Brought to you by ...

Priti Srinivas Sajja

Logic Circuits

Professor

P G Department of Computer Science,
Sardar Patel University, Vidyanagar-388 120, Gujarat, India.

PS01CMCA02
Course Content

Tutorial
Practice Material

Acknowldgement
References

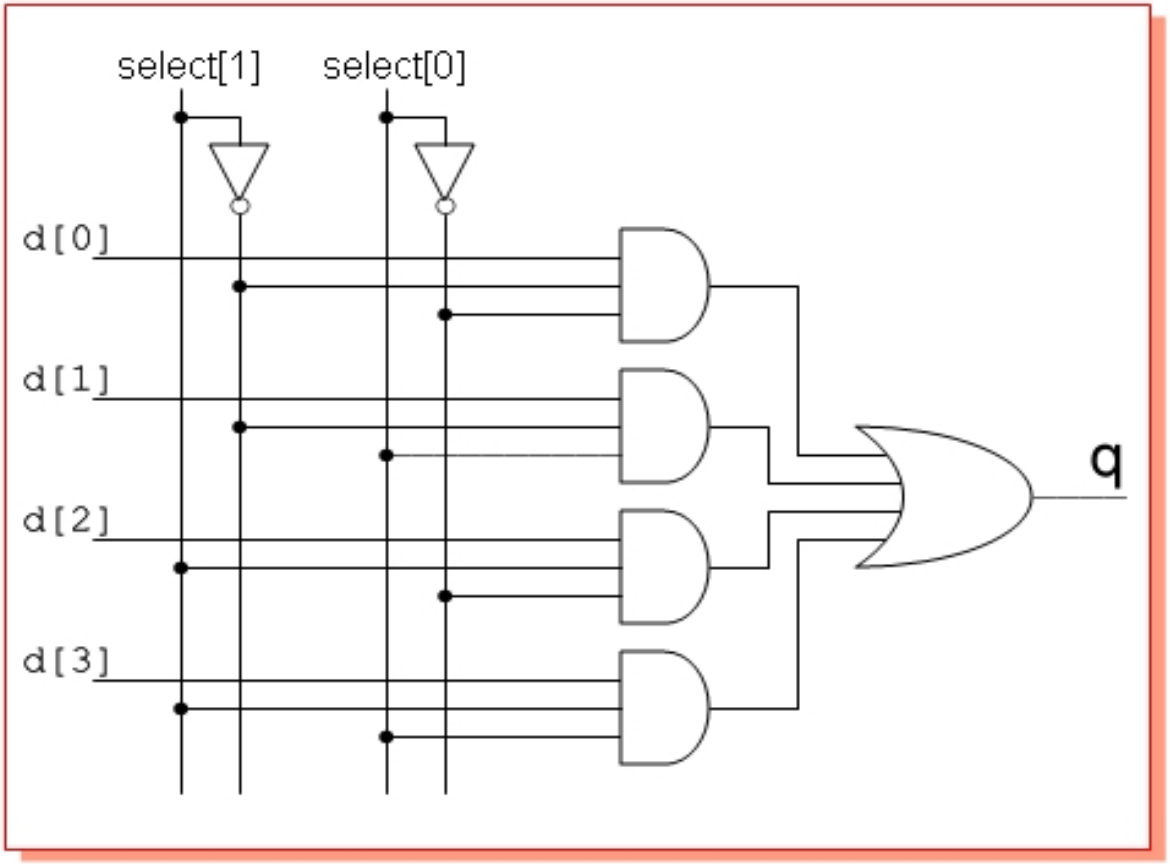
Website
pritisajja.info

Topics

- Integrated Circuits
- Combinational & Arithmetic Circuits
- Latches and Flip-flops
- Registers and Counters

Multiplexer

- Means many into one , also called data selector

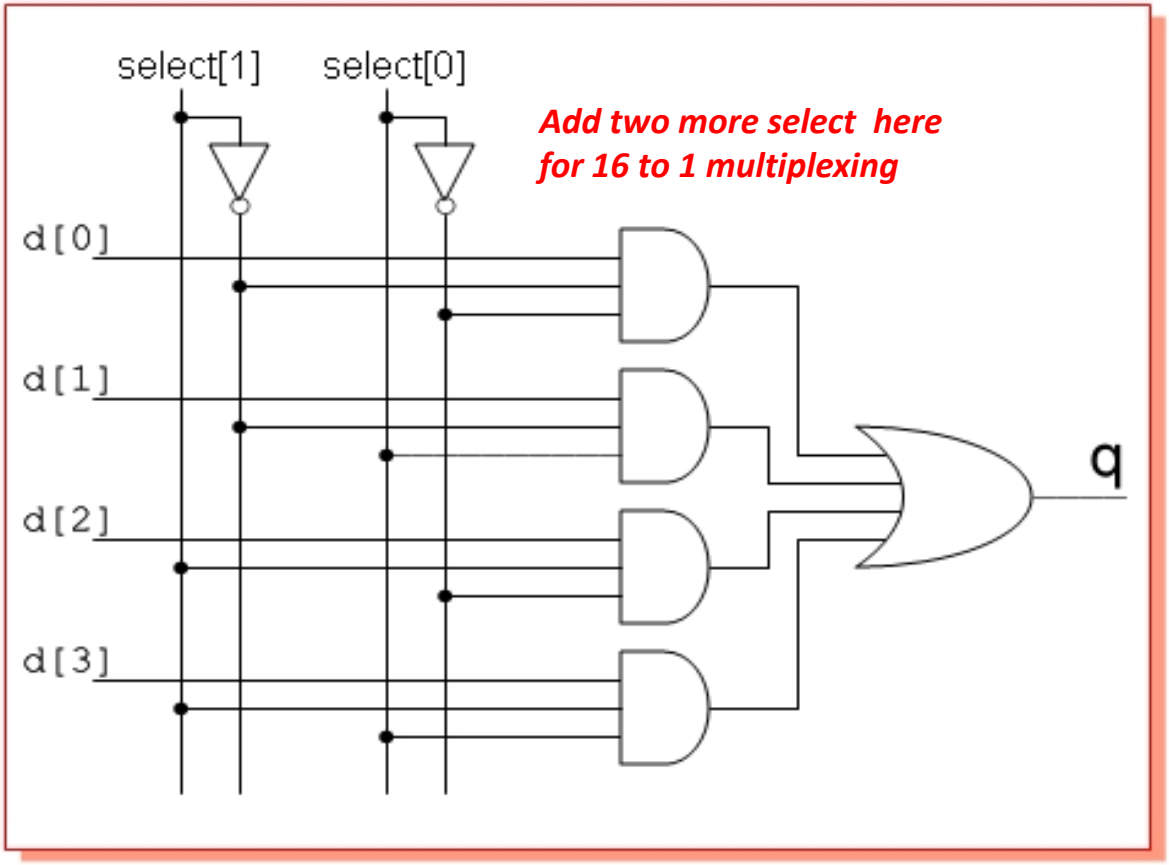


4 to 1 multiplexer circuit

http://electrosofts.com/verilog/mux.html

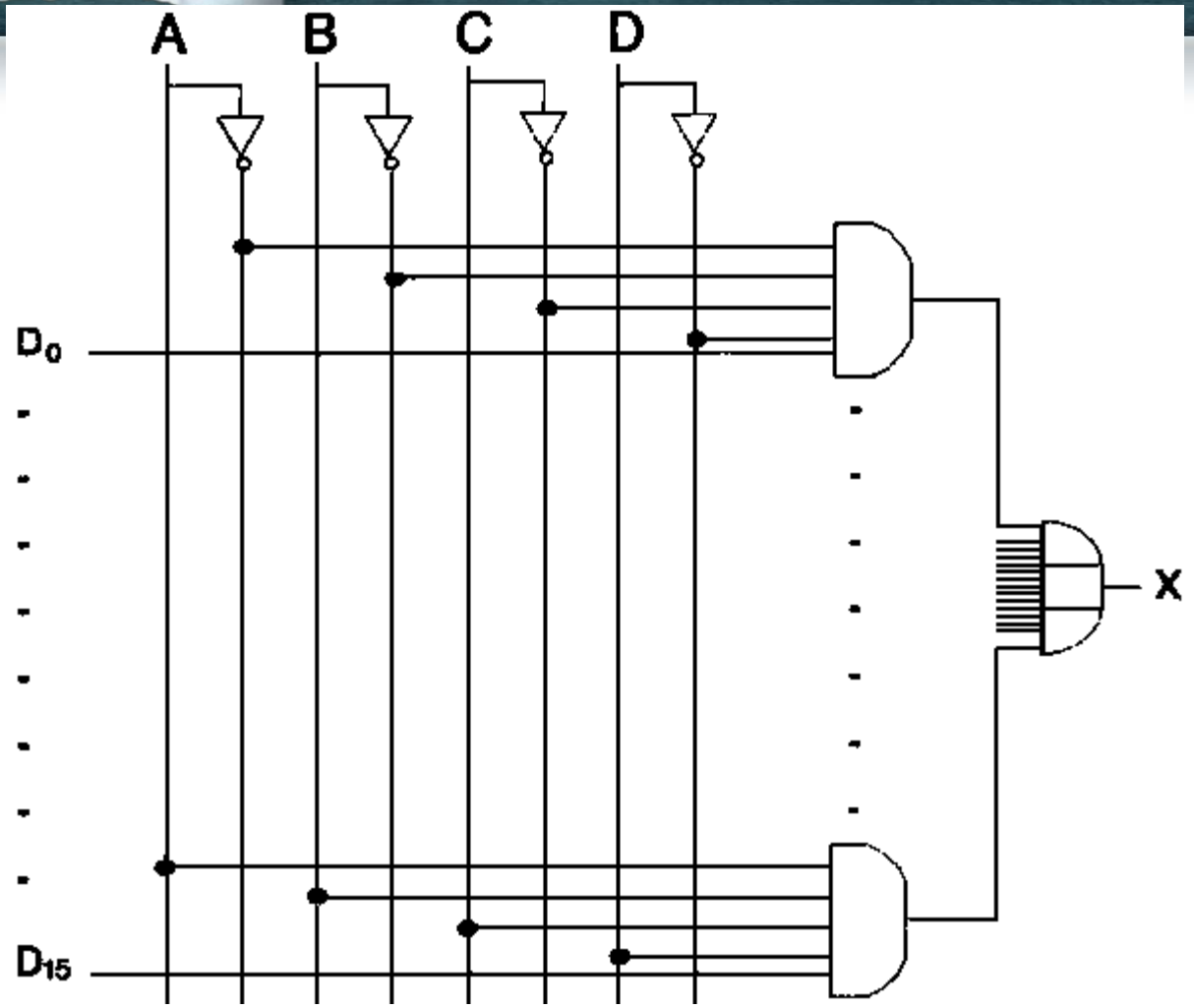
Multiplexer

- Means many into one , also called data selector



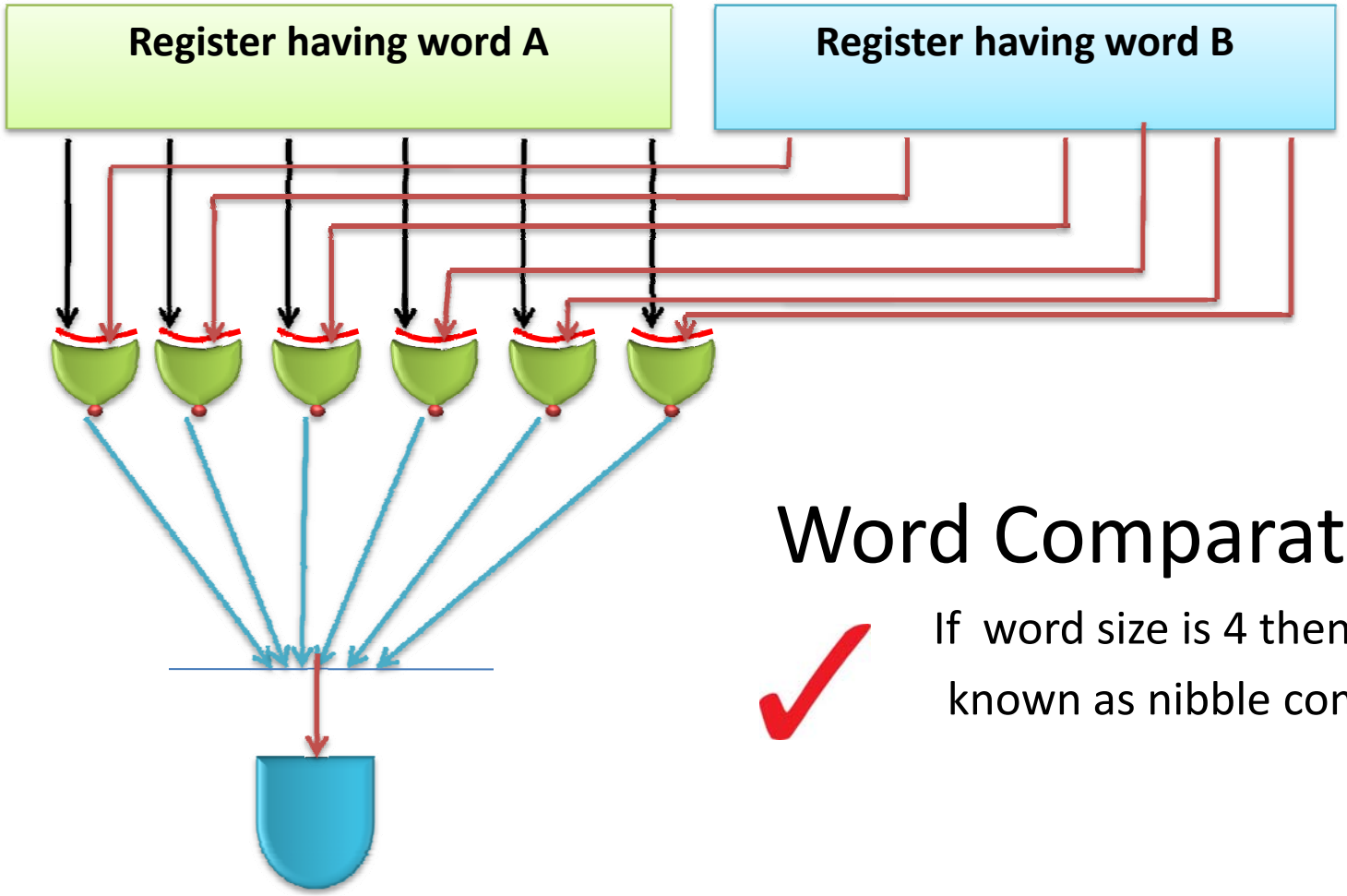
4 to 1 multiplexer circuit

Multiplexer



16 to 1 multiplexer circuit

Word Comparator

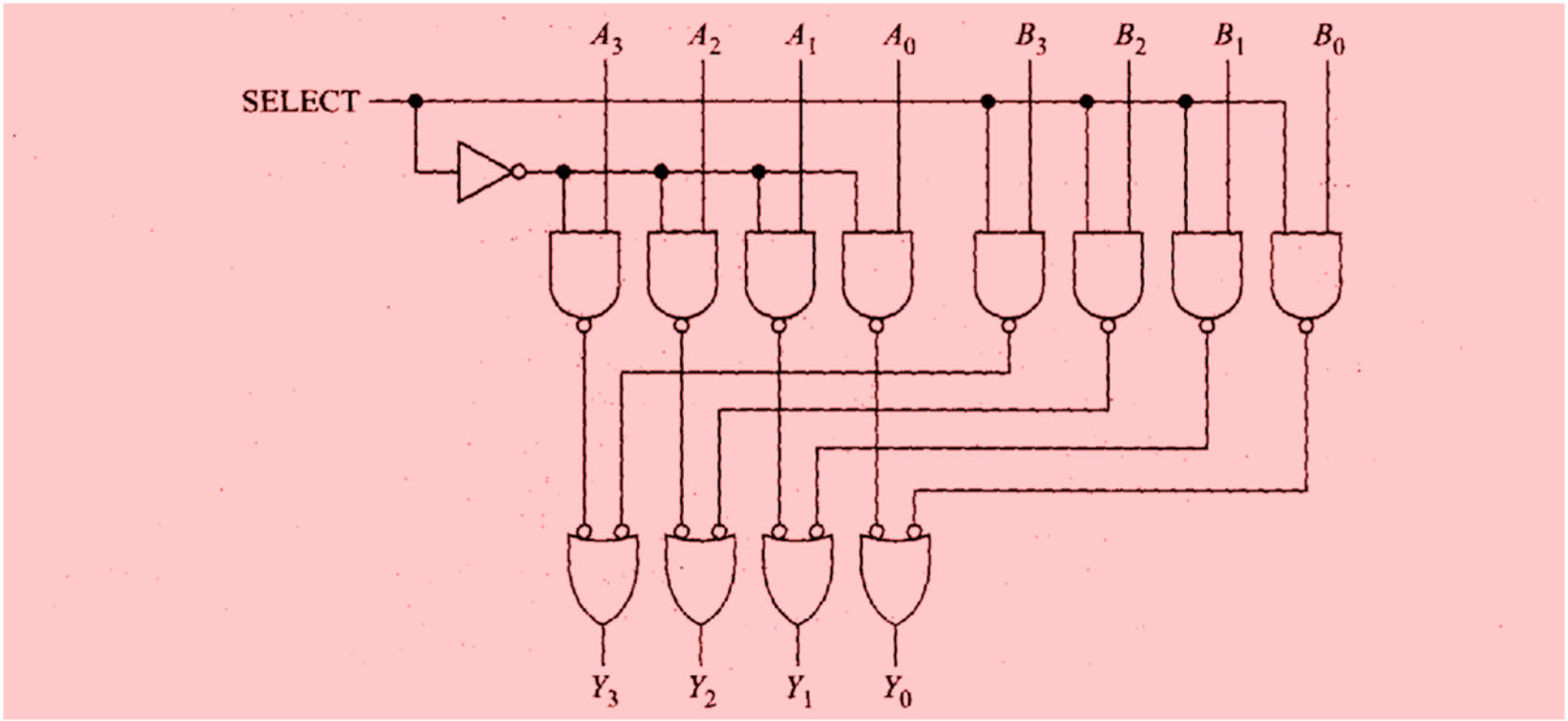


Word Comparator

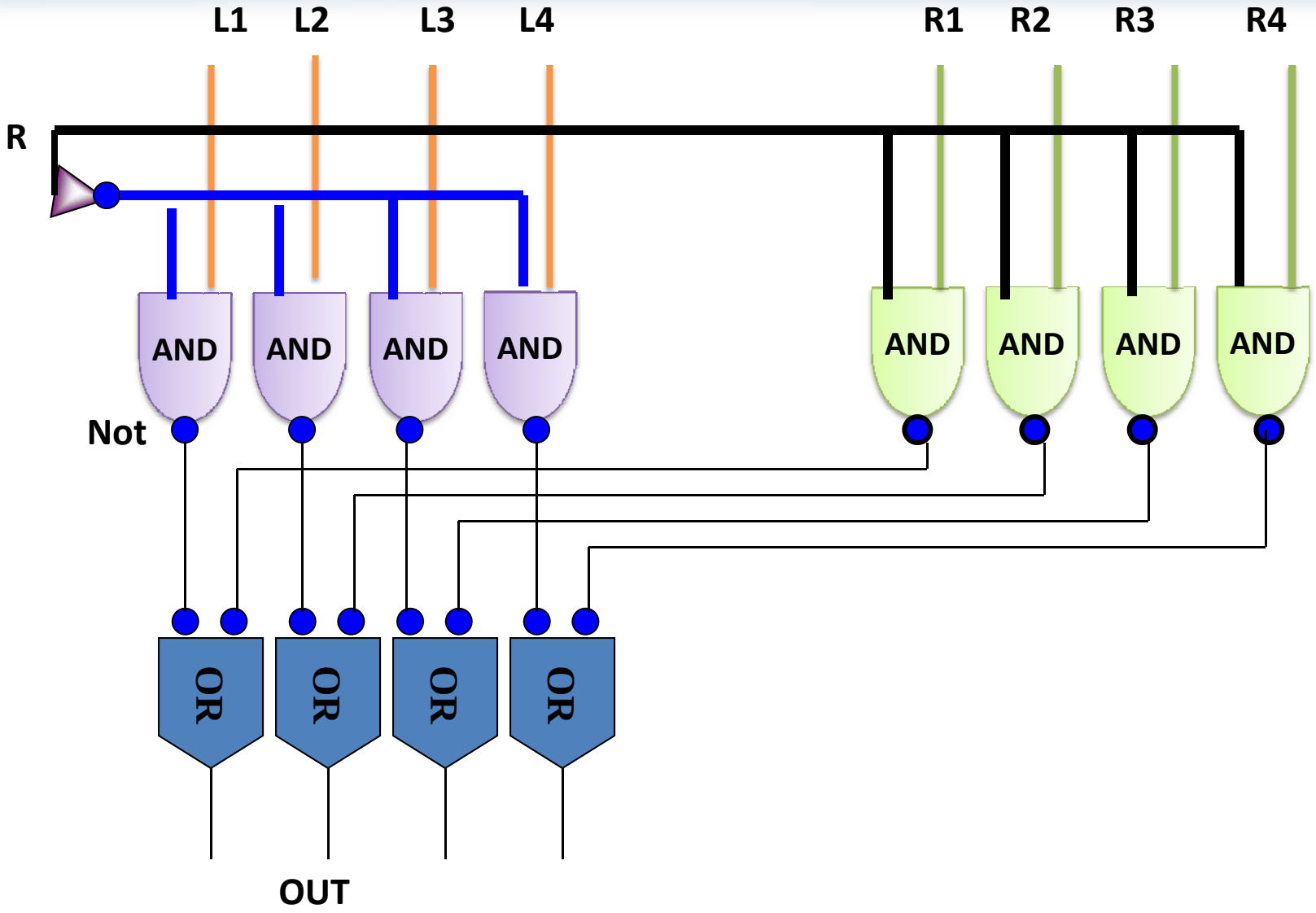
- If word size is 4 then it is also known as nibble comparator.

Nibble Multiplexer

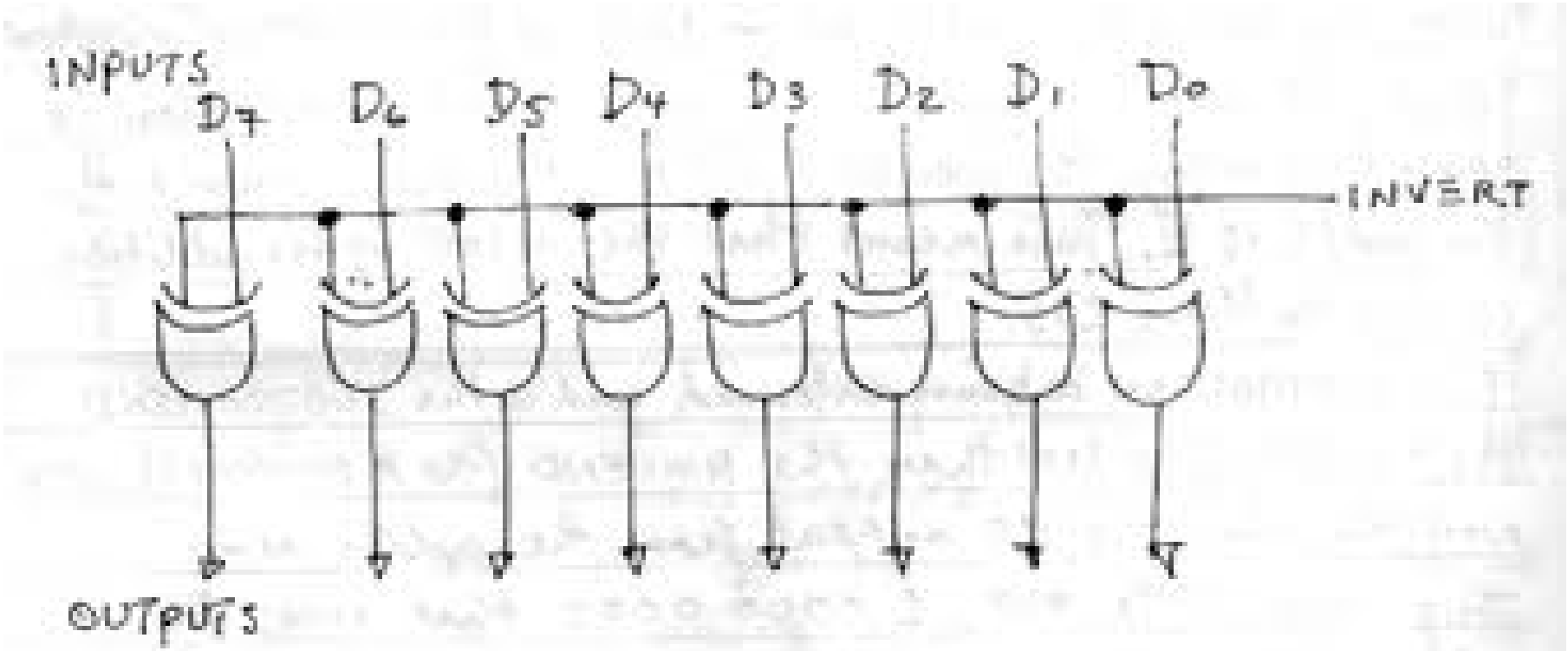
- Find out what this circuit does???



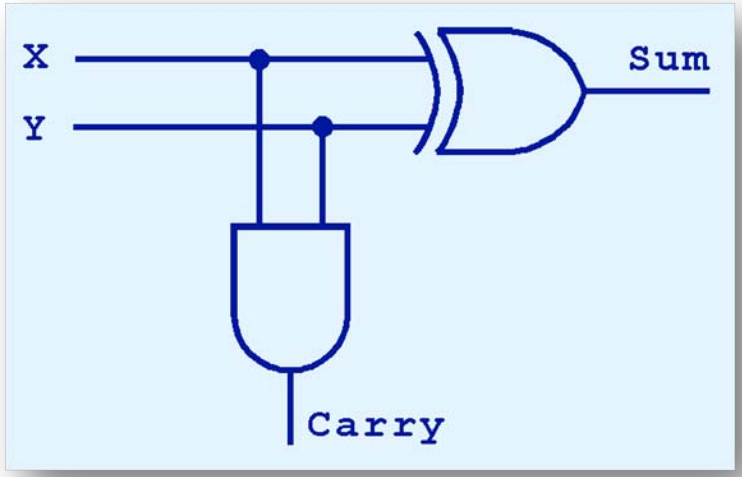
Nibble Multiplexer



Control Inverter

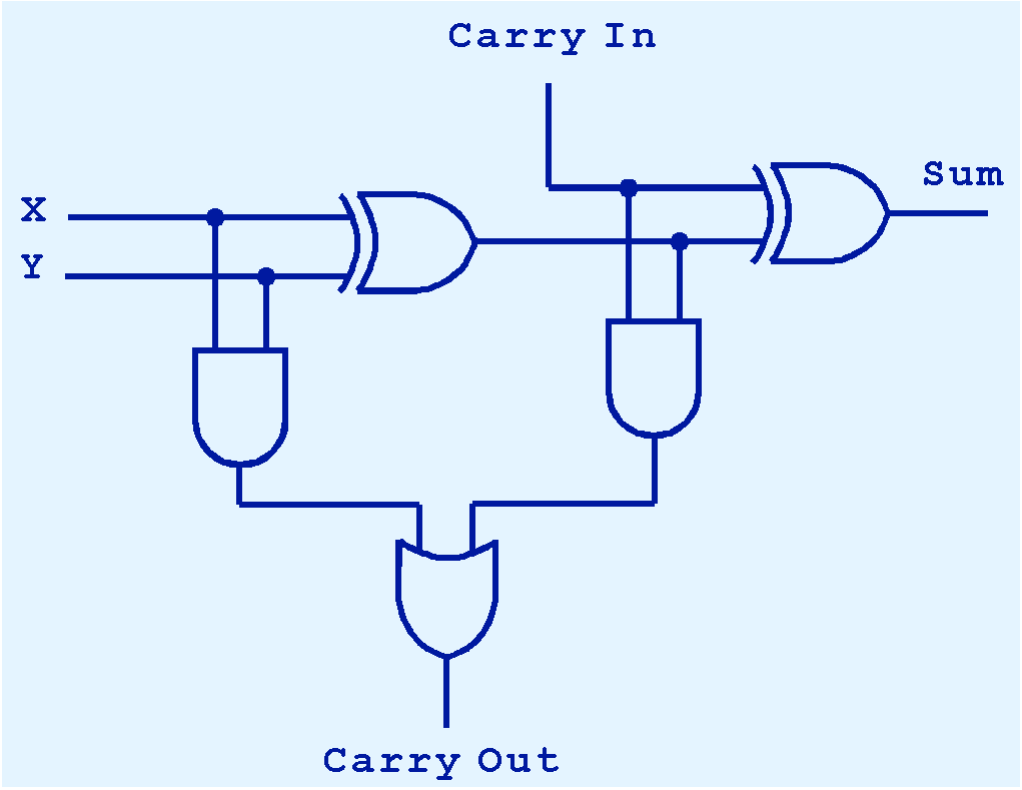


- As we see, the sum can be found using the XOR operation and the carry using the AND operation.



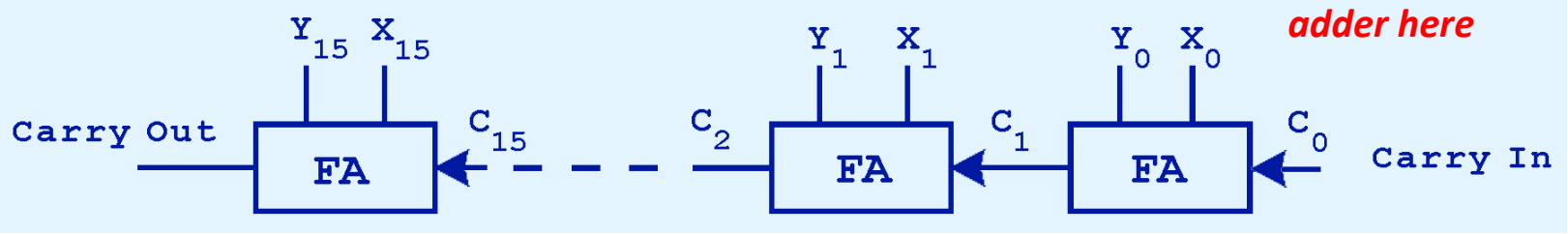
Inputs		Outputs	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- The completed full adder is as follows.



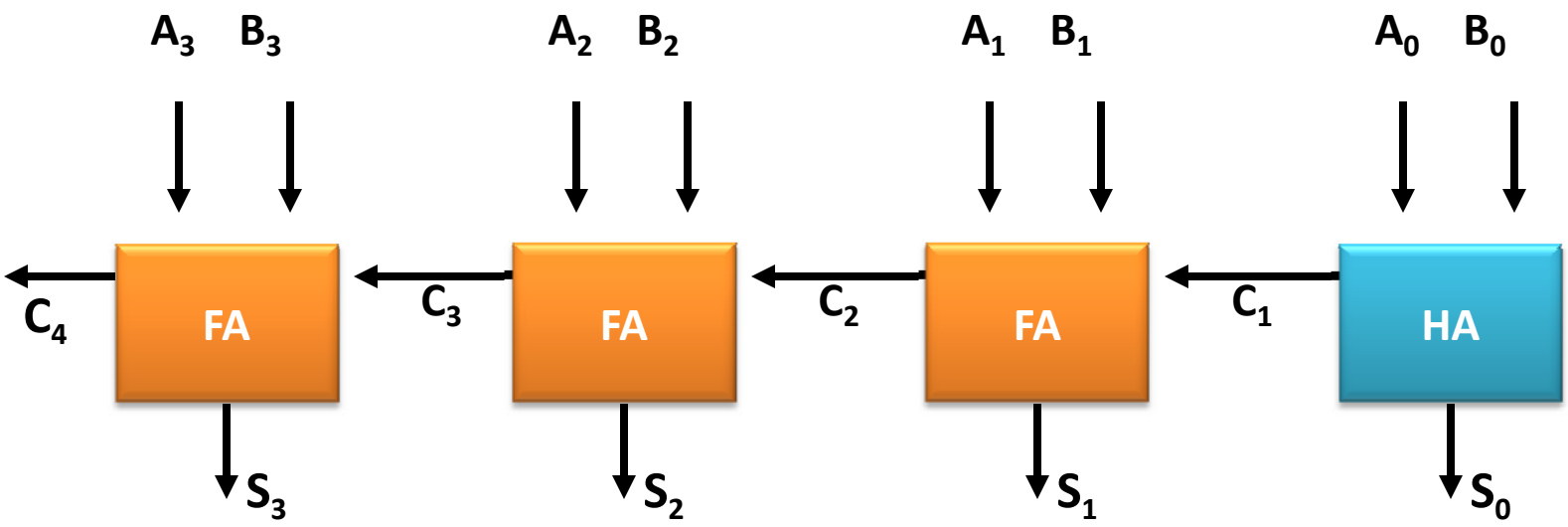
Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- Just as we combined half adders to make a full adder, full adders can connected in series.
- The carry bit “ripples” from one adder to the next; hence, this configuration is called a *ripple-carry adder*.

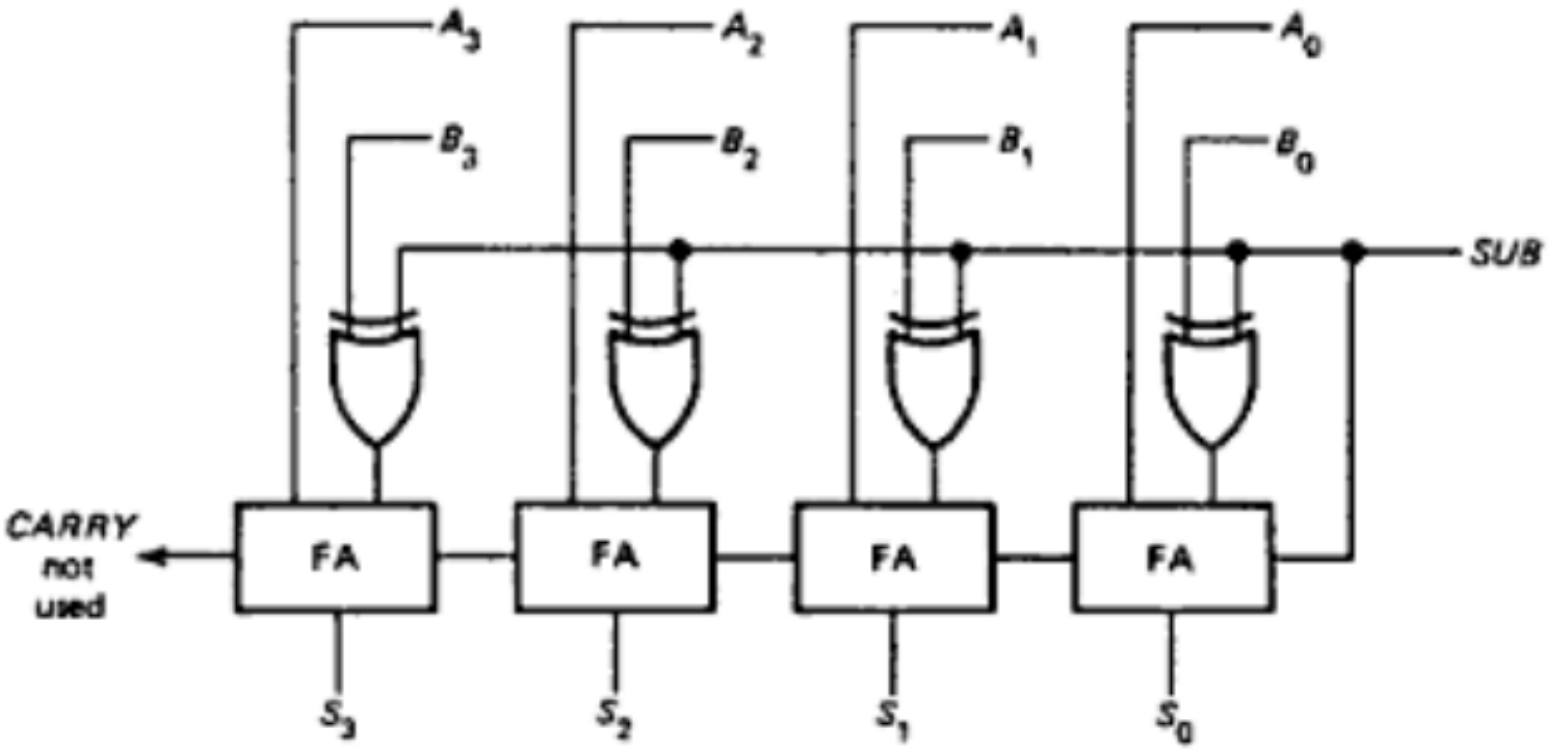


Today’s systems employ more efficient adders.

Binary Adder

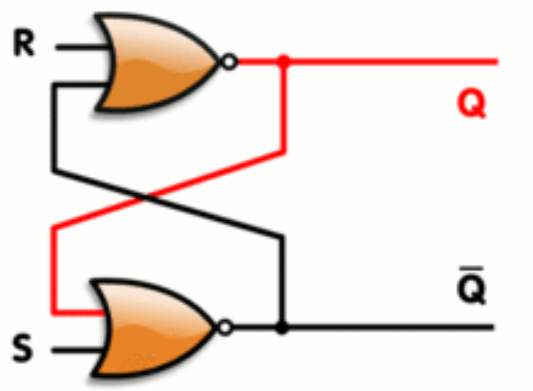


2's complement adder subtractor



Try with A=1001 and B=0010.

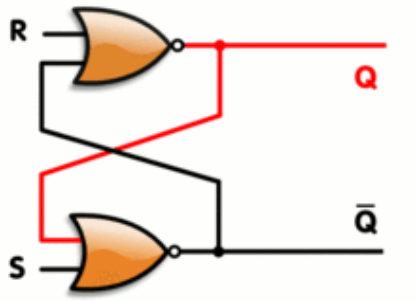
- A flip-flop is a device that with **two stable states**, it remains in one till it is triggered to the other. It is also known as SR latch (set-reset).



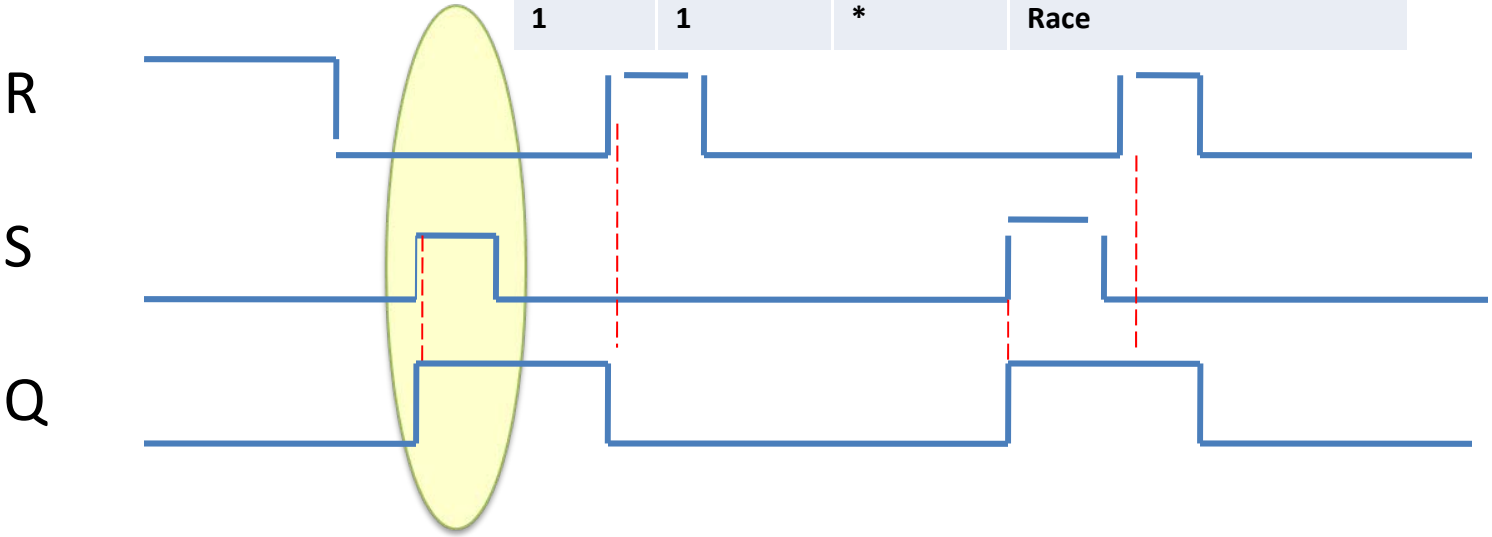
R	S	Q	Comments
0	0	NC	No change
0	1	1	Set
1	0	0	Reset
1	1	*	Race

Latches and Flip-flops

- A latch is a 1 bit memory that remembers previous input values.



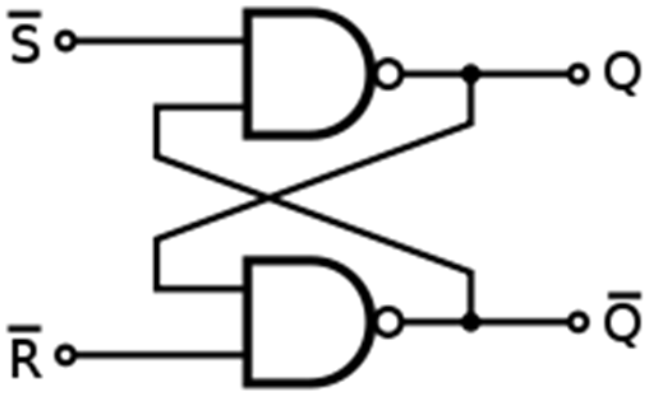
R	S	Q	Comments
0	0	NC	No change
0	1	1	Set
1	0	0	Reset
1	1	*	Race

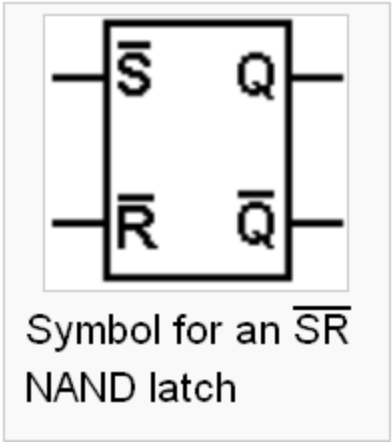


SR latch

- The most fundamental latch
- where S and R stand for *set* and *reset*.
- It can be constructed from a pair of cross-coupled NOR logic gates.
- The stored bit is present on the output marked Q.
- While the S and R inputs are both low, feedback maintains the Q and Q outputs in a constant state, with Q the complement of Q.
- If S (*Set*) is pulsed high while R (*Reset*) is held low, then the Q output is forced high, and stays high when S returns to low; similarly, if R is pulsed high while S is held low, then the Q output is forced low, and stays low when R returns to low.

- This is an **alternate model of the simple SR latch** which is built with NAND (not AND) logic gates.

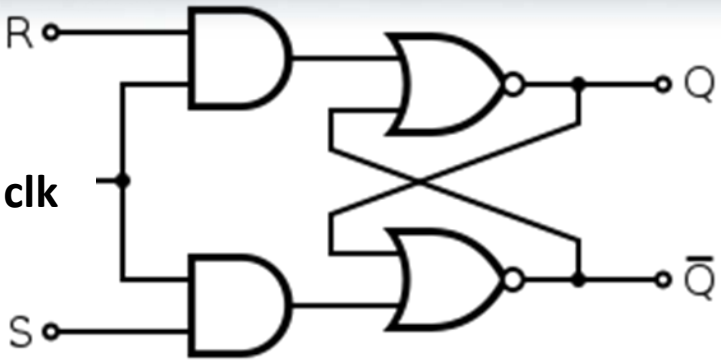




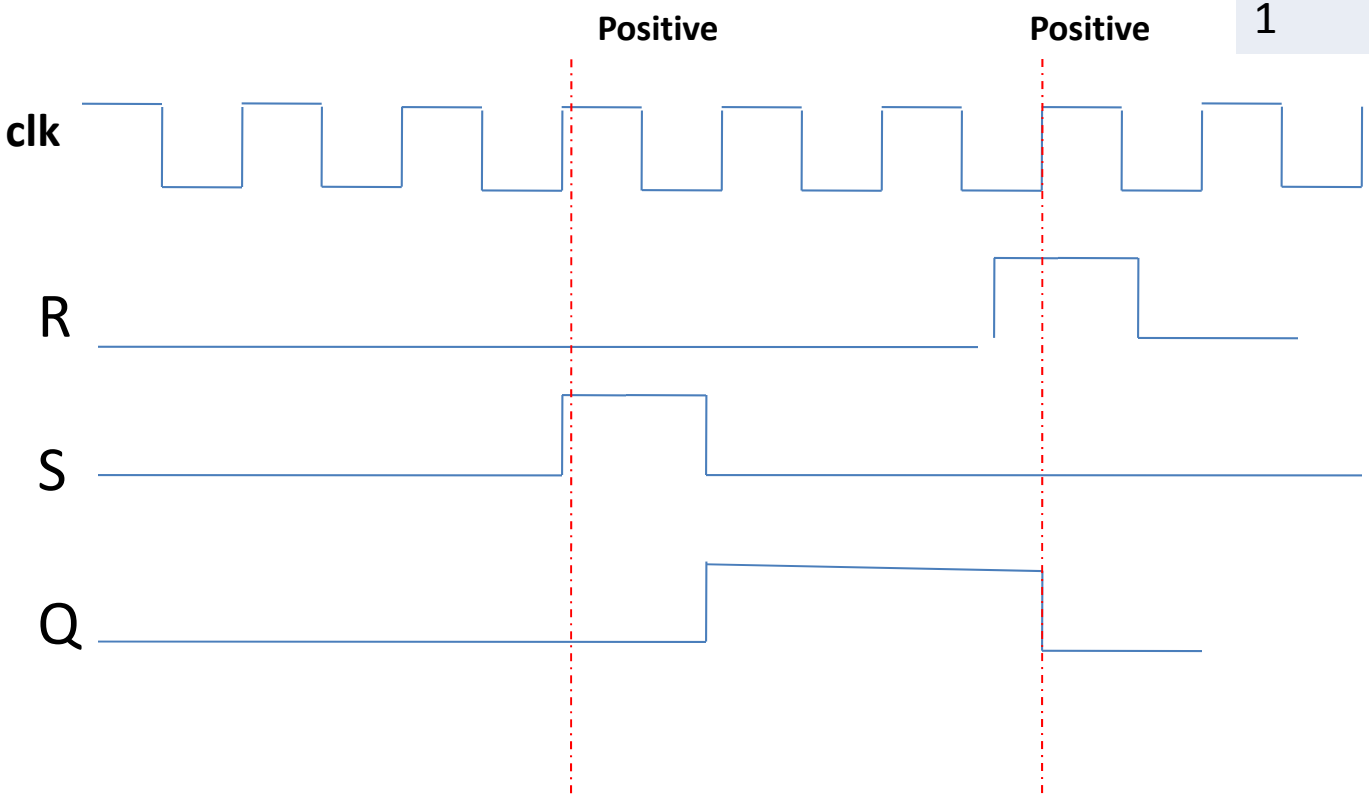
Symbol for an $\overline{SR}$ NAND latch

$\overline{SR}$ latch operation		
$\overline{S}$	$\overline{R}$	Action
0	0	Restricted combination
0	1	Q = 1
1	0	Q = 0
1	1	No Change

Clocked Latch

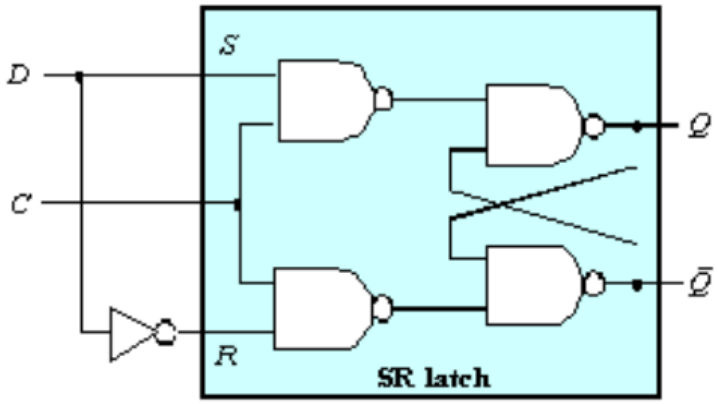


CLK	R	S	Q
0	0	0	**NC**
0	0	1	**NC**
0	1	0	**NC**
0	1	1	**NC**
1	0	0	NC
1	0	1	1
1	1	0	0
1	1	1	*

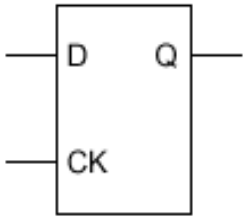


**** Race,***
never used

Clocked D Latch



NAND implementation



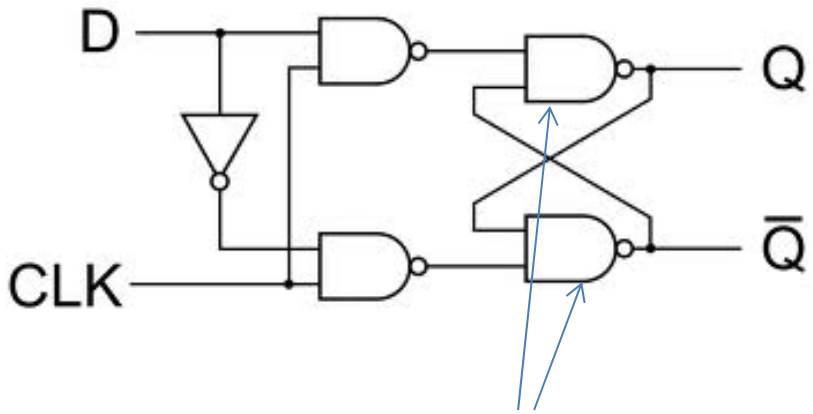
CLK	D	Q
0	X	NC
1	0	0
1	1	1

- Most widely used.
- In the D latch, when the CLK input is logic 1, the Q output will always reflect the logic level present at the D input, no matter how that changes.
- When the CLK input falls to logic 0, the last state of the D input is trapped and held in the latch, for use by whatever other circuits may need this signal.
- Since the RS flip flop is susceptible to race condition, its design is modified as D latch

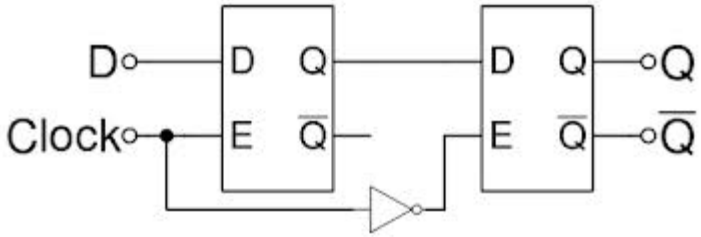
Edge Triggered D Flipflop

An edge-triggered flip-flop changes states either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock pulse on the control input. The three basic types are introduced here: S-R, J-K and D.

*Such gates are sensitive to inputs only near **edge** of clock signal (not while steady)*



The NAND gate can be changed to bubbled OR gate.

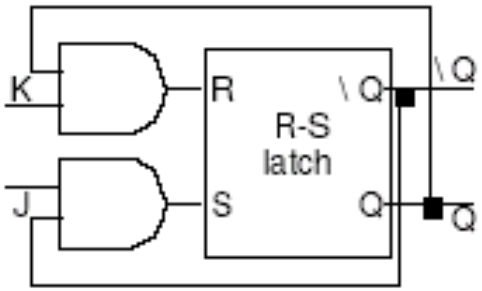


CLK	D	Q
0	X	NC
1	X	NC
Down (falling edge)	X	NC
Up (rising edge)	0	0
Up	1	1

Edge triggered jk flip-flop

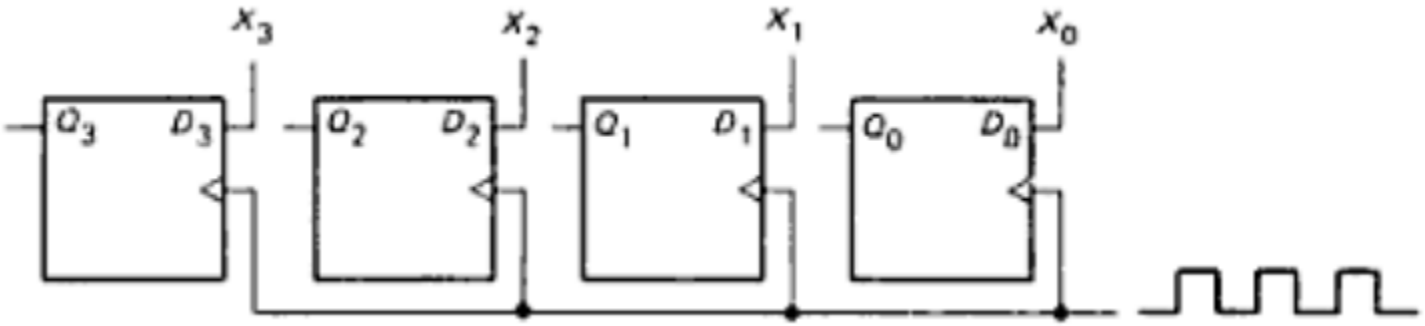
- Feeding the output Q (Q') back to the inputs using AND gate whose the other input is the J(K) input
- J is called SET input while K is the RESET input
- The two AND gates cannot produce the output 1 simultaneously (provided that the outputs Q and Q' are in proper state) and ensure that the R and S input do not becomes 1 simultaneously
- Characteristic equation: $Q^+ = JQ' + K'Q$

J	K	Q+	
0	0	Q	hold
0	1	0	reset
1	0	1	set
1	1	Q'	toggle



Registers

- Buffer Register



Registers

Example 8.1 : *Determine the number of flip-flops needed to construct a register capable of storing.*

i) A 6-bit binary number

ii) Decimal numbers upto 32

iii) Hexadecimal numbers upto F

iv) Octal numbers upto 10

Solution :

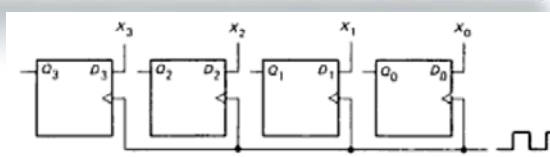
i) A 6-bit binary number requires register with 6 flip-flops.

ii) $(32)_{10} = (100000)_2$. The number of bits required to represent 32 in binary are six, therefore, 6 flip-flops are needed to construct a register capable of storing 32 decimal.

iii) $(F)_{16} = (1111)_2$. The number of bits required to represent $(F)_{16}$ in binary are four, therefore four flip-flops are needed to construct a register capable of storing $(F)_{16}$.

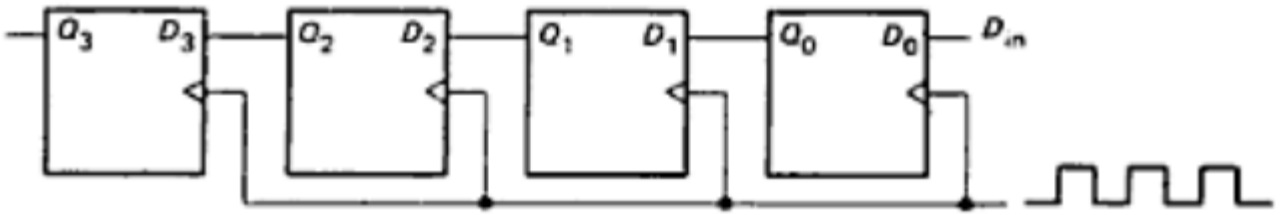
iv) $(10)_8 = (1000)_2$. The number of bits required to represent $(10)_8$ in binary are four, therefore, four flip-flop are needed to construct a register capable of storing $(10)_8$.

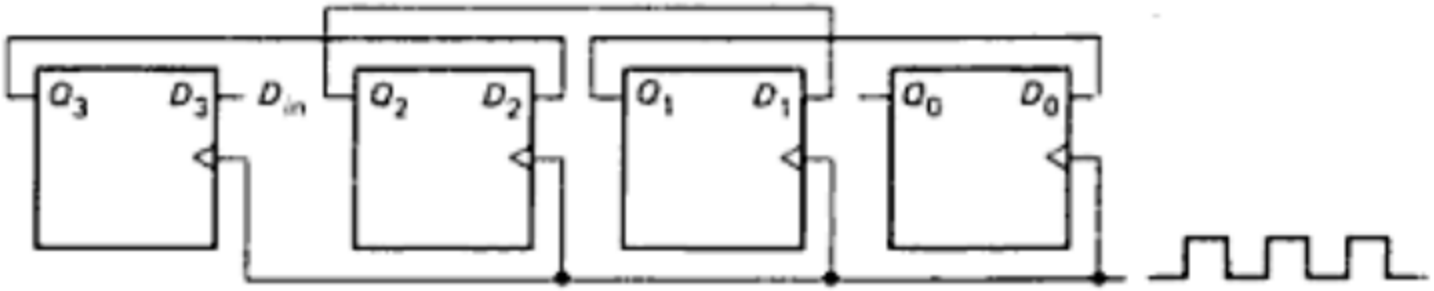
Registers



Simple buffer register

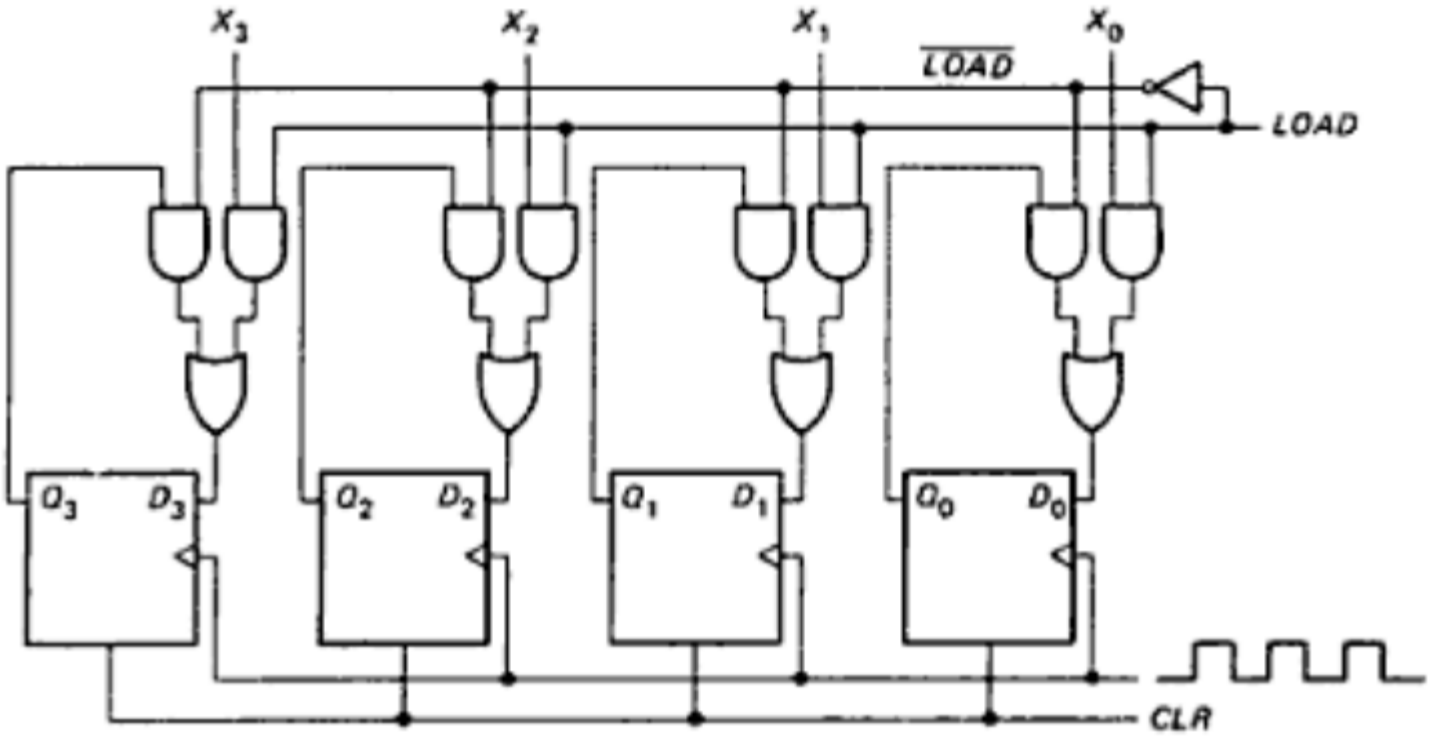
- Shift Left and Shift Right Registers





Registers

- Controlled Buffer Register



Registers

- **Controlled Shift Register**

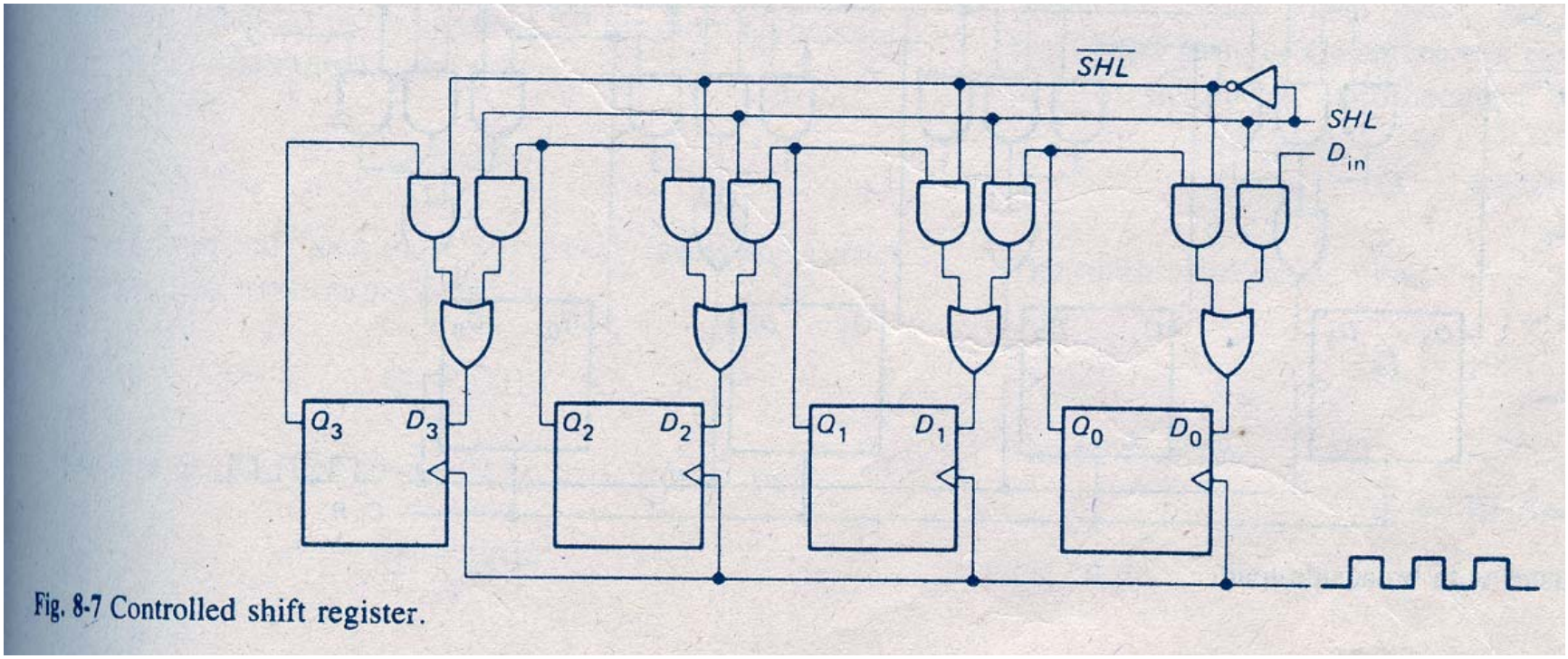
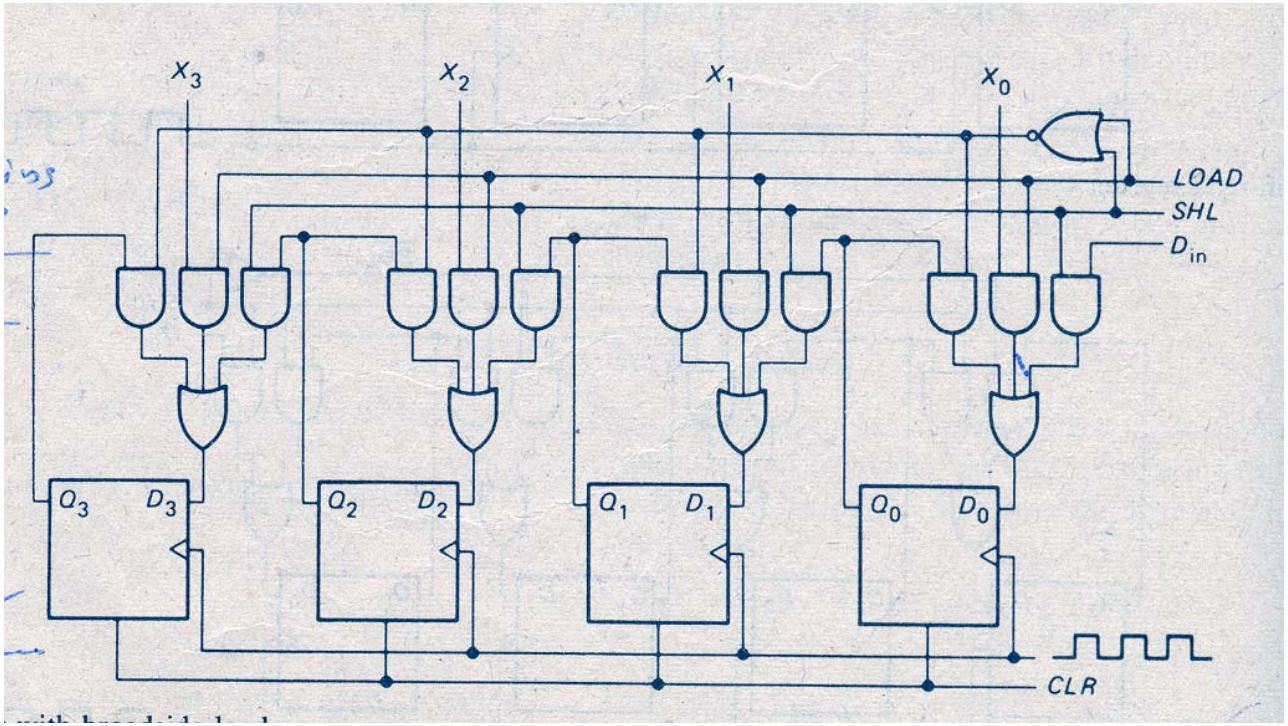


Fig. 8-7 Controlled shift register.

- It has controlled input that determines what it does on the next clock pulse.
- SHL is control signal to shift left.
- When **SHL is high**, Din sets up the right flip flop, Q_0 sets the second flip flop and so on.
- **Each positive clock edge shifts the stored bits one position to the left**.

Registers

- Controlled Shift Register **Controlled Shift Register with Broadside Load**



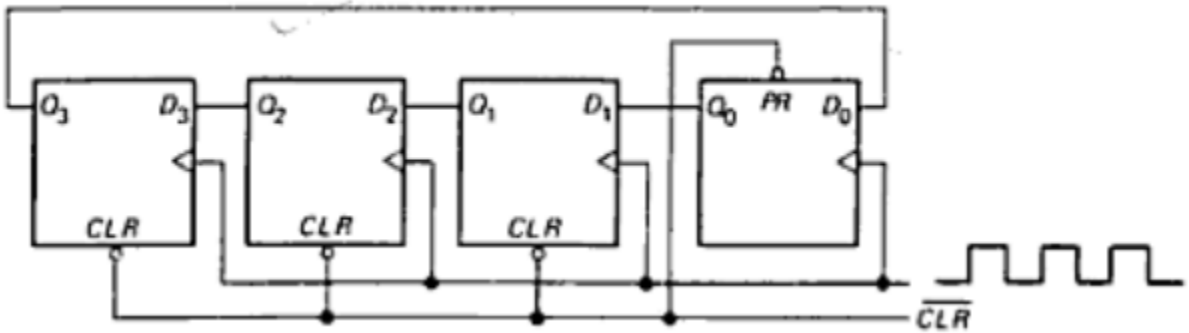
- This circuit can **load X bits directly into the flip flops**.
- This kind of loading is known as **parallel loading** or broadside loading.
- It takes **only one clock pulse to store a digital word**.
- If LOAD and SHL is low, the output of nor gate is high and flip flop returns to their data input. That is the register is inactive, when LOAD and SHL are low, and the content is stored indefinitely.
- By adding more flip flops, one can build a controlled shift register of any length.

Counters

- A *counter* is a sequential circuit that goes through a predetermined sequence of states upon the application of clock pulses.
- Counters are categorized as:
  - Synchronous Counter:
    All Flip-flops (FF) receive the common clock pulse, and the change of state is determined from the present state.
  - Ripple Counters:
    The FF output transition serves as a source for triggering other FFs.
    No common clock.

Counters

Ring Counter

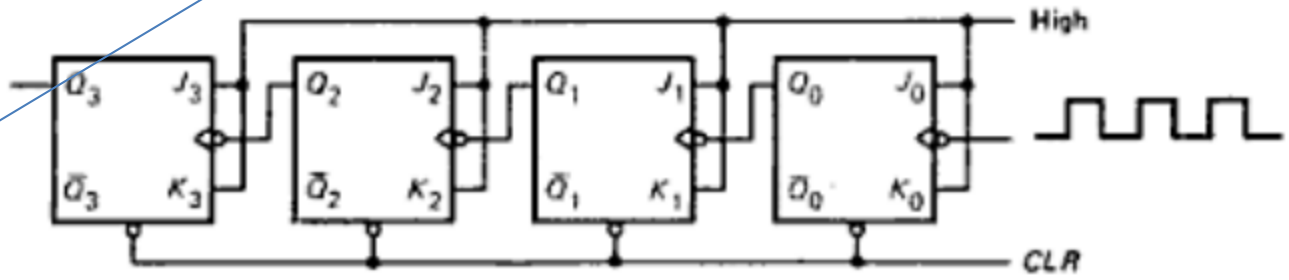


- Ring counter is build with D flip flops, where each word is having only one high bit.
- It resembles a shift left register.
- When CLR goes low then back to high, the initial output word is **Q=0001**
- The first positive clock edge shifts the most significant bit(MSB) into the Last Significant position, and the output word becomes **Q=0010**
- The second positive clock edge causes another rotate left and output is **Q=0100**
- After third positive clock edge **Q=1000** and
- After forth positive clock edge **Q=0001**

- To activate one of the several devices, or sequence of operation, ring counter is used.

Ripple Counters

- **Ripple counter** is build with 4 J K flip flops.(carry moves on the flip flop like a water ripple)
- If CLR goes low, the register becomes
  - Q=0000
  - on next pulse Q=0001
  - ….
  - On tenth pulse
  - Q=1010 and so on…



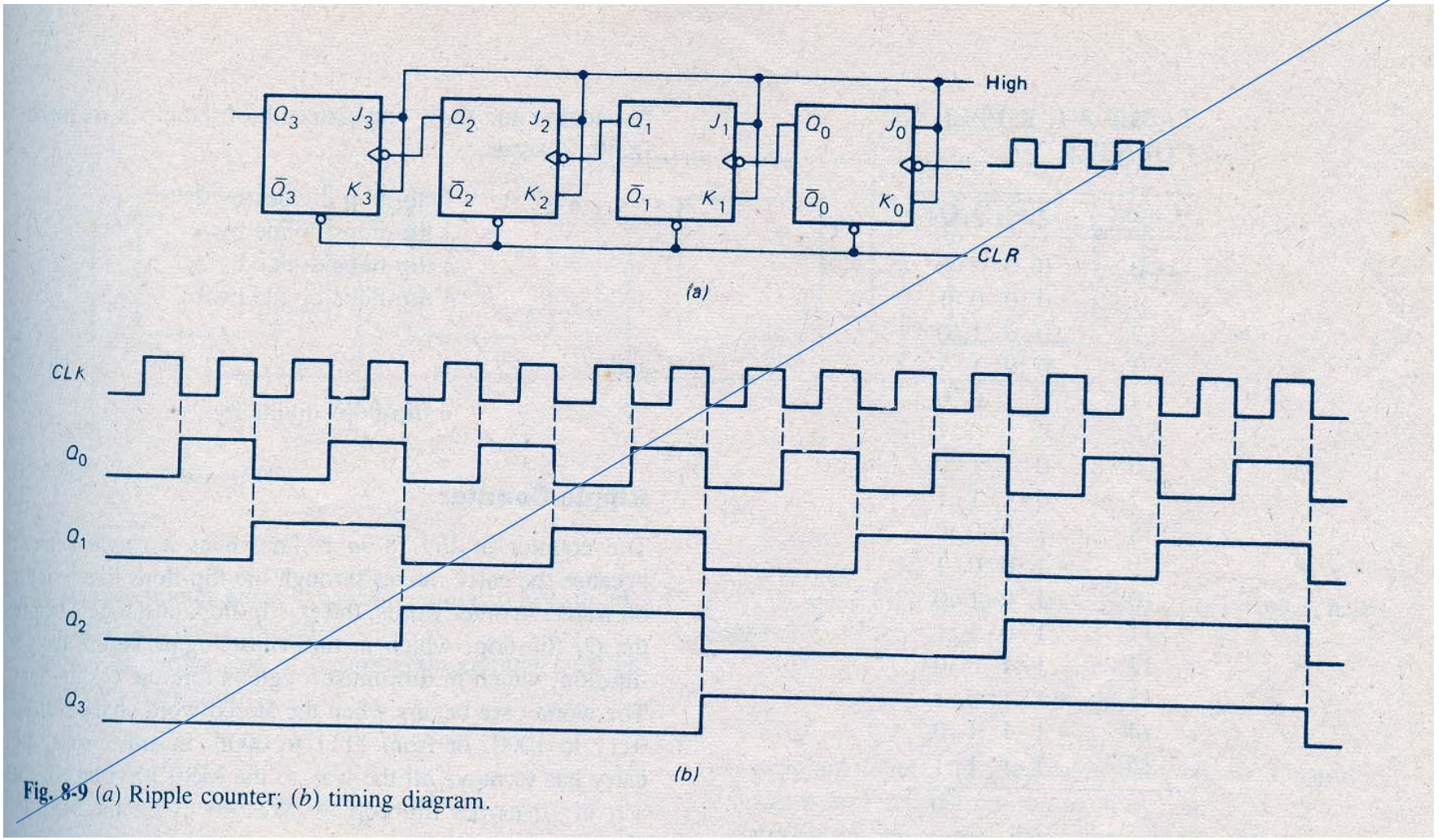


Fig. 8-9 (*a*) Ripple counter; (*b*) timing diagram.

Acknowledgement

- *"Intelligent technologies for web applications",* CRC Press (Taylor & Francis Group), Boka Raton, FL, USA (2012)
- AP Godse and DA Godse, Digital principles and systems design, Technical publications, Pune
- clker.com
- http://electrosofts.com/verilog/mux.html
- http://www.briarcliff.edu/departments/cis/CSCI280/Index-2010c.htm
- Malvino A. P.: Digital Computer Electronics,2nd Edition, Tata McGraw, Hill Pub. Co. Ltd.,New Delhi, 1990.
- nzdl.org