Laboratory work Nr 3

Registers

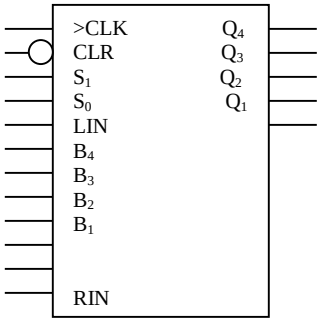
Purpose

The purpose of this laboratory work is to design and implement multifunctional registers..

INTRODUCTION

Registers are often used to store and to process binary information. They contain flip-flops, controlled by combinational circuits.

Logic Symbol for 74-194 register :



Register can perform the following functions:

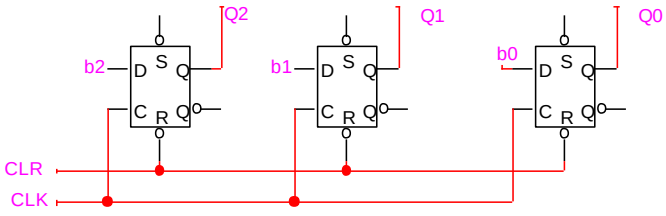
1. hold
2. shift right
3. shift left
4. load

This register contain 4 D flip-flops with a common clock signal CLK
CLR-is an asynchronous clear input, active law.
S0,S1 – control inputs
LIN – (left in) – serial input for left shifts
RIN – (right in) – serial input for right shift
A,B,C,D – informational inputs
QA,QB,QC,QD – outputs of the register, its current stat.

The register can work as a serial-in, serial-out, parallel-in, parallel-out device.
For example the structure for a serial-in, serial-out, shift register.

Parallel registers

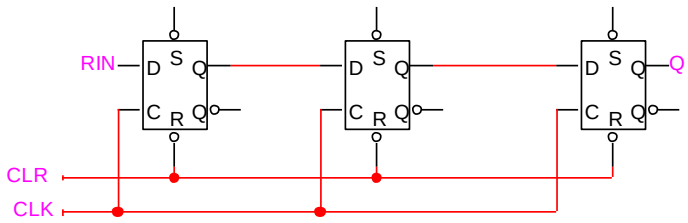
Binary words can be writing or reading in such registers at the same time. The main function of them is to store binary information.



Serial registers

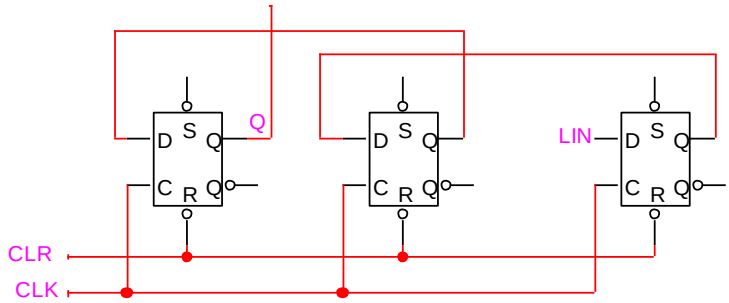
In these registers binary words can be written only through serial inputs LIN or RIN. The binary word can be read through serial output Q. Such registers are used to shift binary words.

Shift right serial register

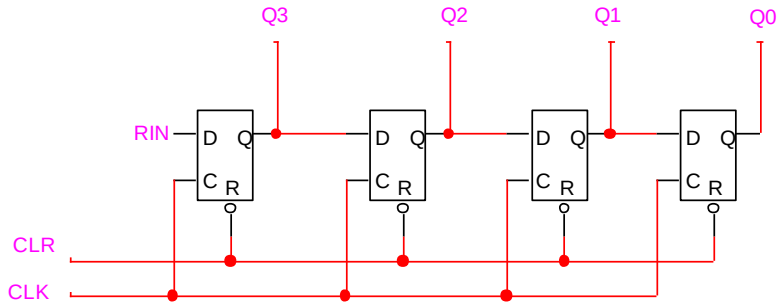


The RIN specifies a new bit to be shifted into one end at each clock tick. This bit appears at the serial output after n clock ticks. Thus an n-bit serial-in, serial-out shift register can be used to delay a signal by n clock ticks

Shift left serial register



A serial-in, parallel-out shift register has outputs for all of its stored bits making them available to other circuits. Such a shift register can be used to perform serial-to-parallel conversion.



A parallel-in, serial-out shift register can be used to perform parallel-to-serial conversion. The device uses a 2-input multiplexer on each flip-flop's D inputs to select between shift or load, according to the control signal S.

So at each clock tick the Rg either loads new data from inputs D1 – Dn or it shifts its current contents, depending on the value of S.

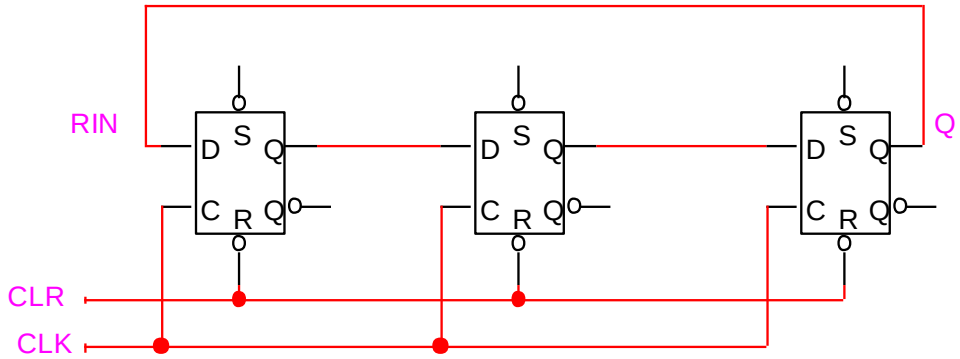
The register that can shift only in one direction is called a **unidirectional shift register**.

The register that can shift in either of two directions – left or right is called a **bidirectional shift register.**

Shift left on n-bit is equivalent with multiplication on 2^n. Shift right on n-bit is equivalent with division on 2^n.

D shift register can be combined with a combinational logic to form a state machine whose state diagram is cyclic.

Such a circuit is called a **shift-register counter (ring counter)**. Unlike a binary counter a shift-register counter does not count in an ascending or descending binary sequence, but it is useful in many control application.



The 194 is sometimes called a universal shift register because it can be made to function like any of the leas general shift-register types unidirectional; (serial-in, serial-out, parallel-in, parallel-out) …

A transition table for 74-194

Functions	Inputs		t			
	S1	S0	$Q4^*$	$Q3^*$	$Q2^*$	$Q1^*$
Hold	0	0	Q4	Q3	Q2	Q1
Shift-right	0	1	RIN	Q4	Q3	Q2
Shift-left	1	0	Q3	Q2	Q1	LIN
Load	1	1	B4	B3	B2	B1
Arithmetic shift right	0	1	Q4	Q4	Q3	Q2
Round shift rigth	0	1	Q1	Q4	Q3	Q2
Arithmetic shift left	1	0	Q3	Q2	Q1	0
Logic shift left	1	0	Q3	Q2	Q1	0
Round shift left	1	0	Q3	Q2	Q1	Q4

This functions table is highly compressed since it does not contain columns for most of the inputs B4 – B1, RIN, LIN, S1, and S0. In this case we can have 2^8 possible combinations -> a 256 row table.

$D4 = nS_1nS_0 * Q_4 + nS_1S_0\ RIN + S_1nS_0 * Q_3 + S_1S_0 * B_4$

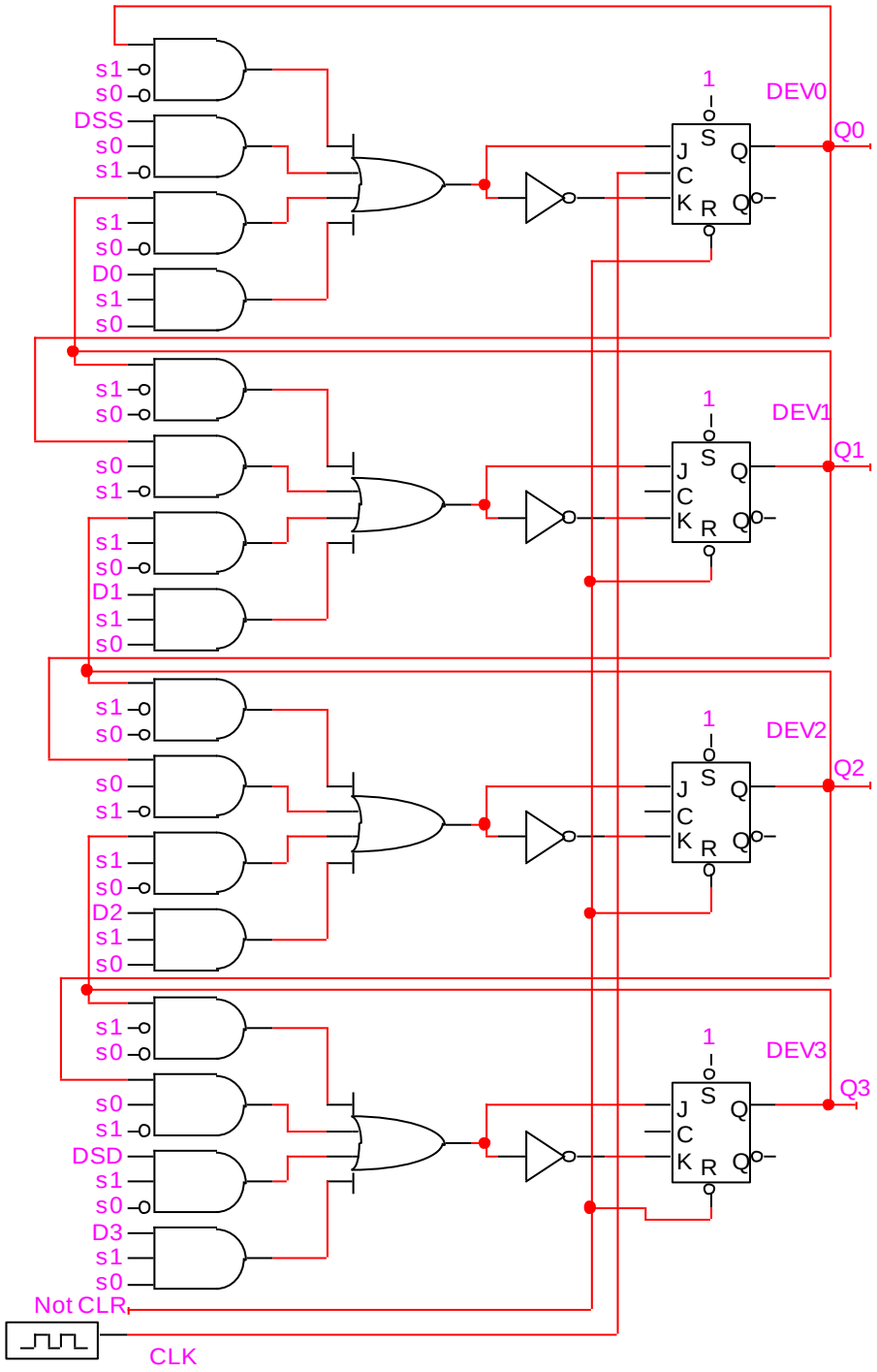
$D3 = nS_1nS_0 * Q_3 + nS_1S_0\ Q_4 + S_1nS_0 * Q_2 + S_1S_0 * B_3$

$D2 = nS_1nS_0 * Q_2 + nS_1S_0\ Q_3 + S_1nS_0 * Q_1 + S_1S_0 * B_2$

$D1 = nS_1nS_0 * Q_1 + nS_1S_0\ Q_2 + S_1nS_0 * LIN + S_1S_0 * B_1$

$J4=nK4 = nS_1nS_0 * Q_4 + nS_1S_0\ RIN + S_1nS_0 * Q_3 + S_1S_0 * B_4$
$J3=nK3 = nS_1nS_0 * Q_3 + nS_1S_0\ Q_4 + S_1nS_0 * Q_2 + S_1S_0 * B_3$
$J2=nK2 = nS_1nS_0 * Q_2 + nS_1S_0\ Q_3 + S_1nS_0 * Q_1 + S_1S_0 * B_2$
$J1=nK1 = nS_1nS_0 * Q_1 + nS_1S_0\ Q_2 + S_1nS_0 * LIN + S_1S_0 * B_1$

Logic circuit



Task

According to the variant indicated in Table 1, perform the synthesis of a 4-bit register based on the JK flip-flops and NAND gates.

Table 1

Nr. var.	Flip flop	Functions	Nr. var.	Flip flop
1	JK	Hold Arithmetic shift left Load	14	D
2	JK	Hold Arithmetic shift right Load	15	D
3	JK	Hold Logic shift left Load	16	D
4	JK	Hold Logic shift right Load	17	D
5	JK	Hold Round shift left Load	18	D
6	JK	Hold Round shift left Load	19	D
7	JK	Hold Arithmetic shift left Arithmetic shift right	20	D
8	JK	Hold Logic shift left Logic shift right	21	D
9	JK	Hold Round shift left Round shift right	22	D
10	JK	Load Arithmetic shift left Arithmetic shift right	23	D
11	JK	Load Logic shift left Logic shift right	24	D
12	JK	Load Round shift left Round shift right	25	D
13	JK	Load Round shift left Arithmetic shift right	26	D

Report

The report for the laboratory work will include:

1. Individual task according to the variant;
2. Transition table
3. Logic functions
4. Designed logic circuits of the register. Cost and delay time. Timing diagramm;
5. Conclusions.