Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

# ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №2

# МІНІМІЗАЦІЯ ПЕРЕМИКАЛЬНИХ ФУНКЦІЙ

Виконав:

студент групи ІО-64

Андрійчук Д. А.

Залікова книжка № IO-6401

Перевірив:

[Верба О. А.](http://rozklad.kpi.ua/Schedules/ViewSchedule.aspx?v=3616fe25-c15f-4d3e-986b-deb3928e21b8)

Київ 2016

Хід роботи

Метод Квайна (*f*1)

Випишемо всі конституенти одиниці.

Виконаємо склеювання та поглинання

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Табл. 2.8*  *Таблиця істинності* | | | | | |
| *x*4 | *x*3 | *x*2 | *x*1 | *f*1 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | *1* |
| 0 | 0 | 1 | 0 | *0* |
| 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 | 1 |
| 1 | 0 | 1 | 1 | *0* |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | *1* |
| 1 | 1 | 1 | 1 | 0 |

СДНФ=

Створимо таблицю покриття. Знайдемо ядро функції та з’ясуємо чи немає лишніх імплікант.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Як ми бачимо без імпліканти можна обійтись оскільки її конституенти покриваються ядром функції.

МДНФ=

Метод Квайна-Мак-Класки (*f*2)

Випишемо всі конституенти одиниці.

Виконаємо склеювання та поглинання

|  |  |  |  |
| --- | --- | --- | --- |
| к-ть одиниць |  | позиція х |  |
| 0 | ~~0000~~ | x\_\_\_ | **x110** |
| 1 | ~~0010~~ | \_x\_\_ | **0x10**  **1x11** |
| 2 | ~~0110~~ | \_\_x\_ | **00x0** |
| 3 | ~~1011~~  ~~1110~~ |  |
| \_\_\_x | **111x** |
| 4 | ~~1111~~ |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Табл. 2.8*  *Таблиця істинності* | | | | | |
| *x*4 | *x*3 | *x*2 | *x*1 | *f*2 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | *1* |
| 0 | 0 | 1 | 1 | *0* |
| 0 | 1 | 0 | 0 | *0* |
| 0 | 1 | 0 | 1 | *0* |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | *0* |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | *0* |
| 1 | 0 | 1 | 0 | *0* |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | *0* |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | *1* |

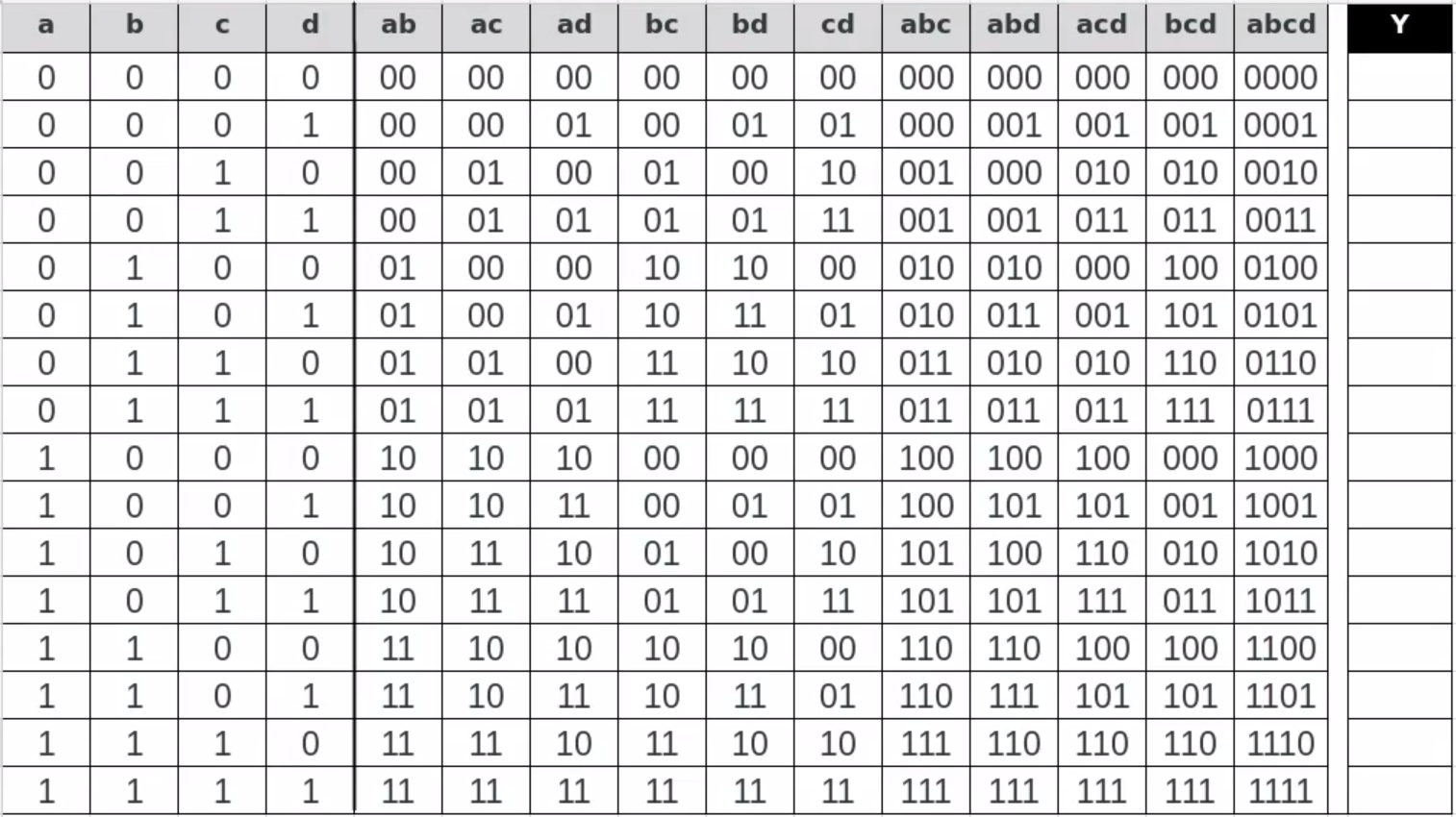
СДНФ=

Створимо таблицю покриття. Знайдемо ядро функції та з’ясуємо чи немає лишніх імплікант.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  | v |  | v |  |
|  |  | v | v |  |  |  |
|  |  |  |  | v |  | v |
|  | v | v |  |  |  |  |
| 111x |  |  |  |  | v | v |

ТДНФ=

МДНФ=



Метод невизначених коефіцієнтів (*f3*)

Заповнимо таблицю комбінаційних наборів. Викрислемо всі комбінації конституент одиниці. Виконаємо поглинання залишившихся конституент.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Таблиця істинності* | | | | | |
| *a* | *b* | *c* | *d* |  |
| *x*4 | *x*3 | *x*2 | *x*1 | *f*3 |
| 0 | 0 | 0 | 0 | *1* |
| 0 | 0 | 0 | 1 | *0* |
| 0 | 0 | 1 | 0 | *0* |
| 0 | 0 | 1 | 1 | *0* |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | *0* |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | *0* |
| 1 | 0 | 0 | 0 | *0* |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | *0* |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | *1* |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Ядро -

Не покривається ядром лише одна функція.

Однаково коротко можна їх покрити функціями і

ТКНФ=

МКНФ=

|  |
| --- |
|  |
| 1 |
| 0 |
| 0 |
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 1 |
| 0 |
| 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |

x3

x2

x4

Метод діаграм Вейча (*f4*)

Намалювмти діаграму, знайдемо найбільші прямокутики, які покривають всі функції 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Табл. 2.8*  *Таблиця істинності* | | | | | |
| *x*4 | *x*3 | *x*2 | *x*1 | *f*4 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | *0* |
| 0 | 0 | 1 | 1 | *0* |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | *0* |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | *0* |
| 1 | 0 | 0 | 0 | *0* |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | *0* |
| 1 | 0 | 1 | 1 | *0* |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | *1* |
| 1 | 1 | 1 | 0 | *1* |
| 1 | 1 | 1 | 1 | 1 |

x1

МКНФ=

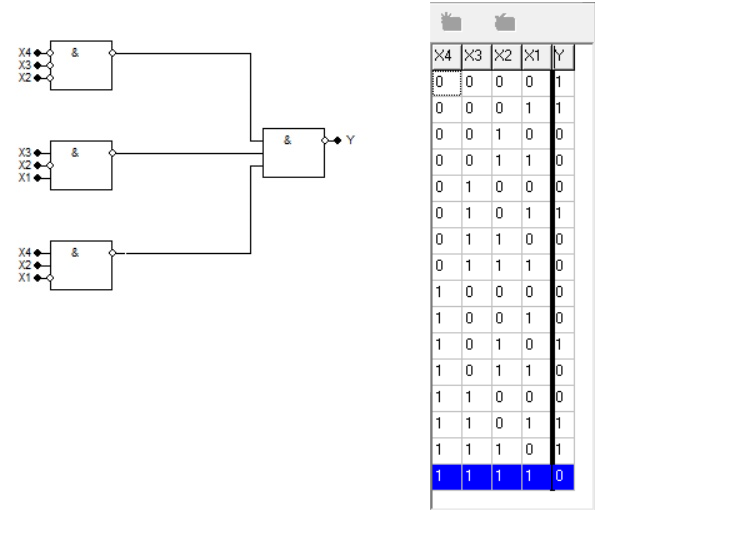
Спробуємо представити *f1* в операторній формі, яку можна представити елементами 3І-НЕ

– І\АБО

=

– І-НЕ\І-НЕ

Створимо задану комбінаційну схему.



Складність за Квайном - 12

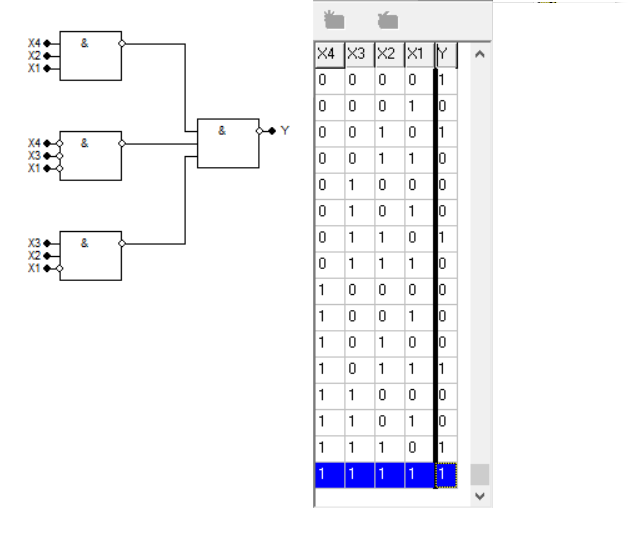
Спробуємо представити *f2* в операторній формі, яку можна представити елементами 3І-НЕ

І\АБО

=

І-НЕ\І-НЕ

Створимо задану комбінаційну схему.



Складність за Квайном - 12

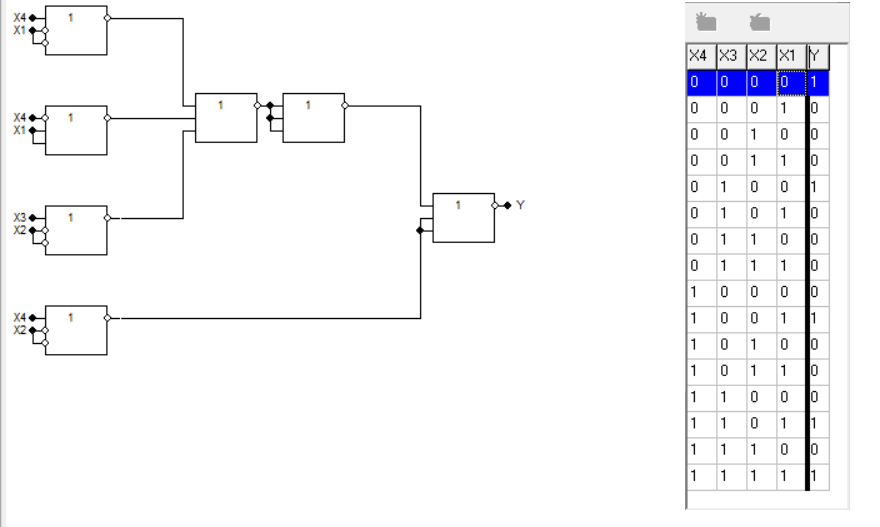
Спробуємо представити *f3* в операторній формі, яку можна представити елементами 3АБО-НЕ.

– АБО\І

=

– АБО-НЕ\АБО-НЕ

Створимо задану комбінаційну схему



Складність за Квайном - 21

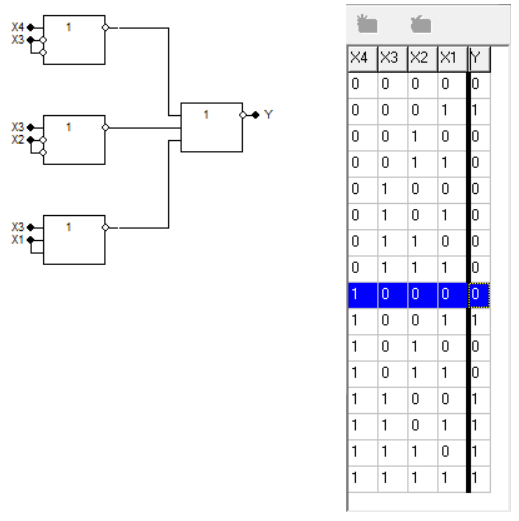
Спробуємо представити *f4* в операторній формі, яку можна представити елементами 3АБО-НЕ.

– АБО\І

=

– АБО-НЕ\АБО-НЕ

Створимо задану комбінаційну схему



Складність за Квайном - 12

Висновок: завдяки мінімізації перемикальних функцій ми можемо не лише значно спростити схему, але й зменшити витрати коштів, збільшити швидкодію схеми.