CHAPTER 2

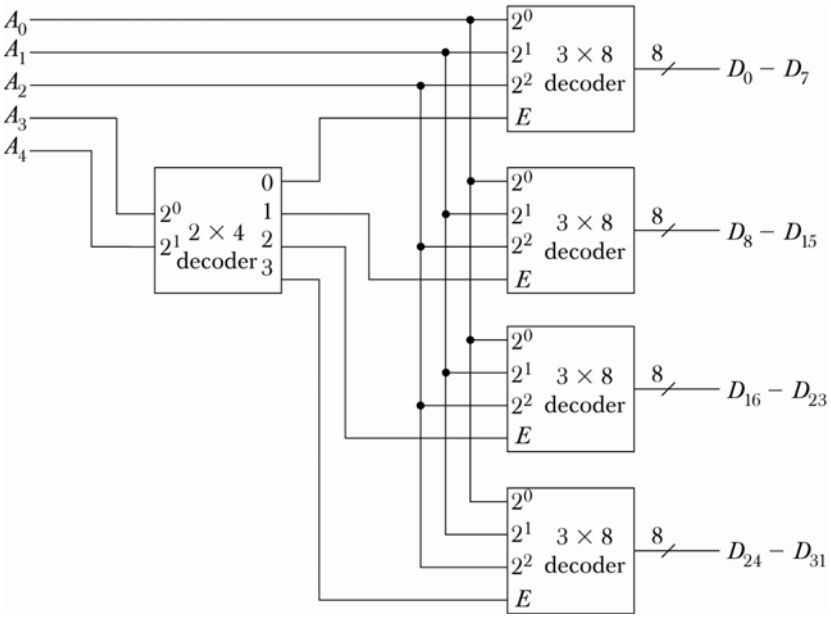
2.1

			TTL JC
(a)	Inverters – 2 pins each	12/2 = 6 gates	7404
(b)	2-input XOR – 3 pins each	12/3 = 4 gates	7486
(c)	3-input OR – 4 pins each	12/4 = 3 gates	
(d)	4-input AND – 5 pins each	12/5 = 2 gates	7421
(e)	5-input NOR – 6 pins each	12/6 = 2 gates	74260
(f)	8-input NAND – 9 pins	1 gate	7430
(g)	JK flip-flop – 6 pins each	12/6 = 2 FFs	74107

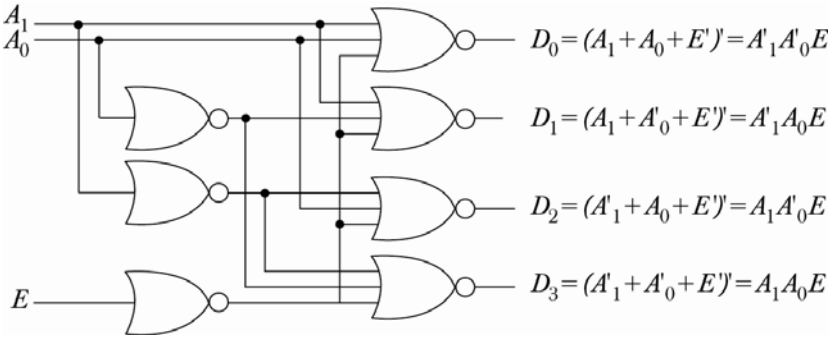
2.2

(a) 74155 – Similar to two decoders as in Fig. 2.2.
(b) 74157 – Similar to multiplexers of Fig. 2.5.
(c) 74194 – Similar to register of Fig. 2.9.
(d) 74163 – Similar to counter of Fig. 2.11

2.3



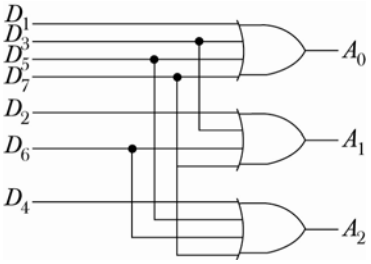
2.4



2.5

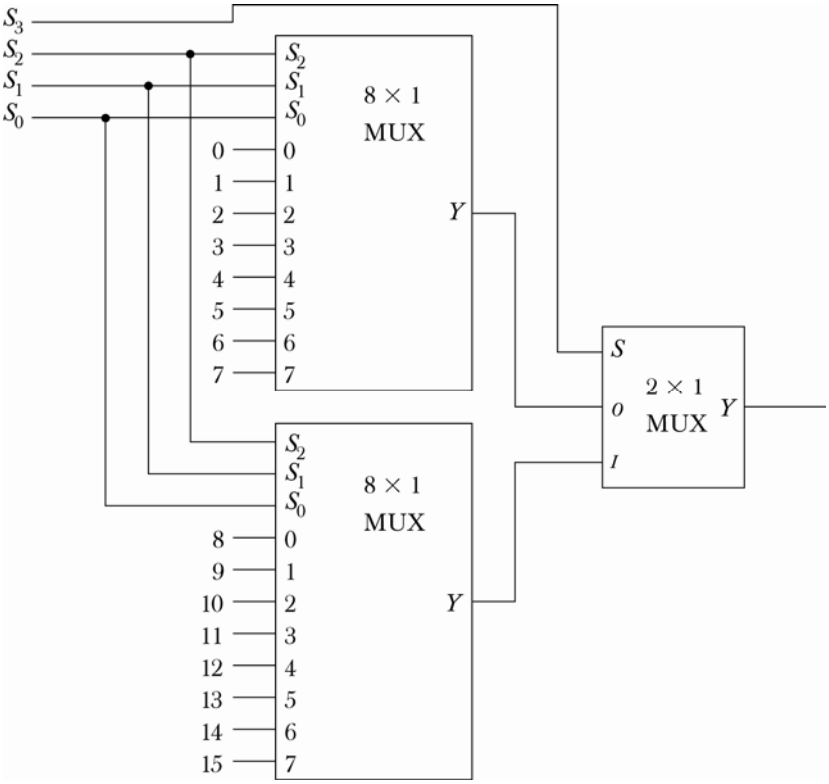
Remove the inverter from the E input in Fig. 2.2(a).

2.6

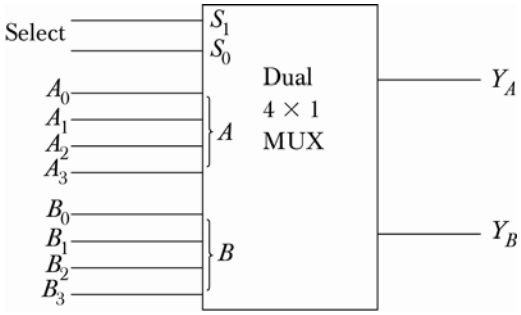


If all inputs equal 0 or if only $D_0 = 1$: the outputs $A_2A_1A_0 = 000$.
Needs one more output to recognize the all zeros input condition.

2.7



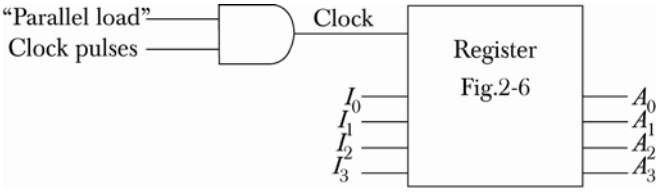
2.8



S_1S_0	$Y_A Y_B$
0 0	A_0 B_0
0 1	A_1 B_1
1 0	A_2 B_2
1 1	A_3 B_3

Function table

2.9

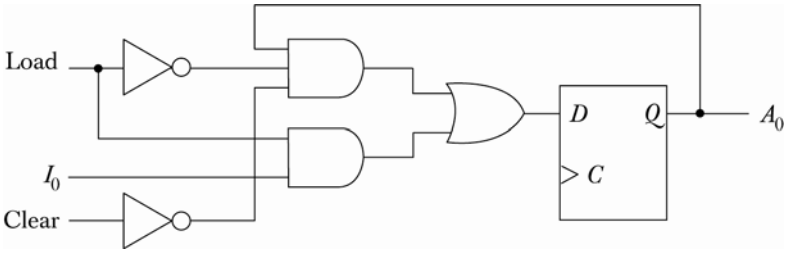


When the parallel load input = 1, the clock pulses go through the AND gate and the data inputs are loaded into the register when the parallel load input = 0, the output of the AND gate remains at 0.

2.10

The buffer gate does not perform logic. It is used for signal amplification of the clock input.

2.11

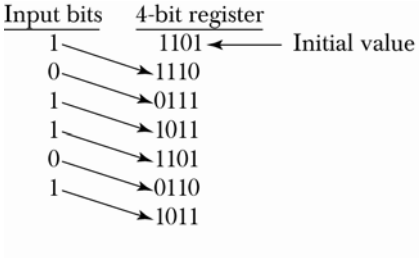


One stage of Register Fig. 2.7

Load	Clear	D	Operation
0	0	Q(t)	no change
0	1	0	Clear to 0
1	x	I_0	load I_0

Function table

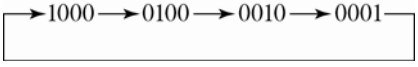
2.12



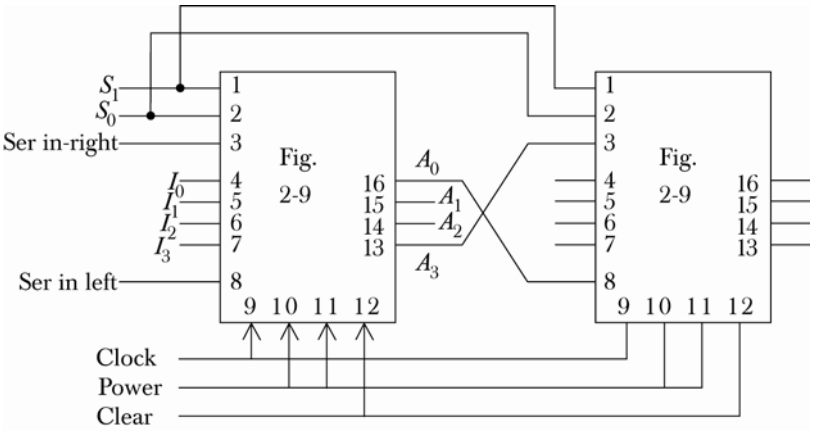
2.13

Serial transfer: One bit at a time by shifting. Parallel transfer: All bits at the same time. Input serial data by shifting–output data in parallel. Input data with parallel load–output data by shifting.

2.14

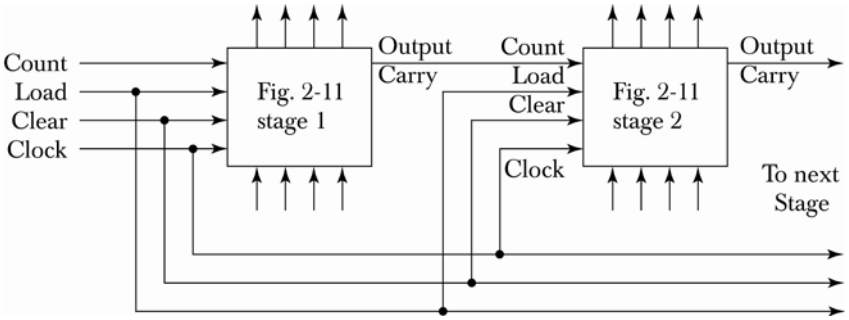


2.15

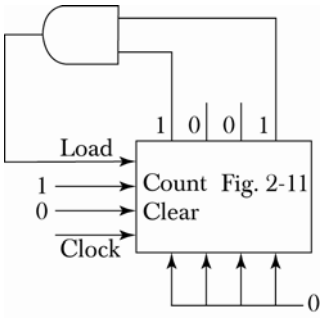


2.16 (a) 4 ; (b) 9

2.17



2.18
After the count reaches N – 1 = 1001, the register loads 0000 from inputs.



2.19

		Address lines	Data lines
(a)	2K × 16 = 2^{11} × 16	11	16
(b)	64K × 8 = 2^{16} × 16	16	8
(c)	16M × 32 = 2^{24} × 32	24	32
(d)	4G × 64 = 2^{32} × 64	32	64

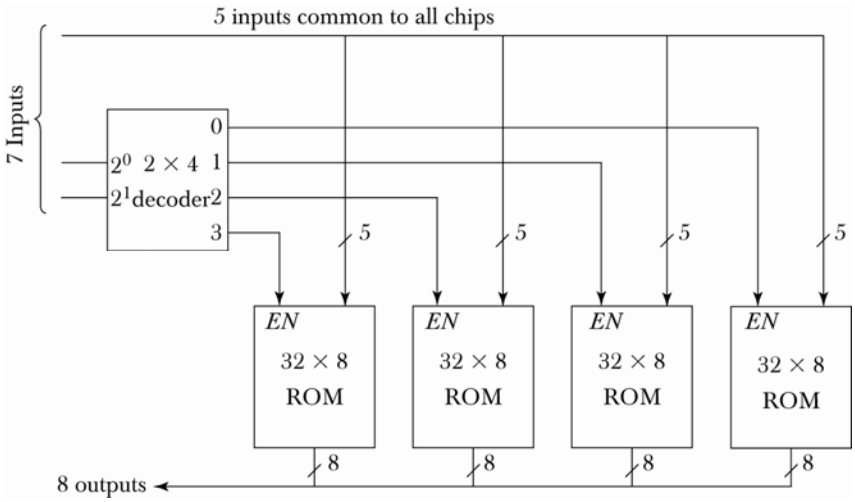
2.20

(a) 2K × 2 = 4K = 4096 bytes
(b) 64K × 1 = 64K = 2^{16} bytes
(c) $2^{24} \times 4 = 2^{26}$ bytes
(d) $2^{32} \times 8 = 2^{35}$ bytes

2.21

$$\frac{4096 \times 16}{128 \times 8} = \frac{2^{12} \times 2^4}{2^7 \times 2^3} = 2^6 = 64 \text{ chips}$$

2.22



2.23

12 data inputs + 2 enable inputs + 8 data outputs + 2 for power = 24 pins.

CHAPTER 3

3.1

$(101110)_2 = 32 + 8 + 4 + 2 = 46$
$(1110101)_2 = 64 + 32 + 16 + 4 + 1 = 117$
$(110110100)_2 = 256 + 128 + 32 + 16 + 4 = 436$

3.2

$(12121)_3 = 3^4 + 2 \times 3^3 + 3^2 + 2 \times 3 + 1 = 81 + 54 + 9 + 6 + 1 = 151$
$(4310)_5 = 4 \times 5^3 + 3 \times 5^2 + 5 = 500 + 75 + 5 = 580$
$(50)_7 = 5 \times 7 = 35$
$(198)_{12} = 12^2 + 9 \times 12 + 8 = 144 + 108 + 8 = 260$

3.3

$(1231)_{10} = 1024 + 128 + 64 + 15 = 2^{10} + 2^7 + 2^6 + 2^3 + 2^2 + 2 + 1 = (10011001111)_2$
$(673)_{10} = 512 + 128 + 32 + 1 = 2^9 + 2^7 + 2^5 + 1 = (1010100001)_2$
$(1998)_{10} = 1024 + 512 + 256 + 128 + 64 + 8 + 4 + 2$
$= 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^3 + 2^2 + 2^1 = (11111001110)_2$

3.4

$(7562)_{10} = (16612)_8$
$(1938)_{10} = (792)_{16}$
$(175)_{10} = (10101111)_2$

3.5

$(F3A7C2)_{16} = (1111\ 0011\ 1010\ 0111\ 1100\ 0010)_2$
$= (74723702)_8$

3.6

$(x^2 - 10x + 31)_r = [(x - 5)(x - 8)]_{10}$
$= x^2 - (5 + 8)_{10}\, x + (40)_{10}\, x$
Therefore: $(10)_r = (13)_{10}$ r = 13
Also $(31)_r = 3 \times 13 + 1 = (40)_{10}$
(r = 13)

3.7

$(215)_{10} = 128 + 64 + 16 + 7 = (11010111)_2$

(a)	000011010111				Binary
(b)	000	011	010	111	Binary coded octal
	0	3	2	7	
(c)	0000	1101	0111		Binary coded hexadecimal
	0	D	7		
(d)	0010	0001	0101		Binary coded decimal
	2	1	5		

3.8

$(295)_{10} = 256 + 32 + 7 = (100100111)_2$

(a)	0000	0000	0000	0001	0010	0111
(b)	0000	0000	0000	0010	1001	0101
(c)	10110010		00111001		00110101	

3.10

JOHN DOE

3.11
87650123; 99019899; 09990048; 999999.

3.12
876100; 909343; 900000; 000000

3.13

01010001;	01111110;	01111111;	11111110;	11111111
01010010;	01111111;	10000000;	11111111;	00000000

3.14

(a) 5250 + 8679 → $1\overline{)3929}$

(b) 1753 + 1360 → $0\overline{)3113}$ ↓ = 10's complement → – 6887

(c) 020 + 900 → $0\overline{)920}$ → – 080

(d) 1200 + 9750 → $1\overline{)0950}$

3.15

(a) 11010 + 10000 → $1\overline{)01010}$ (26 – 16 = 10)

(b) 11010 + 10011 → $1\overline{)01101}$ (26 – 13 = 13)

(c) 000100 + 010000 → $0\overline{)010100}$ ↓ –101100 (4 – 48 = –44)

(d) 1010100 + 0101100 → $1\overline{)0000000}$ (84 – 84 = 0)

3.16

+ 42 =	0101010	+13 =	0001101
– 42 =	1010110	–13 =	1110011
(+42)	0101010	(– 42)	1010110
(–13)	1110011	(+ 13)	0001101
(+29)	0011101	(– 29)	1100011

3.17 01 ← last two carries → 1 0

+ 70	01000110	– 70	10111010
+ 80	01010000	– 80	10110000
+150	10010110	– 150	01101010
↑	↑	↑	↑
greater than +127	negative	less than – 128	positive

3.18

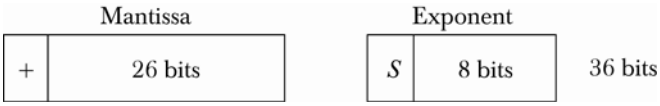
(a)

(–638)		9362
(+785)	+	0785
(+147)		0147

(b)

(–638)		9362
(–185)	+	9815
(–823)		9177

3.19



Largest: + 0.1111 ….1	+ 11111111	
$1–2^{-26}$	+ 255	$(1–2^{–26}) \times 2^{+255}$
Smallest: + 0.1000….0	–11111111	
(normalized) 2^{-1}	–255	$2^{–256}$

3.20

$46.5 = 32 + 8 + 4 + 2 + 0.5 = (101110.1)_2$

Sign

0101110100000000	00000110
24-bit mantissa	8-bit exponent (+6)

3.21 (a)

Decimal	Gray code
16	11000
17	11001
18	11011
19	11010
20	11110
21	11111
22	11101
23	11100
24	10100
25	10101
26	10111
27	10110
28	10010
29	10011
30	10001
31	10000

(b)

Decimal	Exess-3	Gray
9	0010	1010
10	0110	1010
11	0110	1110
12	0110	1111
13	0110	1101
14	0110	1100
15	0110	0100
16	0110	0101
17	0110	0111
18	0110	0110
19	0110	0010
20	0111	0010

3.22 8620

(a)	BCD	1000	0110	0010	0000
(b)	XS-3	1011	1001	0101	0011
(c)	2421	1110	1100	0010	0000
(d)	Binary	10000110101100 (8192 + 256 + 128 + 32 + 8 + 4)			

3.23

Decimal	BCD with even parity	BCD with odd parity
0	00000	10000
1	10001	00001
2	10010	00010
3	00011	10011
4	10100	00100
5	00101	10101
6	00110	10110
7	10111	00111
8	11000	01000
9	01001	11001

3.24

3984 = 0011 1111 1110 0100
= 1100 0000 0001 1011 = 6015

3.25

A B	Y = A ⊕ B
0 0	0
0 1	1
1 0	1
1 1	0

C D	Z = C ⊕ D
0 0	0
0 1	1
1 0	1
1 1	0

y	z	x = y ⊕ z
0	0	0
0	1	1 ← AB = 00 or 11; CD = 01 or 10
1	0	1 ← AB = 01 or 10; CD = 00 or 11
1	1	0

ABCD

0001, 0010, 1101, 1110

0100, 0111, 1000, 1011

Always odd number of 1's

3.26

Same as in Fig. 3.3 but without the complemented circles in the outputs of the gates.
P = x ⊕ y ⊕ z
Error = x ⊕ y ⊕ z ⊕ P