

Exercise 2.13

(a) $(1001010100011110)_2 = (1001\ 0101\ 0001\ 1110)_2 = (951E)_{16}$

(b) $(3456)_8 = (011\ 100\ 101\ 110)_2 = (011100101110)_2$

(c) To convert from radix-2 to radix- 2^k we consider groups of k bits. The digits in radix- 2^k are obtained converting each group (binary representation of the digit) into a single value in the new radix.

To convert from radix- 2^k to radix-2, the digits in radix- 2^k are converted to binary. The final vector, that corresponds to the concatenation of all digit representations in binary, is the radix-2 representation of the number.

Exercise 2.14

We could directly build the tables of the functions and show that the identity holds by comparing the tables. However, this would be time consuming since the table would have 32 rows. Therefore, we look at different assignments as follows:

- i) if $a = 0$ and $b = 0$ then both sides of the identity have value 0, since $M(x, y, z) = 0$ when two of its arguments are 0.
- ii) if $a = 1$ and $b = 1$ then both sides have value 1, since $M(x, y, z) = 1$ when two arguments are 1.
- iii) the two cases $a \neq b$ are equivalent due to the symmetry of the majority function and the fact that these variables appear together in all functions in the expression. Consequently, we check only the case $a = 0$ and $b = 1$. For this assignment,

$$M[0, 1, M(c, d, e)] = M(c, d, e)$$

and

$$M[M(0, 1, c), d, M(0, 1, e)] = M[c, d, e]$$

The identity is true for all assignments of a and b and is therefore true.

Exercise 2.15

(a) Prove that $f_{\text{XOR}}(f_{\text{AND}}(x_1, x_0), f_{\text{AND}}(x_1, x_0)) = f_{\text{EQUIVALENCE}}(x_1, x_0)$

x_1	x_0	$f_{\text{AND}}(x_1, x_0)$	$f_{\text{XOR}}(f_{\text{AND}}(x_1, x_0), f_{\text{AND}}(x_1, x_0))$	$f_{\text{EQUIVALENCE}}(x_1, x_0)$
0	0	0	0	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

The conclusion is:

$$f_{\text{XOR}}(f_{\text{AND}}(x_1, x_0), f_{\text{AND}}(x_1, x_0)) \neq f_{\text{EQUIVALENCE}}(x_1, x_0)$$

Using one-set and dc-set representation:

- $z_7 = \text{one_set}(11, 12), \text{dc_set}(10, 11, 12, 13, 14, 15)$
- $z_6 = \text{one_set}(8, 9, 10), \text{dc_set}(10, 11, 12, 13, 14, 15)$
- $z_5 = \text{one_set}(3, 4, 5, 6, 7, 9, 10, 12), \text{dc_set}(10, 11, 12, 13, 14, 15)$
- $z_4 = \text{one_set}(3, 4, 5, 6, 7, 8, 10, 11, 12), \text{dc_set}(10, 11, 12, 13, 14, 15)$
- $z_3 = \text{one_set}(6, 7, 8, 9, 10), \text{dc_set}(10, 11, 12, 13, 14, 15)$
- $z_2 = \text{one_set}(4, 5, 10, 11, 12), \text{dc_set}(10, 11, 12, 13, 14, 15)$
- $z_1 = \text{one_set}(3, 5, 7, 11), \text{dc_set}(10, 11, 12, 13, 14, 15)$
- $z_0 = \text{one_set}(3, 4, 5, 6, 7, 9, 11), \text{dc_set}(10, 11, 12, 13, 14, 15)$

Exercise 2.22 Using 2-out-of-5 code, the function specified in Exercise 2.1 is described by the following table. As the table has 32 entries, only the 10 entries that are useful are shown, the others have don't care values:

x	z_9	z_8	z_7	z_6	z_5	z_4	z_3	z_2	z_1	z_0
3	0	0	0	1	1	0	0	0	1	1
5	0	1	0	0	1	1	1	0	0	0
6	0	1	1	0	0	0	0	1	1	0
9	0	0	1	1	0	1	0	0	1	0
10	1	0	1	0	0	0	1	0	1	0
12	0	0	0	1	1	0	0	1	1	0
17	1	0	0	1	0	0	0	1	0	1
18	0	0	0	1	1	1	0	0	0	1
20	0	0	0	1	1	1	0	0	1	0
24	0	0	0	1	1	1	0	1	0	0

Using a two-dimensional table:

$x_4x_3x_2$	x_1x_0			
	00	01	10	11
000	-	-	-	0001100011
001	-	0100111000	0110000110	-
010	-	0011010010	1010001010	-
011	0001100110	-	-	-
100	-	1001000101	0001101001	-
101	0001110010	-	-	-
110	0001110100	-	-	-
111	-	-	-	-

Using one-set and d.c.-set representation:

- $z_9 = \text{one_set}(10, 17),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$

- $z_8 = \text{one_set}(5, 6),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$
- $z_7 = \text{one_set}(6, 7, 8),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$
- $z_6 = \text{one_set}(3, 9, 12, 17, 18, 20, 24),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$
- $z_5 = \text{one_set}(3, 5, 12, 18, 20, 24),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$
- $z_4 = \text{one_set}(5, 9, 18, 20, 24),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$
- $z_3 = \text{one_set}(5, 10),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$
- $z_2 = \text{one_set}(6, 12, 17, 24),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$
- $z_1 = \text{one_set}(3, 6, 9, 10, 12, 20),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$
- $z_0 = \text{one_set}(3, 17, 18),$
 $\text{dc_set}(0, 1, 2, 4, 7, 8, 11, 13, 14, 15, 16, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31)$

Exercise 2.23

The number of values of n binary variables is 2^n . For each of these, the function can take three possible values (0, 1 or *dc*) resulting in 3^{2^n} functions. Out of these 2^{2^n} are completely specified functions.

Exercise 2.24

(a)

$$\begin{aligned}
 a'b' + ab + a'b &= a'(b + b') + ab \\
 &= a'1 + ab \\
 &= a' + b
 \end{aligned}$$

(b)

$$\begin{aligned}
 a' + a(a'b + b'c)' &= a' + (a'b + b'c)' \\
 &= a' + (a + b')(b + c') \\
 &= a' + ab + ac' + bb' + b'c' \\
 &= a' + b + c' + b'c' \\
 &= a' + b + c'
 \end{aligned}$$

abc	$a\#(b\&c)$	$(a\#b)\&(a\#c)$
000	$0\#0 = 0$	$0\&0 = 0$
001	$0\#1 = 0$	$0\&0 = 0$
002	$0\#2 = 0$	$0\&0 = 0$
010	$0\#1 = 0$	$0\&0 = 0$
011	$0\#1 = 0$	$0\&0 = 0$
012	$0\#2 = 0$	$0\&0 = 0$
020	$0\#2 = 0$	$0\&0 = 0$
021	$0\#2 = 0$	$0\&0 = 0$
022	$0\#2 = 0$	$0\&0 = 0$
100	$1\#0 = 0$	$0\&0 = 0$
101	$1\#1 = 1$	$0\&1 = 1$
102	$1\#2 = 1$	$0\&1 = 1$
110	$1\#1 = 1$	$1\&0 = 1$
111	$1\#1 = 1$	$1\&1 = 1$
112	$1\#2 = 1$	$1\&1 = 1$
120	$1\#2 = 1$	$1\&0 = 1$
121	$1\#2 = 1$	$1\&1 = 1$
122	$1\#2 = 1$	$1\&1 = 1$
200	$2\#0 = 0$	$0\&0 = 0$
201	$2\#1 = 1$	$0\&1 = 1$
202	$2\#2 = 2$	$0\&2 = 2$
210	$2\#1 = 1$	$1\&0 = 1$
211	$2\#1 = 1$	$1\&1 = 1$
212	$2\#2 = 2$	$1\&2 = 2$
220	$2\#2 = 2$	$2\&0 = 2$
221	$2\#2 = 2$	$2\&1 = 2$
222	$2\#2 = 2$	$2\&2 = 2$

Table 2.2: Proof of distributivity for system in Exercise 2.29

- (a) $(\#)$, $(\&)$ are commutative because the table is symmetric about the main diagonal, so postulate 1 ($P1$) is satisfied.
- (b) For distributivity, we must show that $a\#(b\&c) = (a\#b)\&(a\#c)$. Let us prove that this postulate is true for the system by perfect induction, as shown in Table 2.2.
- (c) The additive identity element is 0 since $a\&0 = 0\&a = a$. The multiplicative identity element is 2 since $a\#2 = 2\#a = a$.
- (d) For 1 to have a complement $1'$ we need $1\&1' = 2$ and $1\#1' = 0$. Consequently, from the table we see that 1 has no complement.

Exercise 2.30

Let us call the four elements 0, x , y , and 1 with 0 the additive identity, 1 the multiplicative identity and let x be the complement of y . The corresponding operation tables are

+	0	x	y	1
0	0	x	y	1
x	x	x	1	1
y	y	1	y	1
1	1	1	1	1

*	0	x	y	1
0	0	0	0	0
x	0	x	0	x
y	0	0	y	y
1	0	x	y	1

Exercise 2.31

To show that the NAND operator is not associative, that means, $((a.b).c)' \neq (a.(b.c))'$ we just need to find a case when the expressions are different. Take $a = 0$, $b = 0$ and $c = 1$. For these values we have:

$$((a.b).c)' = ((0.0).1)' = (1.1)' = 0$$

$$(a.(b.c))' = (0.(0.1))' = 1$$

So, the NAND operator is not associative.

To show that the NOR operator is not associative, that means, $((a + b)' + c)' \neq (a + (b + c))'$ we take $a = 0$, $b = 0$ and $c = 1$. For these values we have:

$$((a + b)' + c)' = ((0 + 0)' + 1)' = 0$$

$$(a + (b + c))' = (0 + (0 + 1))' = (0 + 0)' = 1$$

So, the NOR operator is not associative.

Exercise 2.32**Part (a)**

(i) show that XOR operator is commutative:

a	b	$a \oplus b$	$b \oplus a$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

(ii) show that XOR operator is associative: $a \oplus (b \oplus c) = (a \oplus b) \oplus c$

abc	$a \oplus b$	$b \oplus c$	$(a \oplus b) \oplus c$	$a \oplus (b \oplus c)$
000	0	0	0	0
001	0	1	1	1
010	1	1	1	1
011	1	0	0	0
100	1	0	1	1
101	1	1	0	0
110	0	1	0	0
111	0	0	1	1

Exercise 2.42

a) We first convert to a sum of products

$$\begin{aligned}
 [(a + b + a'c')c + d]' + ab]' &= \\
 &= (((a + b + a'c')c + d)')'(ab)' \\
 &= ((a + b + a'c')c + d)(a' + b') \\
 &= (ac + bc + d)(a' + b') \\
 &= a'bc + a'd + ab'c + b'd
 \end{aligned}$$

To get the CSP we expand and eliminate repeated minterms

$$\begin{aligned}
 &= a'bc(d + d') + a'd(b + b')(c + c') + ab'c(d + d') + b'd(a + a')(c + c') \\
 &= a'bcd + a'bcd' + a'bc'd + a'b'cd + a'b'c'd + ab'cd + ab'cd' + ab'c'd
 \end{aligned}$$

In m -notation,

$$E(a, b, c, d) = \sum m(1, 3, 5, 6, 7, 9, 10, 11)$$

b) We first get a sum of products

$$\begin{aligned}
 [(w' + (xy + xyz' + (x + z)'))'(z' + w')] &= \\
 &= (w' + (xy + xyz' + x'z'))' + (z' + w')' \\
 &= (w(xy + xyz' + x'z') + zw) \\
 &= wxy + wxyz' + wx'z' + wz
 \end{aligned}$$

Now we expand and eliminate repeated minterms

$$\begin{aligned}
 &= wxy(z + z') + wxyz' + wx'z'(y + y') + wz(x + x')(y + y') \\
 &= wxyz + wxyz' + wx'y'z' + wx'y'z' + wxy'z + wx'y'z + wx'y'z
 \end{aligned}$$

In m -notation,

$$E(w, x, y, z) = \sum m(8, 9, 10, 11, 13, 14, 15)$$

Exercise 2.43

(a) $E(w, x, y, z) = xyz + yw + x'z' + xy'$

$$\begin{aligned}
 E(w, x, y, z) &= x(yz + y') + yw + x'z' \\
 &= x(y' + z) + yw + x'z' \\
 &= (x(y' + z) + yw + x')(x(y' + z) + yw + z') \\
 &= (y' + z + yw + x')(xy' + z' + xz + yw) \\
 &= (w + x' + y' + z)(xy' + x + z' + yw) \\
 &= (w + x' + y' + z)(x + z' + yw) \\
 &= (w + x' + y' + z)(w + x + z')(x + y + z') \\
 &= (x + x' + y' + z)(w + x + y + z')(w + x + y' + z')(w' + x + y + z')
 \end{aligned}$$

$$(b) E(a, b, c, d) = (abc + ab')'(a'b + c')$$

$$\begin{aligned} E(a, b, c, d) &= (a' + b' + c')(a' + b)(a' + c')(b + c') \\ &= (a' + b' + c')(a' + b + c)(a' + b + c')(a + b + c') \end{aligned}$$

Exercise 2.44

$$(a) \sum m(0, 1, 3, 4, 11, 12, 14, 15) = \prod M(2, 5, 6, 7, 8, 9, 10, 13)$$

$$(b) \prod M(0, 1, 3, 5, 6, 9, 10, 13) = \sum m(2, 4, 7, 8, 11, 12, 14, 15)$$

Exercise 2.45

$$f(x, y, z) = \prod M(0, 1, 4, 6, 7)$$

$$g(x, y, z) = (f(x, y, z))'$$

$$g(x, y, z) = \sum m(0, 1, 4, 6, 7) = \prod M(2, 3, 5)$$

Exercise 2.46

(a)

Input: $A, B \in \{0, 1, 2, 3\}$ and $C_{in} \in \{0, 1\}$ Output: $Z \in \{0, 1, 2, 3\}$ and $C_{out} \in \{0, 1\}$

$$Z = (A + B + C_{in}) \bmod 4$$

$$C_{out} = \begin{cases} 1 & \text{if } (A + B + C_{in}) \geq 4 \\ 0 & \text{otherwise} \end{cases}$$