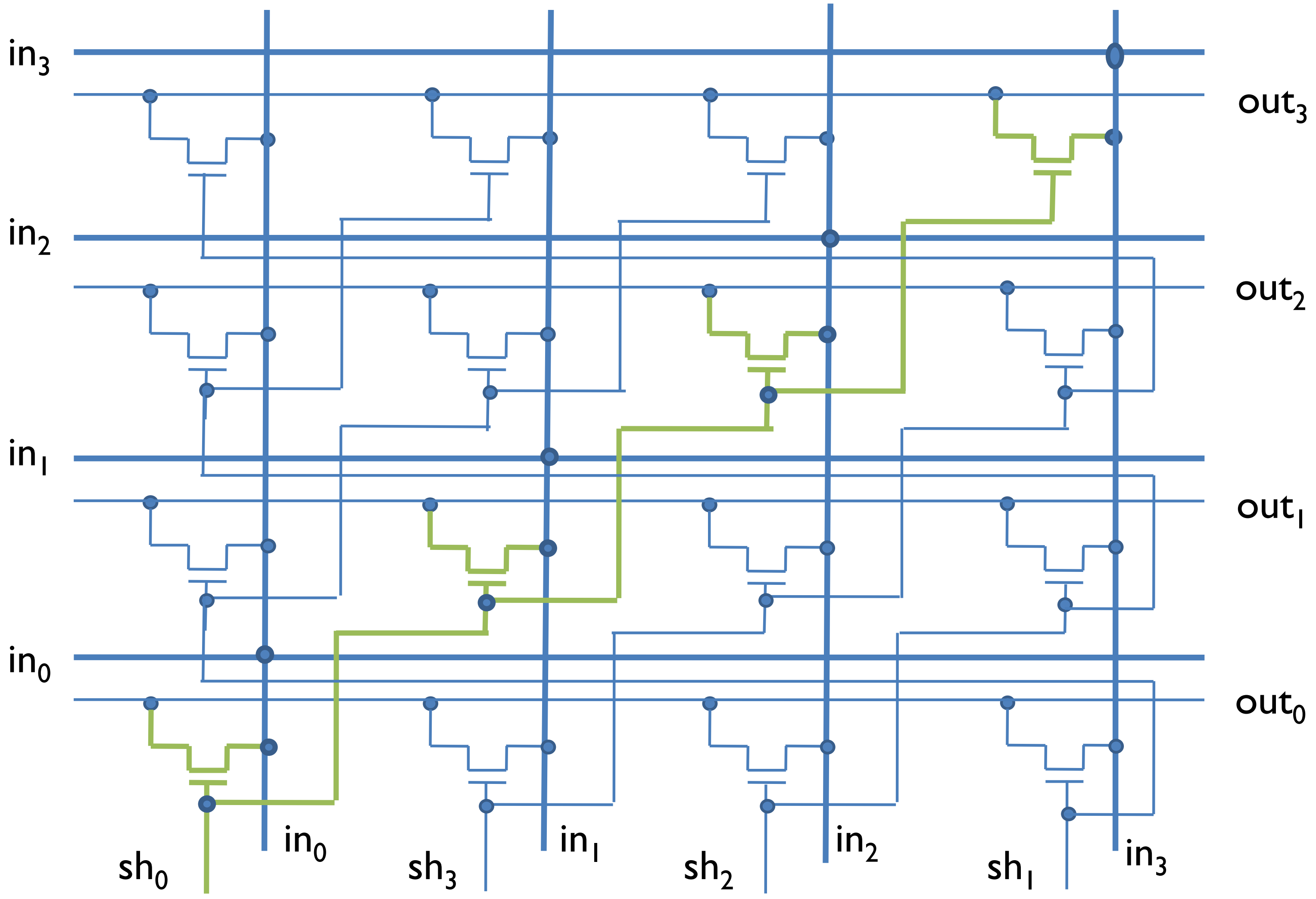


Fig.: 4x4 Crossbar Switch

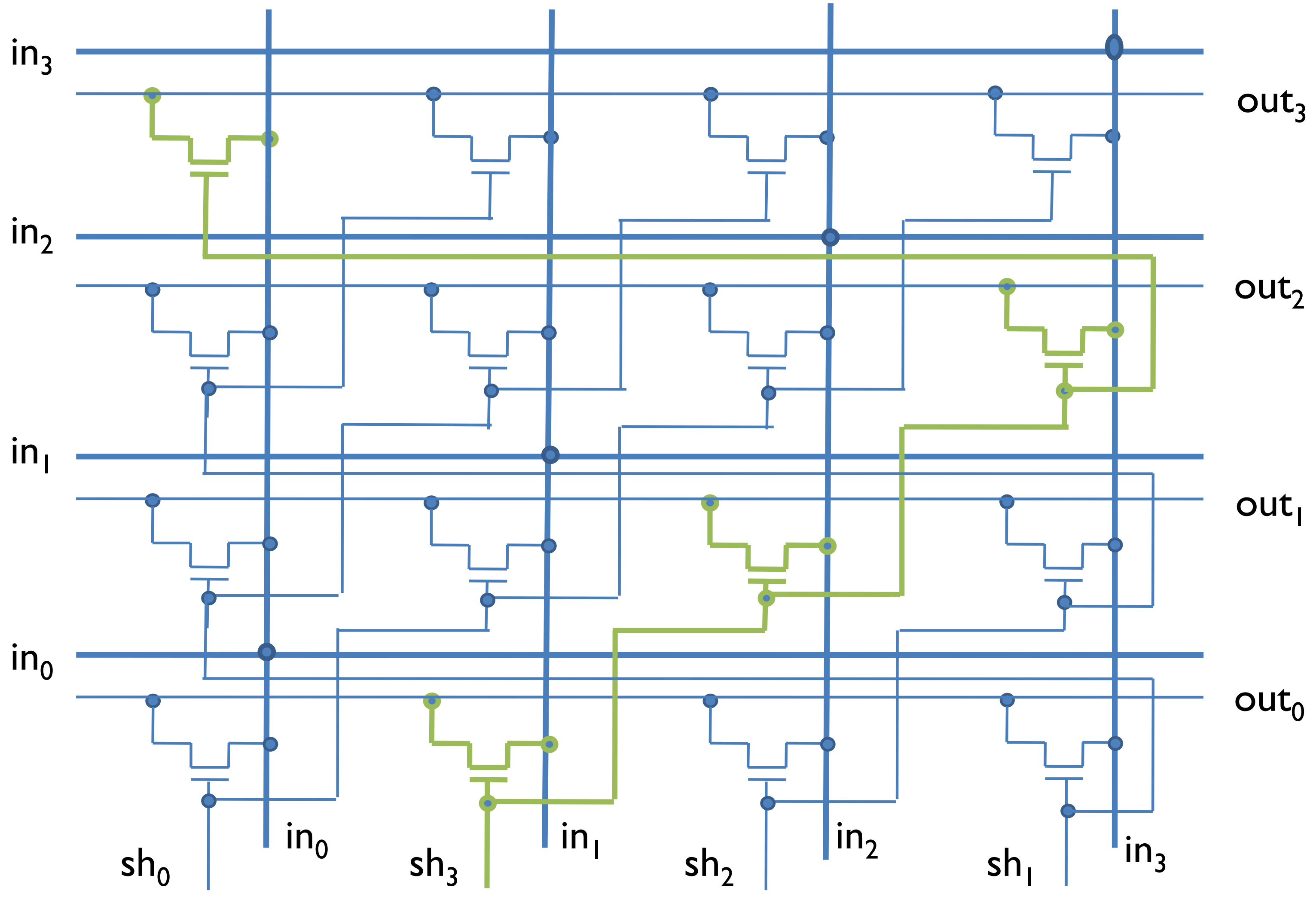
4x4 Barrel Shifter…



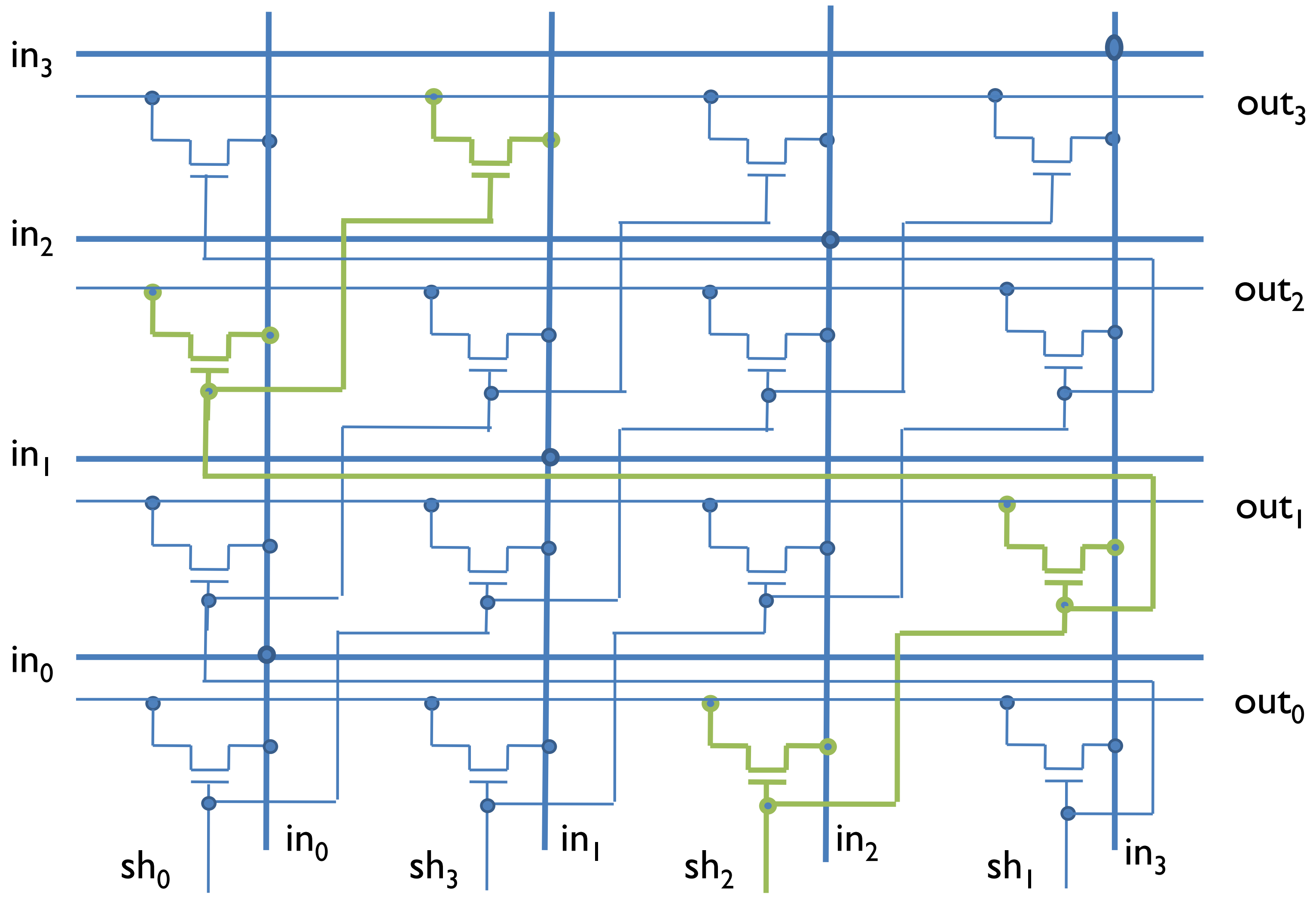
4x4 Barrel Shifter…



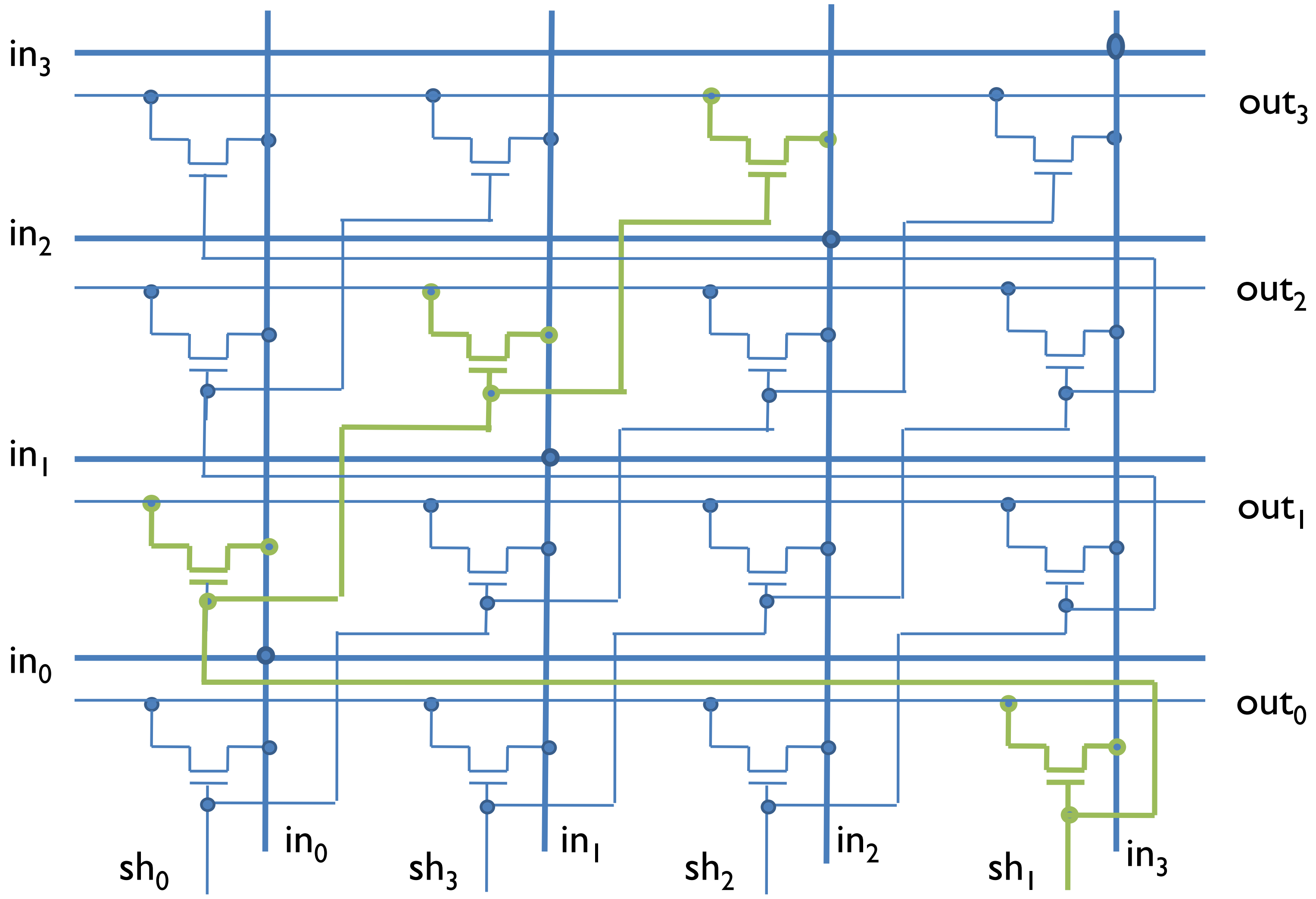
4x4 Barrel Shifter...



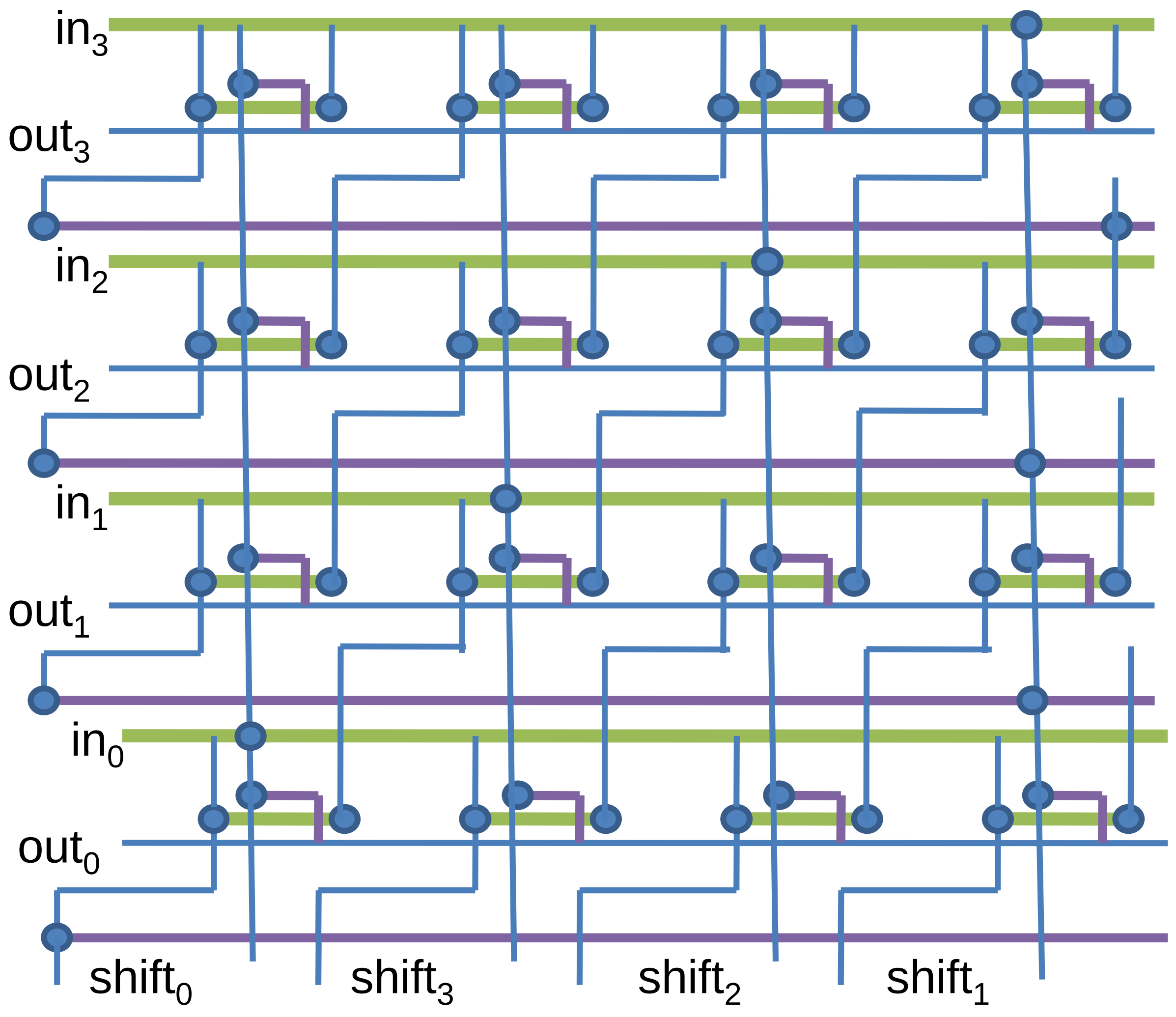
4x4 Barrel Shifter…



4x4 Barrel Shifter...



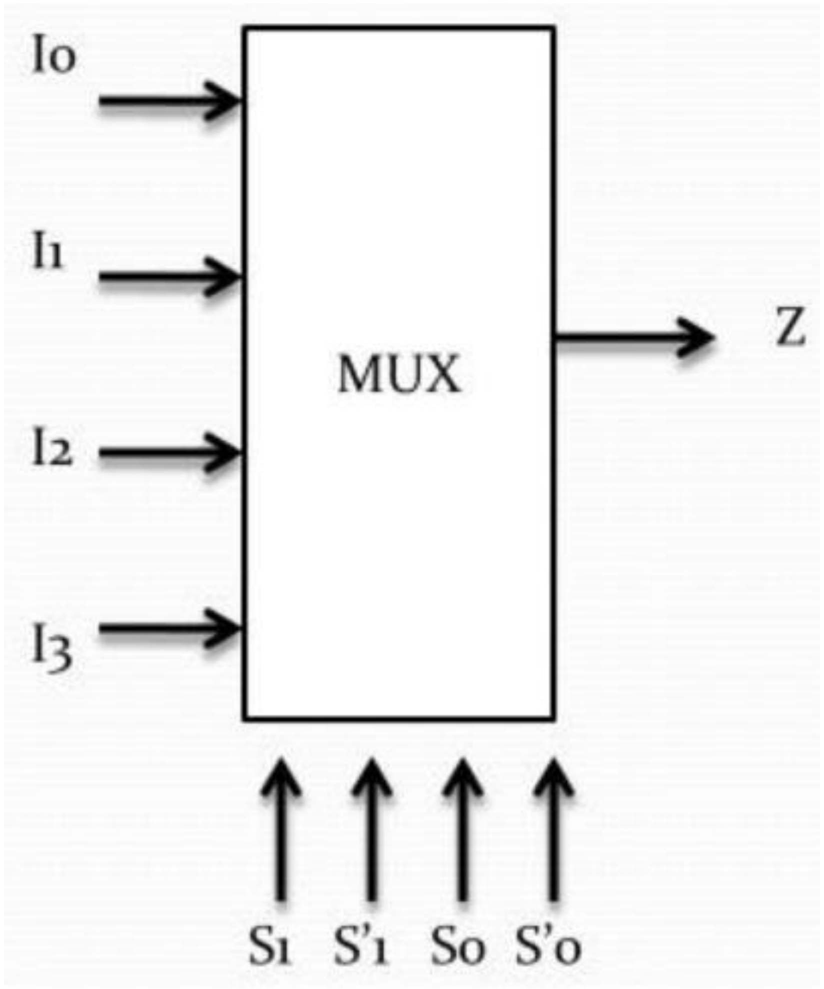
4x4 Barrel Shifter Stick Diagram



Applications

- Multiplication and Division
- Floating point arithmetic
- Alignment
- Microprocessor

Multiplexers (Data Selectors)



S_1	S_0	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Z = I_0.\bar{S}_1.\bar{S}_0 + I_1.\bar{S}_1.S_0 + I_2.S_1.\bar{S}_0 + I_3.S_1.S_0$$

Fig: 4:1 MUX

Multiplexers…

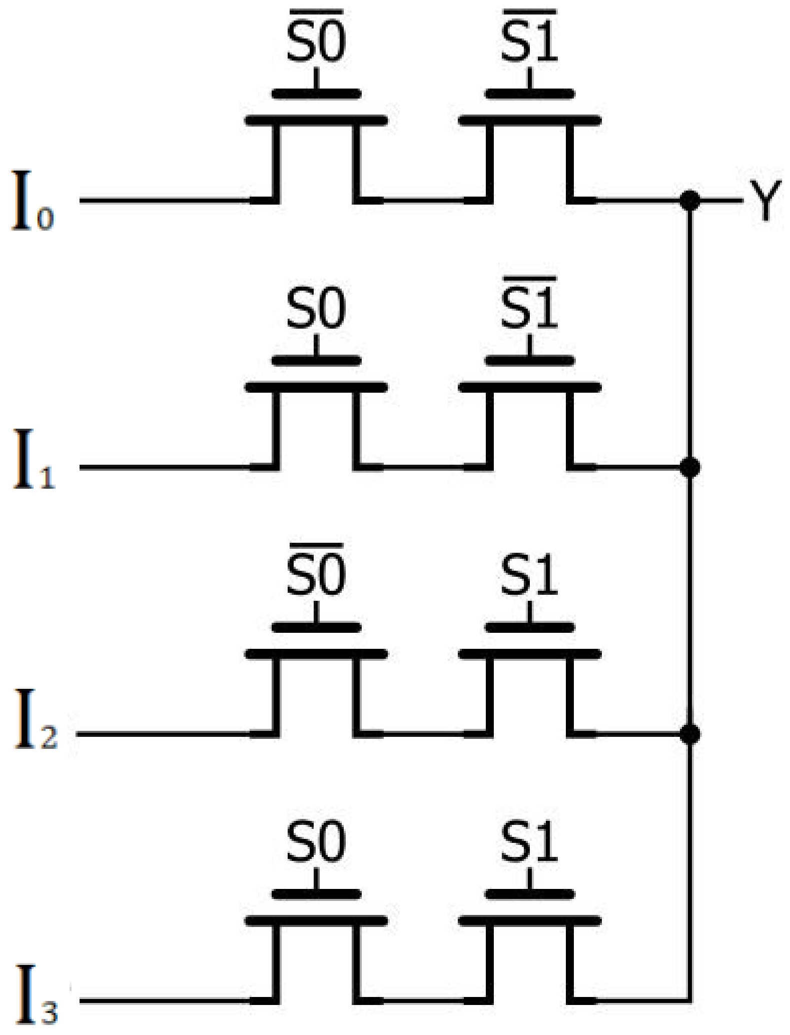
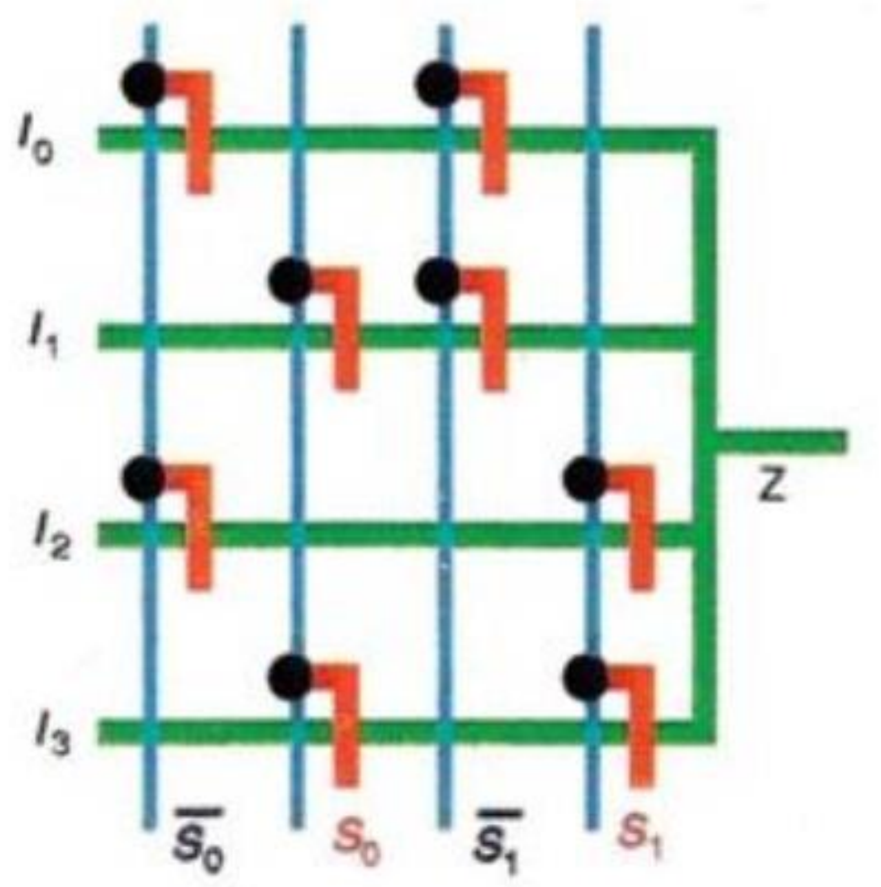


Fig: 4:1 MUX using NMOS Pass Transistor

24

Multiplexers…



$$Z = I_0.\overline{S}_1.\overline{S}_0 + I_1.\overline{S}_1.S_0 + I_2.S_1.\overline{S}_0 + I_3.S_1.S_0$$

Fig: Stick diagram for 4:1 MUX using NMOS Pass Transistor

25

Multiplexers…

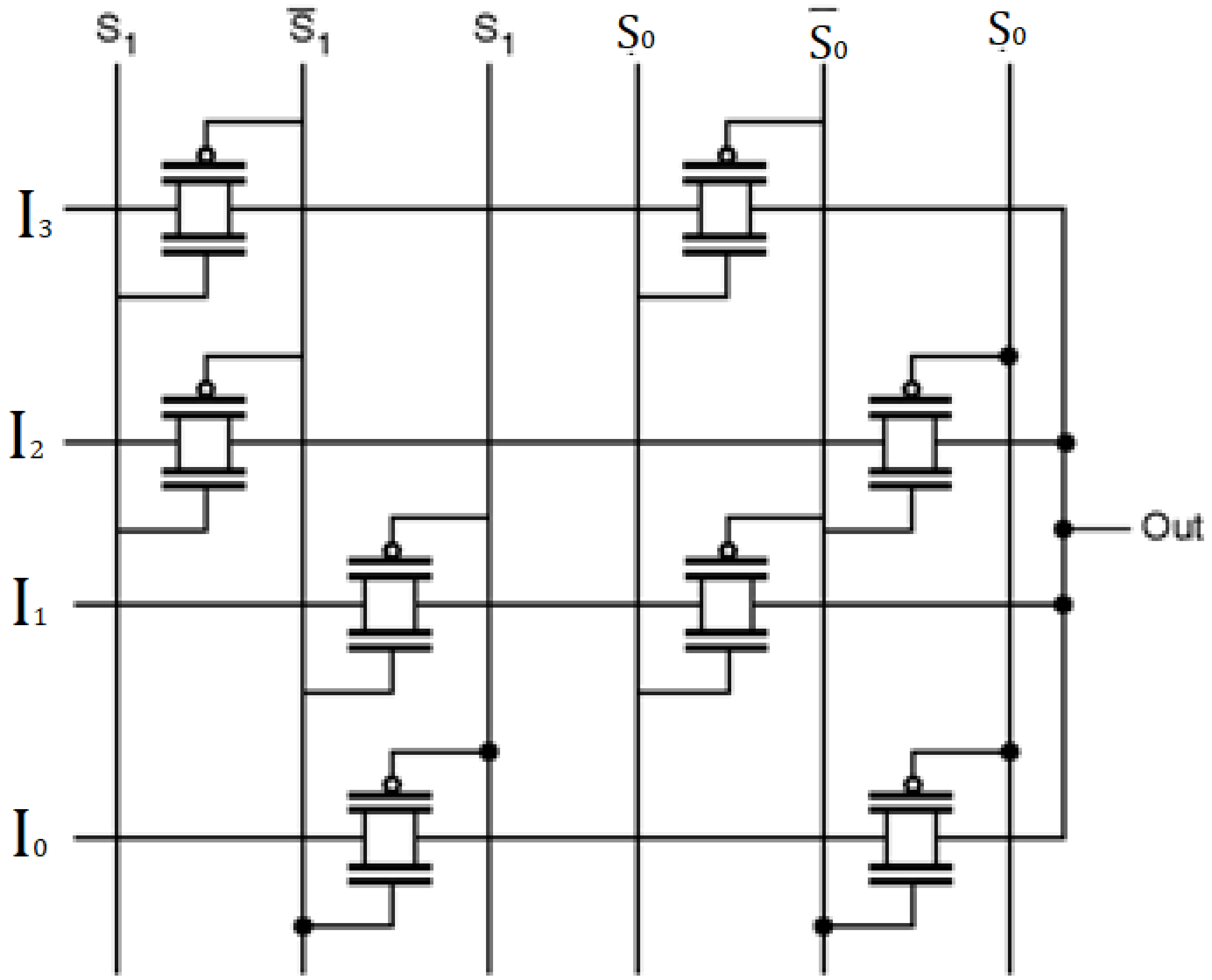


Fig: 4:1 MUX using CMOS Transmission/Pass Gate

Multiplexers…

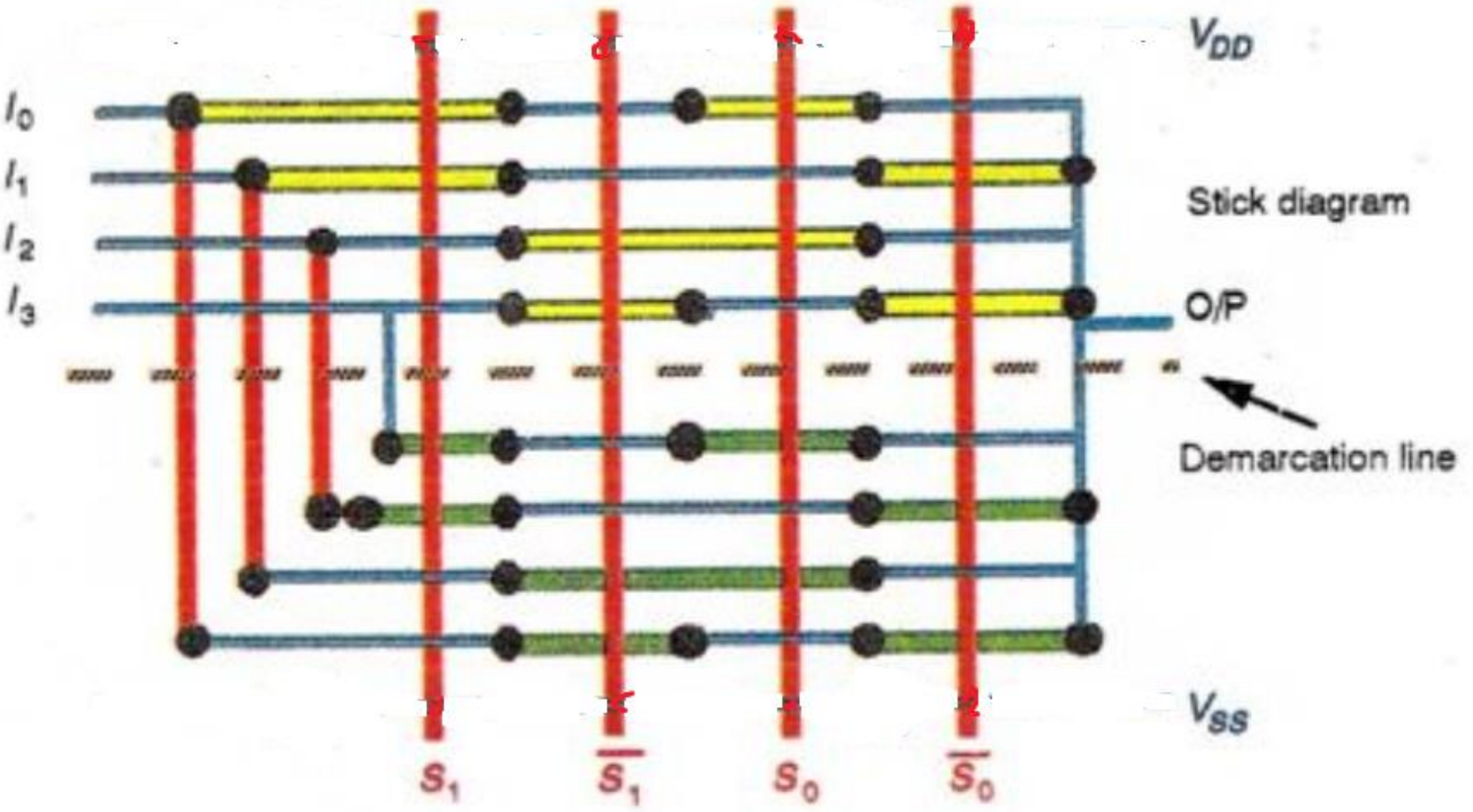
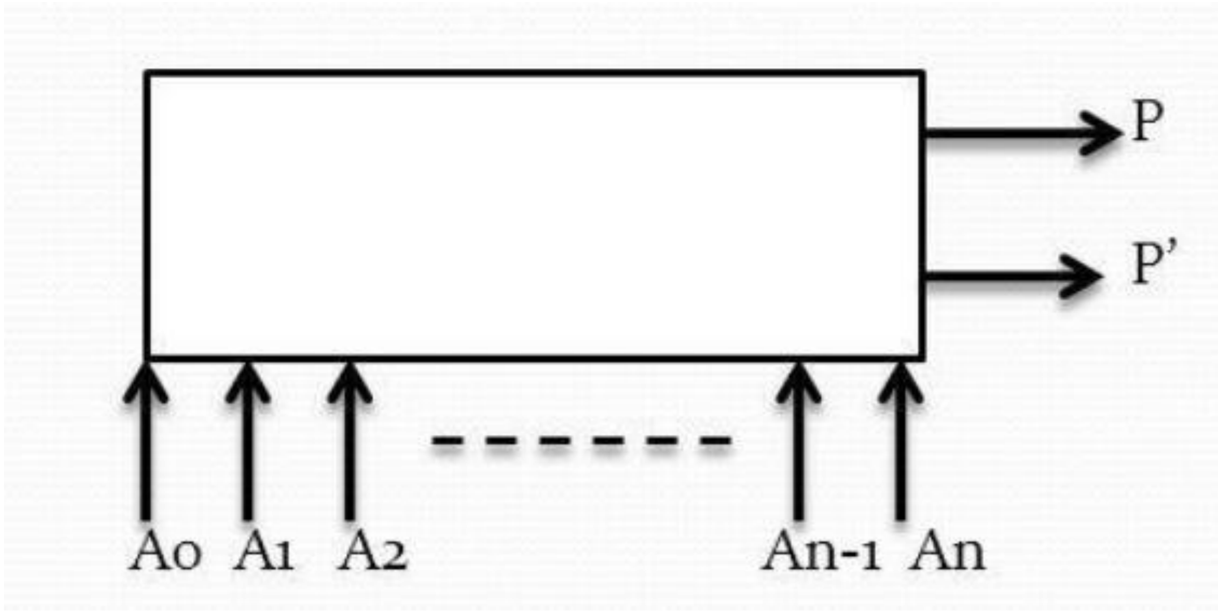


Fig: Stick diagram for 4:1 MUX using CMOS Pass Gate

Parity Generator

- A circuit to be designed to indicate the **parity** of a binary number or word.

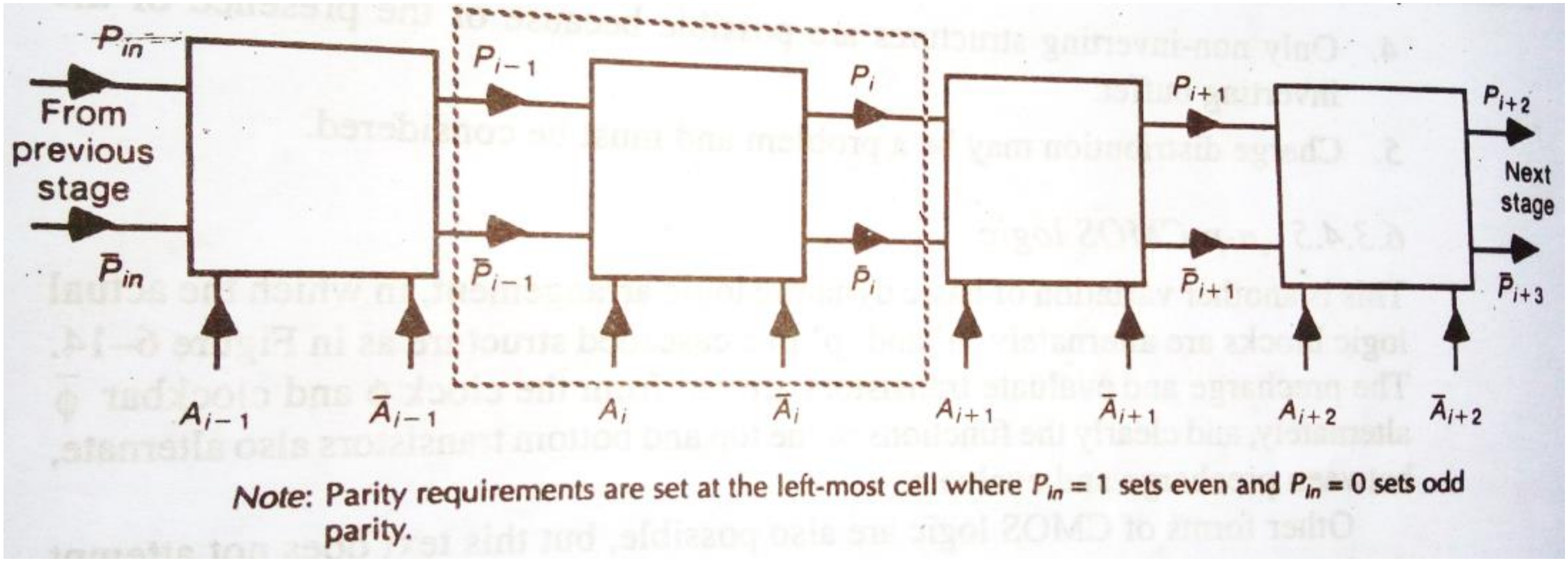


$$p = \begin{cases} 1 & Even\ number\ of\ 1's\ at\ input \\ 0 & Odd\ number\ of\ 1's\ at\ input \end{cases}$$

Fig: Parity Generator Basic Block Diagram

28

Parity Generator…



Note: Parity requirements are set at the left-most cell where $P_{in} = 1$ sets even and $P_{in} = 0$ sets odd parity.

$A_i = 1$ parity is changed, $P_i = \bar{P}_{i-1}$

$A_i = 0$ parity is unchanged, $P_i = P_{i-1}$

Fig: Parity Generator – Structured Design Approach

Parity Generator…

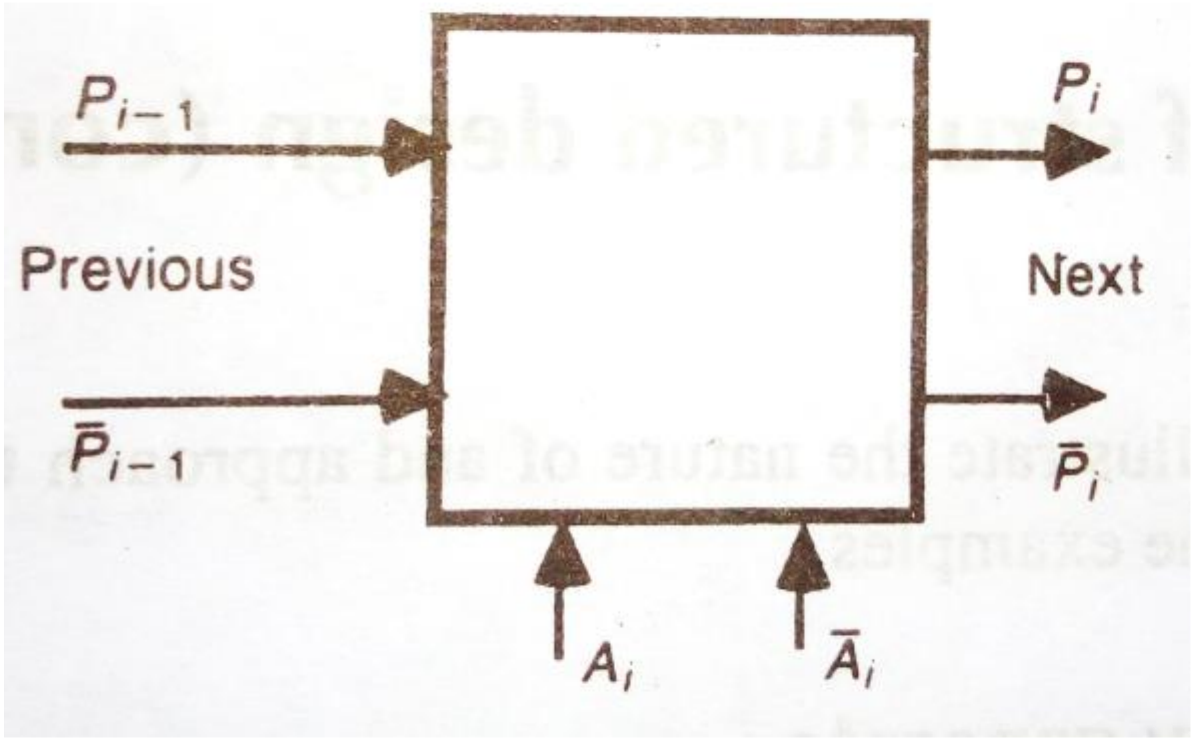


Fig: Parity Generator – Basic One Bit Cell

Parity Generator…

Truth Table of Parity Generator:

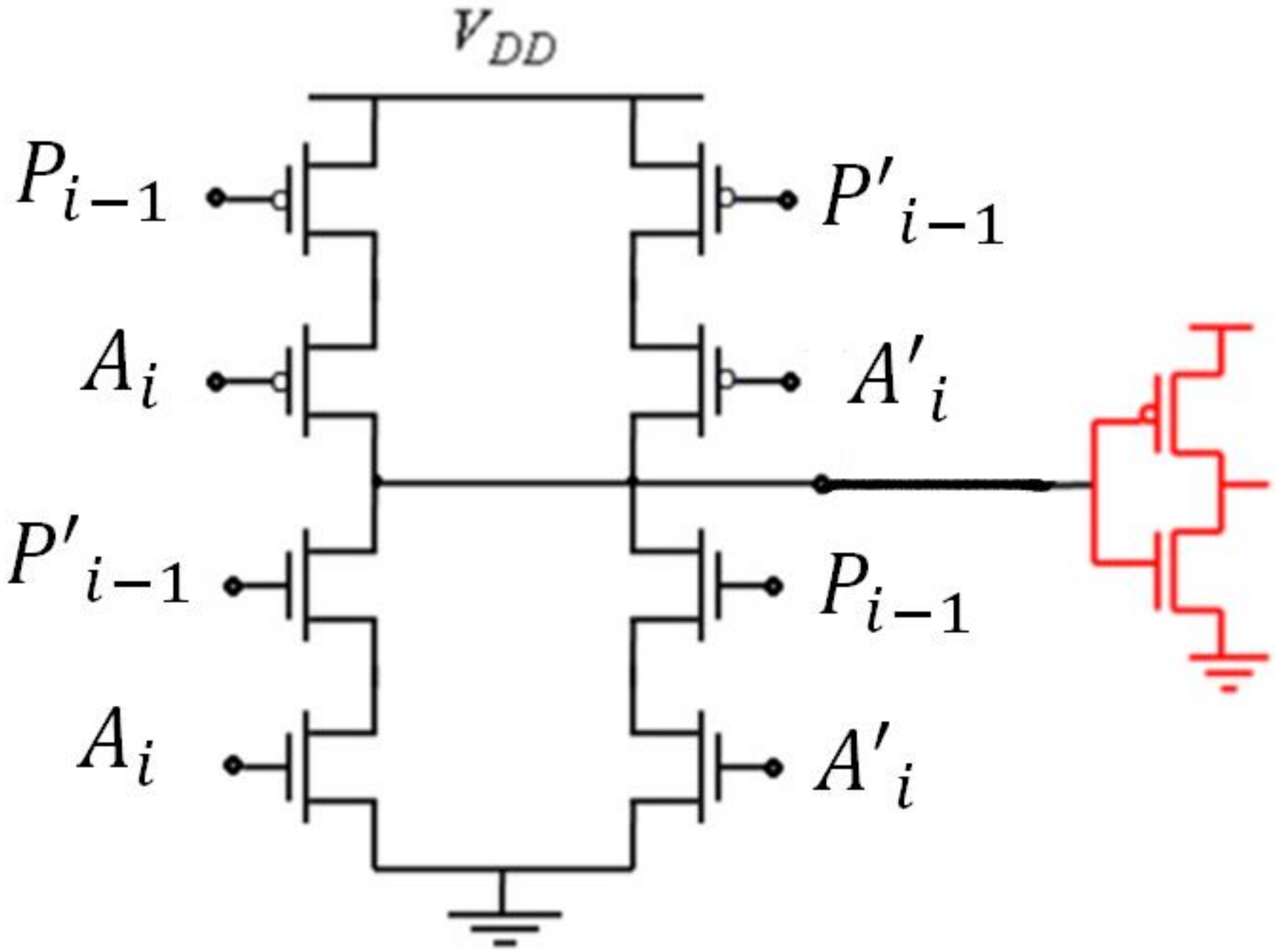
$\boldsymbol{P_{i-1}}$	**A**	$\boldsymbol{P_i}$
0	0	0
0	1	1
1	0	1
1	1	0

$$P_i = P'_{i-1} . A_i + P_{i-1} . A'_i$$

$$i.e. \quad P_i = P_{i-1} \oplus A$$

Parity Generator…

Parity Generator Circuit (CMOS):



32

Parity Generator…

Parity Generator Stick Diagram (CMOS):

Try This Home

33

Acknowledgement

[1] http://bwrcs.eecs.berkeley.edu/Classes/ic541ca/ic541ca_f01/Notes/chapter6.pdf
[2] https://slideplayer.com/slide/13382157/

Thanks

34