Практическая работа №6

**Получение и минимизация Булевых функций (БФ) из карт Карно (КК).**

***1.*** *Цель работы.*

1. Составление полной ТИ ее сокращенной записи.

2. Минимизация БФ в прямом и инверсном виде с помощью КК.

3. Перевод минимальной формы БФ в представление с базисными операциями.

4. Использование функций XOR и XNOR для минимизации БФ.

***2.*** *Подготовка к работе.*

Сдача работы производится на стандартном бланке, образец, которого показан в документе Pract Work#06 Tables.docx.

Карты Карно должны быть заготовлены заранее:

* в 3-х ТИ должны быть проставлены все комбинации булевых переменных в порядке возрастания двоичного кода и их десятичные эквиваленты,
* в 3-х картах Карно должны быть:
* разметка: двоичных кодов по краям,
* десятичных кодов – маленькими числами в углах ячеек.

***!!*** Функции для работы выдаются преподавателем только при наличии бланка (а не черновика в тетрадке).

На бланке приведены примеры заполнения результатов работы. Эта информация, конечно должна быть опущена – только три ТИ и три КК.

***3.*** *Задачи работы*.

Конечная задача по каждому пункту – создание схемы с наименьшим количеством *корпусов* ИМС. Если эта информация еще не дана в лекционном курсе, то справка:

Один корпус ЦИМС содержит:

**6** независимых элементов с **1-м** входом – инверторы,

**4** независимых элемента с **2-мя** входами с любой функцией – базовой или базисной,

**3** независимых элемента с **3-мя** входами, кроме 3ИЛИ(!!!),

**2** независимых элемента с **4-мя** входами, только 4И-НЕ(!!!)

Базовые функции: И, ИЛИ, НЕ.

Базисные функции: И-НЕ, ИЛИ-НЕ.

При неоднозначном по корпусам решении лучшим считается то, которое предусматривает меньшее количество типов ИМС.

*Пример, актуальный для данной П.Р.*

Решение может предусматривать два корпуса И-НЕ и ИЛИ-НЕ. Другое решение этой же задачи предусматривает тоже два корпуса, но оба И-НЕ; оно конечно, лучше.

***4.*** *Этапы работы.*

* преобразование "Сокращенная запись → КК"
* минимизация СДНФ с помощью карт Карно – самостоятельная подготовка,
* минимизация с использованием формул де Моргана,
* минимизация с использованием операций XOR и XNOR,
* минимизация для сокращения типов логических операций (см. Пример выше) – это может и отсутствовать.

***5****. Преобразования (теорема, формула) де Моргана.*

Цель применения – сокращение количества ЛЭ, участвующих в реализации.

В простейшем случае – замена 3-х БЛЭ на один базисный:

* два инвертора и один элемент И – на один элемент 2ИЛИ-НЕ

 (6.1)

* два инвертора и один элемент ИЛИ – на один элемент 2И-НЕ

 (6.2)

Преобразование используется и для переменных составного вида, например,

 (6.3)

Здесь выгода еще более очевидна, чем в (6.1), (6.2), т.к. ***один*** элемент 3И-НЕ заменяет ***три*** типа БЛЭ, на которые потребовались ***три различных корпуса***.

***Рекомендация***. Если нет явного вида формулы де Моргана, используйте *двойную инверсию*, как для всей правой части, так и для отдельных слагаемых.

***!!!*** По пп.1,2 все схемные реализации можно сделать, используя ***только один(!!!)*** корпус, т.е. *четыре элемента 2И-НЕ или четыре элемента 2ИЛИ-НЕ*.

***6****. Использование операций XOR и XNOR.*

Булевы функции операций

 (6.4)

 (6.5)

В отличие от формул де Моргана эта замена не имеет никакого формального обоснования. Главное то, что имеется корпус ИМС с 4-мя ЛЭ и каждый из них сразу производит операцию XOR над 2-мя переменными.

***!!!*** В EWB имеется идеальный элемент XNOR, но следует пользоваться связкой XNOR + инвертор = XOR, так ближе к реальности.

***!!!*** "Cвободный" элемент XOR при подаче "1" на один вход – это инвертор.

Здесь также возможен составной вид переменных, например

 (6.6)

***!!!*** По п3 все схемные реализации можно сделать, используя два корпуса, одним из который является XOR.

***7****. Схемы, реализующие БФ*.

В пп.1,3 дана сокращенная запись прямого вида БФ (позиции "1" в КК). Минимальная форма БФ и схема получаются для *прямой* функции.

В п.2 дана сокращенная запись инверсного вида БФ (позиции "0" в КК). Минимальная форма получается для *инверсной* функции, а перевод в *прямую* уже на уровне схемы.