**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Поволжский государственный университет телекоммуникаций и информатики»** Т.А. Коваленко, А.Г. Солодов

**«ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ЯЗЫКИ ПРОГРАММИРОВАНИЯ»**

**Часть 2**

Учебное пособие

Самара 2022

УДК - 004.772

ББК 3.32.97

К-56 Коваленко Т.А

Рекомендовано к изданию методическим советом ПГУТИ Протокол №48 от 12.04.2022

**Рецензенты:**

к.т.н. начальник отдела сбора информации ГНКЦ ЦСКБ «ПРОГРЕСС» Халилов Р.Р.

Коваленко Т.А

К-56 Вычислительная техника и языки программирования. Часть 2: учебное пособие/ Т.А. Коваленко, А.Г. Солодов – Самара: ПГУТИ, 2022. – 168 с.

Учебное пособие предназначено для студентов второго курса специальности 11.03.02 дневной и заочной формы обучения. В нем рассмотрены вопросы, относящиеся к области цифровой и вычислительной техники, достаточной для последующего изучения специальных дисциплин.

Пособие представлено в двух частях теоретической и практической. В теоретическое части дается представление об основных вопросах цифровой техники. Вторая часть состоит из лабораторных и практических работ, которые направлены на получение практических навыков синтеза цифровых устройств, знания элементов архитектуры современных цифровых сигнальных процессоров и практического их использования.

Пособие позволяет рассмотреть не только теоретические вопросы, но и выполнить самостоятельно лабораторные работы и практические задания. Согласно приказу МОН РФ «19» сентября 2017 г. № 930 и в соответствии с решением Ученого Совета ФГБОУ ВО ПГУТИ от «31» августа 2021г. (протокол № 1) учебное пособие направлено на усвоение принципы работы современных информационных технологий и использован их для решения задач профессиональной деятельности (ОПК-4).

Материал, представленный в учебном пособии, является актуальным. Он изложен доступным для студентов языком.

Учебное пособие является необходимым и полезным в учебном процессе.

ISBN

© Коваленко Т.А., Солодов А.Г., 2022

© Поволжский государственный университет телекоммуникаций и информатики, 2022

Содержание

Введение............................................................................................................... 6 Теоретическая часть............................................................................................ 7 Лекция 1 Логические основы цифровых устройств ........................................ 8

1.1. Двоичная система счисления................................................................... 8 1.2. Понятие функции алгебры логики........................................................ 10 1.3. Основные законы и тождества алгебры логики .................................. 14

Лекция 2 Комбинированные цифровые устройства...................................... 16 2.1. Понятие и последовательность синтеза ............................................... 16 2.2. Способы задания КЦУ ........................................................................... 19

Лекция 3 Методы минимизации ФАЛ ............................................................ 22 3.1 Вывод минимальной ФАЛ...................................................................... 22 3.2 Базисы и минимальные базисы .............................................................. 25 3.3. Построение структурной схемы............................................................ 26

Лекция 4 Типовые КЦУ. Дешифраторы и шифраторы................................. 31 4.1 Дешифратор.............................................................................................. 32 4.2 Шифратор ................................................................................................. 35

Лекция 5 Типовые КЦУ. Мультиплексор, демультиплексор....................... 39 и преобразователь кода..................................................................................... 39 5.1 Мультиплексор......................................................................................... 39 5.2 Демультиплексор ..................................................................................... 42 5.3 Преобразователь кода.............................................................................. 43 Лекция 6 Последовательные цифровые устройства...................................... 46 6.1. Понятие и способ задания ПЦУ............................................................ 46 6.2 Понятие и классификация триггеров..................................................... 49 6.3. Типовые триггеры................................................................................... 50 6.4 Синтез ПЦУ.............................................................................................. 56

3

Лекция 7 Принципы управления микропроцессора ...................................... 59 7.1 Классификация микропроцессоров........................................................ 60 7.2. Декомпозиция МП. ................................................................................. 62 7.3 Принцип аппаратного управления ("жёсткой" логики)....................... 63 7.4. Принцип микропрограммного управления (гибкой логики). ............ 64 7.5. Способы формирования сигналов управления в управляющих автоматах с "гибкой" логикой. ..................................................................... 67

Лекция 8 Элементы архитектуры ЦСП TMS320C6x..................................... 69 8.1. Данные ..................................................................................................... 69 8.2. Методы адресации операндов ............................................................... 72

Лекция 9 Структура ЦСП TMS320C6x........................................................... 74 Лекция 10 Структура командной строки ассемблера, вопросы особенности адресации и команд........................................................................................... 79

10.1 Основные команды процессора ‘C6x для целых чисел ..................... 81 10.2 Особенности методов базирования и индексации в C6x................... 82 10.3 Команды ввода исходных данных ....................................................... 85 10.4 Арифметические команды.................................................................... 86

Лекция 11 Организация стандартных алгоритмических структур .............. 91 Лабораторные работы....................................................................................... 96 Лабораторная работа 1...................................................................................... 97 Исследование логических элементов.............................................................. 97 Лабораторная работа 2...................................................................................... 99 Моделирование работы комбинационных цифровых устройств................. 99 Лабораторная работа 3.................................................................................... 102 Шифраторы и дешифраторы.......................................................................... 102 Лабораторная работа 4.................................................................................... 106 Мультиплексоры и демультиплексоры......................................................... 106 Лабораторная работа 5.................................................................................... 111 Триггеры........................................................................................................... 111

4

Лабораторная работа 6.................................................................................... 116 Знакомство с симулятором TMS320C6201................................................... 116 Лабораторная работа 7.................................................................................... 123

Пересылка данных ....................................................................................... 123 Лабораторная 8 ................................................................................................ 130 Арифметические операции ......................................................................... 130 Лабораторная работа 9.................................................................................... 134 Ветвление с простым условием.................................................................. 134 Практические занятия..................................................................................... 143 Практическая работа 1.................................................................................... 143 Элементарные ФАЛ. Законы и тождества алгебры логики........................ 143 Практическая работа 2.................................................................................... 145 Минимизация ФАЛ методом карт Вейча-Карно и ...................................... 145 минимальные базисы ...................................................................................... 145 Практическая работа 3.................................................................................... 147 Синтез дешифратора и шифратора................................................................ 147 Практическая работа 4.................................................................................... 148 Триггеры........................................................................................................... 148 Практическая работа 5.................................................................................... 149 Синтез двоичных счетчиков........................................................................... 149 Приложение ..................................................................................................... 150 Система команд TMS320С6х для чисел с фиксированной запятой........... 150 Предметный указатель.................................................................................... 165 Список используемой литературы............................................................. 166

5

**Введение**

Целью дисциплины «Вычислительная техника и языки программирования» является обеспечение базовой подготовки студентов очного и заочного отделений специальности 11.03.02 в области цифровой и вычислительной техники, достаточной для последующего изучения специальных дисциплин.

Изучение дисциплины «Вычислительная техника и языки программирования» предполагает самостоятельную работу с литературой, прослушивание лекций, лабораторные и практические занятия. Данный курс направлен на приобретение знания основных теоретических положений цифровой техники и практических навыков синтеза цифровых устройств, знания элементов архитектуры современных цифровых сигнальных процессоров и практического их использования, а также знания современных сетевых технологий.

В пособие рассматриваются вопросы архитектуры микропроцессоров и программного обеспечения микропроцессорных систем. Оно содержит необходимые сведения об архитектуре процессоров фирмы Texas Instruments, информацию о структуре программы на ассемблере и основных директивах, а также сведения о форматах машинных команд и правилах их записи в ассемблере.

Пособие позволяет рассмотреть не только теоретические вопросы, но и выполнить самостоятельно лабораторные работы.

Использование данного учебного пособия является хорошим подспорьем для студентов технических специальностей.

6

**ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

В теоретической части рассматриваются основные теоретические положения цифровой техники. Дается описание комбинированных, последовательных цифровых устройств. Понятия и способ их задания. Вопросы об архитектуре процессоров фирмы Texas Instruments, информация о структуре программы на ассемблере и основных директивах, а также сведения о форматах машинных команд и правилах их записи в ассемблере. Кроме этого обсуждаются некоторые приемы программирования на ассемблере. Приведены также форматы и примеры использования некоторых команд

**Цель теоретической части:** Дать представление в области цифровой и вычислительной техники, достаточной для последующего изучения специальных дисциплин, тем самым обеспечивая базовую подготовку студентов очного и заочного отделений специальностей 11.03.02,

7

**Лекция 1**

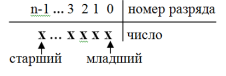
**Логические основы цифровых устройств**

В лекции рассматриваются понятия, двоичная система исчисления, переход от одной позиционной системы в другую, функции алгебры. Даются основные определения и законы, относящиеся к функции алгебры логики. Представлены элементы функции алгебры логики одного и двух аргументов.

**Цель лекции:** Ознакомиться с основными определениями систем исчисления, перевода из одной системы в другую, основами алгебры логики. Построение функций алгебры логики, основные элементы лежащие в их основе. Изучить законы функций алгебры логики.

**1.1. Двоичная система счисления**

**Разряд** – позиция, занимаемая отдельным символом данной системы счисления в изображении числа.



Разряды нумеруются справа налево, начиная с 0.

**Вес разряда** – количественное значение одной единицы, помещенной в данный разряд.

Численно вес разряда определяется через основание системы счисления (ОСС) и номер разряда: **ОСС<НОМЕР РАЗРЯДА>**.

ОСС = количеству символов ее алфавита.

Примеры:

8

**Разрядная сетка** – количество разрядов, отведенное для представления чисел в данной цифровой системе или устройстве. *Переход от одной позиционной системы счисления к другой* 1. Определение десятичного эквивалента:

n-1 n-2 0 N = x 10 n-1 n-2 0

⋅ ⋅ ⋅ ОСС + x ОСС +...+ х ОСС, где

xi – символ из алфавита некоторой системы счисления, размещенный в i-ом разряде соответствующего числа.

Пример для 100102: 1⋅24 + 0⋅23 + 0⋅22 + 1⋅21 + 0⋅20 = 1810

2. Определение двоичного эквивалента десятичного числа: А). Поразрядное взвешивание:

шаг 1. Определить наибольшее n, при котором 2n ≤ N10. Найденное n = номеру старшего разряда искомого двоичного числа, в который помещается 1.

Например, для N10 = 29 n = 4, т.к. 25 = 32 > 29.

шаг 2. Уменьшить n на 1.

шаг 3. Если n = 0 искомое число найдено. В противном случае продолжить.

шаг 4. Если 2n-1 ≤ N10-2n, в разряд (n-1) поместить 1. В противном случае 0.

В нашем примере 23 < 29-24. Следовательно, в 3-й разряд также помещаем 1.

шаг 5. Вернуться к шагу 2.

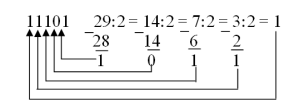
Следуя приведенному алгоритму, 2910 = 111012.

Данный алгоритм удобен для двоичной системы счисления. В случае произвольной системы счисления предпочтительным является следующий алгоритм.

Б). Последовательное деление на ОСС, пока оно больше очередного частного.

9

Например, для ОСС = 2:



**1.2. Понятие функции алгебры логики**

Функция, однозначно определяющая соответствие каждой из всех возможных совокупностей значений аргументов нулю (ложь) или единице (истина), называется функцией алгебры логики (ФАЛ).

Логические переменные – как функция, так и аргументы, могут принимать только два значения. Обычно их обозначают символами 0 и 1, что в вычислительной технике соответствует низкому и высокому уровню напряжения сигнала соответственно.

Для технической реализации ФАЛ используют электрические схемы, называемые логическими элементами. Логический элемент (ЛЭ) выполняет логические операции над одной или более логическими переменными. При этом переменная-функция соответствует выходу ЛЭ, а переменные-аргументы – разрядам двоичных наборов, поступающим на соответствующие входы ЛЭ. Фактически по внутреннему устройству любой ЛЭ это набор транзисторов соединенных по соответствующему принципу.

Закон функционирования ЛЭ представляют в виде таблицы истинности (Таблица 1.1).

10

Таблица 1.1

Закон функционирования ЛЭ

| Номер  двоичного набора | Аргументы | | | Функция |
| --- | --- | --- | --- | --- |
| **хn-1** | **…** | **х0** | **у0** |
| 0 | 0 | … | 0 |  |
| … | … | … | … | .. |
| 2n-1 | 1 | … | 1 |  |

В строках первого раздела этой таблицы по порядку записываются десятичные номера входных двоичных наборов. В строках второго раздела записываются собственно входные наборы, представляющие собой n разрядное двоичное отображение соответствующего десятичного номера. В строках третьего раздела записываются соответствующие входным наборам значения функции.

В общем случае ФАЛ строятся на основе элементарных ФАЛ. Элементарной называется ФАЛ одного или двух аргументов, в логическом выражении которой содержится не более одной логической операции. К элементарным ФАЛ одного аргумента относятся:

1. Константа нуля. Реализуется генератором нуля, который на схемах обозначается соединением соответствующего входа ЛЭ с «землей» (общим проводом источника питания).

2. Константа единицы. Реализуется генератором единицы, который на схемах обозначается соединением соответствующего входа ЛЭ с полюсом источника питания.

3. Повторение. Реализуется логическим элементом повторителем. Его условное графическое обозначение и таблица истинности показаны на рис.1.1. Записывается ФАЛ как у = х. Зачастую возникает вопрос, зачем нужен данный логический элемент? Повторитель: представляет собой просто усилительный каскад. В логическом смысле усиление является аналоговой функцией, а не цифровой, но буферы часто требуются и в цифровых схемах. Например, биты адреса при выходе из процессора

11

оказываются довольно слабыми по нагрузочной способности, и их нельзя подавать в шину адреса без усиления.

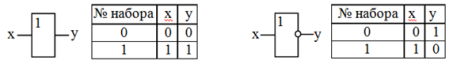
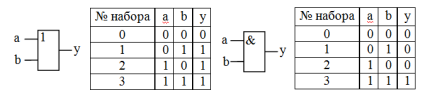
4. Инверсия или логическое отрицание. Реализуется логическим элементом НЕ (инвертором). Его условное графическое обозначение и таблица истинности показаны на рис. 1.2. Записывается ФАЛ как у = х . 

Рис.1.1 – Повторитель Рис.1.2 – Инвертор Основными из элементарных ФАЛ двух аргументов являются: 1. Дизъюнкция (логическое сложение). Реализуется логическим

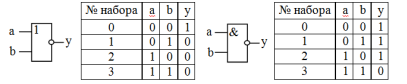
элементом ИЛИ. Его условное графическое обозначение и таблица истинности показаны на рис.1. 3. Записывается ФАЛ как у = a∨b. 2. Конъюнкция (логическое умножение). Реализуется логическим элементом И. Его условное графическое обозначение и таблица истинности показаны на рис.1.4. Записывается ФАЛ как у = a∧b. Рис. 1.3 – ИЛИ Рис.1.4 – И

3. Стрелка Пирса. Реализуется логическим элементом ИЛИ-НЕ. Его условное графическое обозначение и таблица истинности показаны на рис.1.5. Записывается ФАЛ как у = �� ↓ �� и может быть представлена в развернутой форме: �� = �� ⋁��.

4. Штрих Шеффера. Реализуется логическим элементом И-НЕ. Его условное графическое обозначение и таблица истинности показаны на

12

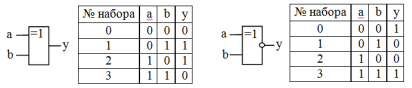
рис.1.6. Записывается ФАЛ как y = a | b и может быть представлена в развернутой форме: �� = �� ⋀��.

Рис.1.5 – ИЛИ-НЕ Рис.1.6 – И-НЕ

5. Исключающее ИЛИ (сложение по модулю два, XOR - eXception OR). Реализуется сумматором по модулю два. Его условное графическое обозначение и таблица истинности показаны на рис. 1.7. Записывается ФАЛ как �� = ��⨁�� и может быть представлена в развернутой форме: �� =

(�� ⋀��)⋁(�� ⋀��).

6. Эквивалентность (равнозначность). Реализуется одноименным ЛЭ. Его условное графическое обозначение и таблица истинности показаны на рис.1.8. Записывается ФАЛ как �� = �� ∼ �� и может быть представлена в развернутой форме:�� = ��⨁�� = (�� ∧ ��) ∨ (�� ∧ ��).

Рис.1.7 – Исключающее ИЛИ Рис. 1.8 – Эквивалентность Функции дизъюнкция и штрих Шеффера с одной стороны, конъюнкция и стрелка Пирса с другой, являются частными случаями функций конституанты (постоянной) нуля и единицы, соответственно. Функция конституанты единицы (нуля) от n аргументов обращается в единицу (ноль) лишь на каком-либо одном наборе аргументов и обращается в ноль (единицу) на остальных наборах.

Все рассмотренные функции могут быть расширены на произвольное число аргументов.

13

**1.3. Основные законы и тождества алгебры логики**

Тождества:

х ∨ х = х, х ∨ ��̅= 1, х ∨ 1 = 1, х ∨ 0 = х,

х ∧ х = х, х ∧ ��̅= 0, х ∧ 1 = х, х ∧ 0 = 0,

х ⊕ х = 0, х ⊕̅�� = 1, х ⊕ 1 = ��̅, х ⊕ 0 = х̅.

Законы:

−двойной инверсии – ��̿ = a;

−сочетательный – a∧(b∧с) = (а∧b)∧с, а∨(b∨с) = (а∨b)∨с,

а ⊕ (b ⊕ с) = (а ⊕ b) ⊕ с;

−переместительный – а∧b = b∧а, а∨b = b∨а, а ⊕ b = b ⊕ а; −распределительный – а∧(b∨с) = а∧b ∨ а∧с,

а∨(b∧с) = (а∨b)∧(а∨с),

а∧(b ⊕ с) = а∧b ⊕ а∧с;

−двойственности (правила де Моргана) – ��̅̅̅⋁̅̅��̅ = ��̅ ∧ ��̅, ��̅̅̅∧̅̅̅��̅ = ��̅ ∨ ��̅; −поглощения – а ∨ а∧с = а, a∧(a∨c) = a;

−склеивания – а∧��̅∨ a∧c = a, (a∨��̅)∧(a∨c) = a.

Все указанные законы и тождества справедливы для любого числа аргументов, причем аргументом может быть как простая переменная, так и функция.

Законы и тождества алгебры логики используются для преобразования ФАЛ в процессе синтеза цифровых устройств (ЦУ). В частности, целью преобразования может быть упрощение ФАЛ:

̅̅̅̅̅̅̅̅̅̅̅̅̅ ∨ �� ∧ ��̅= �� ∨ (�� ∨ ��̅)

�� = ��̅ ∙ (�� ∨ ��̅)

̅̅̅̅̅̅̅̅̅ ∨ �� ∧ ��̅= �� ∨ �� ∧ ��̅∨ �� ∧ ��̅= �� ∨ �� ⊕ ��

Кроме того, в тождествах отражены правила замены логических элементов одного типа логическими элементами другого типа. Минимизация ФАЛ при построении ЦУ на аппаратном уровне приводит к уменьшению количества используемых элементов, а при программной

14

реализации к уменьшению команд, а соответственно уменьшению энергопотребления и увеличению быстродействия.

**Краткие итоги**

Рассмотрены понятия, двоичной системы исчисления и перевода из одной системы в другую. Дано определение функции алгебры логики. Основные элементы ФАЛ: дизъюнкция, конъюнкция, стрелка Пирса, штрих Шеффера, исключающее ИЛИ и эквивалентность. Изучены основные законы тождества алгебры и логики.

**Вопросы для самопроверки**

1. Дайте определение ФАЛ.

2. Сколько значений могут принимать логические переменные? 3. Перечислите логические операции, относящиеся к элементарным ФАЛ одного аргумента.

4. Перечислите логические операции, относящиеся к элементарным ФАЛ двух аргументов.

5. Дайте определение конъюнкции.

6. Дайте определение дизъюнкции.

15

**Лекция 2**

**Комбинированные цифровые устройства**

В лекции рассматриваются цифровые устройства. Отражение двоичных наборов физических значений разрядов. Дается терминология, которая применяется в комбинированных цифровых устройствах (КЦУ). Описывается последовательность синтеза, способы задания КЦУ.

**Цель лекции:** Ознакомиться с основными определениями, применяемыми в КЦУ. Уяснить когда КЦУ называется полностью или не полностью (частично) определённым. Усвоить понятия последовательности синтеза и способы задания КЦУ.

**2.1. Понятие и последовательность синтеза**

Любое цифровое устройство можно рассматривать как преобразователь входных n-разрядных двоичных наборов в выходные m-разрядные двоичные наборы. Поскольку n определяется числом информационных входов цифрового устройства, длина (число разрядов) входных двоичных наборов постоянна и не зависит от их значения. Тогда общее число входных двоичных наборов не превышает 2n и, следовательно, диапазон десятичных чисел, которые могут быть обработаны таким цифровым устройством составляет от 0 до Мmax = 2n – 1. Отсюда легко решается обратная задача: число разрядов, необходимое для представления в цифровом устройстве абсолютных значений десятичных чисел от Мmin до Мmax определяется из соотношения: n = ⎤log2(Мmax + 1)⎡, где ⎤ x ⎡ – наименьшее целое, не меньшее х (округление в большую сторону).

Физически значения разрядов (логических переменных) двоичных наборов отображаются электрическими сигналами в одной из двух форм – потенциальной или импульсной.

16

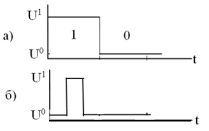
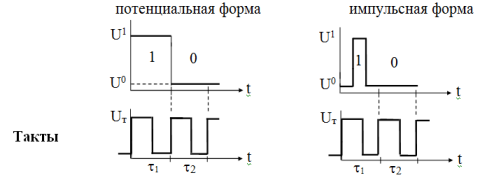


Рис. 2.1 – Физическое представление бит информации

В потенциальной форме (рис. 2.1,а) единичному значению двоичного разряда соответствует высокий уровень напряжения U1, а нулевому – низкий уровень U0, близкий к «земле». В импульсной форме (рис. 2.1,б) единичному значению двоичного разряда соответствует относительно короткий импульс, а нулевому – отсутствие импульса.

Как видно, в любой форме сигналы могут принимать только два значения и потому называются двоичными сигналами. В этой терминологии и определим КЦУ: комбинационным называется цифровое устройство, у которого выходные двоичные сигналы в любой момент времени зависят только от тех двоичных сигналов, которые поступают на вход устройства в тот же момент времени. Таким образом, сигналы на выходе КЦУ изменяются практически сразу после изменения входных сигналов.

Рис. 2.2 – Отображение двоичных сигналов

17

В реальных цифровых устройствах каждые входной и соответствующий ему выходной двоичные наборы отображаются двоичными сигналами лишь в течение определенного интервала времени, называемого тактом (рис. 2.2). При этом говорят, что устройство тактируется, то есть входные и выходные сигналы могут изменяться лишь с началом (окончанием) каждого последующего такта. Сами такты задаются генератором тактовой частоты.

КЦУ называется полностью определённым, если каждому из всех его возможных входных двоичных наборов поставлен в соответствие строго определенный двоичный набор на выходе. Если хотя бы для одного входного набора значение одного или более разрядов соответствующего выходного набора безразлично, КЦУ называется не полностью или частично определённым. На практике частично определенные КЦУ соответствуют ситуациям, когда некоторые двоичные наборы либо никогда не появляются на входе, либо никогда не оказываются востребованными на выходе.

Синтез любого КЦУ проводится в следующей последовательности: 1.Задается закон функционирования.

2.Для каждого из m выходов выводится минимальная ФАЛ, то есть ФАЛ с минимальным числом членов и минимальным числом аргументов в каждом члене.

3.При необходимости каждая минимальная ФАЛ записывается в заданном минимальном базисе.

4.В соответствии с системой минимальных ФАЛ строится структурная схема устройства.

18

**2.2. Способы задания КЦУ**

***Табличный.***

Правила работы задаются таблицей истинности. При этом в случае частично определенного КЦУ строки раздела «Выходной набор» таблицы, соответствующие безразличным входным наборам, заполняются символом тильда (рис. 2.3). Однако при большом значении n таблица истинности становится громоздкой и теряет наглядность.

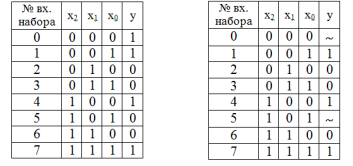


Рис. 2.3 – Пример таблицы истинности при безразличном нулевом наборе

***Скобочная запись таблицы истинности.***

В случае полностью определенного КЦУ для каждого разряда выходного набора в круглых (квадратных) скобках через запятую перечисляются десятичные номера входных наборов, на которых значение этого разряда обращается в 0 (1).

Полностью определенноеЧастично определенное

Рис. 2.4 – Скобочная запись таблицы истинности

На рис. 2.4 показана, запись уi(n = 3) = [0, 1, 4, 7] означает, что i-й

19

разряд выходных наборов имеет значение 1 на 0, 1, 4 и 7 n-разрядных входных наборах. Следовательно, на остальных (23 – 4 = 4) не указанных наборах, этот разряд имеет значение 0.

В случае частично определенного КЦУ используются оба вида скобок, что позволяет не указывать безразличные входные наборы. На рис. 2.4, запись уi(n = 3) = [1, 4, 7, (2, 3, 6)] означает, что i-й разряд выходных наборов принимает значение 1 на первом, четвертом и седьмом входных наборах, значение 0 – на втором, третьем и шестом, и его значение безразлично на остальных (23 – 6 = 2) не указанных входных наборах.

***Аналитический.***

Правила работы КЦУ задаются системой ФАЛ, каждая из которых соответствует определенному выходу КЦУ и записывается на основании таблицы истинности или ее скобочного представления в любой из двух совершенных форм – дизъюнктивной (СДНФ) или конъюнктивной (СКНФ).

***Правила записи ФАЛ в совершенной форме:***

−ФАЛ в СДНФ (СКНФ) представляется дизъюнкцией (конъюнкцией) своих членов, каждый из которых соответствует единственному и строго определенному входному набору, обращающего данную функцию в 1 (0) или соответствующего безразличному ее значению;

−каждый член функции в СДНФ (СКНФ) образуется конъюнкцией (дизъюнкцией) всех аргументов, которые берутся с инверсией при нулевом (единичном) значении в данном входном наборе и без инверсии – при единичном (нулевом). Таким образом, каждый член функции в СДНФ (СКНФ) является функцией конституанты 1 (0) (таб.2.1).

20

Таблица 2.1

Примеры записи ФАЛ в совершенной форме

| № вх.  набора | х2 | х1 | х0 | у |
| --- | --- | --- | --- | --- |
| 0 | **0** | **0** | **0** | **1** |
| 1 | **0** | **0** | **1** | **1** |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | **1** | **0** | **0** | **1** |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | **1** | **1** | **1** | **1** |

| № вх.  набора | х2 | х1 | х0 | у |
| --- | --- | --- | --- | --- |
| 0 | **0** | **0** | **0** | **~** |
| 1 | 0 | 0 | 1 | 1 |
| 2 | **0** | **1** | **0** | **0** |
| 3 | **0** | **1** | **1** | **0** |
| 4 | 1 | 0 | 0 | 1 |
| 5 | **1** | **0** | **1** | **~** |
| 6 | **1** | **1** | **0** | **0** |
| 7 | 1 | 1 | 1 | 1 |

усднф = ~~��~~2 ∙ ~~��~~1 ∙ ~~��~~0 ⋁ ~~��~~2 ∙ ~~��~~1 ∙ ��0⋁ ��2 ∙ ~~��~~1 ∙ ~~��~~0. . . ускнф = (��2 ⋁ ��1 ⋁��0)⋀(��2 ⋁ ~~��~~1 ⋁��0).. набор 0 набор 1 набор 4набор 0набор 2

**Краткие итоги**

Рассмотрены понятия КЦУ. Даны понятие, и последовательность синтеза. Изучены как двоичные наборы отображаются двоичными сигналами и способы задания КЦУ.

**Вопросы для самопроверки**

1. Дайте определение цифрового устройства.

2. Дайте определение КЦУ.

3. Что означает отображение электрическими сигналами двоичных наборов в потенциальной форме?

4. Что означает отображение электрическими сигналами двоичных наборов в импульсной форме?

5. Какова последовательность синтеза любого КЦУ?

6. Перечислите способы задания КЦУ.

7. Опишите табличный способ задания КЦУ.

8. Опишите аналитический способ задания КЦУ.

9. В чем заключается скобочная запись таблицы истинности для задания КЦУ?

10. Перечислите правила записи ФАЛ в совершенной форме.

21

**Лекция 3**

**Методы минимизации ФАЛ**

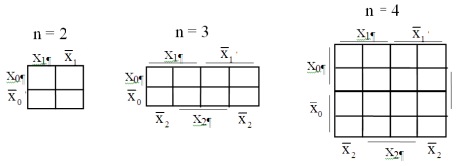
В лекции рассматриваются выводы минимальной ФАЛ методом графическим с помощью карт Вейча-Карно, и алгебраическим – метод Квайна. Понятие минимального базиса как минимальный набор ФАЛ, позволяющий представить любую функцию от произвольного числа аргументов.

**Цель лекции:** Ознакомиться с основными методами минимизации ФАЛ. Уяснить чем отличается графический от алгебраического метода. Усвоить понятия минимального базиса. Рассмотреть построение схемы цифрового устройства(ЦУ)

**3.1 Вывод минимальной ФАЛ**

Существуют графические и алгебраические методы минимизации ФАЛ. Из графических наибольшее распространение получил метод карт Вейча-Карно, а из алгебраических – метод Квайна.

Карта Вейча-Карно представляет собой таблицу истинности специальной формы с числом клеток 2n (рис. 3.1), где n – длина входных наборов КЦУ.

Рис 3.1 – Формат карт Вейча-Карно

Минимизация методом карт Вейча-Карно проводится в следующей

22

последовательности.

1. Каждая клетка карты соответствует определенному члену исходной ФАЛ. Местоположение клетки определяется пересечением строк и столбцов карты, одноименных с аргументами данного члена ФАЛ. Найденная клетка отмечается 1 в случае СДНФ, 0 в случае СКНФ и тильдой, если данный член ФАЛ соответствует безразличному значению функции.

2. Отмеченные и только отмеченные клетки объединяются в замкнутые области. При этом:

– каждая область должна представлять собой прямоугольник с числом клеток 2k, где k = 0, 1, 2, ... Следовательно, во-первых, область может содержать одну клетку (k = 0), две (k = 1), четыре (k = 2), восемь (k = 3) и т.д., но не 3, 5, 6, 7 и т.д. клеток. Во-вторых, нельзя объединять клетки, расположенные по диагонали;

– число клеток в области должно быть максимально возможным, поскольку только тогда число аргументов в соответствующем члене выводимой функции будет минимальным;

– одни и те же клетки могут входить в разные области, т.е. области могут пересекаться;

– при условии участия всех отмеченных клеток в процедуре формирования областей следует стремиться к минимальному числу областей, поскольку только тогда число членов выводимой функции будет минимальным. С этой целью допускается сворачивание карты в цилиндр относительно горизонтальной или вертикальной осей с соединением противоположных граней, также возможно объединение 4-х углов.

Учет безразличных входных наборов (клеток, помеченных тильдой) повышает эффективность минимизации. Однако если эти клетки не способствуют расширению областей, их учитывать не следует. 3. В соответствии с выделенными областями записывается

23

минимальная ФАЛ в дизъюнктивной (если исходной была СДНФ) или конъюнктивной (если исходной была СКНФ) нормальной форме. При этом каждая область определяет отдельный член минимальной функции, который составляется лишь из тех аргументов, которые в данную область входят либо только с инверсией, либо только без инверсии (Рис. 3.2).

Рис. 3.2 – Вывод минимальной ФАЛ методом карт Вейча-Кано Минимизация методом Квайна заключается в следующем.

1. Члены исходной ФАЛ, где учтены и безразличные входные наборы, отличающиеся только в одной переменной, группируются в пары. Две ФАЛ отличаются в одной переменной, если эта переменная в одну из них входит с инверсией, а в другую без инверсии. Других различий нет.

При группировке руководствуются следующими правилами: –пары могут пересекаться, т.е. один и тот же член функции может входить в различные пары;

–при условии участия в процедуре группирования всех членов ФАЛ, число пар должно быть минимальным.

2. К сформированным парам в случае СДНФ применяется операция склеивания, а в случае СКНФ – операция поглощения. 3. По результатам преобразования записывается новая ФАЛ, к которой применяются законы и тождества алгебры логики или повторно все пункты минимизации.

24

Пример для ФАЛ у/n=3 = [0, 1, 7, (2, 6)].

Шаг 1. Запись ФАЛ в любой из совершенных форм.

Шаг 2. Группирование:

1 пара 2 пара 3 пара у = х х х 2 1 0∨х х х 2 1 0∨х х х 2 1 0∨ [х х х 2 1 0∨х х х 2 1 0∨х х х 2 1 0]

Шаг 3. Применить закон склеивания (в случае СДНФ) или поглощения (в случае СКНФ):

1 пара: х х х 2 1 0∨х х х 2 1 0= х х2 1

2 пара: х х х 2 1 0∨х х х 2 1 0= хх1 0

3 пара: х х х 2 1 0∨х х х 2 1 0= х х2 1

Шаг 4. Записать новую ФАЛ и при возможности группирования возврат к шагу 2:

1 пара

(шаг 2) у = х х2 1∨хх1 0∨ [х х2 1];

(шаг 3) 1 пара: х х2 1∨х х2 1= х1;

(шаг 4) новая ФАЛ: у = х1∨хх1 0.

В противном случае продолжить.

Шаг 5. Если возможно, применить законы и тождества алгебры логики:

у = х х х х х х х (х х ) х х 1 1 0 1 1 0 1 1 0 1 0 ∨ = = ∨ == х х 1 0 ∨ . Шаг 6. Объявить конечный результат.

При этом полезным бывает следующее соотношение ��̅∙ ��⋁�� = �� ∙ ��.

**3.2 Базисы и минимальные базисы**

Полным базисом называется система ФАЛ, позволяющая представить любую функцию от произвольного числа аргументов. Полный

25

базис допускает использование логических элементов самых различных типов: И, ИЛИ, И-НЕ, ИЛИ-НЕ, НЕ и т.д.

Минимальным базисом называется минимальный набор ФАЛ, позволяющий представить любую функцию от произвольного числа аргументов. Минимальный базис допускает использование логических элементов только одного строго определенного типа. Из минимальных базисов наибольшее практическое применение получили базисы И-НЕ и ИЛИ-НЕ.

ФАЛ в базисе ИЛИ-НЕ может содержать только операции стрелка Пирса, а в базисе И-НЕ – только операции штрих Шеффера. Вместе с тем допустимо использование операции инверсии, а, следовательно, в технической реализации – логического элемента НЕ.

Для представления минимальной ФАЛ в любом из минимальных базисов используются законы двойной инверсии и двойственности. Например, ФАЛ �� = ��2 ∙ ��3⋁��1 ⋅ ��0 запишется в базисе И-НЕ следующим ̿̿̿̿̿̿̿̿̿̿̿̿̿̿̿̿̿̿̿ = ��̅̅̅̅2̅̅̅̅⋅̅̅̅̅��̅̅3̅̅̅⋀̅̅��̅̅̅1̅̅̅̅⋅̅̅̅̅��̅̅0̅̅.

образом: �� = ��2 ∙ ��3⋁��1 ⋅ ��0

**3.3. Построение структурной схемы**

Структурная схема ЦУ представляет собой совокупность условных графических изображений логических элементов, типовых цифровых устройств и связей между ними.

Элементной базой при технической реализации цифровых устройств являются интегральные схемы различной степени интеграции. На структурных схемах компоненты микросхем нумеруются слева направо и сверху вниз, то есть построчно. При этом используются две латинские буквы DD, за которыми следуют номер микросхемы в устройстве и через точку номер компоненты в микросхеме. Если микросхема содержит один компонент (логический элемент или типовое устройство), то используется одна буква D и номер компоненты в

26

нумерации отсутствует.

Последовательность построения структурной схемы устройства, реализующего данную ФАЛ или систему ФАЛ, определяется приоритетностью логических операций: старшей является инверсия, затем следует конъюнкция и, наконец, операции типа дизъюнкции – дизъюнкция, исключающее ИЛИ и эквивалентность.

При разработке структурной схемы необходимо учитывать такой параметр микросхем, как коэффициент разветвления по выходу Краз, определяющий допустимое число входов интегральных компонент, одновременно подключаемых к выходу компоненты данной микросхемы при сохранении ее работоспособности в заданных условиях эксплуатации. Значение Краз определяется типом выходного каскада компонент микросхемы. Помимо этого тип выходного каскада определяет способ организации мультиплексной линии.

Мультиплексирование – поочередное подключение к линии передачи на время Т выхода каждого из N логических элементов или цифровых устройств.

Выходные каскады выполняются в одном из трех основных вариантов. 1. Обычный каскад (рис. 3.3, а). Краз.= 10.

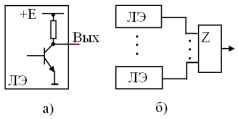


Рис. 3.3 – Обычный выходной каскад

Выход имеет два устойчивых состояния – 0 (транзистор открыт) и 1 (транзистор закрыт).

При организации мультиплексной линии (рис. 3.2,б) требуется дополнительный ЛЭ Z, тип которого (И либо ИЛИ) зависит от значения

27

выходного сигнала логических элементов в состоянии закрытого ключа. 2. Каскад с открытым коллектором (рис. 3.4,а). Краз.= 16.

Для обеспечения устойчивого состояния 1 выход через внешний резистор подключается к источнику питания.

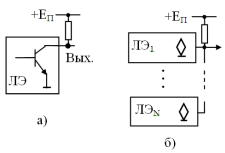
Повышенное значение Краз объясняется возможностью регулирования выходного тока путем изменения сопротивления резистора. 

Рис 3.4 – Каскад с открытым коллектором

Исполнение выходных каскадов с открытым коллектором отмечается на корпусе микросхемы символом в виде подчеркнутого ромба (рис. 3.4,б). Мультиплексная линия образуется соединением выходов всех N логических элементов на одном общем внешнем резисторе (рис. 3.4,б). Такое включение эквивалентно использованию дополнительного логического элемента ИЛИ, поэтому его называют "монтажным (проводным) ИЛИ".

3. Каскад с тремя состояниями (рис. 3.5,а). Краз.= 10.



Рис 3.5 – Каскад с тремя состояниями

28

Здесь возможны три ситуации:

–напряжение логического нуля на выходе соответствует открытому нижнему транзистору;

–напряжение логической единицы на выходе соответствует закрытому нижнему и открытому верхнему транзистору;

–при обоих закрытых транзисторах выход отключен от цепей питания. Это и есть третье (безразличное) состояние.

Исполнение выходных каскадов с тремя состояниями отмечается на корпусе микросхемы символом в виде ромба с поперечной линией (рис. 3.5,б).

Мультиплексная линия образуется «монтажным ИЛИ» без дополнительного резистора (рис. 3.5,б).

**Порядок построения схемы***:*

1. Реализуется инверсия отдельной переменной, а затем группы переменных (штрих Шеффера, стрелка Пирса) в порядке возрастания их численности.

2. Реализуется операция конъюнкции.

3. Реализуются операции типа дизъюнкции (дизъюнкция, сумма по модулю два, эквивалентность) (рис. 3.6).

Рис. 3.6 – Пример реализации ФАЛ у = хх ххх10 123∨.

29

**Краткие итоги**

Изучены основные методы минимизации ФАЛ. Рассмотрены понятия минимального базиса, а так же построение схемы ЦУ. Даны понятия интегральной схемы, мультиплексированию. Изучены три вида выходных каскадов: обычный, с открытым коллектором и тремя состояниями.

**Вопросы для самопроверки**

1. Опишите метод минимизации ФАЛ с помощью Карты Вейча Карно.

2. Опишите метод минимизации Квайна.

3. Что называется полным базисом?

4. Что называется минимальным базисом?

5. Что такое структурная схема ЦУ?

6. Дайте определение интегральной схемы.

7. Дайте определение понятию мультиплексирование.

8. Опишите обычный каскад.

9. Опишите каскад с открытым коллектором.

10. Опишите каскад с тремя состояниями.

30

**Лекция 4**

**Типовые КЦУ. Дешифраторы и шифраторы.**

В лекции рассматриваются типовые КЦУ дешифраторы и шифраторы. Их принцип работы и построения типовых КЦУ. **Цель лекции:** Ознакомиться с основными принципами работы типовых КЦУ дешифраторов, шифраторов. Рассмотреть синтез дешифратора и шифратора. Структуру построения дешифратора.

На входы типовых КЦУ могут подаваться два вида сигналов – информационные сигналы и сигналы управления. Информационные сигналы отображают обрабатываемую информацию, а сигналы управления выполняют одну или несколько из следующих функций:

− переключение входных информационных сигналов на определенные выходы устройства;

− задание порядка формирования выходных сигналов;

− синхронизация работы устройства. Синхросигнал задает временной интервал между любыми двумя соседними моментами срабатывания устройства и отображается тактовыми импульсами (тактами);

− стробирование устройства. Сигнал стробирования задает временной интервал, в течение которого выходы устройства разблокированы и на них передается результат обработки информационных сигналов.

Разрешающее значение управляющего сигнала называется активным. На условном графическом обозначении типовых КЦУ всегда указывается именно активный управляющий сигнал: нулевой знаком инверсии , V

единичный – его отсутствием . Аналогично для некоторых КЦУ V

указываются и активные значения выходных сигналов.

31

К основным из типовых КЦУ относятся дешифраторы, шифраторы, мультиплексоры, демультиплексоры и преобразователи кода.

**4.1 Дешифратор**

**Дешифратором** называется КЦУ, преобразующее n-разрядное двоичное число на входе в активный сигнал на одном из m выходов. Основное назначение дешифраторов – формирование управляющих сигналов.

Дешифратор с n входами и m = 2nвыходами называется полным, а в случае m < 2n – неполным (частично определенное КЦУ). На рис. 4.1 приведено условное графическое обозначение дешифраторов на примере полного дешифратора 2×4 (n = 2, m = 4) со стробированием (вход V) и инверсными выходами (активным нулевым выходным сигналом). Информационные входы обозначаются весовыми коэффициентами двоичных разрядов, что устанавливает однозначное соответствие между номером входа дешифратора и номером разряда двоичного набора. Для выходов используется сквозная нумерация. Таблица истинности полных дешифраторов (Рис. 4.1 (б), ограниченная дешифратором Рис. 4.1 (а)).

Рис. 4.1 – Дешифратор и его таблица истинности

Из таблицы видно, что номер выхода, на котором появляется активный сигнал, является десятичным эквивалентом текущего двоичного набора на информационных входах. В маркировке микросхем

32

дешифраторов используются буквы ИД.

На базе любого дешифратора n×m можно строить дешифратор большего размера, путем каскадной схемы соединения.

Пример: дешифратор 4×16 на базе дешифратора 2×4 (рис.4.2). Количество микросхем первой ступени: К1 = ⎤ N/М ⎡ (в примере К1 = 16/4 = 4). Их информационные входы соединены параллельно и являются информационными входами младших разрядов искомого дешифратора. По правилу работы дешифратора разрешенной должна быть только одна из К1 микросхем. Этот принцип реализуется с помощью второй ступени дешифраторов, управляющих дешифраторами первой ступени по входам стробирования. Число микросхем этой ступени: К2 = ⎤ К1/М ⎡ (в примере К2 = 4/4 = 1). Их информационные входы также соединяются параллельно и являются информационными входами следующих разрядов искомого дешифратора.

Аналогично организуются третья и т.д. ступени, каждая из которых управляет предыдущей ступенью.

Наконец, информационные входы дешифратора последней ступени являются информационными входами старших разрядов искомого дешифратора.



Рис. 4.2 – Дешифратор 4×16 на базе микросхем дешифратора 2×4

33

Таким образом, информационные выходы дешифратора второй ступени используются в качестве управляющих для дешифраторов первой ступени.

**Синтез полного дешифратора**

Пример: синтез дешифратора 2×4 с инверсными выходами в полном базисе (рис.4.3).

Рис.4.3 – Синтез дешифратора 2×4 с инверсными выходами **Синтез неполного дешифратора**

Пример: синтез дешифратора 3×5 с прямыми выходами в полном базисе (рис.4.4)

Рис.4.4 – Синтез дешифратора 3×5 с прямыми выходами в полном базисе

34

**4.2 Шифратор**

**Шифратором** называется КЦУ, преобразующее активный сигнал на одном (неприоритетный шифратор) или нескольких (приоритетный шифратор) из n информационных входов в m-разрядное двоичное число на выходе.

Шифраторы используются в устройствах ввода информации в цифровые системы и устройства.

Шифратор с m выходами и n = 2m входами называется полным, в противном случае – неполным (частично определенное КЦУ). 

Рис. 4.5 – Полный шифратор

На рис. 4.5 приведено условное графическое обозначение шифраторов на примере полного шифратора 4×2 (n = 4, m = 2) с инверсными входами (с активным нулевым сигналом на входе). Для входов используется сквозная нумерация, а выходы обозначаются весовыми коэффициентами двоичных разрядов, что позволяет правильно определить двоичное число на выходе шифратора.



Рис. 4.6 – Таблица истинности полного шифратора

Таблица истинности неприоритетного и приоритетного (значения

35

для входов указаны в скобках) полных шифраторов, ограниченная шифратором 4×2 показана на рис. 4.6. Из таблицы следует, что двоичное число на выходе неприоритетного шифратора соответствует десятичному номеру активного входа. Для приоритетного шифратора допускается наличие активного сигнала одновременно на нескольких входах. В этом случае двоичное число на выходе соответствует наибольшему по номеру активному входу. Из таблицы также видно, что шифраторам свойственна неопределенность: одно и то же двоичное число на выходе образуется как при активном старшем входе, так и пассивных всех входах. Если к тому же шифратор дополняется входом стробирования, образуется еще одна неопределенность: в режиме блокирования все выходы шифратора тоже будут установлены в единичное значение. Для идентификации этих ситуаций в интегральных шифраторах предусматриваются два служебных выхода.

В маркировке микросхем шифраторов используются буквы ИВ. Рассмотрим пример построения шифратора для преобразования десятичного одноразрядного кода (десятичных чисел от 0 до 9) в двоичный код. При этом предполагается, что сигнал, соответствующий логической единице, в каждый момент времени подаётся только на один вход. Условное обозначение шифратора и таблица соответствия кода (рис.4.7) Рис.4.7 – Шифратор с таблицей истинности

36

На практике часто используют шифратор с приоритетом. В таких шифраторах код двоичного числа соответствует наивысшему номеру входа, на который подан сигнал «1», т. е. на приоритетный шифратор допускается подавать сигналы на несколько входов, а он выставляет на

выходе код числа, соответствующего старшему входу.

Рассмотрим в качестве примера шифратор с приоритетом (приоритетный шифратор) К555ИВЗ серии микросхем К555 (ТТЛШ) (рис.4.8).

Шифратор имеет 9 инверсных входов, обозначенных через RI,..., PR9. Аббревиатура PR обозначает «приоритет». Шифратор имен четыре инверсных выхода В1, ... , B8. Аббревиатура В означает «шина» (от англ. bus). Цифры определяют значение активного уровня (нуля) в соответствующем разряде двоичного числа. Например, В8 обозначает, что ноль на этом выходе соответствует числу 8. Очевидно, что это неполный шифратор.



Рис.4.8 – Шифратор

Если на всех входах — логическая единица, то на всех выходах также логическая единица, что соответствует числу 0 в так называемом инверсном коде (1111).Если хотя бы на одном входе имеется логический ноль, то состояние выходных сигналов определяется наибольшим номером входа, на котором имеется логический ноль, и не зависит от сигналов на входах, имеющих меньший номер.

37

**Краткие итоги:**

Изучили основные принципы работы типовых КЦУ дешифраторов, шифраторов. Рассмотрели понятие управляющего сигнала, таблицы истинности типовых КЦУ. Выяснили в чем различие шифраторов и дешифраторов.

**Вопросы для самопроверки**

1. Какие сигналы подаются на входы типовых КЦУ?

2. Дайте определение информационного сигнала.

3. Дайте определение сигнала управления.

4. Дайте определение дешифратора.

5. Каково основное назначение дешифраторов?

6. Сколько выходов имеет полный дешифратор с тремя входами? 7. Дайте определение шифратору.

8. Каково основное назначение шифратора?

9. Чем отличается работа шифратора от дешифратора?

38

**Лекция 5**

**Типовые КЦУ. Мультиплексор, демультиплексор и преобразователь кода.**

В лекции рассматриваются типовые КЦУ мультиплексор, демультиплексор и преобразователь кода. Их принцип работы и построения.

**Цель лекции:** Ознакомиться с основными принципами работы типовых КЦУ мультиплексора, демультиплексора и преобразователя кода. Рассмотреть синтез мультиплексора, демультиплексора и преобразователя кода. Структуру построения мультиплексора.

**5.1 Мультиплексор**

**Мультиплексором** называется КЦУ, обеспечивающее подключение к единственному выходу одного из n информационных входов, выбор которого производится m-разрядным двоичным числом, поступающим на управляющие (адресные, селективные) входы. Очевидно, число селективных входов m = ⎤ log2(n) ⎡.

Мультиплексоры используются для организации мультиплексной линии или перехода от параллельной передачи двоичных наборов (всех разрядов в одном такте) к последовательной (поразрядной). 

Рис. 5.1 – Мультиплексор

На рис. 5.1 приведены таблица истинности и условное графическое

39

обозначение мультиплексоров на примере мультиплексора 4×1 (n = 4, один выход). Буквой D обозначены информационные входы, а буквой А – адресные. Индекс при букве А обозначает номер разряда соответствующего двоичного числа. Используется и сквозная нумерация входов цифрами от 0 до n+m–1, но селективные (адресные) входы всегда узнаются по полочке, отделяющей их от информационных входов.

В маркировке микросхем мультиплексоров используются буквы КП. На базе любого мультиплексора n×1 можно построить мультиплексор большего размера, путем каскадной схемы соединения. Пример: мультиплексор 16×1 на базе микросхем мультиплексора 4×1:

Количество микросхем первой ступени: К1 = ⎤ N/М ⎡. Их селективные входы соединены параллельно и являются селективными входами младших разрядов искомого мультиплексора.

Последующие ступени используются для постепенного уменьшения числа выходов до одного. Так, число микросхем второй ступени: К2 = ⎤К1/М⎡ (в примере К2 = 4/4 = 1). Их селективные входы также соединяются параллельно и являются селективными входами следующих разрядов искомого мультиплексора. Информационные же входы соединяются с выходами мультиплексоров первой ступени.

Аналогично организуются третья и т.д. ступени.

Наконец, селективные входы мультиплексора последней ступени являются селективными входами старших разрядов искомого мультиплексора (Рис 5.2)

40



Рис. 5.2 Мультиплексор 16×1 на базе микросхем мультиплексора 4×1 **Синтез полного мультиплексора**

Пример: синтеза мультиплексора 4×1 без стробирования в полном базисе (Рис. 5.3).

у = A A0 1D0 ∨A A0 1D1 ∨A A0 1D2 ∨ А1А0D3. 

у = y0D0 ∨ у1D1 ∨ у2D2 ∨ у3D3, где у0, …, у3 –

выходы дешифратора 2×4.

Рис. 5.3 – Синтеза мультиплексора 4×1

41

В результате такой замены синтез мультиплексора сводится к синтезу дешифратора: где у0 соответствует D0, у1 – D1 и т.д. Такой подход позволяет легко записать минимальную ФАЛ для любого из входов частично определенного мультиплексора.

**5.2 Демультиплексор**

**Демультиплексором** называется КЦУ, обеспечивающее подключение единственного информационного входа к одному из m выходов, выбор которого осуществляется n-разрядным двоичным числом на управляющих входах.

Демультиплексоры решают задачу, обратную задаче мультиплексирования.



Рис. 5.4 – Демультеплексор

На рис. 5.4 приведено условное графическое обозначение демультиплексоров на примере демультиплексора 1×4 (один информационный вход и четыре выхода). Управляющие входы обозначаются весовыми коэффициентами двоичных разрядов, а для выходов используется сквозная нумерация.



Рис. 5.5 - Таблица истинности демультиплексора

42

Таблица истинности демультиплексоров, ограниченная демультиплексором 1×4 (рис. 5.4), показана на рис. 5.5. Из таблицы следует, что номер выхода, к которому подключается информационный вход D, является десятичным эквивалентом двоичного числа х1х0 на управляющих входах. Кроме того, нетрудно видеть, что в качестве демультиплексора вполне можно использовать дешифратор со стробированием, если считать стробирующий вход информационным, а информационные входы – управляющими. По этой причине в интегральном исполнении демультиплексоры не выпускаются, а дешифратор со стробированием называют дешифратором демультиплексором, подчеркивая тем самым возможность выполнения им двух функций.

**5.3 Преобразователь кода**

**Преобразователями кода** называются КЦУ, реализующие процедуру кодирования – изменение закона расположения нулей и единиц относительно исходных двоичных наборов.

В интегральном исполнении выпускаются только преобразователи двоичного кода в двоично-десятичный или семисегментный код, а также двоично-десятичного кода в двоичный. В маркировке таких микросхем используются буквы ПР, например, К155ПР6. Другие преобразователи кода либо синтезируются как КЦУ, либо реализуются на базе программируемой логической матрицы (ПЛМ).

ПЛМ – это универсальная комбинационная схема, обеспечивающая преобразование входных n-разрядных кодовых слов в m-разрядные кодовые слова на выходе. Структурно (рис. 5.6) ПЛМ состоит из трех уровней логических элементов и двух матриц соединительных линий М1 и М2.

43



Рис. 5.6 – Структура ПЛМ

В исходном состоянии во всех точках пересечения соединительных линий обеих матриц электрический контакт может быть либо обеспечен, либо отсутствовать. В первом случае место соединения выполняется в виде плавкой перемычки, а во втором – в виде р-n перехода. В соответствии с этим программирование ПЛМ заключается либо в разрушении определенных контактов путем пережигания плавкой перемычки, либо в их установлении путем пробоя р-n перехода.

Закон преобразования кодовых слов ПЛМ представленной структуры описывается системой ФАЛ, записанных в СДНФ: {yi = ∨zj}, где zj – j-й член i-й функции. Соответственно с этим логические элементы первого уровня обеспечивают прямые и инверсные значения входных переменных, необходимые для формирования членов ФАЛ. Кроме того, второй ряд логических элементов этого уровня введен с целью обеспечения минимальной нагрузки для генератора исходного кода. Собственно члены ФАЛ zj формируются логическими элементами второго уровня совместно с матрицей М1. Элементы третьего уровня совместно с матрицей М2 обеспечивают необходимую структуру каждой из m ФАЛ.

44

**Краткие итоги:**

Изучили основные принципы работы типовых КЦУ мультиплексоров, демультиплексоров и преобразователей кода. Рассмотрели понятие управляющего сигнала, таблицы истинности типовых КЦУ. Выяснили в чем различие мультиплексоров и демультиплексоров.

**Вопросы для самопроверки**

1. Дайте определение мультиплексора?

2. Когда применяется мультиплексор?

3. Каким выражением описывается мультиплексор?

4. Дайте определение демультиплексора.

5. Какие задачи решает демультиплексор?

6. Назначение стробирующего входа.

7. Чем отличаются мультиплексор от демультиплексора? 8. Дайте определение преобразователю кода.

9. Опишите работу преобразователя кода.

45

**Лекция 6**

**Последовательные цифровые устройства**

В лекции рассматриваются последовательные цифровые устройства (ПЦУ). Дается понятие основных компонентов ПЦУ. Описывается принцип работы триггеров и синтез ПЦУ.

**Цель лекции:** Ознакомиться с основными принципами работы типовых ПЦУ. Дать понятия способа задания ПЦУ и триггера. Изучить классификацию триггеров. Рассмотреть поэтапно синтез ПЦУ

**6.1. Понятие и способ задания ПЦУ**

Последовательным называется цифровое устройство с памятью, в котором текущие двоичные сигналы на выходе зависят как от текущих двоичных сигналов на входе, так и от предыдущих состояний устройства.

Под состоянием ПЦУ понимается двоичная информация, считываемая из его памяти в данный момент времени.

ПЦУ синхронизируются тактовыми импульсами и работают циклами, автоматически переходя с каждым тактом из одного состояния в другое, формируя при этом на выходе определенные двоичные сигналы. Каждый цикл всегда начинается и заканчивается некоторым заранее определенным начальным состоянием.

Наиболее общей моделью ПЦУ является автомат Мили (рис. 6.1). 

Рис. 6.1 – Автомат Мили

Запоминающие элементы (ЗЭ) хранят все состояния цикла работы

46

ПЦУ в заданной последовательности в виде k-разрядных двоичных наборов. КЦУ1 в зависимости от текущих сигналов Х = {xi} на входе и текущего состояния Q = {qj} ПЦУ вырабатывает двоичные сигналы а1, …, аК управления памятью. Эти сигналы определяют состояние, в которое ПЦУ перейдёт в следующем такте. КЦУ2 в зависимости от текущих двоичных сигналов на входе и текущего состояния ПЦУ вырабатывает выходные двоичные сигналы.

Автомат Мили задаётся двумя системами логических функций: функций переходов Qt+1 = f(Xt, Qt) и функций выходов Yt = ϕ(Xt, Qt), где индекс t соответствует текущему такту цикла, а (t+1) – последующему. Однако на практике часто оказывается достаточной более простая модель, в которой отсутствует связь КЦУ2 с входом ПЦУ. В этом случае функции выходов упрощаются: Yt = ϕ(Qt). Такая модель называется конечным автоматом Мура.

Функции переходов определяют состояние ПЦУ в следующем такте и позволяют синтезировать КЦУ1. Задаются они в виде таблицы переходов (Табл. 6.1).

Таблица 6.1

Таблица переходов

| №  состояния | Вход | | | Состояние ЗЭ | | | Сигнал управления | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| хn | … | x1 | qk | … | q1 | ak | … | a1 |
| QH |  |  |  |  |  |  |  |  |  |
| **.**  **.**  **.** |  |  |  |  |  |  |  |  |  |

Заполняется она, начиная с начального состояния QН, соответствующего нулевому такту. В ее строках записываются соответствующие двоичные наборы. При этом значения управляющих сигналов выбираются из условия обеспечения следующего состояния ПЦУ.

47

Функции выходов определяют выходной двоичный набор в текущем такте и позволяют синтезировать КЦУ2. Задаются они в виде таблицы выходов (Табл. 6.2).

Таблица 6.2

Таблица выходов

| №  состояния | Состояние ЗЭ | | | Выход | | |
| --- | --- | --- | --- | --- | --- | --- |
| qk | … | q1 | ak | … | a1 |
| QH |  |  |  |  |  |  |
| **.**  **.**  **.** |  |  |  |  |  |  |

Обычно обе таблицы объединяются в одну, которая называется автоматной таблицей. Однако автоматная таблица не отражает динамику поведения ПЦУ. Поэтому для пояснения работы ПЦУ автоматную таблицу дополняют временными диаграммами.

Другой тип автомата – автомат Мура. Особенность автомата Мура в том, что в нем выходной сигнал зависит лишь от внутреннего состояния и не зависит от входного сигнала.

**Автомат МилиАвтомат Мура**

****����+1 = ��(����,����) *функции переходов*����+1 = ��(����,����)

���� = ��(����,����) *функции выходов*���� = ��(����)

Рис. 6.2 - Автоматы Мили и Мура

Таблица переходов:Таблица выходов:

48

| №  состо  яния | Вход | | | Сост ЗЭ | | | Сиг. упр. | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| хn | … | x1 | qk | … | q1 | ak | … | a1 |
| QH |  |  |  |  |  |  |  |  |  |
| **.**  **.**  **.** |  |  |  |  |  |  |  |  |  |

| №  состо  яния | Сост. ЗЭ | | | Выход | | |
| --- | --- | --- | --- | --- | --- | --- |
| qk | … | q1 | ym |  | y1 |
| QH |  |  |  |  |  |  |
| **.**  **.**  **.** |  |  |  |  |  |  |

Функции переходов и выходов позволяют легко установить тип ПЦУ – это автомат Мили или автомат Мура (рис. 6.2)

**6.2 Понятие и классификация триггеров**

Основными компонентами ПЦУ являются запоминающие элементы, которые реализуются специальными устройствами – триггерами. Триггер – это одноразрядный элемент памяти, предназначенный для хранения одного бита информации. Основной способ построения триггеров – использование обратных связей. Именно за счёт них обеспечивается возможность запоминания.

Любой триггер имеет два выхода – прямой и инверсный, но состояние триггера определяется сигналом на прямом выходе. Число входов в зависимости от типа триггера может составлять от двух до пяти.

По способу приема информации различают асинхронные и синхронные триггеры. Синхронные триггеры записывают бит информации только при наличии активного (разрешающего) сигнала на входе синхронизации. Пассивное значение синхросигнала определяет режим хранения триггера. Асинхронные триггеры не имеют входа синхронизации и записывают бит информации в момент его подачи на информационные входы.

Синхронные триггеры могут быть со статическим или динамическим управлением по входу синхронизации. При статическом управлении активным является уровень логического 0 (инверсный синхровход) или логической 1 (прямой синхровход). На условном графическом обозначении триггера инверсный синхровход показывается кружочком ,

49

а прямой – без кружочка . При динамическом управлении активн~~ы~~м х

является фронт или срезсинхроимпульса. На условном графическом обозначении триггера фронт показывается прямой косой линией , а

срез – обратной .

х

По функ~~ц~~иональным возможностям различают триггеры с х

раздельной установкой состояний 0 и 1 (RS-триггер), с приёмом информации по одному входу D (D-триггер), универсальный с информационными входами J и K (JK-триггер) и со счётным входом Т (Т триггер).

**6.3. Типовые триггеры**

**Асинхронные RS-триггеры** имеют два информационных входа, один из которых обозначается буквой S (Set – установка), а другой – буквой R (Reset – сброс).

Рис. 6.3 – Асинхронный RS-триггер с прямыми входами

На рис. 6.3 приведены реализация триггера на логических элементах (ЛЭ) ИЛИ-НЕ, его условное графическое обозначение при исполнении в виде интегральной схемы и таблица переходов. Как видно, активным значением информационного сигнала является 1. Поэтому RS-триггер, построенный на логических элементах ИЛИ-НЕ, называют RS-триггером с прямыми входами. Однако асинхронный RS-триггер может быть построен и на логических элементах И-НЕ. Такая реализация триггера, ее условное графическое обозначение при исполнении в виде интегральной схемы и таблица переходов приведены на рис. 6.4.

50

Рис. 6.4 – Асинхронный RS-триггер с инверсными входами

Здесь активным значением информационного сигнала является 0. Поэтому RS- триггер, построенный на логических элементах И-НЕ, называют RS-триггером с инверсными входами.

Из таблиц переходов рис. 6.3 и 6.4 видно, что, во-первых, активный сигнал на входе S приводит к установке триггера в единичное состояние, а на входе R – в нулевое. По этой причине у триггеров любого другого типа асинхронный вход принудительной установки в 1 обозначается буквой S, а в 0 – буквой R. Эти входы являются входами первого приоритета, то есть при активном сигнале на одном из этих входов триггер не реагирует на сигналы по другим входам. Во-вторых, существует запрещенная комбинация входных сигналов. Запрет следует понимать только в информационном смысле. То есть он имеет смысл лишь при использовании триггера в качестве запоминающего элемента. Действительно, активные сигналы одновременно на S и R входах триггера устанавливают его выходы в одинаковое единичное состояние, что приводит к неопределенности относительно записываемого бита информации. Однако такая комбинация входных сигналов вполне естественна при использовании триггера в качестве управляющего элемента устройства. Возникновение запрещенной комбинации не приводит к выходу триггера из строя.

В маркировке микросхем RS-триггеров используются буквы ТР.

51

Триггеры всех других типов выпускаются промышленностью как синхронные.

**D-триггеры** имеют один информационный вход D и вход синхронизации С. В интегральном исполнении D-триггеры выпускаются как со статическим, так и с динамическим управлением по входу С. В любом случае в маркировке микросхем D-триггеров используются буквы ТМ.

D-триггеры со статическим управлением не имеют входов принудительной установки, поскольку используются исключительно в качестве запоминающих элементов. Их условное графическое обозначение (для примера с единичным активным синхросигналом – прямым синхровходом), таблица переходов и временные диаграммы, где крестик означает безразличное состояние триггера, приведены на рис. 6.5.

Рис. 6.5 – D-триггер со статическим управлением

Как видно, пока сигнал на входе С сохраняет активное (в данном случае единичное) значение, непрерывно производится запись информации со входа D.

D-триггеры с динамическим управлением дополняются входами S и R принудительной установки начального состояния. Их условное графическое обозначение (для примера с активным фронтом синхросигнала), таблица переходов и временные диаграммы приведены на рис. 6.6.

52

Рис. 6.6 – D-триггер с динамическим управлением

Как видно, запись информации с входа D производится только по фронту синхросигнала (тактового импульса).

**JК -триггеры** имеют два информационных входа J и К, вход синхронизации С и входы R и S принудительной установки начального состояния. В интегральном исполнении выпускаются только с динамическим управлением по входу синхронизации. В маркировке микросхем JK-триггеров используются буквы ТВ.

Условное графическое обозначение (для примера с активным срезом синхросигнала), таблица переходов и временные диаграммы работы JК - триггера приведены на рис. 6.7

Рис.6.7 – JK-триггер

Счётным называется режим, при котором триггер с каждым активным синхросигналом (тактовым импульсом) переключается в противоположное состояние.

В зависимости от значения активного синхросигнала такт разделяется на предстартовый полутакт и пассивный полутакт. Для

53

триггера рис. 6.7 предстартовым полутактом является импульс такта, а пассивным полутактом – его пауза. В течение пассивного полутакта входы J и К триггера отключены от входных цепей, благодаря чему обеспечивается собственно режим хранения. В течение предстартового полутакта к входным цепям подключается только тот информационный вход, активный сигнал (для триггера рис. 6.7 – единица) на котором может изменить текущее состояние триггера. Так, применительно к триггеру рис. 6.7, в интервале времени t1 – t2 будет подключен вход J, а в интервале времени t3 – t4 будет подключен вход К. Если в течение предстартового полутакта на подключенном информационном входе хоть на мгновение появляется активный сигнал, по активному синхросигналу триггер переключится в соответствующее состояние. В противном случае текущее состояние триггера не изменится.

**Т-триггеры** имеют единственный вход Т, на который поступают тактовые импульсы. По этой причине и поскольку Т-триггеры работают только в режиме счета, Т-вход называют счетным входом триггера.

К Т-триггеру относится и TV-триггер – Т-триггер, дополненный входом управления V. Сигнал по входу V либо разрешает работу Т триггера, либо переводит его в режим хранения независимо от тактовых импульсов по входу Т.

Условное графическое обозначение и таблица переходов Т- и TV триггеров на примере последнего приведены на рис. 6.8.



Рис. 6.8 – TV-триггер

В интегральном исполнении ни Т-, ни TV-триггеры не выпускаются,

54

поскольку легко могут быть построены на основе D- или JK-триггеров (рис.6.9, рис. 6.10, рис. 6.11).

Реализация Т-триггера

На базе D-триггера На базе JK-триггера



Рис.6.9 – Реализация Т-триггера

Реализация ТV-триггера

На базе D-триггера На базе JK-триггера



Рис. 6.10 – Реализация ТV-триггера



Рис. 6.11 – Реализация D-триггера на базе JK-триггера

Пусть D-триггер в состоянии 1. По правилу работы Т-триггера по фронту D-триггер должен переключиться в 0. Для этого 0 должен быть на D-входе. Этот 0 может быть взят с инверсного выхода D-триггера.

Пусть теперь D-триггер в состоянии 0. По правилу работы Т триггера по фронту D-триггер должен переключиться в 1. Для этого 1 должна быть на D-входе. Эту 1 опять-таки можно взять с инверсного выхода D-триггера.

55

Пусть JK-триггер в состоянии 1. По правилу работы Т-триггера по срезу JK-триггер должен переключиться в 0. Для этого 0 должен быть на J входе (J-вход пассивен) и 1 на К-входе (К-вход активен).

Пусть теперь JK-триггер в состоянии 0. По правилу работы Т триггера по срезу JK-триггер должен переключиться в 1. Для этого 1 должна быть на J-входе (J-вход активен) и 0 на К-входе (К-вход пассивен).

Как видно, в любом случае К-вход должен быть инверсным J-входу. **6.4 Синтез ПЦУ**

Синтез ПЦУ проводится в пять этапов.

**На 1 этапе** определяется минимальное число состояний, необходимое для построения устойчивого ПЦУ, соответствующего заданным условиям работы. Когда решение этой задачи не очевидно, следует воспользоваться графом переключений.

Графом переключений называется ориентированный граф, состоящий из узлов и направленных связей между ними (рис. 6.12). 

Рис. 6.12 - Граф переключений

Каждый узел отображает состояние ПЦУ в данном такте цикла. Внутри узла в десятичной или двоичной системе счисления записывается номер его состояния. Около каждого узла указывается десятичный номер выходного двоичного набора или непосредственно двоичный набор, соответствующий данному такту. Связи между узлами отображают переходы ПЦУ из одного состояния в другое. Над каждой связью выписываются входные двоичные сигналы, вызывающие переход ПЦУ из текущего i-го состояния в последующее (i+1)-е состояние.

В терминах графа переключений можно сформулировать условие

56

устойчивости: ПЦУ устойчиво, если его граф переключений не содержит ни одного замкнутого контура, дуги которого помечены одним и тем же значением входного сигнала.

Существует два способа устранения неустойчивости ПЦУ: переход к динамическому способу управления триггерами и увеличение числа состояний. Реализация первого способа тривиальна. Второй способ более сложный, но иногда единственно возможный: для устранения одного замкнутого контура графа переключений требуется дополнительно ввести не менее двух состояний (одного триггера).

**На 2 этапе** определяется количество триггеров и выбирается их тип. Количество триггеров ПЦУ определяется выражением: NT = ⎤ Log2NC ⎡, где NC – число состояний автомата, включая начальное.

Тип триггеров выбирается по принципу наибольшего совпадения их правил работы с правилами работы проектируемого устройства. **На 3 этапе** определяются функции переходов и проводится синтез КЦУ1.

Функции переходов представляются таблицей переходов, где управляющие сигналы записываются на основе правил работы выбранных триггеров, состояния запоминающих элементов – в соответствии с принятым порядком их следования, а входные двоичные наборы – в соответствии с правилами работы проектируемого устройства. Полученная таблица рассматривается как таблица истинности КЦУ1, что позволяет воспользоваться стандартной методикой его синтеза.

**На 4 этапе** определяются функции выходов и проводится синтез КЦУ2.

Функции выходов представляются таблицей выходов, где управляющие сигналы записываются, как и в таблице переходов, а выходные двоичные наборы – в соответствии с правилами работы проектируемого устройства. Построение таблицы выходов основывается

57

на графе переключений. Полученная таблица рассматривается как таблица истинности КЦУ2, что позволяет воспользоваться стандартной методикой его синтеза.

**На 5 этапе** строится структурная схема ПЦУ. При этом входы синхронизации триггеров соединяются параллельно (объединяются) на общий вход синхронизации ПЦУ, называемый тактовым входом устройства. В случае обнаружения в структурной схеме типовых структур производится коррекция схемы путем использования условных графических обозначений этих структур.

Наиболее часто используемые ПЦУ – счетчики и регистры, относятся к категории типовых устройств.

**Краткие итоги:**

Изучили основные принципы работы ПЦУ. Рассмотрели понятие триггера. Выяснили, как классифицируются триггеры. Усвоили различия в работе триггеров, их маркировку. Освоили синтез ПЦУ.

**Вопросы для самопроверки**

1. Дайте определение ПЦУ.

2. Как синхронизируются ПЦУ?

3. Дайте понятие триггера.

4. Какова классификация триггеров?

5. Дайте определение синхронного и асинхронного триггера. 6. Каков принцип работы синхронного и асинхронного триггера? 7. Перечислите типовые триггеры.

8. Каков принцип работы D-триггера?

9. Каков принцип работы JK – триггера?

10.Каков принцип работы T – триггера?

11.Расскажите поэтапно синтез ПЦУ.

58

**Лекция 7**

**Принципы управления микропроцессора**

В лекции рассматриваются принципы управления микропроцессором. Классификация микропроцессоров, их работа.

**Цель лекции:** Ознакомиться с основными принципами построения микропроцессорных систем. Изучить классификацию микропроцессоров, принципы аппаратного управления (жесткой логики и гибкой логики).

В основе построения микропроцессорных систем (МПС) обработки информации лежит модульный принцип.

**Модулем микропроцессорной системы** является её функциональный блок, выполненный в виде конструктивно законченного устройства – обычно в виде одной или нескольких БИС либо в виде плат.

**Модули соединяются** между собой посредством специальных устройств, называемых интерфейсами.

**Основными модулями** МПС являются однокристальный микропроцессор (МП), постоянная и оперативная (основная) памяти, устройства ввода-вывода информации и блоки управления (контроллеры).

**Суть проектирования** микропроцессорных систем на основе выбранного однокристального микропроцессора состоит в следующем: 1)выбор внешних устройств, предназначенных для связи МП с устройствами ввода-вывода информации;

2)организация связи этих устройств с микропроцессором; 3)выбор ёмкости памяти для размещения программы и промежуточных данных, а также способа её организации; 4)программирование МП на выполнение требуемых функций при известной конфигурации МПС с использованием системы команд МП.

59

**7.1 Классификация микропроцессоров.**

**Микропроцессором** называется функционально законченное программно-управляемое устройство обработки цифровой информации. Числовая информация, заданная для обработки, называется **данными.**

Физически **микропроцессоры** (МП) выпускаются в виде одной или нескольких **больших или сверхбольших интегральных схем** (БИС или СБИС), содержащих на одном кристалле сотни тысяч элементарных транзисторов.

МП дополняются БИС для хранения, ввода-вывода данных, управления и синхронизации, сопряжения интерфейсов и т.п. Эти БИС совместимы с МП по архитектуре, конструктивному исполнению и параметрам. Такие наборы БИС называются **микропроцессорными комплектами** (МПК).

МП можно классифицировать по весьма большому числу характеристик. Мы воспользуемся одним из возможных вариантов классификации.

**1.*По назначению*** различают универсальные и специализированные микропроцессоры.

***Универсальные МП*** могут осуществить преобразование информации в соответствии с любым заданным алгоритмом. ***Специализированные МП*** предназначены для решения определённого класса задач и даже, может быть, для решения одной конкретной задачи. Их особенностью являются низкая стоимость, малая потребляемая мощность, компактность, простота управления. Среди специализированных МП выделяют ***микроконтроллеры***, используемые в управлении технологическими процессами, измерениях, научных исследованиях. Их отличительной чертой является работа в реальном масштабе времени.

60

2.**По числу БИС** различают секционные (разрядно-модульные) и однокристальные МП.

***Микропроцессорная секция (модуль)*** представляет собой БИС для обработки небольшого числа разрядов данных (4 или 8). *Такие МП* обладают возможностью наращивания разрядности, что позволяет на их основе строить МП произвольной разрядности. *Кроме того, секционные МП* не имеют фиксированного набора команд, а программируются на микрокомандном уровне. Это позволяет реализовать оптимальный для данной задачи набор команд и отдельных процедур.

***Однокристальные МП*** реализуются в виде одной БИС или СБИС. *Такие МП* имеют фиксированный набор команд и фиксированную разрядность. Благодаря успехам современных технологий в повышении степени интеграции и быстродействия в настоящее время в микропроцессорной технике основное место занимают однокристальные МП (процессоры).

3. **По способу управления** различают МП с микропрограммным и с "жёстким" (аппаратным) управлением.

*Микропрограммное управление* характерно для секционных МП, а *"жёсткое" –* для однокристальных.

4.**По возможности прерываний выполняемой программы** МП могут не иметь, а могут иметь одно- или многоуровневую систему прерывания.

В многоуровневых системах прерывания разрешается прерывание прерывания. Такие системы используются в МП, работающих в реальном масштабе времени.

61

**7.2. Декомпозиция МП.**

Для облегчения понимания принципа работы микропроцессора, представим его в виде некоторого ПЦУ, состоящего из двух устройств: управляющего и операционного (рис. 7.1).



Рис.7.1 – Программное цифровое устройство

**Операционное устройство** производит прием операндов (чисел участвующих в операции), их хранение и преобразование. Кроме того операционный блок выдает во внешнюю среду результат преобразования, а в управляющее устройство – сигналы признаков.

**Сигналы признаков (флаги)** несут информацию об особенностях операндов и их отдельных разрядов, а также особенностях промежуточных и конечных результатов обработки (например, равенство нулю, чётность результата, переполнение разрядной сетки и др.).

**Управляющее устройство**, в зависимости от кода операции и сигналов признаков вырабатывает сигналы управления, обеспечивающие выполнение заданной операции.

Процесс функционирования МП во времени состоит из последовательности элементарных операций. Например, передача кодового слова от одного узла МП к другому, поразрядное инвертирование слова, сдвиг и др.

**Элементарная операция** выполняется за один такт синхросигнала и называется ***микрооперацией***.

62

В течение такта может быть выполнено несколько микроопераций, но только если результат выполнения каждой из них не зависит от результатов выполнения остальных.

Элементарные преобразования информации производятся операционным блоком под воздействием управляющих сигналов. **Совокупность управляющих сигналов**, обеспечивающих выполнение микроопераций в течение одного тактового интервала, называется ***микрокомандой*.**

Последовательность микрокоманд**, обеспечивающая выполнение данной операции (команды), называется** *микропрограммой***.** Управляющее устройство может быть задано как автомат Мили или как автомат Мура, для которых функции переходов и выходов определяются заданной микропрограммой. В связи с этим управляющее устройство часто именуется ***микропрограммным*** или ***управляющим автоматом***.

Для построения управляющего автомата используются принципы "жёсткой" (схемной) и "гибкой" (программируемой) логики.

**7.3 Принцип аппаратного управления ("жёсткой" логики).** Типовой управляющий автомат с "жёсткой" логикой имеет следующую структуру (рис.7.2).



Рис.7.2 – Типовой управляющий автомат с «жесткой» логикой

63

В управляющем устройстве предусматривается ряд управляющих узлов (УУз), представляющих собой набор логических схем Команда, поступающая из внешнего ОЗУ фиксируется в регистре команд (РгК) и с помощью дешифратора команд (ДШ) включает соответствующий управляющий узел.

Управляющий узел вырабатывает определённую последовательность сигналов управления (СУ), обеспечивая выполнение данной операции операционным блоком.

В общем случае значения сигналов управления зависят от сигналов признаков, отражающих ход вычислительного процесса.

При таком способе построения управляющего автомата **микропрограммы операций заложены в однажды выполненные соединения между логическими схемами управляющих узлов.** *Это означает***, что набор команд или, иначе говоря, система команд фиксируется и соответствует числу выполняемых операций.**

Поэтому такие микропроцессоры называются с "жёсткой" логикой управления.

Невозможность изменения системы команд после изготовления МП приводит к его узкой специализации. Вместе с тем МП с "жёсткой" логикой управления обеспечивает наивысшее быстродействие при заданной технологии изготовления.

**7.4. Принцип микропрограммного управления (гибкой логики).** В управляющем автомате с "гибкой" логикой предусматриваются управляющая память (УП) и блок микропрограммного управления (БМУ) (рис. 7.3).

64



Рис.7.3 – Управляющий автомат с «гибкой» логикой

**В УП для каждой операции содержится своя МКП.**

УП может быть постоянной или с произвольным обращением, т.е. допускающая как считывание, так и запись. В последнем случае загрузка УП производится пользователем.

**Команда, поступающая из внешней памяти**, используется БМУ для определения адреса первой МК той МКП, которая реализует заданную операцию.

Далее микрокоманды найденной МКП последовательно считываются из УП. При этом ***адрес следующей МК определяется БМУ на основе предыдущей МК***.

***Для обеспечения такого процесса управления*** в МК предусматриваются три поля (три группы разрядов): поле адреса, поле условных переходов (УсП) и поле сигналов управления (СУ). Два первых поля образуют адресную часть МК, а последнее поле – её операционную часть (рис.7.4)



Рис.7.4 – Поля МК

65

***В поле адреса*** содержится адрес очередной МК*.*

***Поле УсП*** предусматривается для реализации условных и безусловных переходов.

*Один из разрядов этого поля* отводится для указания вида перехода (например, 0 – безусловный переход, 1 – условный переход). *Ещё один разряд* определяет участие данного вида перехода в определении адреса (например, 1 – участвует, 0 – не участвует). *Остальные разряды* используются для указания условий, на которые следует ориентироваться при определении адреса очередной МК. В результате в зависимости от условия образуются два различных адреса и очередная МК, считывается из одной либо из другой ячейки УП. Рассмотренный способ управления получил название микропрограммного, а МП с управляющим автоматом на этом принципе называются МП с программируемой логикой.

**Достоинством такой организации управления** является возможность гибкого изменения системы команд МП с помощью изменения совокупности МКП, реализующих эти команды. Отсюда второе название принципа – принцип "гибкой" логики.

**Вместе с тем использование принципа "гибкой" логики** может привести к снижению быстродействия из-за увеличения числа тактов реализации микропрограммы.

Микропрограммное управление используется не только в секционных МП, но и в устройствах управления периферийным оборудованием МПС, а также как средство для аппаратной реализации фрагментов операционных систем, трансляторов и т.д.

**Управляющие автоматы с "гибкой" логикой различаются** по способу формирования сигналов управления.

66

**7.5. Способы формирования сигналов управления в**

**управляющих автоматах с "гибкой" логикой.**

Возможно горизонтальное, вертикальное и смешанное микропрограммирование.

***При горизонтальном микропрограммировании*** каждому разряду операционной части МК ставится в соответствие определённый управляющий сигнал, т.е. определённая микрооперация.

Так, если в i-ом разряде стоит 1, то соответствующая микрооперация выполняется независимо от значения других разрядов.

При таком способе операционная часть МК содержит m разрядов, где m – общее число микроопераций.

***Достоинствами горизонтального микропрограммирования*** являются

возможность одновременного выполнения в одном такте любого набора микроопераций и простота формирования сигналов управления. ***Однако при этом*** требуется большая длина МК, поскольку число управляющих сигналов может достигать нескольких сотен. Поэтому большее распространение получили другие методы.

**Краткие итоги:**

Изучили принципы работы МП. Их проектирование, классификацию и декомпозицию. Уяснили что такое модуль микропроцессорной системы. Описали принцип аппаратного управления ("жёсткой" логики) и принцип микропрограммного управления (гибкой логики), его достоинства и недостатки.

67

**Контрольные вопросы**

1. Что такое модуль микропроцессорной системы? 2. В чем состоит суть проектирования микропроцессорных систем?

3. Классификация микропроцессоров по назначению. 4. Классификация микропроцессоров по числу БИС 5. Классификация микропроцессоров по способу управления 6. Классификация микропроцессоров по возможности прерываний выполняемой программы

7. Каков принцип декомпозиции микропроцессора. 8. Опишите принцип аппаратного управления ("жёсткой" логики).

9. Опишите принцип микропрограммного управления (гибкой логики), его достоинства и недостатки.

10. Расскажите о способах формирования сигналов управления в управляющих автоматах с "гибкой" логикой, достоинства и недостатки.

68

**Лекция 8**

**Элементы архитектуры ЦСП TMS320C6x**

**Цель лекции: Дать о**пределение данных. Изучить преобразование чисел из одной системы исчисления в другую. Научиться производить арифметические операции с числами в разных системах исчисления. Уяснить, как работают методы адресации операндов.

Компания Texas Instruments (TI) вывела на рынок процессоры с архитектурой VLIW (высоко параллельная архитектура с очень длинным – 32-разрядным, командным словом), предложив для нее термин VelosiTI. 5 семейства (платформ) ЦСП: от TMS320C2х до TMS320C6х.

**8.1. Данные**

**16-ричная система счисления** – используется для компактной записи чисел в программах**.**

*Алфавит*: 10 арабских цифр от 0 до 9 и шесть латинских букв: A~10, B~11, C~ 12, D~13, E~14 и F~15.

*Преобразование* десятичных чисел в 16-ричные и обратно: 

*Переход* от 2-х чисел к 16-ричным: *Переход* от 16-ричных чисел к двоичным:



**Представление целых чисел.**

Числа бывают без знаковые и знаковые.

В n-разрядной двоичной сетке для модуля без знаковых чисел отводятся все n разрядов. При этом диапазон чисел составляет от 0 до 2n-1.

69

В случае знаковых чисел старший разряд отводится под знак (1 – «-», 0 – «+»), а остальные разряды – под модуль. Т.о., диапазон чисел составляет от +(2n-1 – 1) до -(2n-1- 1).

В процессоре знаковые числа представляются в дополнительном коде.

*Дополнительный код* положительного числа, есть само число. *Дополнительный код* отрицательного числа образуется по правилу: символ младшего разряда вычитается из числа, равного основанию системы счисления, а символы остальных разрядов – из числа, на 1 меньшего основания системы счисления.

Примеры при n = 8: -510 = -0000 01012 в прямом двоичном коде; -510 = 1111 10112 в дополнительном коде. -7Е16 = 8216 в дополнительном коде.

**Арифметические операции над целыми числами.**

1. ***Сложение***: 

а) производится поразрядно, начиная с младших разрядов; б) если сумма Si чисел i-го разряда превышает или равна ОСС, то в этот разряд результата записывается разность (Si – k⋅ОСС), а в следующий, более старший разряд (включая и знаковый), переносится k в виде дополнительного слагаемого. Здесь k – целая часть от деления Si на ОСС. Например, в случае двоичной системы счисления и Si = 5, k = [5/2] = 2. Следовательно, в i-й разряд результата запишется 5-k⋅2 = 5-4 = 1, а в следующий разряд переносится (говорим «в уме») дополнительное слагаемое 2 (k = 2);

в) в случае знаковых чисел перенос из знакового разряда не производится(рис.8.1)

70

Рис.8.1 – Примеры

*Результат операции над знаковыми числами* представлен в прямом коде, если он положительный и в дополнительном коде, если отрицательный.

**2. *Вычитание*** заменяется сложением чисел, предварительно представленных в дополнительном коде.

Примеры при n = 4:

616-316 = 616+D16 = 316 (перенос из знакового разряда не производится); 316-616 = 316+А16 = D16 (дополнительный код числа -3);

-616-316 = A16+D16 = 1716 (переполнение разрядной сетки).

**3. *Умножение*** для положительных чисел выполняется обычным образом, а для отрицательных – с промежуточным преобразованием в прямой код.

Примеры: а). Числа без знака (рис.8.2):

Рис.8.2 – Примеры

71

**8.2. Методы адресации операндов**

**Адресация** – обращение к операнду, указание на который содержится в команде.

**Адресный код** (**АК)** – это информация об адресе операнда, содержащаяся в команде.

**Исполнительный адрес** (**АИ)** – это номер физической ячейки памяти, к которой производится обращение.

**Первая группа адресаций** устанавливает АИ по значению АК. *Непосредственная адресация* – операнд указывается в команде константой. Эта адресация используется для ввода исходных данных и при работе с различного рода константами.

*Прямая адресация* – АИ совпадает с АК.

*Регистровая адресация* – в команде указывается имя регистра РОН процессора, в котором хранится операнд.

*Косвенная адресация* – АК указывает имя регистра процессора, в котором находится АИ. Такой регистр называют *регистром адреса* (рис.8.3).



Рис.8.3 – Регистр адреса

*Автоинкрементная* (*автодекрементная*) адресация – в команде указывается имя регистра процессора, содержимое которого автоматически увеличивается (уменьшается) на 1.

**Вторая группа адресаций** вычисляет АИ по АК. При этом АИ определяется алгебраической суммой АК, называемого *базовым адресом* (*базой*), и некоторого числа, называемого смещением (рис.8.4).

72



Рис.8.4 – Вторая группа адресации

«База» задает центр области ячеек памяти, а «смещение» – величину смещения относительно этого центра.

«База» содержится в регистре РОН, называемом регистром адреса или также базой.

2 метода – базирование и индексация. При базировании база не меняется, а при индексации – меняется и становится равной АИ. **Краткие итоги**

Рассмотрены понятия системы исчисления, перевод из одной в другую, арифметические действия. Изучены методы адресации операндов. **Контрольные вопросы**

1. Как представляются целые числа?

2. Как производится сложение?

3. Как делается умножение?

4. Как делается вычитание?

5. Дайте определение адресации, адресному коду и исполнительному адресу.

6. Что устанавливает первая группа адресации?

7. Для чего нужна вторая группа адресации?

73

**Лекция 9**

**Структура ЦСП TMS320C6x**

**Цель лекции:** Ознакомиться со структурой ЦСП TMS320C6x. Понять принцип работы регистровых пар. Процесс обработки команд.

Процессоры платформы ‘С6х сделаны по новой архитектуре VelociTI – высоко параллельная и детерминированная архитектура с очень длинным командным словом. Восемь независимых модулей (функциональных устройств) позволяют параллельно (одновременно) выполнять до восьми команд. Все команды содержат условия их выполнения, что позволяет сократить расходы производительности процессора на выполнение переходов и увеличить степень параллелизма обработки. Характерным для процессоров этой платформы является наличие аппаратного умножителя, позволяющего выполнять умножение двух чисел за один такт.

Три составные части: ядро (ЦПУ), две области памяти – память данных и память команд, внутренняя периферия (рис.9.1). Рис.9.1 – Структура ЦСП TMS320C6x

74

Контроллер прямого доступа в память (ПДП) занимается передачей данных между областями памяти без участия ЦПУ. Кроме того, он используется для начальной загрузки программы в память процессора.

Интерфейс внешней памяти (ИВП) предназначен для обмена данными с внешней памятью и быстродействующими внешними устройствами.

Два 32-разрядных таймера используются для задания временных событий, реализации счетчиков, генерации импульсов, прерывания процессора. Процесс прерывания заключается в прерывании выполнения текущей программы и переходе к выполнению некоторой другой программы. По завершении выполнения этой другой программы процессор возвращается к прерванной программе и продолжает ее выполнение.

Логика снижения энергопотребления (ЛСЭ) включает один из трех возможных «спящих» режимов процессора. В первом режиме тактовые импульсы снимаются только с ядра процессора, во втором – и с периферии, размещенной на кристалле, а в третьем режиме тактовая частота снимается практически со всего кристалла.

ЦПУ включает в себя 8 независимых функциональных устройств (модулей) – два умножителя (устройства .М) и 6 арифметико-логических устройств (устройства .L, .S и .D). Арифметико-логическое устройство (АЛУ) – это КЦУ, реализующее арифметические (кроме умножения и деления) и логические операции.

Каждая четверка модулей связана с набором регистров общего назначения (РОН), создавая тем самым разделение ядра на сторону А и сторону В. РОН программно доступны, т. е. с помощью соответствующих команд программист может управлять их содержимым. Следовательно, они могут быть использованы для самых различных целей – поддержки различных типов адресации памяти данных, хранения промежуточных

75

результатов вычислений и в качестве источников операндов. РОН каждой стороны представляет собой 16 32-разрядных регистров, связанных с памятью данных двумя шинами – шиной загружаемых из внутренней памяти данных и шиной загружаемых в эту память данных.

Два АЛУ – .D1 и .D2, используются только для формирования исполнительного адреса ячейки памяти данных. При этом шины адреса, управляемые D-устройствами, позволяют использовать адрес, сформированный в РОН одной стороны, для операций с данными в РОН другой стороны. Данные из функциональных устройств сначала помещаются в РОН, а затем по адресам, формируемым D-устройствами, идет обмен с памятью данных. При этом возможна одновременная выборка из памяти данных до 64 разрядов по двум подаваемым адресам.

Один из модулей .L, .S или .M каждой стороны через соответствующую перекрестную шину (2х или 1х) имеет доступ к РОН противоположной стороны, причем только как к источникам операнда.

В одном и том же такте к РОН своей стороны возможен доступ всех модулей этой стороны одновременно. Однако к РОН противоположной стороны может иметь только один модуль данной стороны.

Поскольку модули независимы, в каждом такте процессор может выдавать им до восьми 32-разрядных команд, которые могут выполняться параллельно (одновременно), последовательно или параллельно последовательно. Блок команд, которые выполняются параллельно (в одном такте), называется выполняемым пакетом. Блок команд, содержащий до 8 выполняемых пакетов, называется пакетом выборки. Таким образом, пакет выборки состоит из восьми выполняемых пакетов, если 8 команд выполняются последовательно, из одного выполняемого пакета, если 8 команд выполняются параллельно, и из двух – семи выполняемых пакетов, если 8 команд выполняются параллельно

последовательно.

76

Процесс обработки команд в ядре начинается загрузкой из программной памяти пакета выборки. Диспетчер команд распределяет каждую из 32-разрядных команд пакета на свой модуль для исполнения. Очередной выполняемый пакет размещается для исполнения в функциональных устройствах в каждом такте. Пакет же выборки из программной памяти не загружается до окончания выполнения текущего пакета выборки.

ПДП – передача данных между областями памяти и начальная загрузка программы в программную память.

ИВП – обмен данными между памятью данных и внешней памятью, внешней памятью и внешней периферией, памятью данных и внешней периферией.

Таймеры – задание временных событий, генерация импульсов, реализация счетчиков, прерывание процессора.

ЛСЭ – включает «спящие» режимы. Режим 1 – тактирование снимается только с ядра, режим 2 – тактирование снимается с ядра и периферии, режим 3 – тактирование снимается практически со всего кристалла.

Имена регистров РОН: А0, …, А15, В0, …, В15. Регистровые пары представлены в таблице 9.1.

Таблица 9.1

Регистровые пары

| Сторона А | | Сторона В | |
| --- | --- | --- | --- |
| А1:А0 | А9:А8 | В1:В0 | В9:В8 |
| А3:А2 | А11:А10 | В3:В2 | В11:В10 |
| А5:А4 | А13:А12 | В5:В4 | В13:В12 |
| А7:А6 | А15:А14 | В7:В6 | В15:В14 |

Ограничения ресурсов:

77

∙только один модуль (.L, .S или .М) и только один операнд можно взять с противоположной стороны РОН;

∙результат выполнения операции модулем всегда размещается в регистре РОН своей стороны;

∙в одном и том же такте к РОН одной стороны возможен доступ одновременно всех модулей этой стороны;

∙для обмена данными между РОН одной стороны и памятью данных можно использовать модуль .D противоположной стороны. Процесс обработки команд в ядре включает в себя:

∙пакет выборки загружается в программную память;

∙диспетчер команд формирует последовательность выполняемых пакетов;

∙для каждой команды выполняемого пакета декодер выделяет код операции и передает его на соответствующий модуль.

**Краткие итоги**

В лекции рассмотрена структура ЦСП TMS320C6x. Даны определения ПДП, ЛЭС, ИВП. Дано объяснение, что такое регистровые пары. Описан процесс обработки команд в ядре.

**Контрольные вопросы**

1. Перечислите три составных части структуры ЦСП TMS320C6x 2. Опишите ядро ЦПУ

3. Дайте определение ПДП,ИВТ и ЛЭС.

4. Перечислите имена Регистров РОН по парно.

5. Какие ограничения ресурсов существует?

6. Что включает в себя обработки команды в ядре?

78

**Лекция 10**

**Структура командной строки ассемблера, вопросы особенности адресации и команд.**

**Цель лекции:** Ознакомиться со структурой командной строки. Уяснить ее оформление, функциональные свойства. Понять, как отделяются поля, какие существуют запреты.

Программа на языке ассемблера должна быть оформлена в виде текстового файла. Любая командная строка ассемблерной программы может включать вплоть до семи полей в следующей последовательности: метка, параллельные полосы, условие, команда, функциональное устройство, операнды и комментарий. Поля отделяются друг от друга одним или более пробелами. Внутри поля пробелы запрещены (рис.10.1):

Рис.10.1 – Ассемблерные предложения

*Пробелы внутри каждого из 6 первых полей запрещены.*

**Поле метки**

правильная запись **L…: \_G**...**:**

неправильная запись **1**…**: \_1**…**: \_:**

**Поле параллельных линий** ||

Команда А

| Такт  (выполняемый пакет) | Команды | | |
| --- | --- | --- | --- |
| 1 | A |  |  |
| 2 | B |  |  |
| 3 | C | D | E |

Команда В

Команда С переход || Команда D ← || Команда E

79

**Поле условия**

Регистры использующиеся в поле условия: A1, A2, B0, B1 и B2 Принцип выполнения условных команд:

|  | Условие Команда выполняется, если |
| --- | --- |
| [A1] | содержимое А1 не нулевое |
| [!A1] | содержимое А1 равно нулю |

**Поле команды**

Мнемоника директивы или команды.

Пример директивы ассемблера: .set (на машинный язык не переводятся).

Пример команды ассемблера: add (переводятся на машинный язык). **Поле модуля**

Имя конкретного функционального устройства, например **.D1** или **.L1Х,** если один операнд берется с противоположной стороны. Функциональный тип устройства, например **.М**.

**Поле операндов**

Имена регистров РОН, содержащие данные.

Адресация операндов.

Данные в числовой форме:

1. *Константы*:

−двоичное целое, например 01**b**, 0100**В**;

−восьмеричное целое, например 236**q**, 10**Q**;

−десятичное целое:

знаковые числа от -2 147 483 648 до 2 147 483 647,

числа без знака от 0 до 4 294 967 295;

−шестнадцатеричное целое, например 9АH, 5Dh.

Требование: число должно начинаться с цифры. Так, при вводе А5h следует указывать как **0**А5h;

80