

LAB I: Logic Gates



Logic gates

Introduction:

Logic gates are the basic building blocks of any digital system. It is an electronic circuit having one or more than one input and only one output. The relationship between the input and the output is based on a certain logic. Based on this, logic gates are named as AND gate, OR gates, NOT gate etc.

① AND gate:

A circuit which performs an AND operation is shown in figure. It has n input ($n \geq 2$) and one output.

Logic diagram



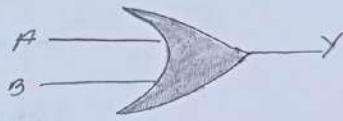
Truth table

Inputs		Output
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

② OR gate

A circuit which performs an OR operation. It has n input ($n \geq 2$) and one output.

Logic diagram



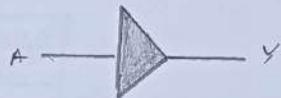
Truth Table

INPUT		OUTPUT
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(3) NOT Gate

NOT Gate is also known as INVERTER. It has one input A and one output Y.

Logic diagram



Truth table

INPUT		OUTPUT
A	B	
0		1
1		0

(4) NAND Gate

A NOT-AND operation is known as NAND operation. It has one n input ($n \geq 2$) and one output.

Logic diagram



3) Truth Table

INPUTS		OUTPUT
A	B	\overline{AB}
0	0	1
0	1	1
1	0	1
1	1	0

⑤ NOR GATE

A NOT-OR operation is known as NOR operation. It has n input pins ($n \geq 2$) and one output.

Logic Diagram



Truth Table

INPUTS		OUTPUTS
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

⑥ XOR GATE

XOR or EX-OR gate is a special type of gate. It can be used in the half adder, full adder and subtractor. It has n input pins ($n \geq 2$) and one output.

Logic Diagram



4

Truth Table

INPUTS		OUTPUTS
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(7) XNOR Gate

XNOR gate is a special type of gate. It can be used in the half adder, full adder and subtractor. It has n inputs ($n \geq 2$) and one output.

Logic diagramTruth Table

INPUTS		OUTPUTS
A	B	$A \oplus B$
0	0	1
0	1	0
1	0	0
1	1	1

5

LAB III - universal gates

A universal gate is a gate which can implement any boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. Likewise, an OR gate is typically implemented as a NOR gate followed by an inverter, not the other way around.

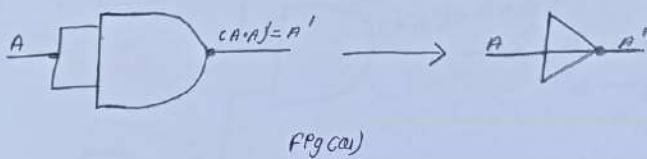
#NAND gate as a Universal Gate

To prove that any boolean function can be implemented using only NAND gates, we will show that the AND, OR and NOT operations can be performed using only these gates.

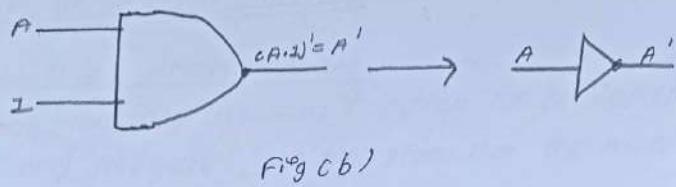
#Implementing an Inverter using only NAND gates

The figure shows two ways in which a NAND gate can be used as an inverter (NOT gate).

- ① All NAND input pins connect to the input signal A gives an output A' .



- ② One NAND input pin is connected to the input signal A while all other input pins are connected to logic 1. The output will be A' .



Implementing AND using only NAND Gates

An AND gate can be replaced by NAND gates as shown in the figure (the AND is replaced by a NAND gate with its output complemented by a NAND gate inverter).

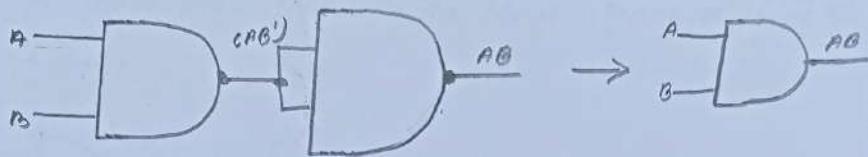
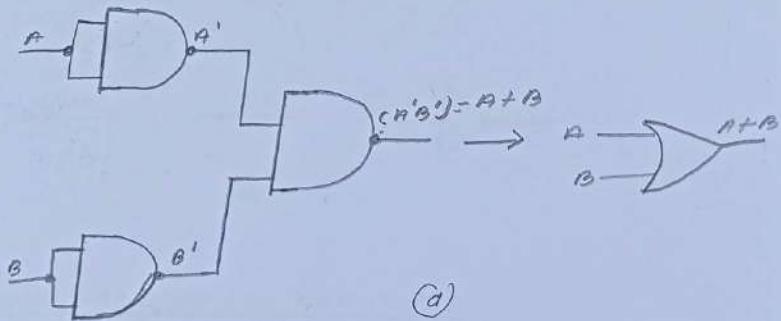


Fig 6.1)

Implementing OR using only NAND Gates

An OR gate can be replaced by NAND gates as shown in the figure (the OR gate is replaced by a NAND gate with all its inputs complemented by NAND gate inverters).



(a)

Fig. Illustration of NAND gate as a universal gate

Thus, the NAND gate is a universal gate since it can implement the AND, OR and NOT function.

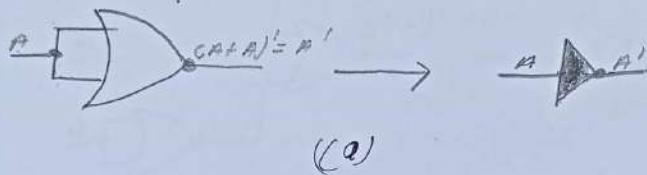
NOR gates is a universal gate

To prove that any boolean function can be implemented using only NOR gates, we will show that the AND, OR and NOT operations can be performed using only these gates.

Implementing an Inverter using only NOR gate

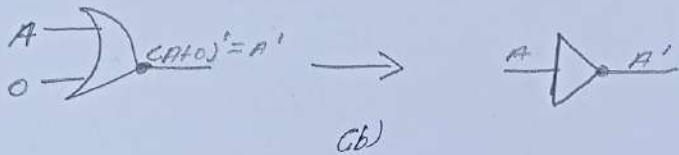
The figure shows two ways in which a NOR gate can be used as an inverter (NOT gate).

1. All NOR input pins connect to the input signal A gives an output A' .



(a)

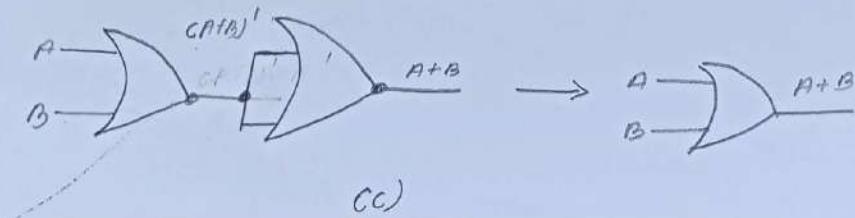
2. One NOR input pin is connected to the input signal A while all other input pins are connected to logic 0. The output will be A' .



(b)

Implementing OR using only NOR gates

An OR gate can be replaced by NOR gates as shown in the figure (The OR is replaced by a NOR gate with its output complemented by a NOR gate inverter)



LAB III: Combinations

*8 Implementing AND using only NOR gates

An AND gate can be replaced by NOR gates as shown in the figure
(The AND gate is replaced by a NOR gate with all its inputs
complemented by NOR gate inverters)

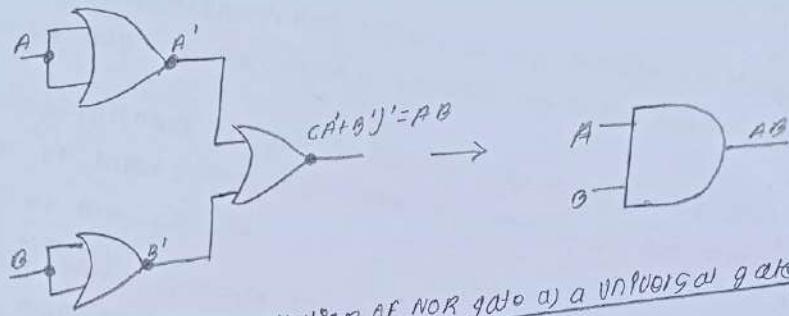


Fig. Illustration of NOR gate as a universal gate

Thus, the NOR gate is a universal gate since it can implement
the AND, OR and NOT functions.

Ques

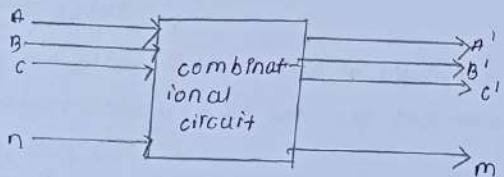
Inputs to
BLU

L A B T H : combinational logic

Combinational circuit is a circuit in which we combine the different gates in the circuit, for example encoder, decoder, multiplexer and demultiplexer. Some of the characteristics of Combinational circuits are following -

- The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
- The combinational circuit do not use any memory. The previous state of inputs does not have any effect on the present state of the circuit.
- A combinational circuit can have an n number of inputs and m number of outputs.

Block diagram

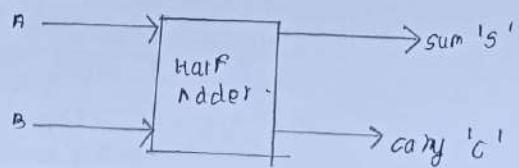


We're going to elaborate few important combinational circuit as follows.

① Half Adder

Half adder is a combinational logic circuit with two inputs and two outputs. The half adder circuit is designed to add two single bit binary numbers A and B.

Block diagram



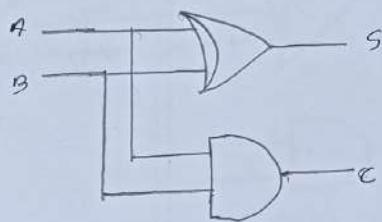
10

1 (P) Half Adder

Truth table

INPUTS		OUTPUTS	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

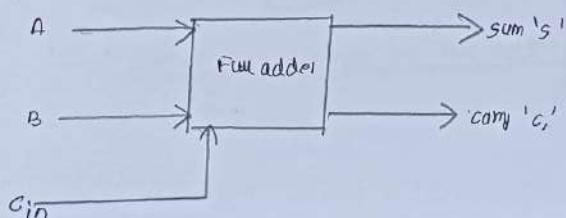
Circuit Diagram



② Full adder

Full adder is developed to overcome the drawback of half adder circuit. It can add two one-bit numbers A and B, and carry C.

Block diagram



Truth table

INPUT			OUTPUT	
A	B	Cin	S	C0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0

				Sum
0	1	0	0	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

11-12 L P/N/M

Circuit Diagram

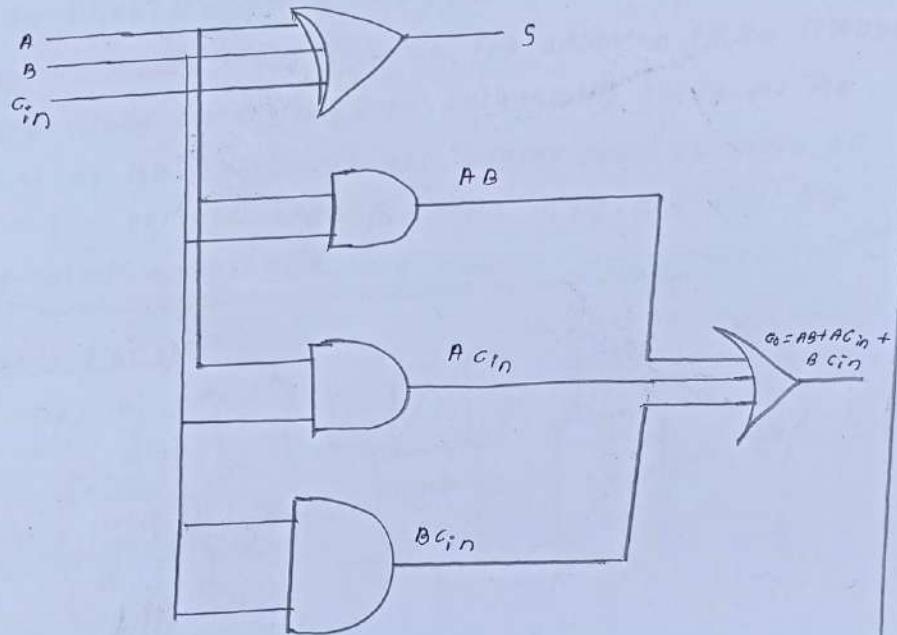


fig: circuit diagram of full adder.

12

B N-Bit parallel Adder

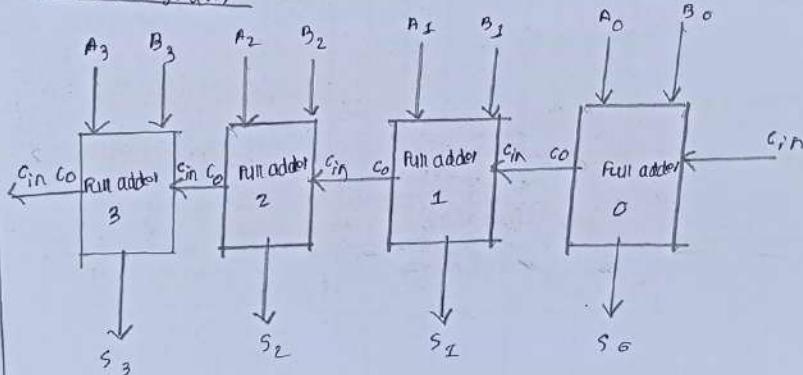
The full adder is capable of adding only two single digit binary number along with a carry input. But in practical we need to add binary numbers which are much longer than just one bit.

The carry output of the previous full adder is connected to carry input of the next full adder.

4-bit parallel Adder

In the block diagram, A_0 and B_0 represents the LSB of the four bit words A and B. Hence full adder-0 is the lowest stage. Hence its C_{in} has been permanently made 0. The rest of the connection are exactly same as those of n-bit parallel adder is shown in fig. The four bit parallel adder is a very common logic circuit.

Block diagram



fr Block diagram

N-bit Subtractors

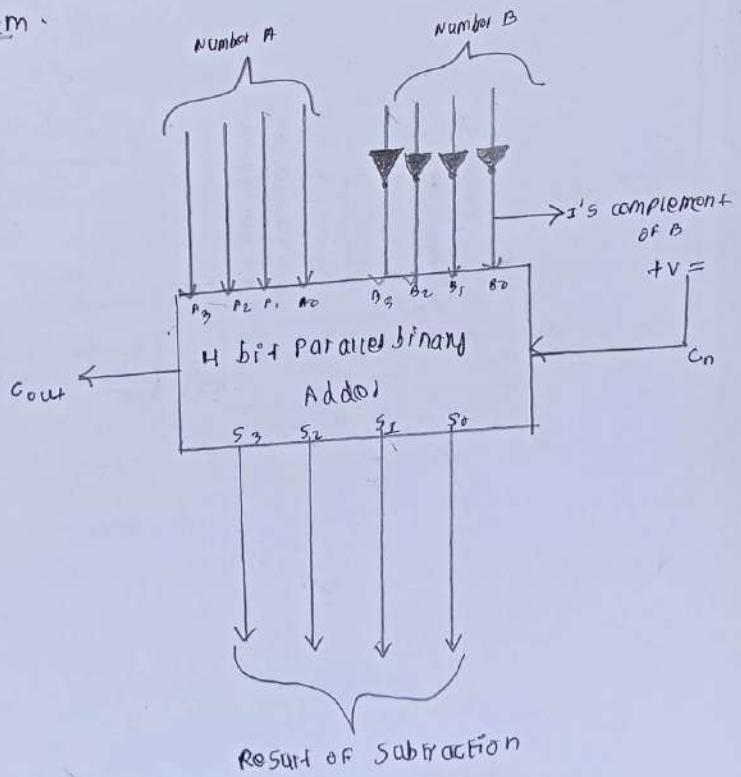
⑤ N-bit parallel subtractor

The subtraction can be carried out by taking the 1's or 2's complement of the number to be subtracted. For example we can perform the subtraction $(A-B)$ by adding either 1's or 2's complement of B to A .

⑥ 4-bit parallel subtractor

The number to be subtracted (B) is first passed through inverters to obtain its 1's complement. The 4-bit adder then adds A and 1's complement of $B+0$ produce the subtraction $S_3 S_2 S_1 S_0$ which represents the results of binary subtraction $(A-B)$. The end carry output C_{out} represents the polarity of the result.

Block diagram:



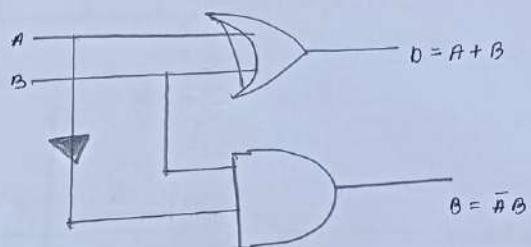
Half Subtractor

Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow). It produces the difference between the two binary bits at the input and also produces an output (borrow) to indicate if a 1 has been borrowed. In the subtraction $(A-B)$, A is called as minuend bit and B is called as subtrahend bit.

Truth Table

INPUTS		OUTPUTS	
A	B	$(A-B)$ BORROW	
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Circuit Diagram



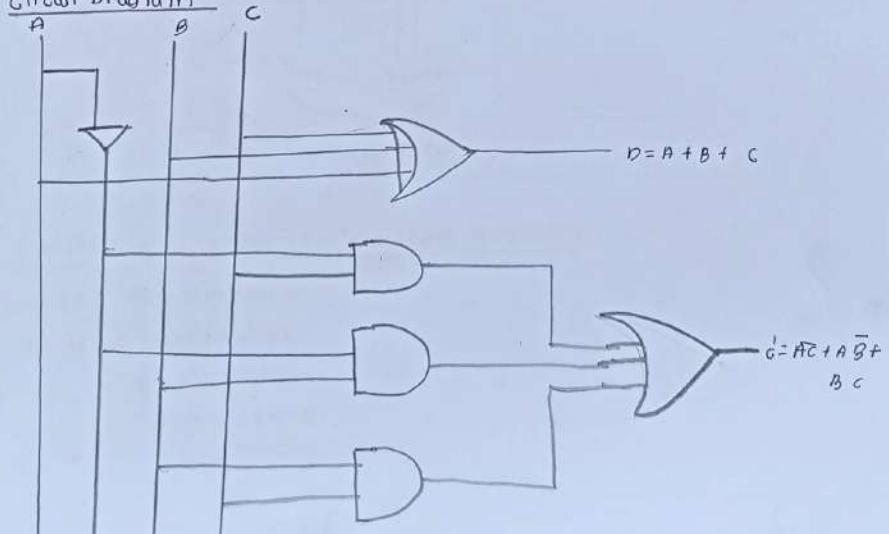
③ Full subtractors

The disadvantage of a half subtractor is overcome by full subtractor. The full subtractor is a combinational circuit with three inputs A, B, C and two output D and C' . A is the "minuend", B is "subtrahend", C is the "borrow" produced by the previous stage, D is the difference output and C' is the borrow output.

Truth Table

INPUTS			OUTPUTS	
A	B	C	(B-C)	C'
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Circuit Diagram



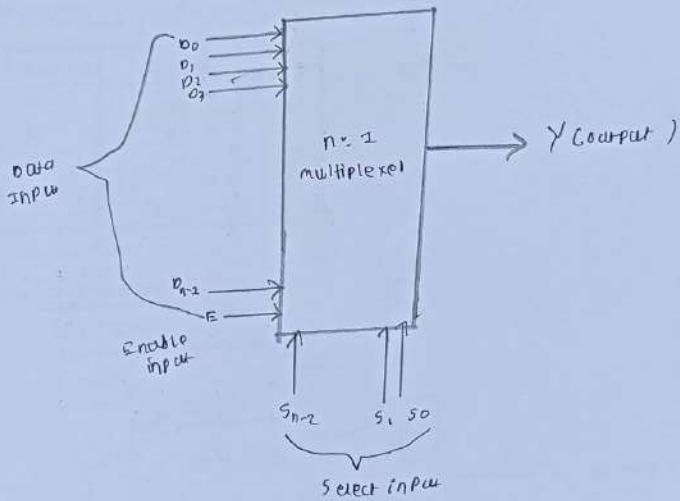
16 LAB IV = multiplexer, Demultiplexer, Encoder & decoder

(1) Multiplexers

Multiplexer is a special type of combinational circuit.

There are n -data inputs, one output and m select inputs with $2^m = n$. It is a digital circuit which selects one of the n data inputs and routes it to the output.

Block diagram

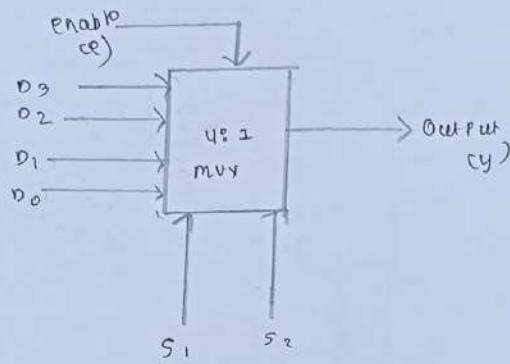


Multiplexers come in multiple variations

- ① 2:1 multiplexers
- ② 4:1 multiplexers
- ③ 16:1 multiplexers
- ④ 32:1 multiplexers

Q3 4 : 1 multiplexer

Block diagram



Truth table

Select data inputs		Output
S ₂	S ₁	y
0	0	D ₀
0	1	D ₁
1	0	D ₂
1	1	D ₃

Logic diagram

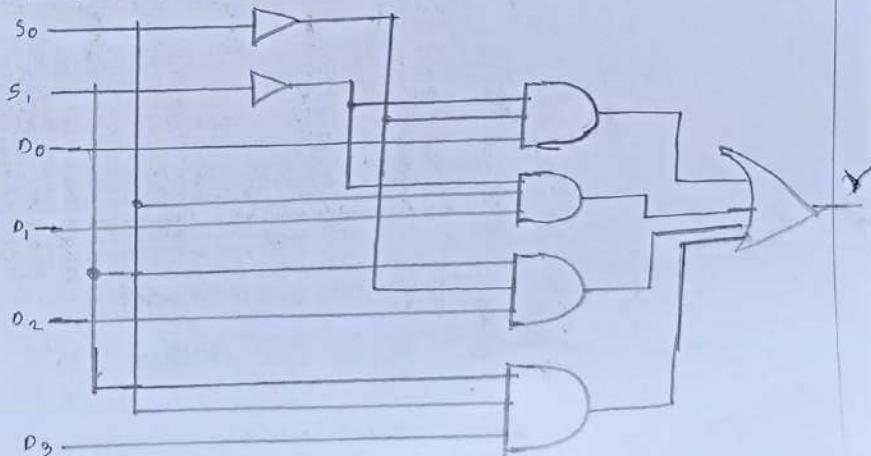
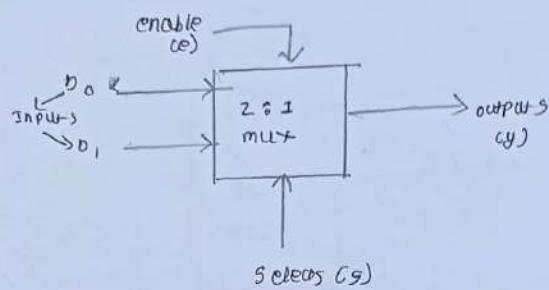


Fig: logic diagram of 4x1 mux.

Truth Table

① 2:1 multiplexer

Block diagram



Truth table

Select	Inputs	Output 1	Output 2
0	0	0	0
0	0	1	1
1	1	0	0
1	1	1	1

Logic diagram

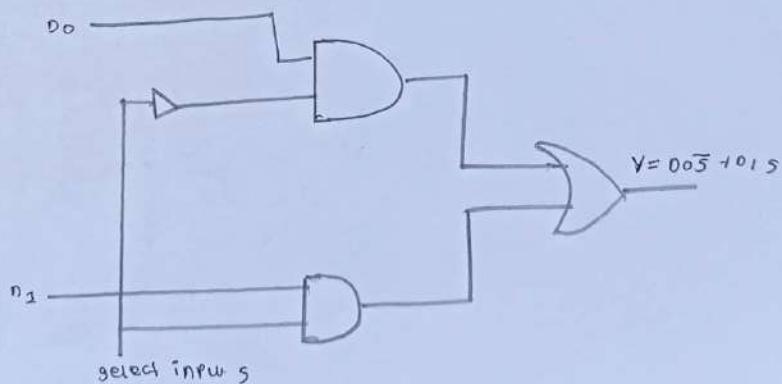
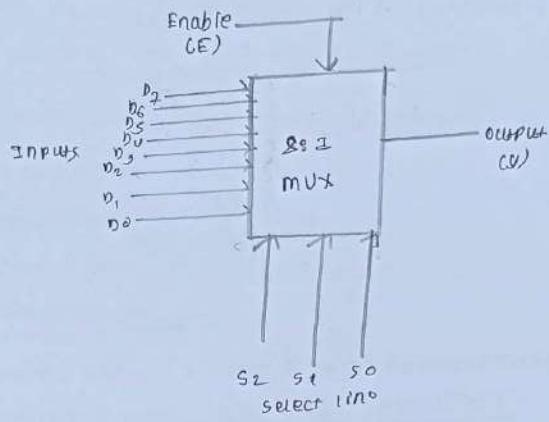


Fig:- logic diagram of 2x1 mux.

Truth Table

8:1 multiplexer

Block diagram



Truth Tab 10

SELECT DATA INPUTS			OUTPUT
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Logic Diagram

Truth Table

① De-multiplexer

A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs. A de-multiplexer is equivalent to single pole multiple way switch as shown.

De-multiplexer comes in multiple variation.

① 1:2 demultiplexer

② 1:4 demultiplexer

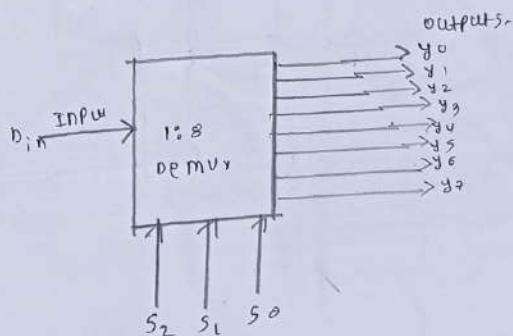
③ 1:16 demultiplexer

④ 1:32 demultiplexer

In book, there is only 1 to 8 demultiplexer and I didn't write it only, so I write only 1:8 demultiplexer.

① 1 to 8 multiplexer

Block diagram

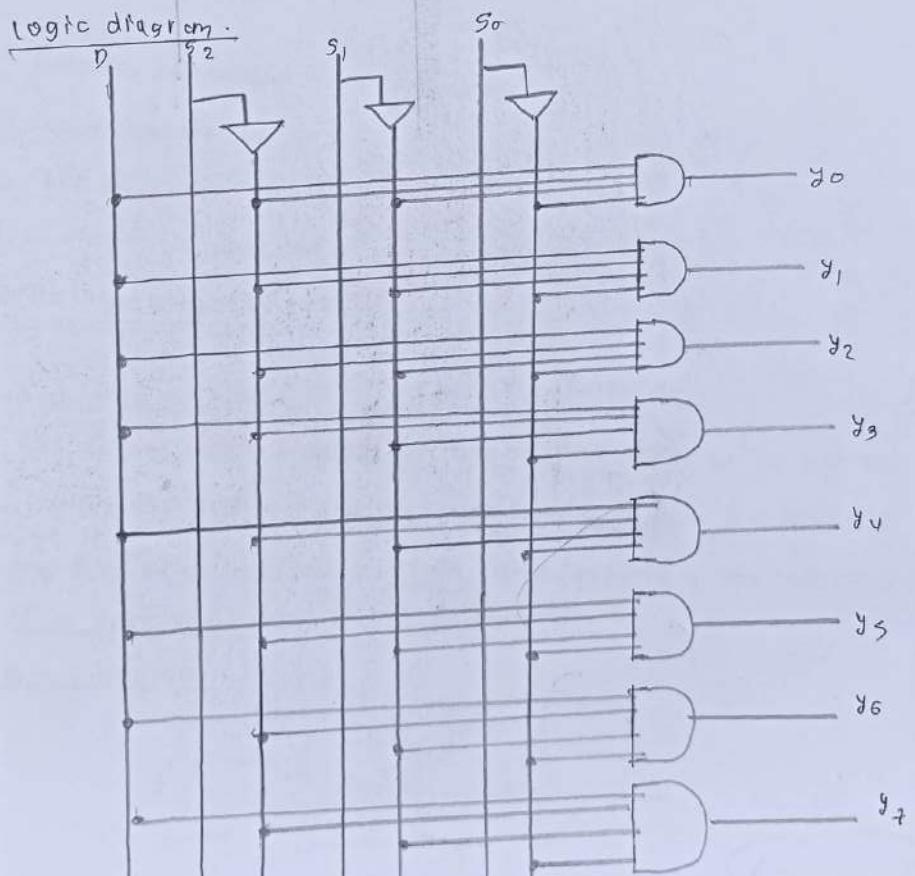


Truth Table

Data Inputs	Select Inputs			Outputs							
	s_2	s_1	s_0	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
D	0	0	0	0	0	0	0	0	0	0	D
D	0	0	1	0	0	0	0	0	0	0	0
D	0	1	0	0	0	0	0	0	0	0	0
D	0	1	1	0	0	0	0	0	0	0	0
D	1	0	0	0	0	0	0	0	0	0	0
D	1	0	1	0	0	0	0	0	0	0	0
D	1	1	0	0	0	0	0	0	0	0	0
D	1	1	1	0	0	0	0	0	0	0	0

Fig: Truth table of 1×8 DEMUX

Logic diagram:

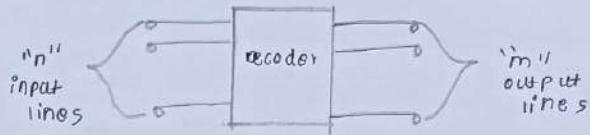


Truth Table

Decoder

A decoder is a combinational circuit. It has n input and $m = 2^n$ outputs. Decoder is identical to a demultiplexer without data input. It performs operations which are exactly opposite to those of an encoder.

Block diagram



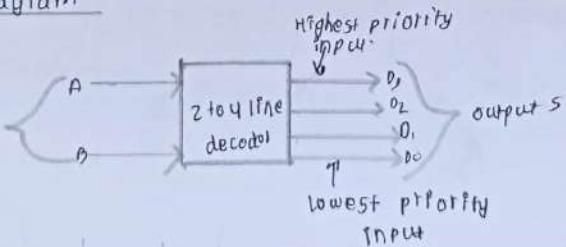
Examples of decoders are following.

- ① Code converters
- ② BCD to seven segment decoders
- ③ Nixie tube decoders
- ④ Relay actuator

2 to 4 line Decoder

The block diagram of 2 to 4 line decoder is shown in the fig. A and B are the two inputs where D through Q are the four outputs. Truth Table explains the operations of a decoder.

Block diagram

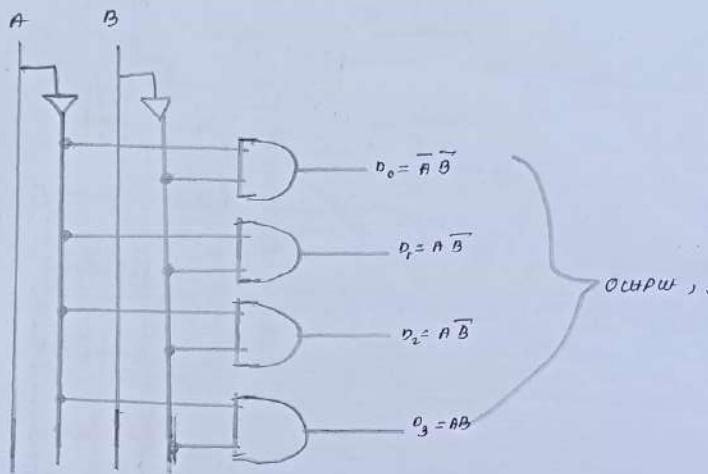


Truth Table

23 Truth Table

INPUTS		OUTPUT			
A	B	D ₀	D ₁	D ₂	D ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	1	0	0	1

Logic circuit



Q 3: 8 Decoder

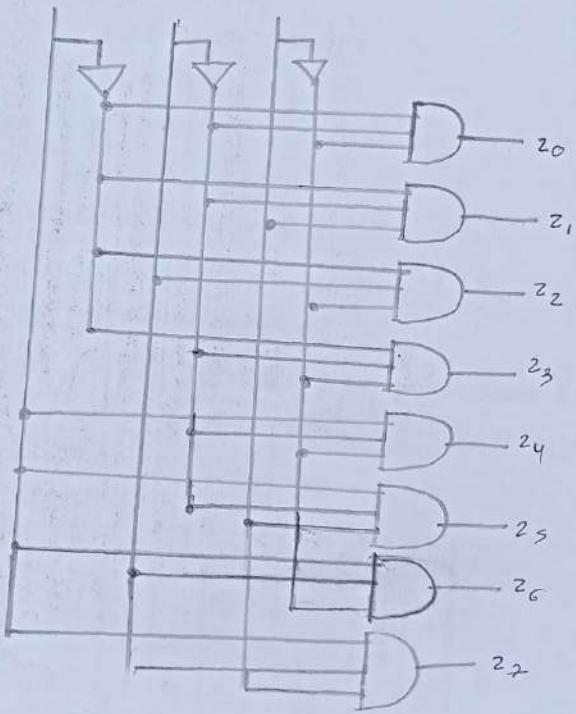
It uses all AND gates, and therefore, the outputs are active-high. For active-low outputs, NAND gates are used. It has 3 input lines and 8 output lines. It is also called as binary to octal decoder. It takes a 3-bit binary input code and activates one of the 8 (octal) outputs corresponding to that code.

Truth Table

INPUT

A_2	A_1	A_0	Z_7	Z_6	Z_5	Z_4	Z_3	Z_2	Z_1	Z_0
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0

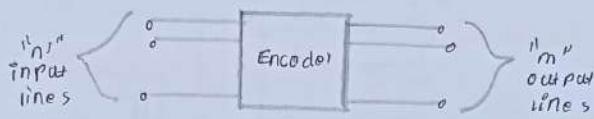
Logic Diagram



④ Encoder

Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder. An encoder has a number of input lines and m number of output lines.

Block diagram



Examples of encoders are following -

- ① Priority encoders
- ② Decimal to BCD encoders
- ③ Octal to binary encoder
- ④ Hexadecimal to binary encoder

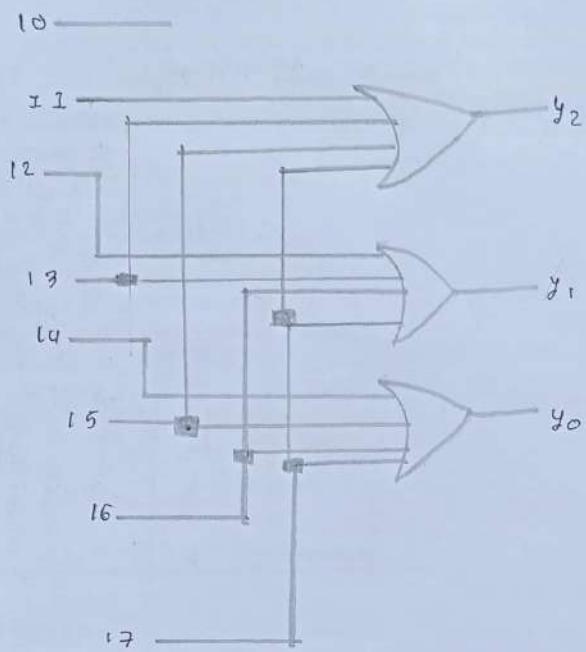
① Octal to binary encoder

Octal-to-binary takes 3 inputs and provides 3 outputs, thus doing the opposite of what the 3-to-8 decoder does.

Truth Table

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	y_2	y_1	y_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

42 59
26 A logic diagram



LAB: V FLIP FLOP

A PEEP FLOP IS A SEQUENTIAL CIRCUIT WHICH GENERALLY SAMPLES ITS INPUTS AND CHANGES ITS OUTPUTS ONLY AT PARTICULAR INSTANTS OF TIME AND NOT CONTINUOUSLY. FLIP FLOP IS SAID TO BE EDGE SENSITIVE OR EDGE TRIGGERED RATHER THAN LEVEL TRIGGERED LIKE LATCHES.

S-R PEEP FLOP

IT IS BASICALLY S-R LATCH USING NAND GATES WITH AN ADDITIONAL ENABLE INPUT. IT IS ALSO CALLED AS LEVEL TRIGGERED SR-FF. FOR THIS, CIRCUIT IN OUTPUT WILL TAKE PLACE IF AND ONLY IF THE ENABLE INPUT (E) IS MADE ACTIVE.

42 99

Flutn

INP

E

I

1

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

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

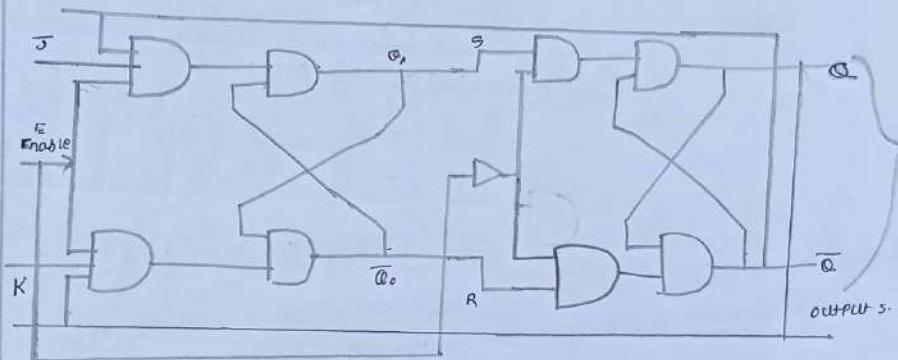
321

322

Truth Table

INPUTS			OUTPUTS		COMMENT
E	J	K	Q_{nxt}	\bar{Q}_{nxt}	
1	0	0	0	1	No change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	0	1	Toggle

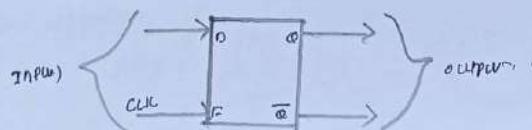
Circuit diagram



Delay flip flop / D flop flop

Delay flip flop or D flop flop is the simple gated S-R latch with a NAND inverter connected between S and R inputs. If has only one input. The input data is appearing at the output after some time.

Block diagram



Truth Tabl

INPUTS	
E	T
1	0
0	1

L A O VII

O ASync

The
in Figu
can use
Perman
clock
the Clo

logical
logic I

CLIK

Truth Tabl

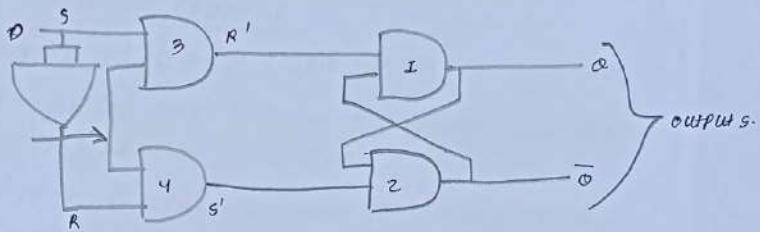
CLOK IS

INITIALLY
25 +
2nd
3rd
4th

TRUTH TABLE

INPUTS		OUTPUTS		Comments
E	D	Q_{n+1}	\bar{Q}_{n+1}	
1	0	0	1	
1	1	1	0	Reset Set

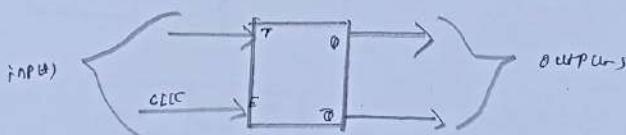
Circuit diagram



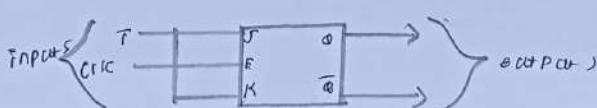
Toggle Flipp Flop / T Flipp Flop

Toggle flip flop is basically a JK flip flop with J and K terminals permanently connected together. It has only one input denoted by T as shown in the symbol diagram.

Symbol diagram



Block diagram



Truth Table

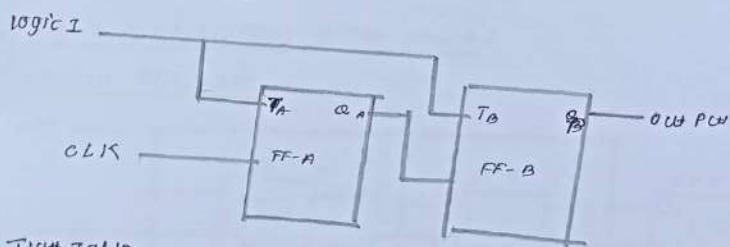
INPUTS	OUTPUTS	Comment		
E	T	Q_{A+}	Q_{B+}	
1	0	0	1	No change
0	1	1	0	Toggle

LAB III - A SYNCHRONOUS COUNTER

① ASYNCHRONOUS OR RPPPLA UP COUNTER

The logic diagram of a 2-bit RPPPLA UP COUNTER is shown in figure. The toggle (T_2) FLIP-FLOP are being used. But we can use the JK FLIP-FLOP also with J and K connected permanently to logic 1. External clock is applied to the clock input of flip-flop A and Q_A output is applied to the clock input of the next flip-flop i.e. FF-B.

LOGICAL DIAGRAM

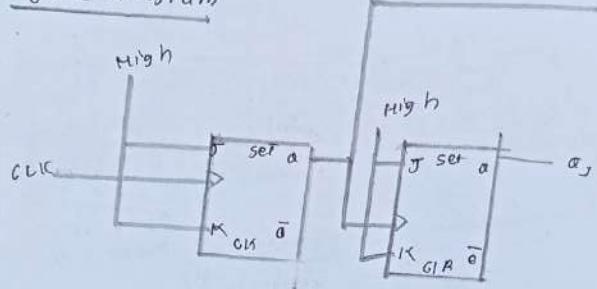


Truth Table

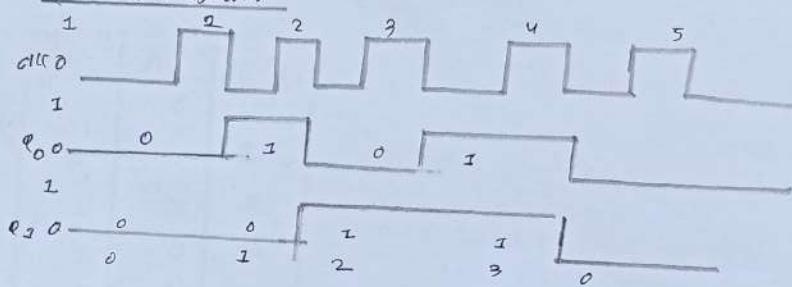
CLOCK	COUNTER OUTPUT		STATE NUMBER	DECIMAL COUNTER OUTPUT
	Q_3	Q_2		
Initially	0	0	-	0
1st	0	1	1	1
2nd	1	0	2	2
3rd	1	1	3	3
4th	0	0	4	0

① 2-bit ASYNCHRONOUS BINARY COUNTER.

logical diagram

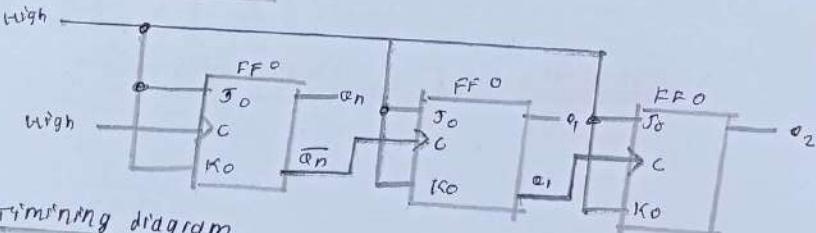


Timing diagram

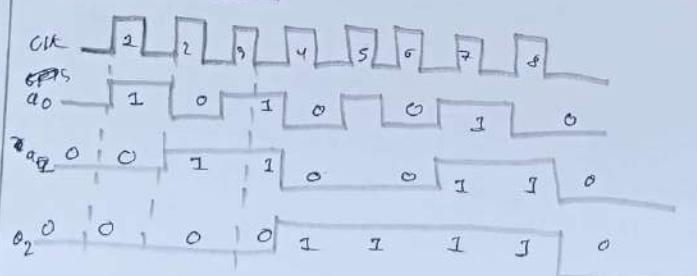


② 3-bit ASYNCHRONOUS BINARY COUNTER

logical diagram



Timing diagram

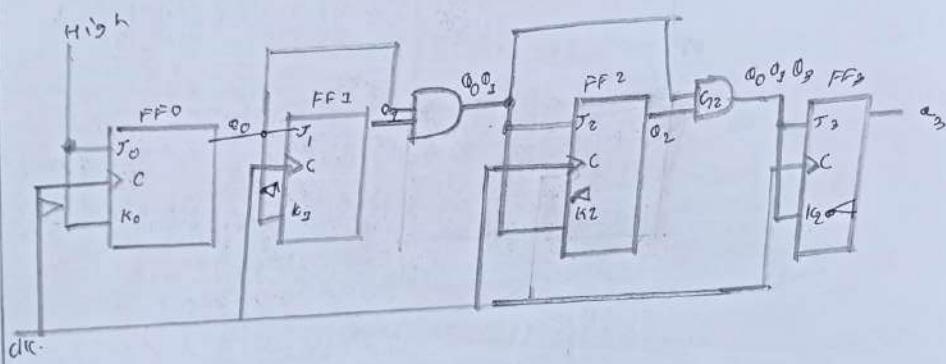


32

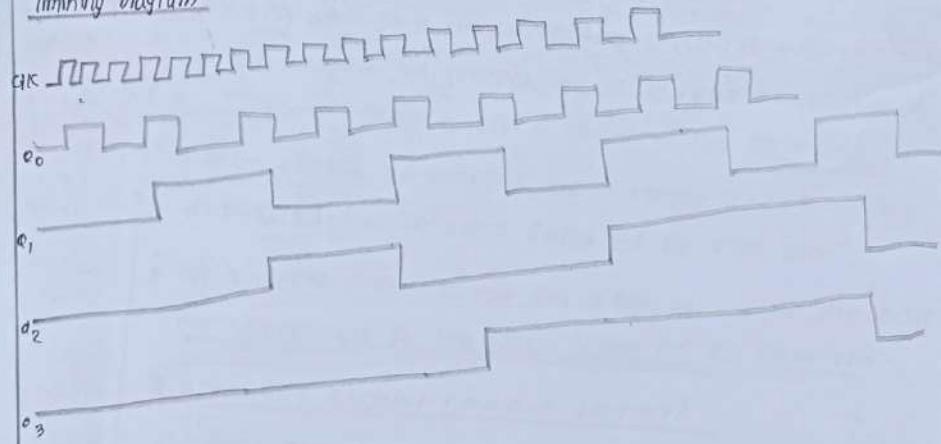
35

Q 4 Bit-Synchronous Binary Counter.

Logic diagram



Timing diagram



in the
TOPS OR
BY:

i's

i's

FF

ind

Psychronous UP-DOWN COUNTER

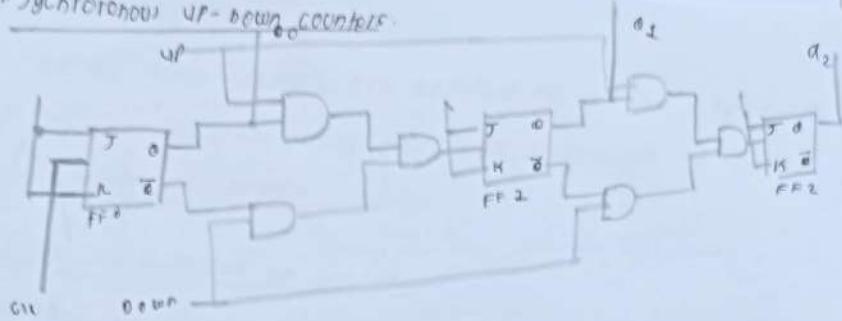


Fig - 3-bit up-down counter

FF2	FF1	FF0
0	0	0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1

Table & count sequence of 3-bit down counter

34

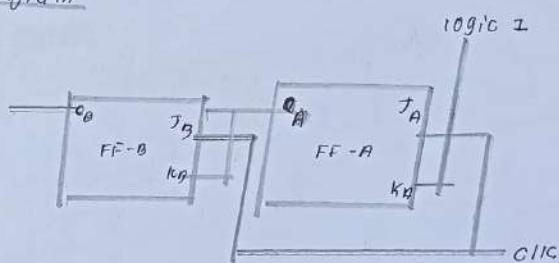
⑥ Synchronous counters

If the "CLOCK" pulses are applied to all the FLIP-FLOPS in a counter simultaneously, then such a counter is called Synchronous counter.

2 bit synchronous UP counter

The JA and KA inputs of FF-A are tied to logic 1. So FF-A will work as a toggle flip-flop. The JB and KB inputs are connected to Q_A.

Logical diagram



Classification of counters

Depending on the way in which the counting progresses, the synchronous or asynchronous counters are classified as follows -

UP counters

DOWN counters

UPDOWN counters

⑦ UPDOWN counter

UP counter and down counter is combined together to obtain an updown counter. A mode control CMJ input is also provided to select either up or down mode.

TYPE OF UP/DOWN COUNTERS

- ① UP/DOWN RIPPLE COUNTERS
- ② UP/DOWN SYNCHRONOUS COUNTER

UP/DOWN RIPPLE COUNTERS

IN THE UP/DOWN RIPPLE COUNTER ALL THE FFs OPERATE IN THE TOGGLE MODE. SO EITHER T FLIP-FLOPS OR JK FLIP-FLOPS ARE TO BE USED. THE LSB FLIP-FLOP RECEIVES CLOCK DIRECTLY.

Example

3-bit binary up/down ripple counter

- 3-bit - hence three FFs are required.
- UP/DOWN - so a mode control input is essential.
- For a Ripple Up Counter, the Q output of preceding FF is connected to the clock input of the next one.
- For a Ripple Up Counter, the Q output of preceding FF is connected to the clock input of the next one.
- For a Ripple Down Counter, the Q-bar output of preceding FF is connected to the clock input of the next one.

③ # modulus counter (Modulo-N counter)

The 2-bit Ripple Counter is called as MOD-4 counter and 3-bit Ripple Counter is called as MOD-8 counter.

TYPES OF MODULUS

- 2-bit up or down (Mod-4)
- 3-bit up or down (Mod-8)
- 4-bit up or down (Mod-16)

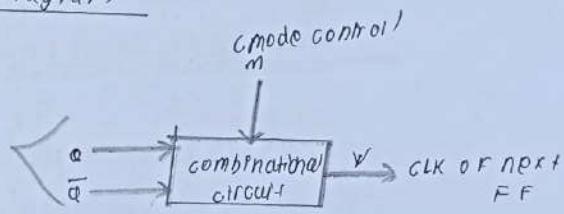
1. Waveforms

42 89 36

Application of counters

- Frequency counters
- Digital clock
- Time measurement
- A to D converter
- Frequency divider circuits
- Digital triangular wave generator

Block diagram



Truth Table

m	a	\bar{a}	outputs
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$Y = a$
for up counter

$Y = \bar{a}$
for up counter.

LAB VIII = Registers

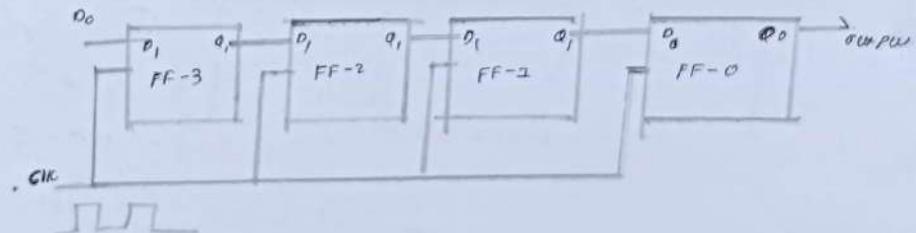
FLIP-FLOP is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop.

There are four mode of operations of a shift register:

- serial input serial output
- serial input parallel output
- parallel input serial output
- parallel input parallel output

Serial Input Serial Output

Block Diagram

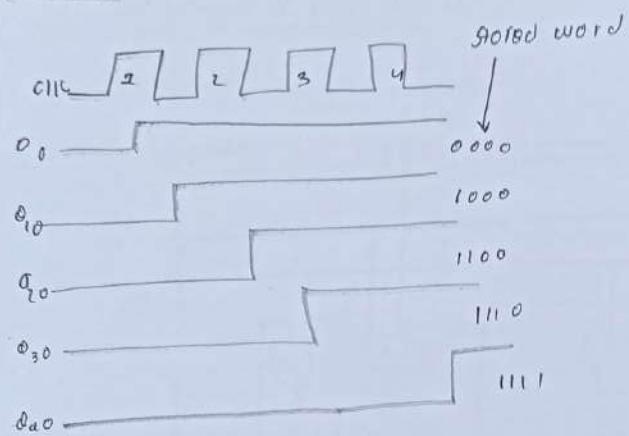


Operation Truth Table

Initially	CLK	$D_0 = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
i)	↓	I → 1	1 → 0	0 → 0	0 → 0	0
ii)	↓	I → 1	1 → 1	1 → 0	0 → 0	0
iii)	↓	I → 1	1 → 1	1 → 1	1 → 0	0
iv)	↓	I → 1	1 → 1	1 → 1	1 → 1	1

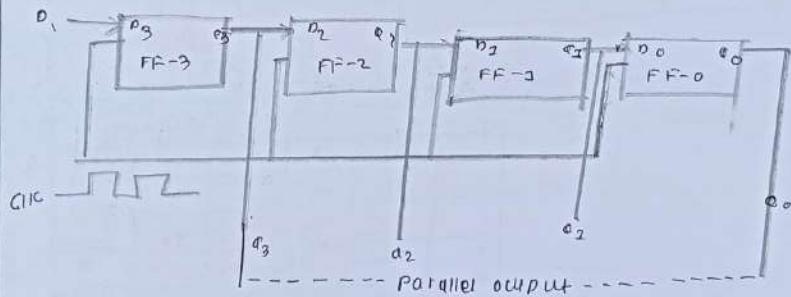
38

Waveforms



SERIAL INPUT PARALLEL OUTPUT

Block diagram

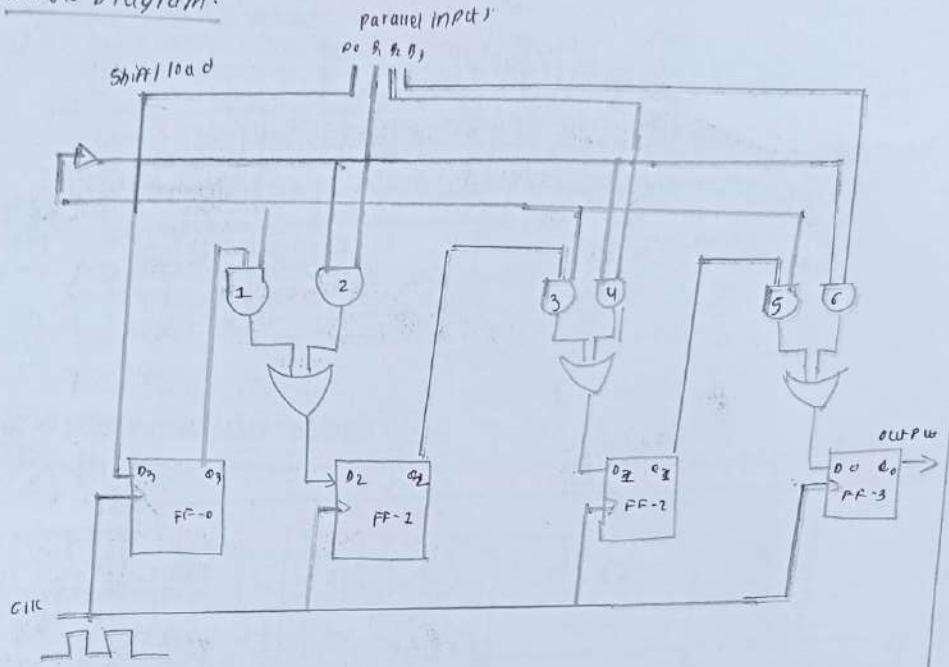


CLEAR	FF0	FF1	FF2	FF3
1001	0	0	0	0
	1	0	0	0
	0	1	0	0
	0	0	1	0
	1	0	0	1

3. Plot Table & shift operation in serial in parallel out
shift register.

① Parallel In-Series Out Shift Registers

LOGIC Diagram:

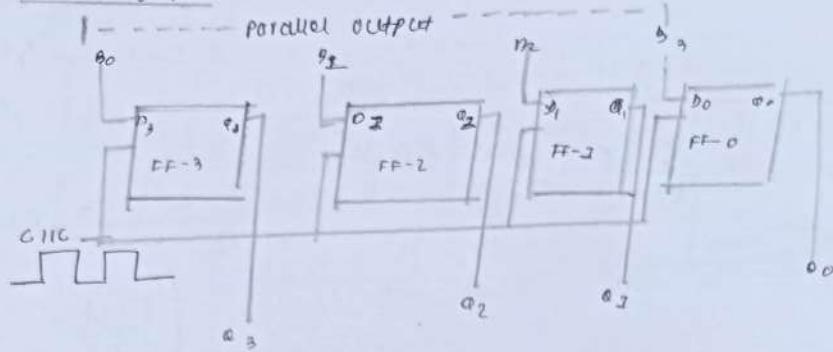


.	Q_0	Q_1	Q_2	Q_3
Clear	0	0	0	0
Write	1	0	0	1
Shift+	1	0	0	1
	1	1	0	0
	1	1	1	0
	1	1	1	1
	1	1	1	1

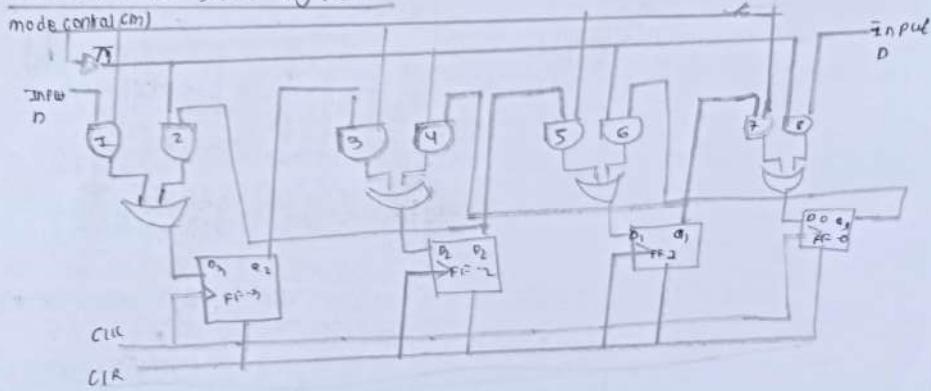
Table : Shift operation in parallel in serial out shift register.

Q) Parallel In - Parallel Out shift register

Block diagram

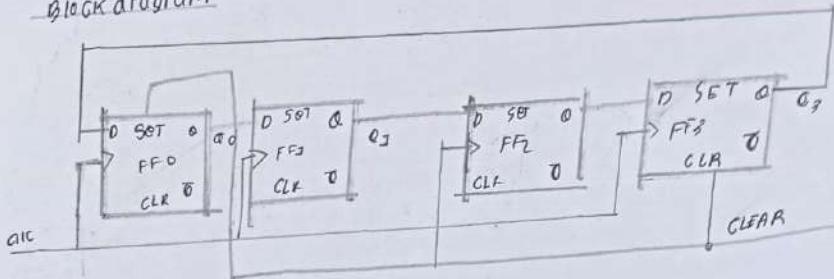


Bi-directional shift register



Ring Counters

Block diagram

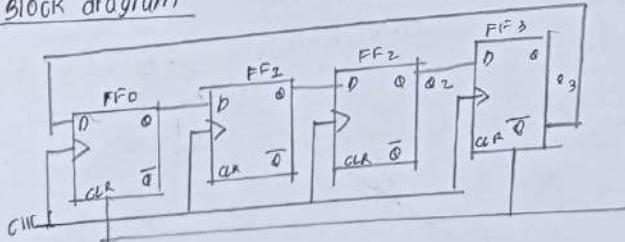


CLOCK PULSE	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0

Table: Count sequence of a 4-bit Ring counter

Johnson Counters

Block diagram



CLOCK PULSE	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1
5	1	1	1	0
6	1	1	0	0
7	1	0	0	0

Table: Count sequence of Johnson counter