Homework Solution #1

Chapter 2
2.6, 2.17, 2.24, 2.30, 2.39, 2.42, 2.48

Grading policy is as follows:
- Total 100

Exercise 2.6 (total 10)
- Column 3 : 5 point
- Column 4 : 5 point

Exercise 2.17 (total 25)
- a), d), g) : 5 point
- others : 2 point each

Exercise 2.24 (total 15)
- each of sub-item : 2.5 point

Exercise 2.30 (total 10)
- Each one functional table has 5 point

Exercise 2.39 (total 10)
- minterm/maxterm expression has 5 point each

Exercise 2.42 (total 10)
- a), b) has 5 point each

Exercise 2.48 (total 20)
- a), b), c) : 5 point each
- One set result has 5 point
A tabular representation would not be practical because there are $4 \times 4 = 16$ variables, each of which can have 4 values, resulting in a table of $4^{16}$ rows.

**Exercise 2.5**

Inputs: Integer $0 \leq x < 2^{16}$, binary control variable $d \in \{0, 1\}$

Output: Integer $0 \leq z < 2^{16}$.

$$z = \begin{cases} 
(x + 1) \mod 2^{16} & \text{if } d = 1 \\
(x \leftrightarrow 1) \mod 2^{16} & \text{if } d = 0
\end{cases}$$

Tabular representation: Requires $2^{16}$ rows for $d = 1$ and $2^{16}$ rows for $d = 0$.

Total rows $= 2 \times 2^{16} = 2^{17}$. A tabular representation is out of question.

**Exercise 2.6**

<table>
<thead>
<tr>
<th>$a$</th>
<th>$b$</th>
<th>$f_1(a, b)$</th>
<th>$f_2(b, f_1(a, b))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Exercise 2.7**

Range from 10 to 25 $\Rightarrow$ 16 values.

Minimum number of binary variables: $[\log_2 16] = 4$ variables.

Input: integer $10 \leq x \leq 25$  
Output: integer $0 \leq z \leq 15$

$$z = (x \leftrightarrow 10)$$

In the next table the output $z$ is represented by a vector $z = (z_3, z_2, z_1, z_0)$, with $z_i \in \{0, 1\}$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000</td>
<td>0001</td>
<td>0010</td>
<td>0011</td>
<td>0100</td>
<td>0101</td>
<td>0110</td>
<td>0111</td>
<td>1000</td>
<td>1001</td>
<td>1010</td>
<td>1011</td>
<td>1100</td>
<td>1101</td>
<td>1110</td>
<td>1111</td>
</tr>
</tbody>
</table>

**Exercise 2.8**

(a) Month: $[\log_2 12] = 4$ bits

<table>
<thead>
<tr>
<th>Month</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>0000</td>
<td>0001</td>
<td>0010</td>
<td>0011</td>
<td>0100</td>
<td>0101</td>
<td>0110</td>
<td>0111</td>
<td>1000</td>
<td>1001</td>
<td>1010</td>
<td>1011</td>
</tr>
</tbody>
</table>

(b) Day: $[\log_2 31] = 5$ bits

Year (assuming 0-2500): $[\log_2 2501] = 12$ bits

Month: 4 bits

A total of 21 bits would be needed.

(c) Each decimal digit needs 4 bits. Two digits are necessary to represent the day, and 4 digits to represent the year. The number of bits to represent these fields would be $6 \times 4 = 24$ bits. Adding
(b) Prove that \( f_{\text{NAND}}(f_{\text{NAND}}(x_1, x_0), f_{\text{NAND}}(x_1, x_0)) = f_{\text{AND}}(x_1, x_0) \)

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_0 )</th>
<th>( f_{\text{NAND}}(x_1, x_0) )</th>
<th>( f_{\text{NAND}}(f_{\text{NAND}}(x_1, x_0), f_{\text{NAND}}(x_1, x_0)) )</th>
<th>( f_{\text{AND}}(x_1, x_0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The conclusion is:

\[ f_{\text{NAND}}(f_{\text{NAND}}(x_1, x_0), f_{\text{NAND}}(x_1, x_0)) = f_{\text{AND}}(x_1, x_0) \]

Exercise 2.16

Each variable can have 2 values (0 or 1).

Total number of \( n \)-variable inputs: \( 2 \cdot 2 \cdot \ldots \cdot 2 = 2^n \)

For each input, the output function can have 2 values.

Total number of functions: \( 2 \cdot 2 \cdot \ldots \cdot 2 = 2^{2^n} \)

Exercise 2.17

a) Let \( f \) be a symmetric switching function of three variables, \( x, y, \) and \( z \). Since the function is symmetric, \( f(0, 0, 1) = f(0, 1, 0) = f(1, 0, 0) \) and \( f(0, 1, 1) = f(1, 0, 1) = f(1, 1, 0) \) so that we have the following table:

<table>
<thead>
<tr>
<th>( x )</th>
<th>( y )</th>
<th>( z )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( a )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( b )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>( b )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( c )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( b )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( c )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>( c )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( d )</td>
</tr>
</tbody>
</table>

where \( a, b, c, \) and \( d \) are binary variables. From this description any particular example can be generated by assigning values to \( a, b, c, \) and \( d \).

b) Since \( a, b, c, \) and \( d \) can each take two values (0 or 1), the number of symmetric functions of three variables is \( 2^4 = 16 \).

c) A symmetric switching function has the same value for all argument values that have the same number of 1’s, since all these values can be obtained by permuting the arguments. Consequently, the values of the function of \( n \) arguments are decomposed into \( n + 1 \) classes defined by the number of 1’s in the argument vector (four classes in part a). Therefore the set \( A \) completely defines the function.

d) The function has value 1 whenever 0, 2 or 3 arguments have value 1. The table is
e) In part (c) we saw that the argument values of symmetric switching function of \(n\) variables are divided into \(n+1\) classes. For each of these classes the function can have value 1 or 0. Consequently, the number of symmetric switching functions of \(n\) variables is \(2^{n+1}\).

f) No, the composition of symmetric switching functions is not necessarily a symmetric function. Consider as counterexample

\[
f(x,y,z) = f_{\text{AND}}(f_{\text{OR}}(x,y),z)
\]

and interchange the variables \(x\) and \(z\).

g) The table is

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>(f_1)</th>
<th>(f_2)</th>
<th>(f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

h) Let \(U\) be the set \(\{0,1,\ldots,n\}\) where \(n\) is the number of variables of the symmetric function, and let \(A\) be the set describing the function \(f\). Since the complement of \(f\) is 0 when \(f\) is 1 and vice versa, it is represented by the set \(A_c\) such that

\[
A_c = U \iff A
\]

For example, considering a 4-variable symmetrical function with \(A = \{0,1\}\), we have \(A_c = \{2,3,4\}\).

**Exercise 2.18**

(a) The threshold switching function of three variables with \(w_1 = 1\), \(w_2 = 2\), \(w_3 = \equiv 1\), and \(T = 2\) is:
• $z_8 = \text{one} \set(5,6)$,  
  $\set(0,1,2,4,7,8,11,13,14,15,16,19,21,22,23,25,26,27,28,29,30,31)$

• $z_7 = \text{one} \set(6,7,8)$,  
  $\set(0,1,2,4,7,8,11,13,14,15,16,19,21,22,23,25,26,27,28,29,30,31)$

• $z_6 = \text{one} \set(3,9,12,17,18,20,24)$,  
  $\set(0,1,2,4,7,8,11,13,14,15,16,19,21,22,23,25,26,27,28,29,30,31)$

• $z_5 = \text{one} \set(3,5,12,18,20,24)$,  
  $\set(0,1,2,4,7,8,11,13,14,15,16,19,21,22,23,25,26,27,28,29,30,31)$

• $z_4 = \text{one} \set(5,9,18,20,24)$,  
  $\set(0,1,2,4,7,8,11,13,14,15,16,19,21,22,23,25,26,27,28,29,30,31)$

• $z_3 = \text{one} \set(5,10)$,  
  $\set(0,1,2,4,7,8,11,13,14,15,16,19,21,22,23,25,26,27,28,29,30,31)$

• $z_2 = \text{one} \set(6,12,17,24)$,  
  $\set(0,1,2,4,7,8,11,13,14,15,16,19,21,22,23,25,26,27,28,29,30,31)$

• $z_1 = \text{one} \set(3,6,9,10,12,20)$,  
  $\set(0,1,2,4,7,8,11,13,14,15,16,19,21,22,23,25,26,27,28,29,30,31)$

• $z_0 = \text{one} \set(3,17,18)$,  
  $\set(0,1,2,4,7,8,11,13,14,15,16,19,21,22,23,25,26,27,28,29,30,31)$

**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)  

\[ a'b' + ab + a'b = a'(b+b') + ab = a'1 + ab = a' + b \]

(b)  

\[ a' + a(a'b + b'c)' = a' + (a'b + b'c)' = a' + (a + b')(b + c') = a' + ab + ac' + bb' + b'c' = a' + b + c' + b'c' = a' + b + c' \]
(c)
\[(a'b + c)(a + b)(b' + ac)' = (a'b' + c)(a + b)b(a' + c')
= (a'b' + c)b(a' + c')
= (b'd' + bc)(a' + c')
= bc(a' + c')
= a'bc\]

(d)
\[a'b + b'd' + a'c' = a'b + (a + a')b'd' + a'c'
= a'b + a'b'd' + a'b'c' + a'c'
= ab(1 + c') + a'(1 + b')c'
= ab' + ac'\]

(e)
\[wx'y(xz + yz') + x'(zw + y'z') + z(x'w' + y'x) = wx'y + u'xy + z(x'(w + y') + x'w' + y'x)
= (w + w')xy + z(x'(w + y' + u') + y'x)
= xy + z(x' + xy')
= xy + z(x' + y')
= xy + z(xy')'
= xy + z\]

(f)
\[abc' + bc'd + a'bd = abc' + (a + a')bc'd + a'bd
= abc' + abc'd + a'bc'd + a'bd
= abc'(1 + d') + a'bd(1 + c')
= abc' + a'bd\]

Exercise 2.25
\[xz' + x'z =
= x(xy' + x'y') + x'(xy' + x'y)
= x(xy')' + x'xy + x'x'y
= x(x' + y)(x + y') + x'y
= (xx' + xy)(x + y') + x'y
= (xyx + xy'y') + x'y
= xy + x'y
= (x + x')y
= 1 \cdot y\]
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
Exercise 2.31
To show that the NAND operator is not associative, that means, \((a \cdot b)' \cdot c)' \neq (a \cdot (b \cdot c)')'\) 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:
\[
((a \cdot b)' \cdot c)' = ((0.0)' \cdot 1)' = (1.1)' = 0
\]
\[
(a \cdot (b \cdot c)')' = (0.(0.1)')' = 1
\]
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:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>a + b</th>
<th>b ⊕ a</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>abc</th>
<th>a ⊕ b</th>
<th>b ⊕ c</th>
<th>(a ⊕ b) ⊕ c</th>
<th>a ⊕ (b ⊕ c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>110</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Exercise 2.37

<table>
<thead>
<tr>
<th></th>
<th>Expr a</th>
<th>Expr b</th>
<th>Expr c</th>
<th>Expr d</th>
<th>Expr e</th>
</tr>
</thead>
<tbody>
<tr>
<td>x=0</td>
<td>y=0</td>
<td>z=0</td>
<td>x′y′ + xz + xz′</td>
<td>xy + x′y′ + yz′</td>
<td>xyz + x′y′z + xz′ + xy′z′</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x=0</td>
<td>y=0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x=0</td>
<td>y=0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x=0</td>
<td>y=0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x=0</td>
<td>y=0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Equivalent expressions: Expr a and Expr d, Expr b and Expr c.

Exercise 2.38

(a) \( E(a, b, c, d) = ab'c'd + ab'c + be'd + ab'c'd' + acd + a'b'c'd \)

\[
E(a, b, c, d) = \begin{align*}
&= c'd(ab + b) + ab'(c + d') + (a + a'b)c'd \\
&= be'd + ab' + (a + b)cd \\
&= bd(c + d') + ab' + acd \\
&= bd + ab' + a(b + b')cd \\
&= bd(1 + ac) + ab'(1 + cd) \\
&= ab' + bd
\end{align*}
\]

(b) \( E(a, b, c, d) = acb + ac'd' + be'd' + ab'c'd' + be'd \)

\[
\begin{align*}
E(a, b, c, d) &= bc'(d + d') + b'c'(a' + ad') + acb + ac'd' \\
&= bc' + bc'a' + b'c'd' + acb + ac'd' \\
&= c'(b + b'a' + b'd' + ad) + acb \\
&= c'(b + a' + d + d') + acb \\
&= c' + acb \\
&= ab + c'
\end{align*}
\]

Exercise 2.39

\( f = d'b'c + d'bc' + d'bc + ab'c = m_1 + m_2 + m_3 + m_5 = \sum m(1, 2, 3, 5) \)

\( f = (a + b + c)(a' + b + c)(d' + b' + c)(a' + b' + c') = M_0 + M_4 + M_6 + M_7 = \prod M(0, 4, 6, 7) \)

Exercise 2.40

(a) \( a'b + ac + bc \)

\[
\begin{align*}
a'b + ac + bc &= a'b(c + d') + a(b + b')c + (a + a'd)bc \\
&= a'bc + a'dc' + abc + ab'c + abc + a'dbc \\
&= m_3 + m_2 + m_7 + m_5 + m_7 + m_3 \\
&= \sum m(2, 3, 5, 7) = \prod M(0, 1, 4, 6)
\end{align*}
\]
Exercise 2.42
a) We first convert to a sum of products

\[
[(a + b + a'c')c + d]' + ab' = \\
= (((a + b + a'c')c + d)'(ab')) \\
= ((a + b + a'c')(c + d)(a' + b)) \\
= (ac + bc + d)(a' + b') \\
= a'bc + a'd + ab'c + b'd
\]

To get the CSP we expand and eliminate repeated minterms

\[
= a'bc(d + d') + a'd(b + b')(c + c') + ab'c(d + d') + b'd(a + a')(c + c') \\
= a'bc + a'bcd' + a'bc'd + a'bd' + ab'cd + ab'cd' + ab'cd + ab'cd'
\]

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

\[
[(w' + (xy + xyz' + (x + z)')(z' + w'))]' = \\
= (w' + (xy + xyz' + x'z')(z' + w'))' \\
= (w(xy + xyz' + x'z') + zw) \\
= wxy + wxyz' + wx'y'z + wz
\]

Now we expand and eliminate repeated minterms

\[
= wxy(z + z') + wxyz' + wx'y'(y + y')wz(x + x')(y + y') \\
= wxyz + wxyz' + wx'yz' + wx'y'z + wxy'z + wx'y'z + wz
\]

In m-notation,

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

Exercise 2.43
a) \(E(w, x, y, z) = xyz + yw + x'z' + xy'\)

\[
E(w, x, y, z) = x(zy + y') + yw + x'z' \\
= x(y' + z) + yw + x'z' \\
= (xy' + z + yw + x')(x(y' + z) + yw + z') \\
= (y' + z + yw + x')(xy' + z' + xz + yw) \\
= (w + x' + y' + z)(xy' + x + z' + yw) \\
= (w + x' + y' + z)(x + z' + yw) \\
= (w + x' + y' + z)(w + x + z')(x + y + z') \\
= (x + x' + y' + z)(w + x + y + z')(w + x + y' + z')(w' + x + y + z')
\]
zero-set($z_1$)=$\{1, 2, 5, 6, 9, 10, 13, 14, 16, 19, 20, 23, 24, 27, 28, 31\}$
zero-set($z_0$)=$\{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 19, 31\}$

Exercise 2.48
(a) A high-level description is

- Input: $\mathcal{I} = (x_3, x_2, x_1, x_0)$, and $x_i \in \{0, 1\}$
- Output: $z \in \{0, 1, 2, 3, 4\}$
- Function:

$$z = \sum_{i=0}^{3} x_i$$

(b) The table for arithmetic function and (c) the table for the switching functions considering binary code for $z$ are shown next:

<table>
<thead>
<tr>
<th>$x_3\hspace{0.1cm}x_2\hspace{0.1cm}x_1\hspace{0.1cm}x_0$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>1</td>
</tr>
<tr>
<td>0011</td>
<td>2</td>
</tr>
<tr>
<td>0100</td>
<td>1</td>
</tr>
<tr>
<td>0101</td>
<td>2</td>
</tr>
<tr>
<td>0110</td>
<td>2</td>
</tr>
<tr>
<td>0111</td>
<td>3</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>1001</td>
<td>2</td>
</tr>
<tr>
<td>1010</td>
<td>2</td>
</tr>
<tr>
<td>1011</td>
<td>3</td>
</tr>
<tr>
<td>1100</td>
<td>2</td>
</tr>
<tr>
<td>1101</td>
<td>3</td>
</tr>
<tr>
<td>1110</td>
<td>3</td>
</tr>
<tr>
<td>1111</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_3\hspace{0.1cm}x_2\hspace{0.1cm}x_1\hspace{0.1cm}x_0$</th>
<th>$z_{25\hspace{0.1cm}30}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>000</td>
</tr>
<tr>
<td>0001</td>
<td>001</td>
</tr>
<tr>
<td>0010</td>
<td>0010</td>
</tr>
<tr>
<td>0011</td>
<td>0111</td>
</tr>
<tr>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>0101</td>
<td>0101</td>
</tr>
<tr>
<td>0110</td>
<td>0110</td>
</tr>
<tr>
<td>0111</td>
<td>0111</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1001</td>
<td>1001</td>
</tr>
<tr>
<td>1010</td>
<td>1010</td>
</tr>
<tr>
<td>1011</td>
<td>1011</td>
</tr>
<tr>
<td>1100</td>
<td>1100</td>
</tr>
<tr>
<td>1101</td>
<td>1101</td>
</tr>
<tr>
<td>1110</td>
<td>1110</td>
</tr>
<tr>
<td>1111</td>
<td>1111</td>
</tr>
</tbody>
</table>

The one-sets that correspond to this table are:

one-set($z_2$)=$\{15\}$
one-set($z_1$)=$\{3, 5, 6, 7, 9, 10, 11, 12, 13, 14\}$
one-set($z_0$)=$\{1, 2, 4, 7, 8, 11, 13, 14\}$

Exercise 2.49
(a)

- Inputs: $x, y$ where $x, y \in \{0, 1, 2, 3\}$
- Output: $z \in \{0, 1, 2, 3, 4, 6, 9\}$
- Function: $z = x \cdot y$

(b) The table for the arithmetic function is