Boolean algebra Homework. Dmitry Semenov, M3100, ISU 409537

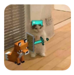
1. Perform the following steps:

(a) Calculate the SHA-256 hash h of the string s = “DM Fall 2023 HW3” (without quotes, with all spaces, encoded in UTF-8). Convert hash h to a 256-bit binary string b (prepend leading zeros if necessary). Cut the binary string b into eight 32-bit slices $r_1, \dots, r_8$, e.g. $r_2 = b_{33 \dots 64}$. XOR all slices into a 32-bit string $d = r_1 \oplus \cdots \oplus r_8$. Compute $w = d \oplus$ 0x24d03294.

The code that calculates the hash and based on it, it performs the operations described above, is located in the repository:

GitHub - W-y-l-t/Discrete-Math
Contribute to W-y-l-t/Discrete-Math development by creating an account on GitHub.
https://github.com/W-y-l-t/Discrete-Math

W-y-l-t/**Discrete-Math**



1 Contributor · 0 Issues · 0 Stars · 0 Forks

As a result, the value of w = 00010011101000100101010101010101

(b) Draw the Karnaugh map (use a template below) for a function $f\ (A, B, C, D, E)$ defined by the truth table $w = (w_1 \dots w_{32})$, where MSB corresponds to $f(0) = w_1$ and LSB to $f(1) = w_{32}$.

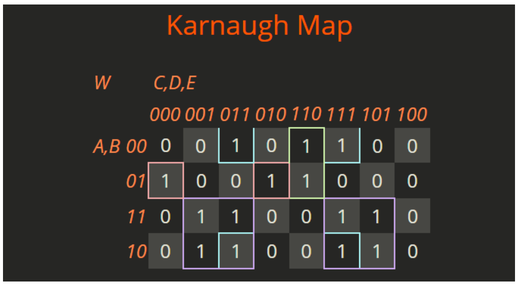
Using the above code we get the Karnaugh map:

```
The Karnaugh map
       000   001   011   010   110   111   101   100
    ------------------------------------------------
00 |    0     0     1     0     1     1     0     0
01 |    1     0     0     1     1     0     0     0
11 |    0     1     1     0     0     1     1     0
10 |    0     1     1     0     0     1     1     0
```

(c) Use K-map to find the minimal DNF and minimal CNF for the function f.

Minimal DNF:

$$f(A,B,C,D,E) = (A \wedge E) \vee (\overline{A} \wedge B \wedge \overline{C} \wedge \overline{E}) \vee (\overline{A} \wedge C \wedge D \wedge \overline{E}) \vee (\overline{B} \wedge D \wedge E)$$



Karnaugh Map

W	C,D,E: 000	001	011	010	110	111	101	100
A,B 00	0	0	1	0	1	1	0	0
01	1	0	0	1	1	0	0	0
11	0	1	1	0	0	1	1	0
10	0	1	1	0	0	1	1	0

Minimal CNF:

$$f(A,B,C,D,E) = \overline{(A \wedge \overline{E}) \vee (\overline{B} \wedge \overline{C} \wedge \overline{E}) \vee (\overline{A} \wedge \overline{D} \wedge E) \vee (\overline{A} \wedge B \wedge E) \vee (\overline{A} \wedge C\overline{D})}$$

$$f(A,B,C,D,E) = (B \vee C \vee E) \wedge (A \vee \overline{B} \vee \overline{E}) \wedge (A \vee D \vee \overline{E}) \wedge (A \wedge \overline{C} \vee D) \wedge (\overline{A} \vee E)$$

Karnaugh Map

W: A,B \ C,D,E	000	001	011	010	110	111	101	100
00	0	0	1	0	1	1	0	0
01	1	0	0	1	1	0	0	0
11	0	1	1	0	0	1	1	0
10	0	1	1	0	0	1	1	0

(d) Use K-map to find the number of prime implicants, i.e. the size of BCF.

BCF:

$f(A, B, C, D, E) = (A \wedge E) \vee (\overline{A} \wedge B \wedge \overline{C} \wedge \overline{E}) \vee (\overline{A} \wedge C \wedge D \wedge \overline{E}) \vee (\overline{B} \wedge D \wedge E) \vee (\overline{A} \wedge B \wedge D \wedge \overline{E}) \vee (\overline{A} \wedge \overline{B} \wedge C \wedge D)$

Hence, size of BCF = 6.

Karnaugh Map

W: A,B \ C,D,E	000	001	011	010	110	111	101	100
00	0	0	1	0	1	1	0	0
01	1	0	0	1	1	0	0	0
11	0	1	1	0	0	1	1	0
10	0	1	1	0	0	1	1	0

2. For each given function f_i of 4 arguments, draw the Karnaugh map and use it to find BCF, minimal DNF, and minimal CNF. Additionally, construct ANF (Zhegalkin polynomial) using either the K-map, the tabular ("triangle") method or the Pascal method — use each method at least once.

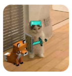
The code that can build a Karnaugh Map and Zhegalkin polynomial using a Boolean function specified in four different ways is located in the repository:

GitHub - W-y-l-t/Discrete-Math

Contribute to W-y-l-t/Discrete-Math development by creating an account on GitHub.

https://github.com/W-y-l-t/Discrete-Math

W-y-l-t/**Discrete-Math**



1 Contributor · 0 Issues · 0 Stars · 0 Forks

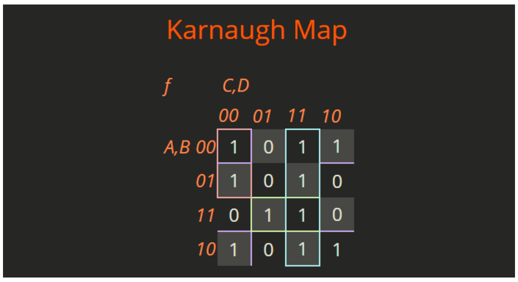
(a) $f_1 = f^{(4)}_{47541}$

Karnaugh Map (using a program from the repository):

```
The Karnaugh map
        00    01    11    10
     ------------------------
00 |    1     0     1     1
01 |    1     0     1     0
11 |    0     1     1     0
10 |    1     0     1     1
```

Minimal DNF:

$f_1(A, B, C, D) = (\overline{B} \wedge \overline{D}) \vee (C \wedge D) \vee (A \wedge B \wedge D) \vee (\overline{A} \wedge \overline{C} \wedge \overline{D})$



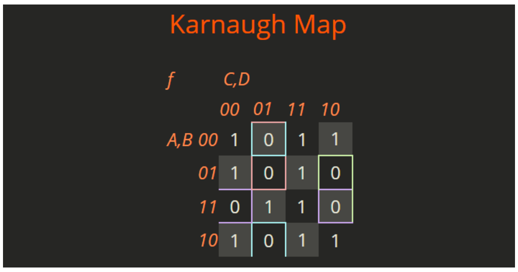
Karnaugh Map

A,B \ C,D	00	01	11	10
00	1	0	1	1
01	1	0	1	0
11	0	1	1	0
10	1	0	1	1

Minimal CNF:

$$f_1(A,B,C,D) = \overline{(A \wedge B \wedge \overline{D}) \vee (\overline{B} \wedge \overline{C} \wedge D) \vee (\overline{A} \wedge \overline{C} \wedge D) \vee (B \wedge C \wedge \overline{D})}$$

$$f_1(A,B,C,D) = (B \vee C \vee \overline{D}) \wedge (\overline{B} \vee \overline{C} \vee D) \wedge (A \vee C \vee \overline{D}) \wedge (\overline{A} \vee \overline{B} \vee D)$$

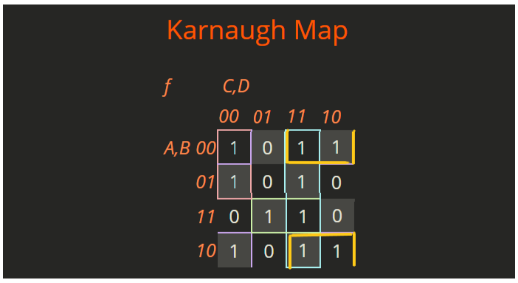


Karnaugh Map

A,B \ C,D	00	01	11	10
00	1	0	1	1
01	1	0	1	0
11	0	1	1	0
10	1	0	1	1

BCF:

$$f_1(A,B,C,D) = (\overline{B} \wedge \overline{D}) \vee (C \wedge D) \vee (A \wedge B \wedge D) \vee (\overline{A} \wedge \overline{C} \vee \overline{D}) \vee (\overline{B} \wedge C)$$



Karnaugh Map

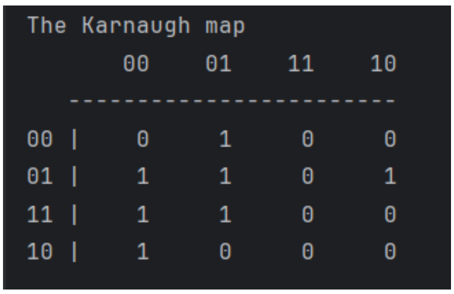
A,B \ C,D	00	01	11	10
00	1	0	1	1
01	1	0	1	0
11	0	1	1	0
10	1	0	1	1

Construct ANF (Zhegalkin polynomial) using Karnaugh map (using a program from the repository):

$$f_{1\oplus}(A,B,C,D) = 1 \oplus D \oplus AB \oplus BC \oplus CD \oplus ABC \oplus BCD$$

(b) $f_2 = \sum m(1,4,5,6,8,12,13)$

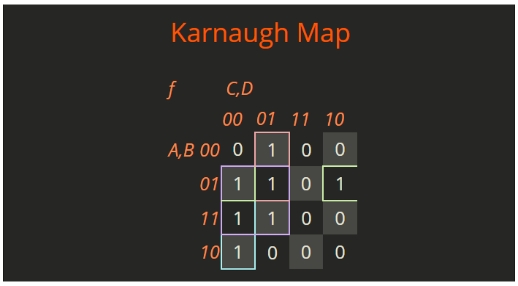
Karnaugh Map (using a program from the repository):



```
The Karnaugh map
        00    01    11    10
   ------------------------
00 |    0     1     0     0
01 |    1     1     0     1
11 |    1     1     0     0
10 |    1     0     0     0
```

Minimal DNF:

$$f_2(A,B,C,D) = (\overline{A} \wedge B \wedge \overline{D}) \vee (B \wedge \overline{C}) \vee (A \wedge \overline{C} \wedge \overline{D}) \vee (\overline{A} \wedge \overline{C} \wedge D)$$



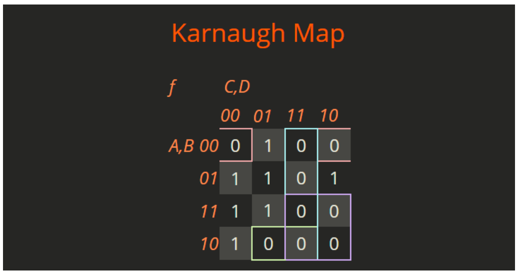
Karnaugh Map

f (A,B \ C,D)	00	01	11	10
00	0	1	0	0
01	1	1	0	1
11	1	1	0	0
10	1	0	0	0

Minimal CNF:

$$f_2(A,B,C,D) = \overline{(A \wedge C) \vee (C \wedge D) \vee (\overline{A} \wedge \overline{B} \wedge \overline{D}) \vee (A \wedge \overline{B} \wedge D)}$$

$$f_2(A,B,C,D) = (\overline{A} \vee \overline{C}) \wedge (\overline{C} \vee \overline{D}) \wedge (A \vee B \vee D) \wedge (\overline{A} \vee B \vee \overline{D})$$

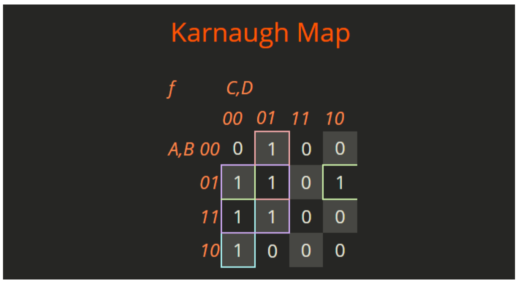


Karnaugh Map

f (A,B \ C,D)	00	01	11	10
00	0	1	0	0
01	1	1	0	1
11	1	1	0	0
10	1	0	0	0

BCF:

$$f_2(A,B,C,D) = (\overline{A} \wedge B \wedge \overline{D}) \vee (B \wedge \overline{C}) \vee (A \wedge \overline{C} \wedge \overline{D}) \vee (\overline{A} \wedge \overline{C} \wedge D)$$



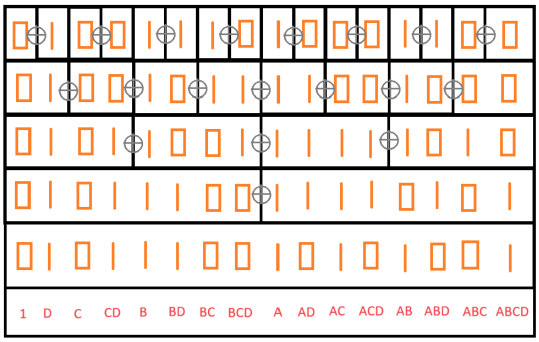
Karnaugh Map

f (A,B \ C,D)	00	01	11	10
00	0	1	0	0
01	1	1	0	1
11	1	1	0	0
10	1	0	0	0

Construct ANF (Zhegalkin polynomial) using Pascal Method:

Using the program from the repository, we get the values of column f from the truth table:

$$f = 0100'1110'1000'1100$$



$$f_{2\oplus}(A,B,C,D) = D \oplus CD \oplus B \oplus BD \oplus A \oplus AC \oplus AB \oplus ABCD$$

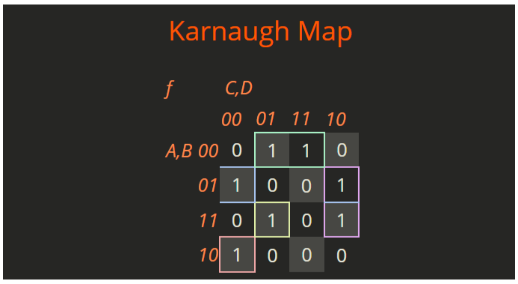
(c) $f_3 = f^{(4)}_{51011} \oplus f^{(4)}_{40389}$

Karnaugh Map (using a program from the repository):

```
The Karnaugh map
        00    01    11    10
   ----------------------
00 |    0     1     1     0
01 |    1     0     0     1
11 |    0     1     0     1
10 |    1     0     0     0
```

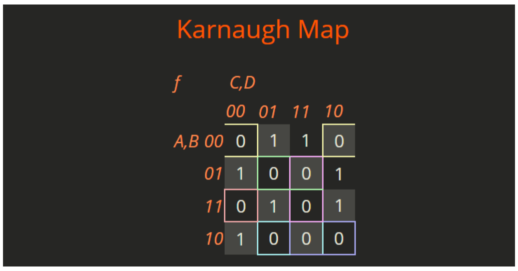
Minimal DNF:

$$f_3(A,B,C,D) = (\overline{A}\wedge\overline{B}\wedge D)\vee(B\wedge C\wedge\overline{D})\vee(\overline{A}\wedge B\wedge\overline{D})\vee(A\wedge B\wedge\overline{C}\wedge D)\vee(A\wedge\overline{B}\wedge\overline{C}\wedge\overline{D})$$



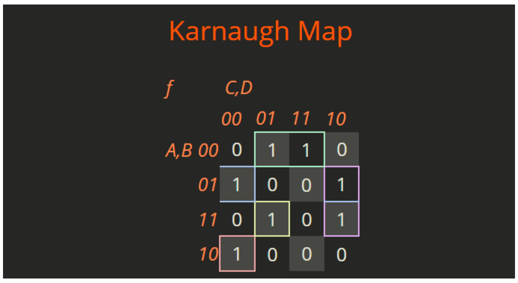
Minimal CNF:

$$f_3(A,B,C,D) = \overline{(\overline{A}\wedge\overline{B}\wedge\overline{D})\vee(\overline{A}\wedge B\wedge D)\vee(A\wedge B\wedge\overline{C}\wedge\overline{D})\vee(A\wedge\overline{B}\wedge C)\vee(A\wedge\overline{B}\wedge D)\vee(B\wedge C\wedge D)}$$
$$f_3(A,B,C,D) = (A\vee B\vee D)\wedge(A\vee\overline{B}\vee\overline{D})\wedge(\overline{A}\vee\overline{B}\vee C\vee D)\wedge(\overline{A}\vee B\vee\overline{C})\wedge(\overline{A}\vee B\vee\overline{D})\wedge(\overline{B}\vee\overline{C}\vee\overline{D})$$



BCF:

$$f_3(A,B,C,D) = (\overline{A}\wedge\overline{B}\wedge D)\vee(B\wedge C\wedge\overline{D})\vee(\overline{A}\wedge B\wedge\overline{D})\vee(A\wedge B\wedge\overline{C}\wedge D)\vee(A\wedge\overline{B}\wedge\overline{C}\wedge\overline{D})$$



Construct ANF (Zhegalkin polynomial) using the tabular ("triangle") method:

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
1	D	C	CD	B	BD	BC	BCD	A	AD	AC	ACD	AB	ABD	ABC	ABCD
0	1	0	0	1	0	0	0	1	0	1	1	0	0	0	1
1	1	0	1	1	0	0	1	1	1	0	1	0	0	1	
0	1	1	0	1	0	1	0	0	1	1	1	0	1		
1	0	1	1	1	1	1	0	1	0	0	1	1			
1	1	0	0	0	0	1	1	1	0	1	0				
0	1	0	0	0	1	0	0	1	1	1					
1	1	0	0	1	1	0	1	0	0						
0	1	0	1	0	1	1	1	0							
1	1	1	1	1	0	0	1								
0	0	0	0	1	0	1									
0	0	0	1	1	1										
0	0	1	0	0											
0	1	1	0												
1	0	1													
1	1														
0															

$f_{3\oplus}(A,B,C,D) = D \oplus B \oplus A \oplus AC \oplus ACD \oplus ABCD$

(d) $f_4 = A\overline{B}D + \overline{AC}D + \overline{B}C\overline{D} + A\overline{C}D$

The truth table of the function f_4:

A	B	C	D	f_4
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Karnaugh Map (using a program from the repository):

```
The Karnaugh map
        00    01    11    10
     -----------------------
00 |    0     1     0     1
01 |    0     1     0     0
11 |    0     1     0     0
10 |    0     1     1     1
```

Minimal DNF:

$f_4(A,B,C,D) = (\overline{C} \wedge D) \vee (A \wedge \overline{B} \wedge C) \vee (\overline{B} \wedge C \wedge \overline{D})$

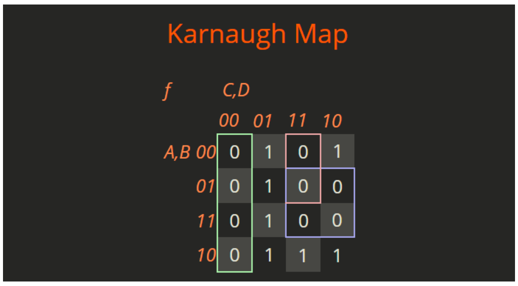
Karnaugh Map

f	C,D			
	00	01	11	10
A,B 00	0	1	0	1
01	0	1	0	0
11	0	1	0	0
10	0	1	1	1

Minimal CNF:

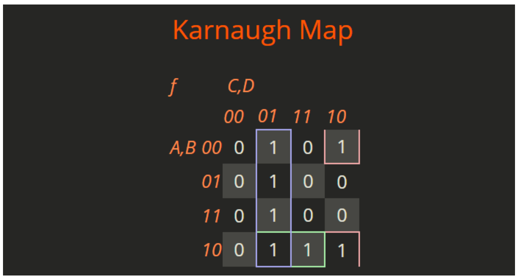
$f_4(A,B,C,D) = \overline{(\overline{C} \wedge \overline{D}) \vee (B \wedge C) \vee (\overline{A} \wedge C \wedge D)}$

$f_4(A,B,C,D) = (C \vee D) \wedge (\overline{B} \vee \overline{C}) \wedge (A \vee \overline{C} \vee \overline{D})$



BCF:

$f_4(A,B,C,D) = (\overline{C} \wedge D) \vee (A \wedge \overline{B} \wedge C) \vee (\overline{B} \wedge C \wedge \overline{D}) \vee (A \wedge \overline{B} \wedge D)$



Construct ANF (Zhegalkin polynomial) using Karnaugh map (using a program from the repository):

$f_{4\oplus}(A,B,C,D) = D \oplus C \oplus BC \oplus BCD \oplus ACD \oplus ABCD$

3. Convert the following formulae to CNF.

(a) $X \leftrightarrow (A \wedge B)$

$(X \to (A \wedge B)) \wedge ((A \wedge B) \to X)$

$(\overline{X} \vee (A \wedge B)) \wedge (\overline{(A \wedge B)} \vee X)$

$(\overline{X} \vee (A \wedge B)) \wedge (\overline{A} \vee \overline{B} \vee X)$

$(\overline{X} \vee A) \wedge (\overline{X} \vee B) \wedge (\overline{A} \vee \overline{B} \vee X)$

(b) $Z \leftrightarrow \vee_i C_i$

$(Z \to \vee_i C_i) \wedge (\vee_i C_i \to Z)$

$(\overline{Z} \vee \vee_i C_i) \wedge (\overline{\vee_i C_i} \vee Z)$

$(\overline{Z} \vee \vee_i C_i) \wedge (\wedge_i \overline{C_i} \vee Z)$

$(\overline{Z} \vee \vee_i C_i) \wedge (\wedge_i (\overline{C_i} \vee Z))$

$(\overline{Z} \vee \vee_i C_i) \wedge \wedge_i (\overline{C_i} \vee Z)$

(c) $D_1 \oplus \cdots \oplus D_n$

Use the XOR operator's definition: $A \oplus B$ = $\overline{(A \wedge B)} \wedge (A \vee B) = (\overline{A} \vee \overline{B}) \wedge (A \vee B)$

Apply the definition to the first two variables:
$D_1 \oplus D_2 = (D_1 \vee D_2) \wedge (\overline{D_1} \vee \overline{D_2})$

Apply the definition to the first three variables:

$$((D_1 \vee D_2) \wedge (\overline{D_1} \vee \overline{D_2})) \oplus D_3 = (((D_1 \vee D_2) \wedge (\overline{D_1} \vee \overline{D_2})) \vee D_3) \wedge (\overline{((D_1 \vee D_2) \wedge (\overline{D_1} \vee \overline{D_2}))} \vee \overline{D_3}) = (D_1 \vee D_2 \vee D_3) \wedge (\overline{D_1} \vee \overline{D_2} \vee D_3) \wedge (\overline{(D_1 \vee D_2)} \vee \overline{(\overline{D_1} \vee \overline{D_2})} \vee \overline{D_3}) = (D_1 \vee D_2 \vee D_3) \wedge (\overline{D_1} \vee \overline{D_2} \vee D_3) \wedge ((\overline{D_1} \wedge \overline{D_2}) \vee (D_1 \wedge D_2) \vee \overline{D_3}) = (D_1 \vee D_2 \vee D_3) \wedge (\overline{D_1} \vee \overline{D_2} \vee D_3) \wedge ((\overline{D_1} \wedge \overline{D_2}) \vee (D_1 \vee \overline{D_3}) \wedge (D_2 \vee \overline{D_3})) = (D_1 \vee D_2 \vee D_3) \wedge (\overline{D_1} \vee \overline{D_2} \vee D_3) \wedge (D_1 \vee \overline{D_3} \vee (\overline{D_1} \wedge \overline{D_2})) \wedge (D_2 \vee \overline{D_3} \vee (\overline{D_1} \wedge \overline{D_2})) = (D_1 \vee D_2 \vee D_3) \wedge (\overline{D_1} \vee \overline{D_2} \vee D_3) \wedge (\overline{D_1} \vee D_1 \vee \overline{D_3}) \wedge (\overline{D_2} \vee D_1 \vee \overline{D_3}) \wedge (\overline{D_1} \vee D_2 \vee \overline{D_3}) \wedge (\overline{D_2} \vee D_2 \vee \overline{D_3}) = (D_1 \vee D_2 \vee D_3) \wedge (\overline{D_1} \vee \overline{D_2} \vee D_3) \wedge (D_1 \vee \overline{D_2} \vee \overline{D_3}) \wedge (\overline{D_1} \vee D_2 \vee \overline{D_3})$$

Continuing to apply the XOR operation, you will notice that in each clause there will be an even number of negations of literals.

Let $D_i^0 = D_i$ and $D_i^1 = \overline{D_i}$

So we can write the original formula like this:

$\wedge_{[\oplus p_i = 0,\, i \in [1;n]]}(D_1^{p_1}, D_2^{p_2}, \dots D_n^{p_n})$

(d) $majority(X_1, X_2, X_3)$

$(X_1 \wedge X_2) \vee (X_2 \wedge X_3) \vee (X_1 \wedge X_3)$

$(X_1 \vee X_2) \wedge (X_2 \vee X_3) \wedge (X_1 \vee X_3)$

(e) $R \to (S \to (T \to \wedge_i F_i))$

$R \to (S \to (\overline{T} \vee \wedge_i F_i))$

$R \to (\overline{S} \vee (\overline{T} \vee \wedge_i F_i))$

$\overline{R} \vee (\overline{S} \vee (\overline{T} \vee \wedge_i F_i))$

$(\overline{R} \vee \overline{S} \vee \overline{T}) \vee \wedge_i F_i$

$\wedge_i(\overline{R} \vee \overline{S} \vee \overline{T} \vee F_i)$

(f) $M \to (H \leftrightarrow \vee_i D_i)$

$M \to ((H \to \vee_i D_i) \wedge (\vee_i D_i \to H))$

$M \to ((\overline{H} \vee \vee_i D_i) \wedge (\overline{\vee_i D_i} \vee H))$

$\overline{M} \vee ((\overline{H} \vee \vee_i D_i) \wedge (\overline{\vee_i D_i} \vee H))$

$\overline{M} \vee ((\overline{H} \vee \vee_i D_i) \wedge (\wedge_i \overline{D_i} \vee H))$

$\overline{M} \vee ((\overline{H} \vee \vee_i D_i) \wedge (\wedge_i (\overline{D_i} \vee H)))$

$(\overline{H} \vee \vee_i D_i \vee \overline{M}) \wedge \wedge_i (\overline{D_i} \vee H \vee \overline{M})$

4. For each given system of functions F_i, determine whether it is functionally complete using Post's criterion. For each basis F_i, use it to represent the majority(A, B, C) function. Draw a combinational Boolean circuit for each resulting formula.

The function majority(A, B, C) takes the value 1 for the following sets of values: (A, B, C): (0, 1, 1), (1, 0, 1), (1, 1, 0), (1, 1, 1)

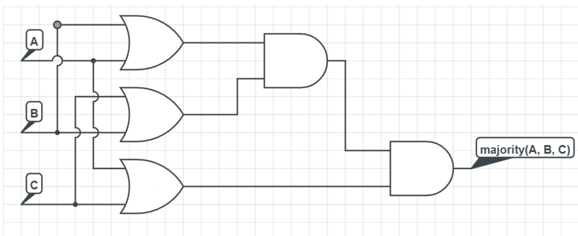
(a) $F_1 = \{\wedge, \vee, \neg\}$

Let's check whether the function is functionally complete using the Post criterion:

- $f_\neg(0) = 1 \Rightarrow f_\neg \notin T_0$
- $f_\neg(1) = 0 \Rightarrow f_\neg \notin T_1$
- $\begin{cases} f_\wedge(x, y) = 0001 \\ \overline{f_\wedge}(\overline{x}, \overline{y}) = 0111 \end{cases} \Rightarrow f_\wedge \notin S$
- $f_\neg = 10 \Rightarrow f_\neg \notin M$
- $f_{\oplus\wedge}(x, y) = xy \Rightarrow f_\wedge \notin L$

$\Rightarrow F_1 = \{\wedge, \vee, \neg\}$ is functionally complete

$majority(A, B, C) = (A \wedge B) \vee (A \wedge C) \vee (B \wedge C)$



(b) $F_2 = \{f_{14}^{(2)}\}$

Let's check whether the function is functionally complete using the Post criterion:

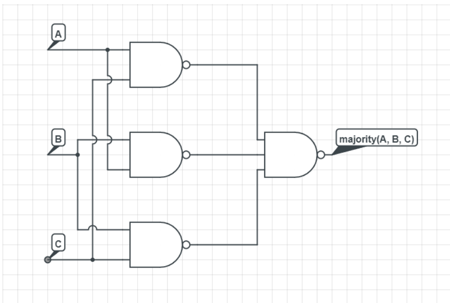
$f_{14}^{(2)} = 1110 = \overline{(x \wedge y)} = x\ NAND\ y = NAND(x, y)$

$F_2 = \{NAND\}$

- $f_{NAND}(0,0) = 1 \Rightarrow f_{NAND} \notin T_0$
- $f_{NAND}(1,1) = 0 \Rightarrow f_{NAND} \notin T_1$
- $\begin{cases} f_{NAND}(x,y) = 1110 \\ \overline{f_{NAND}}(\overline{x},\overline{y}) = 1001 \end{cases} \Rightarrow f_{NAND} \notin S$
- $f_{NAND}(x,y) = 1110 \Rightarrow f_{NAND} \notin M$
- $f_{\oplus NAND}(x,y) = 1 \oplus xy \Rightarrow f_{NAND} \notin L$

$\Rightarrow F_2 = \{NAND\} = \{f_{14}^{(2)}\}$ is functionally complete

$majority(A,B,C) = NAND(NAND(A,B), NAND(A,C), NAND(B,C))$



(c) $F_3 = \{\to, \not\to\}$

$f_{\to}(x,y) = 1101$

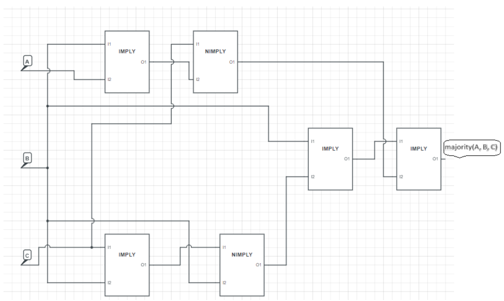
$f_{\not\to}(x,y) = 0010$

Let's check whether the function is functionally complete using the Post criterion:

- $f_{\to}(0,0) = 1 \Rightarrow f_{\to} \notin T_0$
- $f_{\not\to}(1,1) = 0 \Rightarrow f_{\not\to} \notin T_1$
- $\begin{cases} f_{\to}(x,y) = 1101 \\ \overline{f_{\to}}(\overline{x},\overline{y}) = 0100 \end{cases} \Rightarrow f_{\to} \notin S$
- $f_{\to}(x,y) = 1101 \Rightarrow f_{\to} \notin M$
- $f_{\oplus\to}(x,y) = 1 \oplus x \oplus xy \Rightarrow f_{\to} \notin L$

$\Rightarrow F_3 = \{\to, \not\to\}$ is functionally complete

$majority(A,B,C) = (A \to ((C \to B) \not\to B)) \to (C \not\to (B \to A))$



*I1 - input#1, I2 - input#2, O1 - output#1

(d) $F_4 = \{1, \leftrightarrow, \wedge\}$

$f_{\leftrightarrow}(x,y) = 1001$

$f_{\wedge}(x,y) = 0001$

Let's check whether the function is functionally complete using the Post criterion:

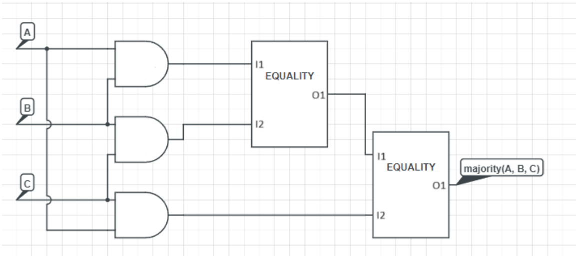
- $f_{\leftrightarrow}(0,0) = 1 \Rightarrow f_{\leftrightarrow} \notin T_0$
- $-f_{\leftrightarrow}(1,1) = 1 \Rightarrow f_{\leftrightarrow} \in T_1$

  $-f_{\land}(1,1) = 1 \Rightarrow f_{\land} \in T_1$

  $-1 \in T_1$

$\Rightarrow F_4 = \{1, \leftrightarrow, \land\}$ is not functionally complete

$majority(A,B,C) = (A \land B) \leftrightarrow (A \land C) \leftrightarrow (B \land C)$



*I1 - input#1, I2 - input#2, O1 - output#1

5. Show—without using Post's criterion—that the Zhegalkin basis $\{\oplus, \land, 1\}$ is functionally complete.

To show that the Zhegalkin basis $\{\oplus, \land, 1\}$ is functionally complete, we need to demonstrate that we can express any Boolean function using only these operations.

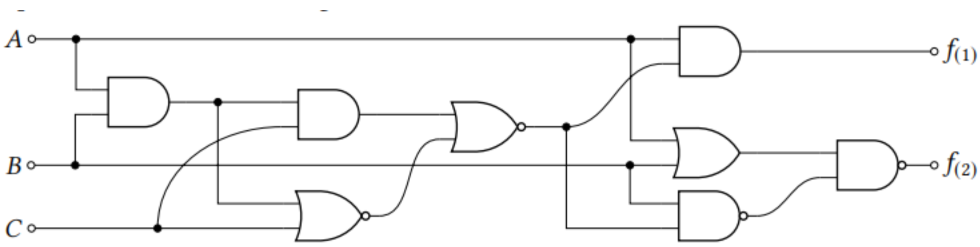
1) The operation of negation $\neg$ can be expressed using the Zhegalkin basis as follows:
$\neg X = X \oplus 1$

2) The AND operation $\land$ is already included in the basis.

3) The OR operation $\lor$ using the Zhegalkin basis as follows:
$A \lor B = (A \oplus B) \oplus (A \land B)$

By demonstrating that we can express negation, OR, and AND operations using only the operations in the Zhegalkin basis $\Rightarrow$ we have shown that the basis is functionally complete.

6. Compute the truth table for the function $f : B^3 \to B^2$ (with the semantics $\langle A, B, C\rangle \mapsto \langle f_{(1)}, f_{(2)}\rangle$) represented with the following circuit



$f_{(1)} = A \land \overline{(((A \land B) \land C) \lor \overline{\overline{((A \land B) \lor C)}})} = A \land \overline{(((A \land B) \land C) \lor \overline{(\overline{(A \land B)} \land \overline{C})})} = A \land \overline{(((A \land B) \land C) \lor ((\overline{A} \lor \overline{B}) \land \overline{C}))} = A \land (\overline{((A \land B) \land C)} \land \overline{((\overline{A} \lor \overline{B}) \land \overline{C})}) = A \land ((\overline{(A \land B)} \lor \overline{C}) \land (\overline{(\overline{A} \lor \overline{B})} \lor \overline{\overline{C}})) = A \land (\overline{A} \lor \overline{B} \lor \overline{C}) \land ((A \land B) \lor C)$

$f_{(2)} = \overline{(A \lor B) \land \overline{(B \land \overline{(((A \land B) \land C) \lor \overline{\overline{((A \land B) \lor C)}})})}} = \overline{(A \lor B) \land \overline{(B \land (\overline{A} \lor \overline{B} \lor \overline{C}) \land ((A \land B) \lor C))}} = (\overline{A} \land \overline{B}) \lor (B \land (\overline{A} \lor \overline{B} \lor \overline{C}) \land ((A \land B) \lor C)))$

A	B	C	$f_{(1)}$	$f_{(2)}$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	0

0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

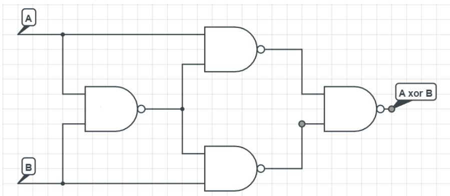
7. Construct a minimal Boolean circuit that implements the conversion of 4-bit binary numbers to Gray code, i.e. the function $f : B^4 \to B^4$ with the semantics $(b_3, b_2, b_1, b_0) \to (g_3, g_2, g_1, g_0)$, e.g. $0000_2 \to 0000_{Gray}$, and $1001_2 \to 1101_{Gray}$. Use only NAND and NOR logic gates.

To convert an unsigned binary number to reflected binary Gray code, we can follow the algorithm:

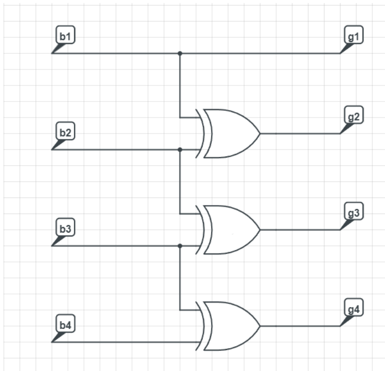
1) Start with the given unsigned binary number.
2) Identify the most significant bit and assign it as the corresponding bit of the Gray code.
3) For each subsequent bit, starting from the MSB: XOR the current bit with the previous bit of the unsigned binary number. Assign the result to the corresponding bit of the Gray code.
4) Repeat step 3 for all the bits, moving from left to right.

In other words, we can find the Gray code with a given number by applying XOR to num and (num >> 1), where (num >> 1) is the bit shift of num by 1 to the right.

XOR gate we can express using NAND gate as follows:



The logic circuit itself will look like this:



8. A half subtractor is a circuit that has two bits as input and produces as output a difference bit and a borrow. A full subtractor is a circuit that has two bits and a borrow as input, and produces as output a difference bit and a borrow.

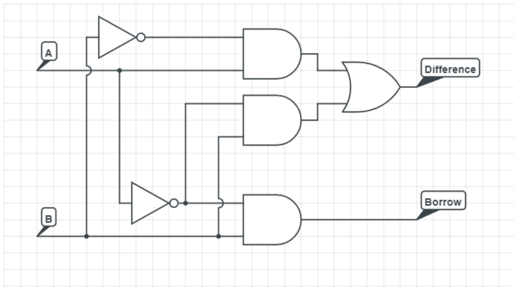
(a) Construct a circuit for a half subtractor using AND gates, OR gates, and inverters.

Truth table for a half substractor:

A	B	Difference bit	Borrow bit
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$\Rightarrow Difference = A \oplus B,\ \ borrow = \overline{A} \wedge B$

Circuit for a half substructor:



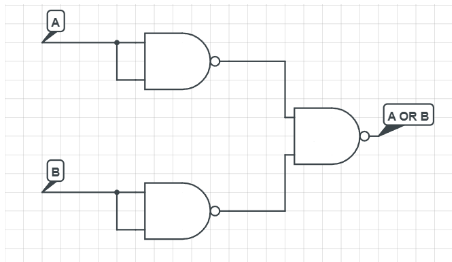
(b) Construct a circuit for a full subtractor using half subtractors and NAND gates.

Truth table for a full substructor:

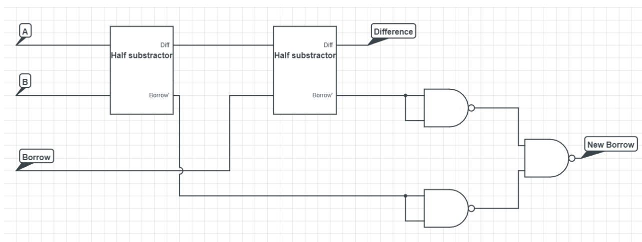
A	B	Borrow	Difference	New Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$\Rightarrow Difference = A \oplus B \oplus borrow,\ \ new\ borrow = (\overline{A} \wedge B) \vee (\overline{(A \oplus B)} \wedge borrow)$

We can replace OR gate with the following equivalent circuit using only NAND gates:



Circuit for a full substructor:

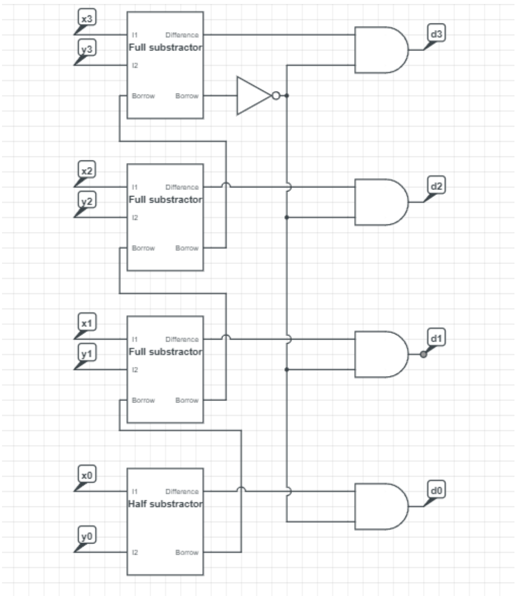


(c) Construct a circuit that computes the saturating difference of two four-bit integers $(x_3, x_2, x_1, x_0)_2$ and $(y_3, y_2, y_1, y_0)_2$ using half/full subtractors, AND gates, OR gates, and inverters. When $x \geq y$, the output bits $d_3, \ldots, d_0$ should represent $d = x - y$, and when $x < y$, the output must be zero.

Let's take half subtractors to x_0 and y_0.

For the next three pairs x_i and y_i, we apply the full subtractor using borrow from the previous operation.

If the borrow obtained at the last stage is equal to one, $x < y$, output zero. Otherwise, we return the difference from the subtractors.



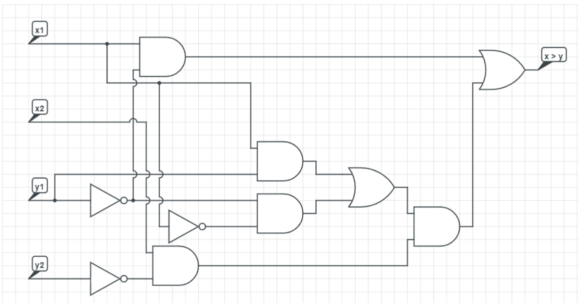
9. Construct a circuit that compares the two-bit integers $(x_1, x_0)_2$ and $(y_1, y_0)_2$, and outputs 1 when $x > y$ and 0 otherwise.

$(x_1, x_2)_2 > (y_1, y_2)_2$ iff:

- $x_1 > y_1 \Rightarrow x_1 \nrightarrow y_1 \Rightarrow \overline{\overline{x_1} \vee y_1} \Rightarrow x_1 \wedge \overline{y_1}$
- $x_1 = y_1$ and $x_2 > y_2 \Rightarrow ((x_1 \wedge y_1) \vee (\overline{x_1} \wedge \overline{y_1})) \wedge (x_2 \wedge \overline{y_2})$

$(x_1, x_2)_2 > (y_1, y_2)_2 \Leftrightarrow (x_1 \wedge \overline{y_1}) \vee ((x_1 \wedge y_1) \vee (\overline{x_1} \wedge \overline{y_1})) \wedge (x_2 \wedge \overline{y_2})$

The logic circuit itself will look like this:



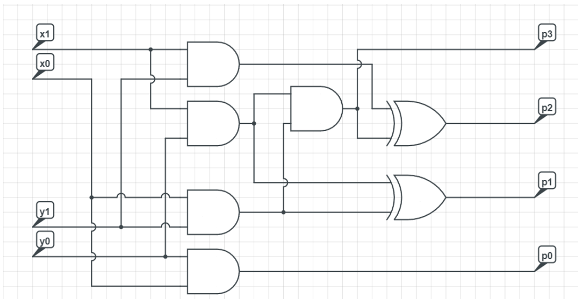
10. Construct a circuit that computes the product of the two-bit integers $(x_1, x_0)_2$ and $(y_1, y_0)_2$. The circuit should have four-bit output $(p_3, p_2, p_1, p_0)_2$ representing the product $p = xy$.

x_1	x_0	y_1	y_0	p_3	p_2	p_1	p_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1

1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

- $p_0 = 1 \Leftrightarrow (x_0 \wedge y_0)$
- $p_1 = 1 \Leftrightarrow (x_0 \wedge y_1) \vee (x_1 \wedge y_0) \wedge \overline{(x_0 \wedge y_1) \wedge (x_1 \wedge y_0)} \Leftrightarrow (x_0 \wedge y_1) \oplus (x_1 \wedge y_0)$
- $p_2 = 1 \Leftrightarrow ((x_0 \wedge y_1) \wedge (x_1 \wedge y_0)) \oplus (x_1 \wedge y_1)$
- $p_3 = 1 \Leftrightarrow ((x_0 \wedge y_1) \wedge (x_1 \wedge y_0)) \wedge (x_1 \wedge y_1) \Leftrightarrow x_0 \wedge x_1 \wedge y_0 \wedge y_1$

The logic circuit itself will look like this:



11. Consider a Boolean function $ITE : B^3 \to B$ defined as follows: $ITE(c, x, y)$ = $\begin{cases} x \ \ if \ \ c = 0 \\ y \ \ if \ \ c = 1 \end{cases}$

    Construct a formula for it using the standard Boolean basis $\{\wedge, \vee, \neg\}$. Determine whether the set ITE is functionally complete.

c	x	y	$ITE(c,x,y)$
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

    $\Rightarrow ITE(c, x, y) = (\bar{c} \wedge x) \vee (c \wedge y)$

    Determine whether the set ITE is functionally complete using Post's criterian:

    - $ITE(0, 0, 0) = 0 \Rightarrow ITE(c, x, y) \in T_0$

    $\Rightarrow ITE(c, x, y)$ is'nt functionally complete

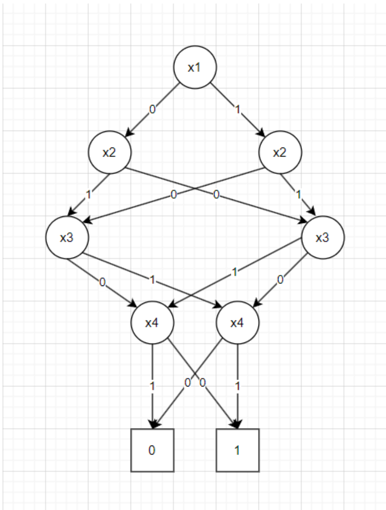
12. For each given function f_i, construct a Reduced Ordered Binary Decision Diagram (ROBDD) using the natural variable order $x_1 \prec x_2 \prec \cdots \prec x_n$. Determine whether the ROBDD can be reduced even further by using a different variable order — if so, show it.

(a) $f_1(x_1, ..., x_4) = x_1 \oplus x_2 \oplus x_3 \oplus x_4 = 0110'1001'1001'0110 = (\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4) \vee (\overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4}) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4}) \vee (\overline{x_1} \wedge x_2 \wedge x_3 \wedge x_4) \vee (x_1 \wedge \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4}) \vee (x_1 \wedge \overline{x_2} \wedge x_3 \wedge x_4) \vee (x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4) \vee (x_1 \wedge x_2 \wedge x_3 \wedge \overline{x_4})$

x_1	x_2	x_3	x_4	f_1
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

- $f_{1_0} = (\overline{x_2} \wedge \overline{x_3} \wedge x_4) \vee (\overline{x_2} \wedge x_3 \wedge \overline{x_4}) \vee (x_2 \wedge \overline{x_3} \wedge \overline{x_4}) \vee (x_2 \wedge x_3 \wedge x_4)$
  $f_{1_1} = (\overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4}) \vee (\overline{x_2} \wedge x_3 \wedge x_4) \vee (x_2 \wedge \overline{x_3} \wedge x_4) \vee (x_2 \wedge x_3 \wedge \overline{x_4})$
- $f_{1_{00}} = (\overline{x_3} \wedge x_4) \vee (x_3 \wedge \overline{x_4})$
  $f_{1_{01}} = (\overline{x_3} \wedge \overline{x_4}) \vee (x_3 \wedge x_4)$
  $f_{1_{10}} = (\overline{x_3} \wedge \overline{x_4}) \vee (x_3 \wedge x_4)$
  $f_{1_{11}} = (\overline{x_3} \wedge x_4) \vee (x_3 \wedge \overline{x_4})$
- $f_{1_{000}} = f_{1_{110}} = x_4$
  $f_{1_{001}} = f_{1_{111}} = \overline{x_4}$
  $f_{1_{010}} = f_{1_{100}} = \overline{x_4}$
  $f_{1_{011}} = f_{1_{101}} = x_4$

$\Rightarrow ROBBD$ using natural variables order for f_1:



$ROBDD$ can't be reduced even further by using a different variable order

(b) $f_2(x_1, ..., x_5) = majority(x_1, ..., x_5) = (\overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge x_4 \wedge x_5) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge x_4 \wedge x_5) \vee (\overline{x_1} \wedge x_2 \wedge x_3 \wedge \overline{x_4} \wedge x_5) \vee (\overline{x_1} \wedge x_2 \wedge x_3 \wedge x_4 \wedge \overline{x_5}) \vee (\overline{x_1} \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5) \vee (x_1 \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \wedge x_5) \vee (x_1 \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4} \wedge x_5) \vee (x_1 \wedge \overline{x_2} \wedge x_3 \wedge x_4 \wedge \overline{x_5}) \vee (x_1 \wedge \overline{x_2} \wedge x_3 \wedge x_4 \wedge x_5) \vee (x_1 \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \wedge x_5) \vee (x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4 \wedge \overline{x_5}) \vee (x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4 \wedge x_5) \vee (x_1 \wedge x_2 \wedge x_3 \wedge \overline{x_4} \wedge \overline{x_5}) \vee (x_1 \wedge x_2 \wedge x_3 \wedge \overline{x_4} \wedge x_5) \vee (x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge \overline{x_5}) \vee (x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5)$

x_1	x_2	x_3	x_4	x_5	f_2
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

- $f_{2_0} = (\overline{x_2} \wedge x_3 \wedge x_4 \wedge x_5) \vee (x_2 \wedge \overline{x_3} \wedge x_4 \wedge x_5) \vee (x_2 \wedge x_3 \wedge \overline{x_4} \wedge x_5) \vee (x_2 \wedge x_3 \wedge x_4 \wedge \overline{x_5}) \vee (x_2 \wedge x_3 \wedge x_4 \wedge x_5)$

  $f_{2_1} = (\overline{x_2} \wedge \overline{x_3} \wedge x_4 \wedge x_5) \vee (\overline{x_2} \wedge x_3 \wedge \overline{x_4} \wedge x_5) \vee (\overline{x_2} \wedge x_3 \wedge x_4 \wedge \overline{x_5}) \vee (\overline{x_2} \wedge x_3 \wedge x_4 \wedge x_5) \vee (x_2 \wedge \overline{x_3} \wedge \overline{x_4} \wedge x_5) \vee (x_2 \wedge \overline{x_3} \wedge x_4 \wedge \overline{x_5}) \vee (x_2 \wedge \overline{x_3} \wedge x_4 \wedge x_5) \vee (x_2 \wedge x_3 \wedge \overline{x_4} \wedge \overline{x_5}) \vee (x_2 \wedge x_3 \wedge \overline{x_4} \wedge x_5) \vee (x_2 \wedge x_3 \wedge x_4 \wedge \overline{x_5}) \vee (x_2 \wedge x_3 \wedge x_4 \wedge x_5)$

- $f_{2_{00}} = (x_3 \wedge x_4 \wedge x_5)$

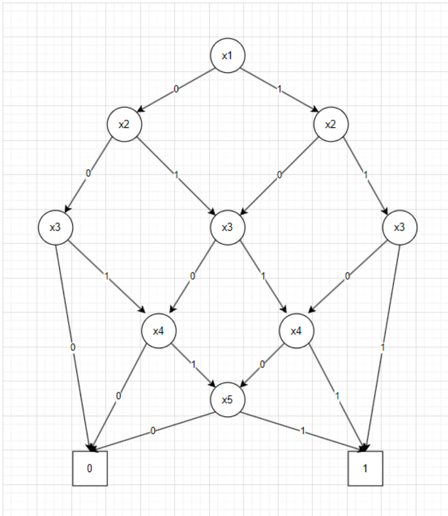
  $f_{2_{01}} = (\overline{x_3} \wedge x_4 \wedge x_5) \vee (x_3 \wedge \overline{x_4} \wedge x_5) \vee (x_3 \wedge x_4 \wedge \overline{x_5}) \vee (x_3 \wedge x_4 \wedge x_5)$

  $f_{2_{10}} = (\overline{x_3} \wedge x_4 \wedge x_5) \vee (x_3 \wedge \overline{x_4} \wedge x_5) \vee (x_3 \wedge x_4 \wedge \overline{x_5}) \vee (x_3 \wedge x_4 \wedge x_5)$

  $f_{2_{11}} = (\overline{x_3} \wedge \overline{x_4} \wedge x_5) \vee (\overline{x_3} \wedge x_4 \wedge \overline{x_5}) \vee (\overline{x_3} \wedge x_4 \wedge x_5) \vee (x_3 \wedge \overline{x_4} \wedge \overline{x_5}) \vee (x_3 \wedge \overline{x_4} \wedge x_5) \vee (x_3 \wedge x_4 \wedge \overline{x_5}) \vee (x_3 \wedge x_4 \wedge x_5)$

- $f_{2_{000}} = 0$
  $f_{2_{001}} = (x_4 \wedge x_5)$
  $f_{2_{010}} = f_{2_{100}} = (x_4 \wedge x_5)$
  $f_{2_{011}} = f_{2_{101}} = (\overline{x_4} \wedge x_5) \vee (x_4 \wedge \overline{x_5}) \vee (x_4 \wedge x_5)$
  $f_{2_{110}} = (\overline{x_4} \wedge x_5) \vee (x_4 \wedge \overline{x_5}) \vee (x_4 \wedge x_5)$
  $f_{2_{111}} = (\overline{x_4} \wedge \overline{x_5}) \vee (\overline{x_4} \wedge x_5) \vee (x_4 \wedge \overline{x_5}) \vee (x_4 \wedge x_5) = 1$
- $f_{2_{0010}} = f_{2_{0100}} = f_{2_{1000}} = 0$
  $f_{2_{0011}} = f_{2_{0101}} = f_{2_{1001}} = x_5$
  $f_{2_{0110}} = f_{2_{1010}} = f_{2_{1100}} = x_5$
  $f_{2_{0111}} = f_{2_{1011}} = f_{2_{1101}} = \overline{x_5} \vee x_5 = 1$

$\Rightarrow ROBBD$ using natural variables order for f_2:



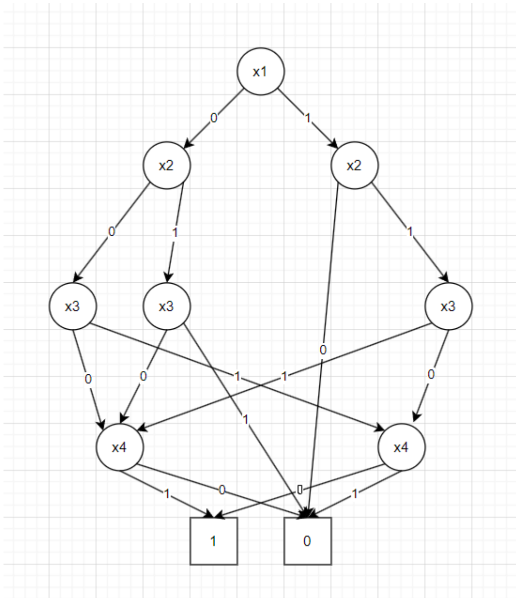
$ROBDD$ can't be reduced even further by using a different variable order

(c) $f_3(x_1, ..., x_4) = \sum m(1, 2, 5, 12, 15) = 0110'0100'0000'1001 = (\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4) \vee (\overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge \overline{x_4}) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge x_4) \vee (x_1 \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4}) \vee (x_1 \wedge x_2 \wedge x_3 \wedge x_4)$

x_1	x_2	x_3	x_4	f_3
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

- $f_{3_0} = (\overline{x_2} \wedge x_3 \wedge \overline{x_4}) \vee (\overline{x_3} \wedge x_4)$
  $f_{3_1} = (x_2 \wedge \overline{x_3} \wedge \overline{x_4}) \vee (x_2 \wedge x_3 \wedge x_4)$
- $f_{3_{00}} = (x_3 \wedge \overline{x_4}) \vee (\overline{x_3} \wedge x_4)$
  $f_{3_{01}} = (\overline{x_3} \wedge x_4)$
  $f_{3_{10}} = 0$
  $f_{3_{11}} = (\overline{x_3} \wedge \overline{x_4}) \vee (x_3 \wedge x_4)$
- $f_{3_{000}} = f_{3_{010}} = f_{3_{111}} = x_4$
  $f_{3_{001}} = f_{3_{110}} = \overline{x_4}$
  $f_{3_{011}} = 0$

$\Rightarrow ROBBD$ using natural variables order for f_3:



$ROBDD$ can't be reduced even further by using a different variable order

(d) $f_4(x_1, ..., x_6) = (x_1 \wedge x_4) \vee (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$

- $f_{4_0} = (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$
  $f_{4_1} = x_4 \vee (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$
- $f_{4_{00}} = x_3 \wedge x_6$
  $f_{4_{01}} = x_5 \vee (x_3 \wedge x_6)$
  $f_{4_{10}} = x_4 \vee (x_3 \wedge x_6)$
  $f_{4_{11}} = x_4 \vee x_5 \vee (x_3 \wedge x_6)$
- $f_{4_{000}} = 0$
  $f_{4_{001}} = x_6$
  $f_{4_{010}} = x_5$
  $f_{4_{011}} = x_5 \vee x_6$
  $f_{4_{100}} = x_4$
  $f_{4_{101}} = x_4 \vee x_6$
  $f_{4_{110}} = x_4 \vee x_5$
  $f_{4_{111}} = x_4 \vee x_5 \vee x_6$
- $f_{4_{1010}} = x_6$
  $f_{4_{1011}} = f_{4_{1101}} = f_{4_{1111}} = 1$

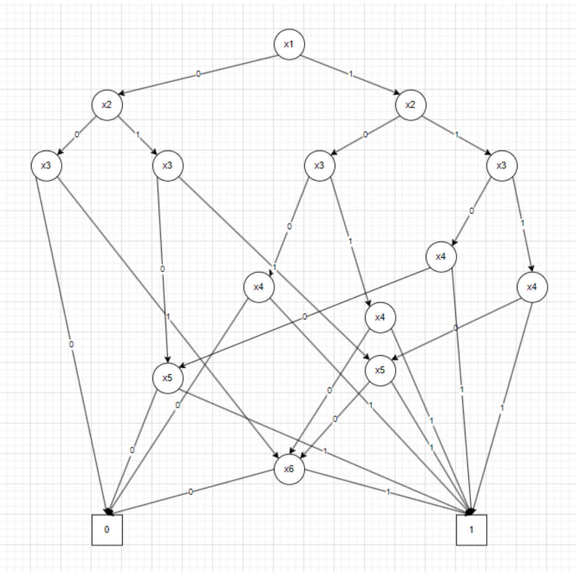
$f_{4_{1100}} = x_5$

$f_{4_{1110}} = x_5 \vee x_6$

- $f_{4_{11100}} = x_6$

  $f_{4_{11101}} = 1$

$\Rightarrow ROBBD$ using natural variables order for f_4:



ROBDD can be reduced even further by using a different variable order:

$x_1 \prec x_4 \prec x_2 \prec x_5 \prec x_6$

$f_4(x_1, ..., x_6) = (x_1 \wedge x_4) \vee (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$

- $f_{4_0} = (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$

  $f_{4_1} = x_4 \vee (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$

- $f_{4_{10}} = (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$

  $f_{4_{11}} = 1$

- $f_{4_{100}} = f_{4_{00}} = (x_3 \wedge x_6)$

  $f_{4_{101}} = f_{4_{01}} = x_5 \vee (x_3 \wedge x_6)$

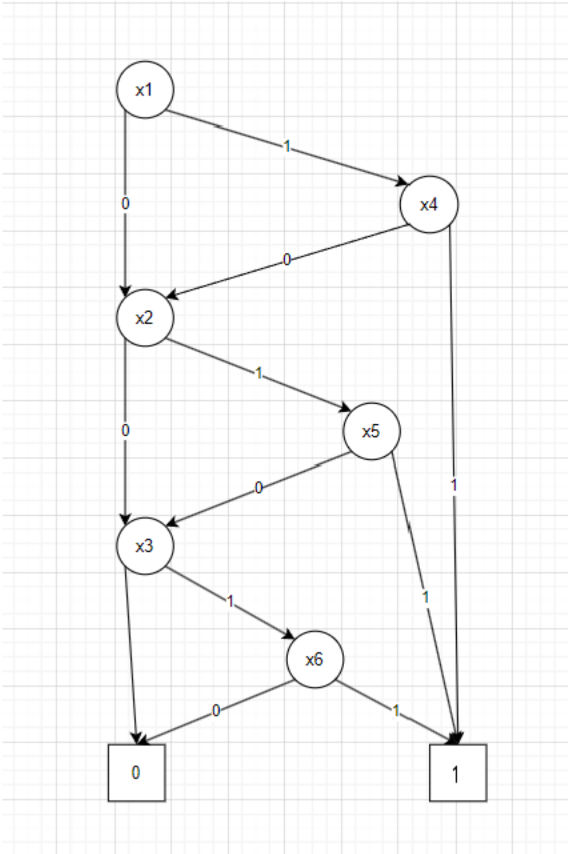
- $f_{4_{1010}} = f_{4_{010}} = (x_3 \wedge x_6)$

  $f_{4_{1011}} = f_{4_{011}} = 1$

- $f_{4_{10100}} = f_{4_{0100}} = f_{4_{1000}} = f_{4_{000}} = 0$

  $f_{4_{10100}} = f_{4_{0100}} = f_{4_{1000}} = f_{4_{000}} = x_6$

$\Rightarrow ROBBD$ using current variables order for f_4:


x1
x4
x2
x5
x3
x6
0
1
1
0
0
1
0
0
1
1
1
1
0
1