Chapter 7

mdc 1/19/07 11:02 AM
Comment [1]: Spell check

7.1 **(a)** $8\text{ K} \times 32 = 2^{13} \times 16$ A = 13 D = 16

(b) $2\text{ G} \times 8 = 2^{31} \times 8$ A = 31 D = 8

(c) $16\text{ M} \times 32 = 2^{24} \times 32$ A = 24 D = 32

(d) $256\text{ K} \times 64 = 2^{18} \times 64$ A = 18 D = 64

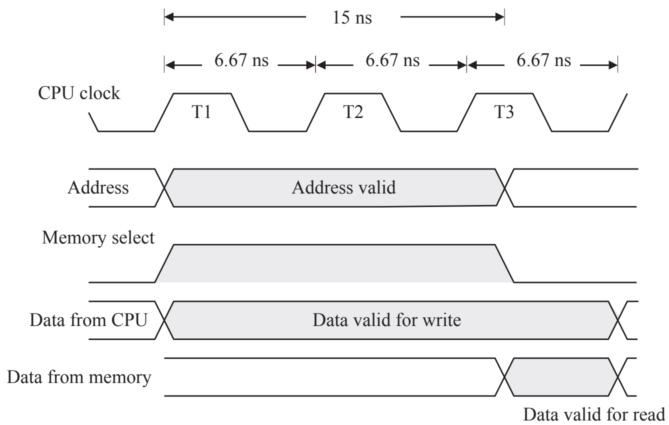
(e)

7.2 **(a)** 2^{13} **(b)** 2^{31} **(c)** 2^{26} **(d)** 2^{21}

7.3 Address: $563_{10} = 10\_0011\_0011_2$

Data word: $1{,}212_{10} = 0000\_0100\_1011\_1100_2$

7.4 $f_{CPU} = 150\text{ MHz}$, $T_{CPU} = 1/f_{CPU} = 6.67^{-9}\text{ Hz}^{-1}$



7.5

Pending

7.7 **(a)** 16 K = 2^{14} = 2^7 x 2^7 = 128 x 128
Each decoder is 7 × 128
Decoders require 256 AND gates, each with 7 inputs

(b) 6,000 = 0101110_1110000
x = 46 *y* = 112

7.8 **(a)** 256 K / 32 K = 8 chips

(b) 256 K = 2^{18} (18 address lines for memory); 32 K = 215 (15 address pins / chip)

(c) 18 – 15 = 3 lines ; must decode with 3 × 8 decoder

7.9 13 + 12 = 25 address lines. Memory capacity = 2^{25} words.

7.10 01011011 = 1 2 3 4 5 6 7 8 9 10 11 12 13
P_1 P_2 0 P_4 1 0 1 P_8 1 0 1 1 P_{13}

P_1 = Xor of bits (3, 5, 7, 9, 11) = 0, 1, 1, 1, 1 = 0 (Note: even # of 0s)
P_2 = Xor of bits (3, 6, 7, 10, 11) = 0, 0, 1, 0, 1 = 0
P_4 = Xor of bits (5, 6, 7, 12) = 1, 0, 1, 1 = 1 (Note: odd # of 0s)
P_8= Xor of bits (9, 10, 11, 12) = 1, 0, 1, 1, = 1

Composite 13-bit code word: 0001 1011 1011 1

7.11 11001001010 = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
P_1 P_2 1 P_4 1 0 0 P_8 1 0 0 1 0 1 0

P_1 = Xor of bits (3, 5, 7, 9, 11, 13, 15) = 1, 1, 0, 1, 0, 0, 0 = 1 (Note: odd # of 0s)
P_2 = Xor of bits (3, 6, 7, 10, 11, 14, 15) = 1, 0, 0, 0, 0, 1, 0 = 0 (Note: even # of 0s)
P_4 = Xor of bits (5, 6, 7, 12, 13, 14, 15) = 1, 0, 0, 1, 0, 1, 0 = 1
P_8= Xor of bits (9, 10, 11, 12, 13, 14, 15) = 1, 0, 0, 1, 0, 1, 0 = 1

Composite 15-bit code word: 101 110 011 001 010

7.12 **(a)** 1 2 3 4 5 6 7 8 9 10 11 12
0 0 0 0 1 1 1 0 1 0 1 0

C_1 (1, 3, 5, 7, 9, 11) = 0, 0, 1, 1, 1, 1 = 0
C_2 (2, 3, 6, 7, 10, 11) = 0, 0, 1, 1, 0, 1 = 1
C_4 (4, 5, 6, 7, 12) = 0, 1, 1, 1, 0 = 1
C_8 (8, 9, 10, 11, 12) = 0, 1, 0, 1, 0 = 0

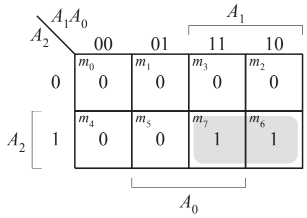
C = 0110
Error in bit 6.
Correct data: 0101 1010

(b) 1 2 3 4 5 6 7 8 9 10 11 12
1 0 1 1 1 0 0 0 0 1 1 0

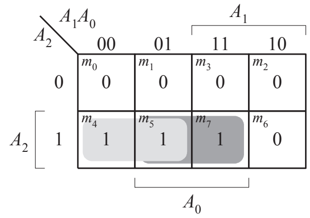
C_1 (1, 3, 5, 7, 9, 11) = 1, 1, 1, 0, 0, 1 = 0
C_2 (2, 3, 6, 7, 10, 11) = 0, 1, 0, 0, 1, 1 = 1
C_4 (4, 5, 6, 7, 12) = 1, 1, 0, 0, 0 = 0
C_8 (8, 9, 10, 11, 12) = 0, 0, 1, 1, 0 = 0

C = 0010

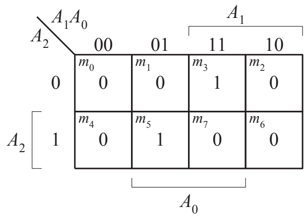
7.21 Note: See truth table in Fig. 7.12(b).



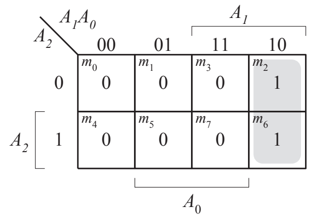
$F_1 = A_2A_1$
$F'_1 = A'_2 + A'_1$



$F_2 = A_2A'_1 + A_2A_0$
$F_2' = A'_2 + A_1A'_0$



$F_3 = A'_2A_1A_0 + A_2A'_1A_0$
$F_3' = A'_0 + A'_2A'_1 + A_2A_1$



$F_4 = A_1A'_0$
$F'_4 = A'_1 + A_0$

	Product term	Inputs $A_2A_1A_0$	Outputs $F_1\ F_2\ F_3\ F_4$
A_2A_1	1	1 1 -	1 - - -
A'_2	2	0 - -	- 1 - -
$A_1A'_0$	3	- 1 0	- 1 - 1
$A'_2A_1A_0$	4	- 1 1	- - 1 -
$A_2A'_1$	5	1 0 1	- - 1 -
			T C T T

Alternative: F'_1, F'_2, F_3, F_4
(5 terms)

7.22

Decimal		w	x	y	z	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0	0	0	0	1
2	4	0	0	1	0	0	0	0	0	0	1	0	0
3	9	0	0	1	1	0	0	0	0	1	0	0	1
4	16	0	1	0	0	0	0	0	1	0	0	0	0
5	25	0	1	0	1	0	0	0	1	1	0	0	1
6	36	0	1	1	0	0	0	1	0	0	1	0	0
7	49	0	1	1	1	0	0	1	1	0	0	0	1
8	64	1	0	0	0	0	1	0	0	0	0	0	0
9	81	1	0	0	1	0	1	0	1	0	0	0	1
10	100	1	0	1	0	0	1	1	0	0	1	0	0
11	121	1	0	1	1	0	1	1	1	1	0	0	1
12	144	1	1	0	0	1	0	0	1	0	0	0	0
13	169	1	1	0	1	1	0	1	0	1	0	0	1
14	196	1	1	1	0	1	1	0	0	0	1	0	0
15	225	1	1	1	1	1	1	1	0	0	0	0	1

Note: $b_0 = z$, and $b_1 = 0$. ROM would have 4 inputs and 6 outputs. A 4 x 8 ROM would waste two outputs.