



Федеральное агентство по рыболовству
БГАРФ ФГБОУ ВО «КГТУ»
Калининградский морской рыбопромышленный колледж

Утверждаю
Заместитель начальника колледжа
по учебно-методической работе
М.С. Агеева

Учебно-методические указания по организации самостоятельной работы по
дисциплине

ОП.04 ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

по дисциплине

**11.02.03 Эксплуатация оборудования радиосвязи и электрорадионавигации
судов**

МО-11 02 03-ОП.04.СР

РАЗРАБОТЧИК
ЗАВЕДУЮЩИЙ ОТДЕЛЕНИЕМ
ГОД РАЗРАБОТКИ

Н.И. Прийма
Д.В. Холоденин
2023

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.2/72

Содержание

Введение.....	3
ПЕРЕЧЕНЬ САМОСТОЯТЕЛЬНЫХ РАБОТ.....	5
САМОСТОЯТЕЛЬНАЯ РАБОТА №1 ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ	6
САМОСТОЯТЕЛЬНАЯ РАБОТА №2. ВЫПОЛНЕНИЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ.....	11
НАД ДВОИЧНЫМИ ЧИСЛАМИ.....	11
САМОСТОЯТЕЛЬНАЯ РАБОТА №3 МИНИМИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ	22
САМОСТОЯТЕЛЬНАЯ РАБОТА 4 СИНТЕЗ КОМБИНАЦИОННЫХ ЛОГИЧЕСКИХ УСТРОЙСТВ.....	26
САМОСТОЯТЕЛЬНАЯ РАБОТА №5: ПРОРАБОТКА КОНСПЕКТОВ ЗАНЯТИЙ ПО ТЕМЕ 1.2. ПОДГОТОВКА К ЗАЩИТЕ ЛР И ПР.....	32
САМОСТОЯТЕЛЬНАЯ РАБОТА №6: ПРОРАБОТКА КОНСПЕКТОВ ЗАНЯТИЙ ПО ТЕМЕ 2.1. ПОДГОТОВКА К ЗАЩИТЕ ЛР И ПР.....	33
САМОСТОЯТЕЛЬНАЯ РАБОТА №8. ДЕСЯТИЧНЫЕ СУММАТОРЫ.....	41
САМОСТОЯТЕЛЬНАЯ РАБОТА №9: ПРОРАБОТКА КОНСПЕКТОВ ЗАНЯТИЙ ПО ТЕМЕ 2.2. ПОДГОТОВКА К ЗАЩИТЕ ЛР И ПР.....	46
САМОСТОЯТЕЛЬНАЯ РАБОТА №10. ТИПОВЫЕ УЗЛЫ ЦИФРОВЫХ УСТРОЙСТВ, ВЫПОЛНЕННЫЕ НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ УСТРОЙСТВАХ С МАТРИЧНОЙ СТРУКТУРОЙ.....	47
САМОСТОЯТЕЛЬНАЯ РАБОТА №11: ПРОРАБОТКА КОНСПЕКТОВ ЗАНЯТИЙ ПО ТЕМЕ 2.450	
САМОСТОЯТЕЛЬНАЯ РАБОТА №12. ИЗУЧЕНИЕ НАБОРА МИКРОСХЕМ СИСТЕМНОЙ ПЛАТЫ КОМПЬЮТЕРА.....	53
САМОСТОЯТЕЛЬНАЯ РАБОТА №13. ПРОРАБОТКА КОНСПЕКТОВ ЗАНЯТИЙ ПОДГОТОВКА К ЗАЩИТЕ ПР ПО РАЗДЕЛУ 3.	71
ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ ЛИТЕРАТУРЫ	72

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.3/72

Введение

Методические указания по организации самостоятельной работы обучающихся составлены в соответствии с рабочей программой дисциплины Вычислительная техника по специальности 11.02.03 «Эксплуатация оборудования радиосвязи и электрорадионавигации судов».

Самостоятельная работа – это деятельность обучающихся в процессе обучения и во внеаудиторное время, выполняемая по заданию преподавателя, но без его непосредственного участия.

На самостоятельную внеаудиторную работу по дисциплине «Вычислительная техника» 46 академических часов в 4 семестре.

Цель внеаудиторной самостоятельной работы;

- *закрепить знания и умения по темам и разделам дисциплины;*
- *расширить знания по отдельным темам;*
- *формировать умения самостоятельного изучения элементов дисциплины, пользоваться дополнительной и учебной литературой, интернетом;*
- *развитие самостоятельности, организованности, ответственности;*
- *работать над формированием общих и профессиональных компетенций, необходимых для работы в данной специальности.*

Также освоение программы дисциплины предусматривает формирование компетенций

-профессиональные компетенции:

ПК 1.2. Нести радиовахту с использованием процедуры связи в подсистемах Глобальной морской системы связи при бедствии.

ПК 1.3. Вести вахтенный журнал радиостанции и оформлять техническую документацию радиооборудования.

ПК 1.4. Пользоваться программным обеспечением микропроцессоров радиооборудования и методами устранения сбоев программного обеспечения;

Внеаудиторная самостоятельная работа выполняется в отдельных тетрадах в виде *конспекта*.

Критериями оценки результатов самостоятельной работы являются:

- уровень усвоения учебного материала;

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.4/72

- умение использовать теоретические знания при выполнении практических задач;

- обоснованность и чёткость изложения ответа;

- оформление материала в соответствии с требованиями.

Итоговая оценка по дисциплине выставляется с учётом результатов выполнения самостоятельной внеаудиторной работы.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.5/72

Перечень самостоятельных работ

№ работы	Тема самостоятельной работы	Количество часов
Раздел 1. Арифметические и логические основы ЭВМ		
Тема 1.1. Арифметические основы ЭВМ		
1.	Самостоятельная работа № 1 Перевод чисел из одной системы счисления в другую.	4
2	Самостоятельная работа №2. Выполнение арифметических операций над двоичными числами	4
3	Самостоятельная работа 3. Минимизация логических функций.	4
4	Самостоятельная работа 4. Синтез комбинационных логических устройств.	4
5	Самостоятельная работа №5: Проработка конспектов занятий по теме 1.2. Подготовка к защите ЛР и ПР.	4
Раздел 2. Типовые узлы и устройства вычислительной техники		
6	Самостоятельная работа №6: Проработка конспектов занятий по теме 2.1. Подготовка к защите ЛР и ПР.	4
7	Самостоятельная работа 7. Построение схем делителей частоты с заданным коэффициентом деления Построение схем счётчиков и временных диаграмм, объясняющих их работу.	2
8	Самостоятельная работа 8. Десятичные сумматоры.	2
9	Проработка конспектов занятий по теме 2.2. Подготовка к защите ЛР и ПР.	4
10	Самостоятельная работа 10. Типовые узлы цифровых устройств, выполненные на программируемых логических устройствах с матричной структурой.	4
11	Проработка конспектов занятий по теме 2.4 Самостоятельная работа №11: Проработка конспектов занятий по теме 2.4	2
12	Самостоятельная работа 12. Изучение набора микросхем системной платы компьютера.	2
Раздел 3. Основы микропроцессорных систем		
13	Самостоятельная работа: №13 Проработка конспектов занятий Подготовка к защите ПР по разделу 3.	6
Итого		46

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.6/72

Раздел 1 Арифметические и логические основы ЭВМ
Тема 1.1 Арифметические основы ЭВМ
Самостоятельная работа №1 Перевод чисел из одной системы счисления в другую

Цель работы:

Закрепление темы 1.1. Арифметические основы ЭВМ

План работы:

1. Перевод чисел из одной системы счисления в другую
2. Правила перевода чисел из одной системы счисления в другую
3. Примеры перевода чисел из одной системы счисления в другую

Правила перевода чисел из одной системы счисления в другую

Перевод чисел из одной системы счисления в другую составляет важную часть машинной арифметики. Рассмотрим основные правила перевода.

1. Для перевода двоичного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 2, и вычислить по правилам десятичной арифметики:

$$X_2 = A_n \cdot 2^{n-1} + A_{n-1} \cdot 2^{n-2} + A_{n-2} \cdot 2^{n-3} + \dots + A_2 \cdot 2^1 + A_1 \cdot 2^0$$

При переводе удобно пользоваться таблицей степеней двойки:

Таблица 1. Степени числа 2

n (степень)	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

Пример. Число 11101000_2 перевести в десятичную систему счисления.

$$11101000_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 232_{10}$$

2. Для перевода восьмеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 8, и вычислить по правилам десятичной арифметики:

$$X_8 = A_n \cdot 8^{n-1} + A_{n-1} \cdot 8^{n-2} + A_{n-2} \cdot 8^{n-3} + \dots + A_2 \cdot 8^1 + A_1 \cdot 8^0$$

При переводе удобно пользоваться таблицей степеней восьмерки:

Таблица 2. Степени числа 8

n (степень)	0	1	2	3	4	5	6
-------------	---	---	---	---	---	---	---

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»						
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА						С.7/72

8^n	1	8	64	512	4096	32768	262144
-------	---	---	----	-----	------	-------	--------

Пример. Число 75013_8 перевести в десятичную систему счисления.

$$75013_8 = 7 \cdot 8^4 + 5 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 3 \cdot 8^0 = 31243_{10}$$

3. Для перевода шестнадцатеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 16, и вычислить по правилам десятичной арифметики:

$$X_{16} = A_n \cdot 16^{n-1} + A_{n-1} \cdot 16^{n-2} + A_{n-2} \cdot 16^{n-3} + \dots + A_2 \cdot 16^1 + A_1 \cdot 16^0$$

При переводе удобно пользоваться таблицей степеней числа 16:

Таблица 3. Степени числа 16

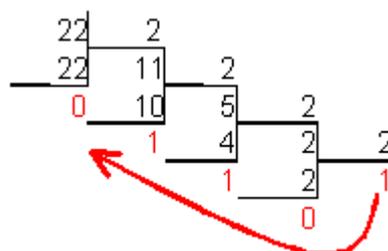
N (степень)	0	1	2	3	4	5	6
16^n	1	16	256	4096	65536	1048576	16777216

Пример: Число $FDA1_{16}$ перевести в десятичную систему счисления.

$$FDA1_{16} = 15 \cdot 16^3 + 13 \cdot 16^2 + 10 \cdot 16^1 + 1 \cdot 16^0 = 64929_{10}$$

4. Для перевода десятичного числа в двоичную систему его необходимо последовательно делить на 2 до тех пор, пока не останется остаток, меньший или равный 1. Число в двоичной системе записывается как последовательность последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 22_{10} перевести в двоичную систему счисления.



$$22_{10} = 10110_2$$

5. Для перевода десятичного числа в восьмеричную систему его необходимо последовательно делить на 8 до тех пор, пока не останется остаток, меньший или

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.9/72

Пример. Число 1011100011_2 перевести в шестнадцатеричную систему счисления.

$$0010\ 1110\ 0011_2 = 2E3_{16}$$

9. Для перевода восьмеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной триадой.

Пример. Число 531_8 перевести в двоичную систему счисления.

$$531_8 = 101011001_2$$

10. Для перевода шестнадцатеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной тетрадой.

Пример. Число $EE8_{16}$ перевести в двоичную систему счисления.

$$EE8_{16} = 111011101000_2$$

11. При переходе из восьмеричной системы счисления в шестнадцатеричную и обратно, необходим промежуточный перевод чисел в двоичную систему.

Пример 1. Число FEA_{16} перевести в восьмеричную систему счисления.

$$FEA_{16} = 111111101010_2$$

$$111\ 111\ 101\ 010_2 = 7752_8$$

Пример 2. Число 6635_8 перевести в шестнадцатеричную систему счисления.

$$6635_8 = 110110011101_2$$

$$1101\ 1001\ 1101_2 = D9D_{16}$$

ПЕРЕВОД ДРОБНЫХ ЧИСЕЛ

перевод дробных чисел рассмотрим на примере десятичного числа 6.125. Это дробное число в десятичной системе счисления представляется так:

$$6.125 = 6 * 10^0 + 1 * 10^{-1} + 2 * 10^{-2} + 5 * 10^{-3} = 6 * 1 + \frac{1}{10} + \frac{2}{100} + \frac{5}{1000}$$

Все просто, не так ли? Та же самая простота сохраняется и при записи дробного числа в любой другой системе счисления. Возьмем, например, двоичную систему и число, например, 110.001. Эта запись есть не что иное как

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.10/72

$110.001 = 1*2^2 + 1*2^1 + 0*2^0 + 0*2^{-1} + 0*2^{-2} + 1*2^{-3} = 1*4 + 1*2 + 0*1 + \frac{0}{2} + \frac{0}{4} + \frac{1}{8} = 6 + \frac{1}{8} = 6.125$
Т.о. число 110.001 в двоичной системе есть 6.125 в десятичной.

Есть только одно но - все-таки из-за того, что здесь участвуют дроби с разными знаменателями, не всегда одно и то же число можно одинаково точно выразить в разных системах счисления.

Возьмем, например, число $0.8 = 0 + \frac{8}{10}$. Отлично смотрится в десятичной системе счисления. Но вот если попробовать получить запись этого числа в двоичной системе счисления - будут проблемы. Попробуем, пока не устанем

$$0 + \frac{1}{2} + \frac{1}{4} + \frac{0}{8} + \frac{0}{16} + \frac{1}{32} + \frac{1}{64} + \dots = 0 + 0.5 + 0.25 + 0.03125 + 0.015625 + \dots = 0.796875 + \dots$$

Продолжать можно еще довольно долго, но уже сейчас видно, что 0.8 в десятичной системе это 0.11001100... (далее очень много цифр) в двоичной.

Поэтому перевод дробного числа из одной системы счисления в другую чаще всего дает погрешность. Погрешность эта зависит от того, сколько разрядов мы используем для записи дробной части переведенного числа. Возьмем пример с числом 0.8 и используем для записи его двоичного представления шесть разрядов после запятой - 0.110011. Полученное число вовсе не 0.8, а 0.796875, разница при этом составляет 0.003125. Это и есть наша погрешность перевода десятичного числа 0.8 в двоичный вид при использовании шести разрядов после запятой.

Вес крайнего правого разряда (самого младшего разряда) называется разрешением (resolution) или точностью (precision), и определяет наименьшее неравное нулю число, которое может быть представлено данным числом разрядов. Для нашего примера это $2^{-6} = 0.015625$. При этом максимально возможная погрешность представления числа, как нетрудно сообразить, не превышает половины этого веса, или 0.0078125. Так что для 0.8 мы имеем еще и не самую плохую погрешность.

Вопросы для самопроверки:

1. Правила перевода целых чисел из одной системы счисления в другую?
2. Правила перевода дробных чисел из одной системы счисления в другую?
3. Привести пример преобразования числа десятичной системы в двоичную.
4. Привести пример преобразования числа двоичной системы в десятичную.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.11/72

5. Привести пример преобразования числа двоичной системы в восьмиричную.

6. Привести пример преобразования числа десятичной системы в восьмиричную.

7. Привести пример преобразования числа двоичной системы в шестнадцатиричную, в восьмиричную.

8. Назовите примеры использования рассмотренных систем счисления.

Самостоятельная работа №2. Выполнение арифметических операций над двоичными числами

Цель работы:

Закрепление темы 1.1. Арифметические основы ЭВМ

План работы:

1. Сложение двоичных чисел.
2. Умножение двоичных чисел.
3. Деление двоичных чисел.
4. Поразрядное сложение двоичных чисел с переносом
5. Вычитание двоичных чисел (столбиком)
6. Арифметические операции над двоичными числами с плавающей точкой

Арифметические действия в двоичной системе производится по тем же правилам что и в десятичной системе счисления. Однако так как в двоичной системе счисления используются только две цифры 0 и 1, то арифметические действия выполняются проще, чем в десятичной системе.

Сложение двоичных чисел.

Сложение выполняется поразрядно столбиком, начиная с младшего разряда и используя таблицы двоичного сложения:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10.$$

При сложении необходимо помнить, что 1+1 дают ноль в данном разряде и единицу переноса в старший.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.12/72

Пример 3.5. Сложить два числа:

$$\begin{array}{r}
 + \quad 1011010 \\
 \quad 111001 \\
 \hline
 10010011
 \end{array}
 \quad
 \begin{array}{r}
 + \quad 90 \\
 \quad 57 \\
 \hline
 147.
 \end{array}$$

Вычитание двоичных чисел.

Вычитание выполняется поразрядно столбиком, начиная с младшего разряда и используя таблицы двоичного вычитания:

$$\begin{array}{l}
 0 - 0 = 0 \\
 1 - 0 = 1 \\
 1 - 1 = 0 \\
 10 - 1 = 1.
 \end{array}$$

Пример 3.6. Найти разность двух чисел:

$$\begin{array}{r}
 \quad 1011010 \\
 - \quad 111001 \\
 \hline
 100001
 \end{array}
 \quad
 \begin{array}{r}
 \quad 90 \\
 - \quad 57 \\
 \hline
 33.
 \end{array}$$

Т.е. при вычитании двоичных чисел в случае необходимости занимает 1 из старшего разряда, которая равна двум единицам младшего разряда.

Умножение двоичных чисел.

Умножение в двоичной системе производится по тому же принципу что и в десятичной системе счисления, при этом используется таблица двоичного умножения:

$$\begin{array}{l}
 0 * 0 = 0 \\
 0 * 1 = 0 \\
 1 * 0 = 0 \\
 1 * 1 = 1.
 \end{array}$$

Пример 3.7. Найти произведение двух чисел:

$$\begin{array}{r}
 \quad 10101 \\
 \times \quad 1001 \\
 \hline
 \quad 10101 \\
 + \quad 10101 \\
 \hline
 10111101 \\
 21 \times 9 = 189
 \end{array}
 \quad
 \begin{array}{r}
 \quad 1101 \\
 \times \quad 11 \\
 \hline
 \quad 1101 \\
 + \quad 1101 \\
 \hline
 100111 \\
 13 \times 3 = 39.
 \end{array}$$

Как видно из приведенных примеров, операция умножения может быть представлена как операции сдвига и суммирования.

Деление двоичных чисел.

Деление в двоичной системе производится вычитанием делителя со сдвигом вправо, если остаток больше нуля.

Пример. Найти частное двух чисел если:

1. Делимое больше делителя:

$$\begin{array}{r}
 110010 \overline{)1010} \\
 - 1010 \\
 \hline
 001010 \\
 - 1010 \\
 \hline
 0 \\
 50:10=5
 \end{array}$$

2. Делимое меньше делителя:

$$\begin{array}{r}
 11001 \overline{)101000} \\
 - 110010 \\
 \hline
 101000 \\
 - 101000 \\
 \hline
 000000
 \end{array}
 \qquad
 \begin{array}{r}
 25 \overline{)40} \\
 - 250 \\
 \hline
 100 \\
 - 80 \\
 \hline
 200 \\
 - 200 \\
 \hline
 000
 \end{array}$$

Как видно из приведенных примеров, операция деления может быть представлена как операции сравнения, сдвига и суммирования.

Поразрядное сложение двоичных чисел с переносом

В данном случае сложение производится как и обычные десятичные числа в столбик.

Если складываем 0 и 1 - получается 1

Если складываем 1 и 1 - получается 0 и единица переходит в старший разряд (влево) где складывается со следующим значением.

Примеры показаны для беззнаковых чисел, в которых старший разряд (крайний левый) не является показателем знака.

$$\begin{array}{r}
 0 \ 0 \ 1 \ 0 \quad (2_{10}) \\
 + 1 \ 0 \ 1 \ 1 \quad (11_{10}) \\
 \hline
 1 \ 1 \ 0 \ 1 \quad (13_{10})
 \end{array}$$

Переполнение

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.14/72

Здесь в качестве примера показано сложение двух двоичных чисел (в десятичном эквиваленте $15 + 1$) в результате которого происходит "переполнение" ячейки памяти в результате которого результат будет не таким, как ожидалось.

$$\begin{array}{r}
 1\ 1\ 1\ 1\ \quad (15_{10}) \\
 +\ 0\ 0\ 0\ 1\ \quad (1_{10}) \\
 \hline
 1\ 0\ 0\ 0\ 0\ \quad 0
 \end{array}$$

Другой пример переполнения памяти, когда результатом сложения двух чисел будет 3, а не 19.

Это происходит потому что в четырех битах максимальное число равно $15_{10} = 1111_2$.

Число 19 не помещается туда физически, как и в примере выше.

$$\begin{array}{r}
 1\ 1\ 1\ 0\ \quad (14_{10}) \\
 +\ \quad 1\ 0\ 1\ \quad (5_{10}) \\
 \hline
 1\ 0\ 0\ 1\ 1\ \quad (3_{10})
 \end{array}$$

2 Умножение «столбиком»

Показан просто пример исполнения - без учета знака числа.

Умножение производится по тому же принципу что и для обычных десятичных чисел.

Умножение на ноль дает единицу, а умножение единицы на единицу дает, естественно, тоже единицу.

После перемножения производится сложение для получения общего результата.

Пример умножения «столбиком» 1110×101 - в десятичном представлении - $14 \times 5 = 70$:

$$\begin{array}{r}
 \mathbf{64\ 32\ 16\ 8\ 4\ 2\ 1} \\
 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0 \\
 \times \quad \quad \quad 1\ 1\ 1\ 0 \\
 \quad \quad \quad \quad 1\ 0\ 1 \\
 + \quad \quad \quad 1\ 1\ 1\ 0 \\
 1\ 1\ 1\ 0 \\
 \hline
 1\ 0\ 0\ 0\ 1\ 1\ 0
 \end{array}$$

3 Умножение посредством сдвига (влево)

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.15/72

Сдвиг - простыми словами это перенос числа влево (в сторону старшего разряда). При этом в младший разряд записывается 0.

Пример:

имеем число 0011_2 (это у нас 3_{10})

0 0 1 1

Сдвигаем его влево и получаем такое число:

0 1 1 0

Это уже число 6_{10}

Что произошло при сдвиге?

Произошло обычное УМНОЖЕНИЕ на 2!

Было 3 - стало 6

Давайте сдвинем это число еще раз:

Имеем:

0 1 1 0

Сдвигаем его влево и получаем:

1 1 0 0

Если перевести это число в десятичную систему, то получим число 12_{10}

Если это число снова подвергнуть сдвигу, то результатом будет переполнение.

Следует заметить, что операция сдвига осуществляется намного быстрее, чем операция умножения (или деления)

Следует запомнить, что:

- производя сдвиг влево на ОДИН РАЗРЯД мы производим умножение на 2
- производя сдвиг влево на ДВА РАЗРЯДА мы производим умножение на 4
- производя сдвиг влево на ТРИ РАЗРЯДА мы производим умножение на 8 и

т.д.

Сдвиг можно осуществлять и вправо

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.16/72

- в этом случае при сдвиге на 1 разряд будет производиться деление на 2,
- сдвигая вправо на 2 разряда будет производиться деление на 4 и т.д.

Что самое интересное, сдвиг можно производить и с десятичными числами!

Взять то же число 10_{10} и сдвинуть его влево - т.е. мы присоединяем дополнительный 0 со стороны младшего разряда. Получается 100_{10}

В основе сдвига операция производится с основной системы счисления к которой относится число, над которым проводятся операции сдвига.

Производится эта операция посредством оператора:

- "<<" для сдвига влево
- ">>" для сдвига вправо

Если применять сдвиг к десятичным числам в Питоне, то операция производится путем битового сдвига, т.е. происходит преобразование десятичного числа в двоичное, далее применяется сдвиг и полученное число снова преобразуется в десятичное.

Т.е. если мы наберем в Терминале:

```
6 << 1 # сдвигаем 6 влево на один разряд
```

То получим **12**, т.е. произошло умножение на 2

Если полученное число 12 сдвинуть влево на один разряд еще раз

```
12 << 1
```

Получим **24**

Соответственно при сдвиге вправо происходит деление:

Сдвинем десятичное число 24 вправо на один разряд

```
24 >> 1
```

Получим – **12**

Чтобы применить сдвиг к другой системе счисления, необходимо использовать соответствующую запись.

Например для двоичного числа:

```
0b0011 << 1
```

В результате получаем **6**

Двоичная форма записи после сдвига преобразовывается в десятичное.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.17/72

Отрицательное число

Эти числа меньше нуля, записываются с приставкой "-" перед числом...

Определение отрицательного числа: при наличии положительного числа "а" при сложении такого же отрицательного числа дает в результате НОЛЬ, т.е. при взаимном сложении они "уничтожаются"

Для двоичного формата это утверждение также справедливо

Естественно данные примеры производятся над **ЗНАКОВЫМИ** числами, в которых старший разряд отведен под указание того, какой знак у числа.

При программировании, как правило, всегда известно, с какого типа данными мы работаем - со знаковыми или беззнаковыми.

В основном, в языках программирования производится так называемое "объявление" переменной, когда мы явным образом указываем с каким типом чисел (в данном случае) мы будем работать.

При объявлении числа типом int - мы будем работать со знаковыми числами, а если число объявить как unsigned int - тогда число будет восприниматься беззнаковым (без какого либо указания на знак в старшем бите).

По умолчанию - числа используются со знаком. Если нам необходимо работать с беззнаковыми числами, то мы это должны указать явно.

Если знаковое число +1 в двоичном формате прибавить к -1, то должен получиться ноль (результат будет с переполнением)

```

+/- знак числа
  0 0 0 1 110
+  1 1 1 1 -110
1  0 0 0 0 010

```

Еще пример

```

+/- знак числа
  0 0 1 1 310
+  1 1 0 1 -310
1  0 0 0 0 010

```

И еще один пример

```

+  +/- знак числа

```

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.18/72

$$\begin{array}{r} 0 \ 1 \ 0 \ 1 \ 5_{10} \\ 1 \ 0 \ 1 \ 1 \ -5_{10} \\ 1 \ 0 \ 0 \ 0 \ 0 \ 0_{10} \end{array}$$

Перевод положительного числа в отрицательное

Для того, чтобы произвести перевод положительного двоичного числа в отрицательно надо сделать следующее:

- в исходном числе меняем все нули на единицы, а единицы на нули
- к младшему байту прибавляем 1

Например: возьмем положительное двоичное число 0101_2 (5_{10}) и сделаем из него отрицательное

- меняем единицы на нули и нули на единицы: 1010_2
- прибавляем единицу к младшему разряду: 1011_2

Если мы говорим о двоичном числе СО ЗНАКОМ (у которого старший разряд является показателем знака: 0 - положительное число, 1 - отрицательное) то полученный результат будет соответствовать -5 в десятичной системе счисления

Возьмем 0110_2 (6_{10}) и тоже поменяем ему знак:

- меняем единицы на нули и нули на единицы: 1001_2
- прибавляем единицу к младшему разряду: 1010_2 (-6_{10})

Вычитание двоичных чисел (столбиком)

Помните, как мы производим вычитание чисел столбиком в нашей привычной десятичной системе счисления?

Вычитание десятичных чисел

Вычтем из десятичного числа 10080 число 1901

$$10080 - 1901 = 8179$$

Курсивом выделены разряды числа

$$\begin{array}{r} 5 \ 4 \ 3 \ 2 \ 1 \\ - \ 1 \ 0 \ 0 \ 8 \ 0 \\ \quad 1 \ 9 \ 0 \ 1 \end{array}$$

$$10080 - 1901 = 8179$$

Вы можете посмотреть разбор пошагового вычитания:

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.19/72

Вычитание двоичных чисел

Для вычитания столбиком двоичных чисел действуют те же самые правила, только в качестве заемного числа выступает число 2, а не 10, поскольку мы оперируем значением двоичной системы счисления – 2

Правила вычитания те же:

$$0 - 0 = 0$$

$$0 - 1 = 1 \text{ и занимает разряд из старшего бита}$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Вычитаем:

$$1101110 \text{ (десятичное 110)}$$

$$- 10111 \text{ (десятичное 23)}$$

$$=1010111 \text{ (десятичное 87)}$$

Еще несколько примеров вычитания двоичных чисел:

В круглых скобках даны эквиваленты в десятичном формате

$$456 - 112 = 344$$

$$111001000 \text{ (456)}$$

$$- 1110000 \text{ (112)}$$

$$=101011000 \text{ (344)}$$

$$234 - 12 = 222$$

$$11101010 \text{ (234)}$$

$$- 1100 \text{ (12)}$$

$$=11011110 \text{ (222)}$$

$$345 - 232 = 113$$

$$101011001 \text{ (345)}$$

$$- 11101000 \text{ (232)}$$

$$=001110001 \text{ (113)}$$

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.20/72

$$16 - 3 = 13$$

$$10000 (16)$$

$$- 11 (3)$$

$$= 1101 (13)$$

Операцию вычитания в ЭВМ заменяют операцией сложения. Сначала запишем правило, по которому это выполняется, а потом рассмотрим на примере.

ПРАВИЛО:

1. Уравниваем количество разрядов в уменьшаемом и вычитаемом.
2. Находим дополнительный код к вычитаемому:
 - Все 0 заменяем на 1, а 1 на 0 получим обратный код;
 - Прибавляем 1 в нулевом разряде обратного кода и получим дополнительный код;
3. Складываем уменьшаемое и дополнительный код.
4. Отбрасываем единицу в старшем разряде.

$$\begin{array}{r}
 \text{разность} \nearrow \\
 1000100 - 100011 = 100001 \\
 \swarrow \quad \searrow \\
 \text{уменьшаемое} \quad \text{вычитаемое}
 \end{array}$$

$$1) 1000100$$

$$0100011$$

$$2) \quad \text{а) } 1011100 - \text{обратный код}$$

$$\text{б) } 1011100$$

$$+ \quad \frac{1 - \text{дополнительный код}}{1011101}$$

$$\begin{array}{r}
 \quad 111 \\
 3) \quad 1000100 \\
 + \quad 1011101 \\
 \hline
 10100001
 \end{array}$$

$$4) \quad \cancel{1}0100001$$

Самостоятельно:

$$111101 - 1001 = 110100$$

Арифметические операции над двоичными числами с плавающей точкой

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.21/72

В современных ЭВМ числа с плавающей точкой хранятся в памяти машин, имея мантиссу и порядок (характеристику) в прямом коде и нормализованном виде. Все арифметические действия над этими числами выполняются так же, как это делается с ними, если они представлены в полулогарифмической форме (мантисса и десятичный порядок) в десятичной системе счисления. Порядки и мантиссы обрабатываются отдельно.

Сложение (вычитание). Операция сложения (вычитания) производится в следующей последовательности.

1. Сравниваются порядки (характеристики) исходных чисел путем их вычитания $p=r_1-p_2$. При выполнении этой операции определяется, одинаковый ли порядок имеют исходные слагаемые.

2. Если разность порядков равна нулю, то это значит, что одноименные разряды мантисс имеют одинаковые веса (двоичный порядок). В противном случае должно проводиться выравнивание порядков.

3. Для выравнивания порядков число с меньшим порядком сдвигается вправо на разницу порядков p . Младшие выталкиваемые разряды при этом теряются.

4. После выравнивания порядков мантиссы чисел можно складывать (вычитать) в зависимости от требуемой операции. Операция вычитания заменяется операцией сложения в соответствии с данными табл. 2.3. Действия над слагаемыми производятся в ОК или ДК по общим правилам.

5. Порядок результата берется равным большему порядку.

6. Если мантисса результата не нормализована, то осуществляются нормализация и коррекция значений порядка.

Пример: Сложить два числа $A_{10}=+1.375$; $B_{10}=-0.625$.

$$A_2=+1.011=0:1011 \cdot 10^1; B_2=-0.101=-0:101 \cdot 10^0.$$

Умножение (деление). Операция умножения (деления) чисел с плавающей точкой также требует разных действий над порядками и мантиссами. Алгоритмы этих операций выполняются в следующей последовательности.

1. При умножении (делении) порядки складываются (вычитаются) так, как это делается над числами с фиксированной точкой.

2. При умножении (делении) мантиссы перемножаются (делятся).

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.22/72

3. Знаки произведения (частного) формируются путем сложения знаковых разрядов сомножителей (делимого и делителя). Возможные переносы из знакового разряда игнорируются

Вопросы для самопроверки:

1. Назвать правила сложения, вычитания, умножения и деления двоичных чисел.
2. Назвать правило вычитания двоичных чисел с применением дополнительного кода.
3. Как в ЭВМ представляются числа с плавающей точкой (запятой)?
4. Как в ЭВМ осуществляются арифметические операции над двоичными числами с плавающей точкой (запятой)

Тема 1.2 Логические основы ЭВМ Самостоятельная работа №3 Минимизация логических функций

Цель работы:

1. Закрепление темы 1.2. *Логические основы ЭВМ*
2. Знакомство с этапами минимизации логических функций.

План работы:

Минимизация логических функций методом Квайна

Минимизация логических функций методом Квайна

Метод Квайна относится к числу таких методов минимизации функции алгебры логики, которые позволяют представлять функции в ДНФ или КНФ с минимальным числом членов и минимальным числом букв в членах. Этот метод содержит два этапа преобразования выражения функции: на первом этапе осуществляется переход от канонической формы (СДНФ или СКНФ) к так называемой *сокращенной форме*, на втором этапе—переход от сокращенной формы логического выражения к *минимальной форме*.

Первый этап (получение сокращенной формы).

Пусть заданная функция f представлена в СДНФ.

Переход к сокращенной форме основан на последовательном применении двух операций: *операции склеивания* и *операции поглощения*.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.23/72

$w \cdot x \vee w \cdot \bar{x} = w \cdot (x \vee \bar{x}) = w$ Для выполнения операции склеивания

выявляются в выражении пары членов вида и, различающихся лишь тем, что один из аргументов в одном из членов представлен без инверсии, в другом—с инверсией. Затем проводится склеивание таких пар членов: и результаты склеивания w вводятся в выражение функции в качестве дополнительных членов. Далее проводится операция поглощения. Она основана на равенстве

$w \vee w \cdot z = w \cdot (1 \vee z) = w$ (член w поглощает член $w \cdot z$). При проведении этой

операции из логического выражения вычеркиваются все члены, поглощаемые членами, которые введены в результате проведения операции склеивания.

Операции склеивания и поглощения проводятся последовательно до тех пор, пока их выполнение оказывается возможным.

Покажем выполнение этих операций применительно к функции, представленной в табл. 3.5.

Записываем СДНФ функции

$$f(x_1, x_2, x_3) = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \vee \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \vee \bar{x}_1 \cdot x_2 \cdot x_3 \quad (3.12)$$

Попарным сравнением членов (каждого из членов со всеми последующими) выявляем склеивающиеся пары членов:

$\bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_1 \cdot \bar{x}_2$ первый и четвертый члены (результат склеивания);

$\bar{x}_1 \cdot \bar{x}_3$ второй и третий члены (результат склеивания);

$\bar{x}_1 \cdot x_3$ второй и четвертый члены (результат склеивания);

$\bar{x}_1 \cdot x_2$ третий и пятый члены (результат склеивания);

четвертый и пятый члены (результат склеивания).

Таблица 3.5

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$f(x_1, x_2, x_3)$	0	0	1	1	0	1	0	1

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.24/72

Результаты операции склеивания вводим в выражение функции и проводим операцию поглощения ими членов исходного выражения: $x_2 \cdot \bar{x}_3 \cdot x_1 \cdot \bar{x}_2 \cdot \bar{x}_2 \cdot \bar{x}_3$ Член поглощает те члены исходного выражения, которые содержат, т. е. первый и четвертый. Эти члены вычеркиваются. Член поглощает второй и третий, а член $x_1 \cdot \bar{x}_3$ – пятый член исходного выражения.

Повторяем операции склеивания и поглощения: $x_1 \cdot \bar{x}_2 \cdot x_1 \cdot \bar{x}_2$

Член операции склеивания $x_1 \cdot \bar{x}_3 \cdot x_1 \cdot \bar{x}_3$ Здесь склеивается лишь пара членов и , (склеивание пары членов и , приводит к тому же результату), результат склеивания x_1 , поглощает второй, третий, четвертый и пятый члены выражения.

Дальнейшее проведение операций склеивания и поглощения оказывается невозможным, сокращенная форма выражения заданной функции (в данном примере она совпадает с минимальной формой)

$$f(x_1, x_2, x_3) = x_2 \cdot \bar{x}_3 \vee x_1 \quad (3.13)$$

$x_2 \cdot \bar{x}_3$ Члены сокращенной формы (в рассмотренном примере такими членами служат $x_2 \cdot \bar{x}_3$ и x_1 называются *простыми импликантами* функции.

Как видим, получено выражение существенно более простое по сравнению с СДНФ функции.

На рис. 3.27 приведена структурная схема логического устройства в базисе И, ИЛИ, НЕ, построенная с использованием выражения (3.13).

Второй этап (получение минимальной формы).

Сокращенная форма может содержать лишние члены, исключение которых из выражения функции не повлияет на значение функции.

Таким образом, дальнейшее упрощение логического выражения достигается исключением из выражения лишних членов. В этом и заключается содержание второго этапа минимизации. Покажем этот этап минимизации логического выражения на примере построения логического устройства для функции в табл. 3.6.

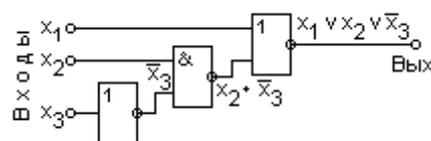


рис 3.27

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.25/72

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.26/72

Таблица 3.6

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$f(x_1, x_2, x_3, x_4)$	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1

Совершенная ДНФ данной функции

(3.14)

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 \vee \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} \vee \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{x_1} \cdot x_2 \cdot x_3 \cdot \overline{x_4} \vee \overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4$$

Вопросы для самоконтроля:

1. Назвать этапы преобразования выражения **логических функций методом Квайна**
2. Что представляет так называемая *сокращенная форма* выражения логических функций?
3. Что представляет собой *минимальная форма* выражения логических функций?

Самостоятельная работа 4 Синтез комбинационных логических устройств

Цель работы:

1. Выполнение синтеза логической схемы цифрового устройства, имеющего 4 входа и 2 выхода.
2. Составление логических уравнений для каждого выхода по таблице истинности.
3. Минимизация функций с помощью карт Карно, выбор оптимального варианта; принципиальная схема.

План работы:

1. Синтез цифровых комбинационных схем по произвольной таблице истинности.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.27/72

2. Совершенная дизъюнктивная нормальная форма (СДНФ)
3. Совершенная конъюнктивная нормальная форма (СКНФ)

Синтез цифровых комбинационных схем по произвольной таблице истинности

Любая цифровая комбинационная схема (логическая схема без памяти) полностью описывается таблицей истинности. При этом не обязательно, чтобы все комбинации входных цифровых сигналов были полезными. Возможна ситуация, когда только часть комбинаций входных логических сигналов является полезной. В этом случае выходные сигналы цифрового устройства для оставшихся комбинаций входных логических сигналов могут быть доопределены произвольно. Обычно при этом стараются выбирать цифровые значения выходных сигналов таким образом, чтобы схема цифрового устройства получилась простейшей.

Для реализации цифровых логических схем с произвольной таблицей истинности используется сочетание простейших логических элементов "И" "ИЛИ" "НЕ". Существует два способа синтеза цифровых схем, реализующих произвольную таблицу истинности. Это СКНФ (логическое произведение суммы входных сигналов) и СДНФ (сумма логических произведений входных сигналов).

При синтезе цифровой схемы, реализующей произвольную таблицу истинности, **каждый выход анализируется (и строится схема) отдельно и независимо**. В настоящее время наиболее распространены цифровые микросхемы, совместимые с ТТЛ технологией, а в этой технологии производства микросхем проще всего получить логические элементы "И". Поэтому первым рассмотрим способ реализации произвольной таблицы истинности, основанный на СДНФ.

Совершенная дизъюнктивная нормальная форма (СДНФ)

Для реализации таблицы истинности при помощи логических элементов "И" (СДНФ) достаточно рассмотреть только те строки таблицы истинности, которые содержат логические "1" в выходном сигнале. Строки, содержащие в выходном сигнале логический 0 в построении цифровой схемы не участвуют. Каждая строка, содержащая в выходном сигнале логическую "1", реализуется схемой логического элемента "И" с количеством входов, совпадающим с количеством входных сигналов в таблице истинности.

Входные сигналы, описанные в таблице истинности логической единицей, подаются на вход этого логического элемента непосредственно, а входные сигналы, описанные в таблице истинности логическим нулем, подаются на вход этого же логического элемента "И" через инверторы. Объединение сигналов с выходов логических элементов "И", реализующих отдельные строки таблицы истинности, производится при помощи логического элемента "ИЛИ". Количество входов в логическом элементе "ИЛИ" определяется количеством строк в таблице истинности, в которых в выходном сигнале присутствует логическая единица.

Рассмотрим конкретный пример. Пусть необходимо реализовать таблицу истинности, приведенную на рисунке 1:

Входы				Выходы	
In0	In1	In2	In3	Out0	Out1
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	1

Рисунок 1. Произвольная таблица истинности.

Для синтеза цифровой схемы, реализующей сигнал Out0, достаточно рассмотреть строки, выделенные красным цветом. В таблице истинности три строки, содержащие единицу в выходном сигнале Out0, поэтому в формуле СДНФ будет содержаться три произведения входных сигналов — **термов**, а точнее минтермов:

$$Out0 = \overline{In0} \cdot \overline{In1} \cdot \overline{In2} \cdot In3 + \overline{In0} \cdot In1 \cdot \overline{In2} \cdot \overline{In3} + \overline{In0} \cdot In1 \cdot In2 \cdot In3$$

Так как количество переменных в каждом терме (ранг терма) данного логического выражения равно, то такое логическое выражение называется совершенным. (Совершенная Дизъюнктивная Нормальная Форма — СДНФ)

Полученное логическое выражение реализуется микросхемой D2 в схеме, приведенной на рисунке 2. Как и в формуле, каждая строка (терм) реализуется своим логическим элементом "И", затем выходы этих логических элементов объединяются при помощи логического элемента "ИЛИ". Количество входов логического элемента "И" (дизъюнкция) в СДНФ однозначно определяется количеством входных сигналов в таблице истинности. Количество этих элементов, а значит и количество входов в логическом элементе "ИЛИ" определяется количеством строк с единичным сигналом на реализуемом выходе цифровой схемы.

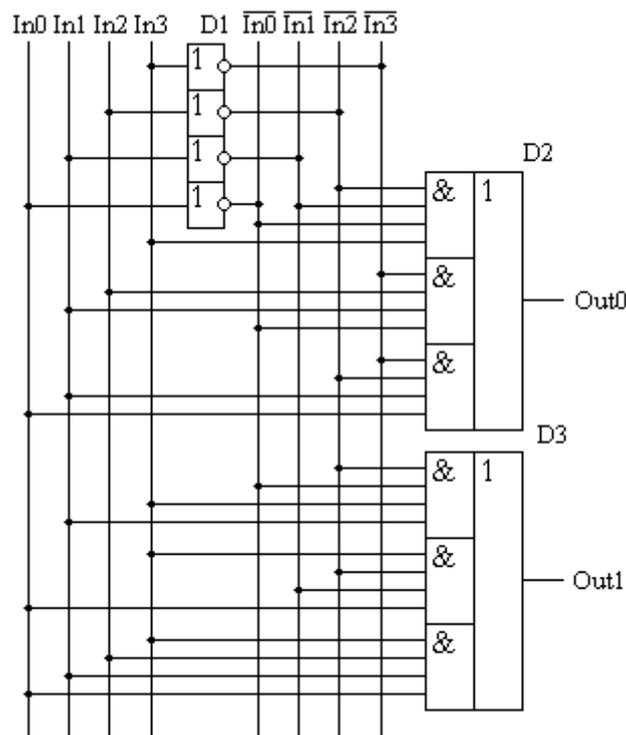


Рисунок 2. Принципиальная схема, реализующая таблицу истинности, приведенную на рисунке 1

Для построения схемы, реализующей сигнал Out1, достаточно рассмотреть строки, выделенные зеленым цветом. Эти строки реализуются микросхемой D3. Принцип построения этой схемы не отличается от примера, рассмотренного выше. В таблице истинности присутствуют всего три строки, содержащие единицу в выходном сигнале Out1, поэтому в формуле СДНФ выхода Out1 будет содержаться три произведения входных сигналов:

$$Out1 = \overline{In0} \cdot In1 \cdot \overline{In2} \cdot In3 + In0 \cdot \overline{In1} \cdot \overline{In2} \cdot In3 + In0 \cdot In1 \cdot In2 \cdot In3$$

Обычно при построении цифровых схем после реализации таблицы истинности производится минимизация схемы, но для упрощения понимания

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»		
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА		С.30/72

материала сейчас минимизация производится не будет. Это оправдано еще и с той точки зрения, что неминимизированные схемы обычно обладают максимальным быстродействием. При реализации схемы на ТТЛ микросхемах быстродействие всего узла будет равно быстродействию одиночного инвертора. По СДНФ форме логического выражения обычно строятся современные микросхемы большой интеграции — программируемые логические интегральные схемы (ПЛИС), такие как программируемые логические матрицы (ПЛМ), программируемые матрицы логики ПМЛ или CPLD.

Совершенная конъюнктивная нормальная форма (СКНФ)

Еще одним способом реализации цифровых комбинационных схем является запись логического выражения в совершенной конъюнктивной нормальной форме (СКНФ). Применение СКНФ оправдано при большом количестве логических единиц в выходном сигнале проектируемой цифровой схемы, как это показано в качестве примера в таблице истинности 2, или в ЭСЛ-микросхемах.

Таблица 2 Пример таблицы истинности 2

№ комбинации	Входы				Выходы	
	8	4	2	1	a	b
0	0	0	0	0	1	1
1	0	0	0	1	0	1
2	0	0	1	0	1	1
3	0	0	1	1	1	1
4	0	1	0	0	0	1
5	0	1	0	1	1	0
6	0	1	1	0	1	0
7	0	1	1	1	1	1
8	1	0	0	0	1	1
9	1	0	0	1	1	1

Для реализации цифрового комбинационного устройства по таблице истинности при помощи логических элементов "ИЛИ" (СКНФ) достаточно рассмотреть только те строки таблицы истинности, которые содержат логические "0" в выходном сигнале. Строки, содержащие в выходном сигнале логическую "1" в построении логического выражения, а, следовательно, и принципиальной схемы цифрового устройства не участвуют. Каждая строка, содержащая в выходном

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.31/72

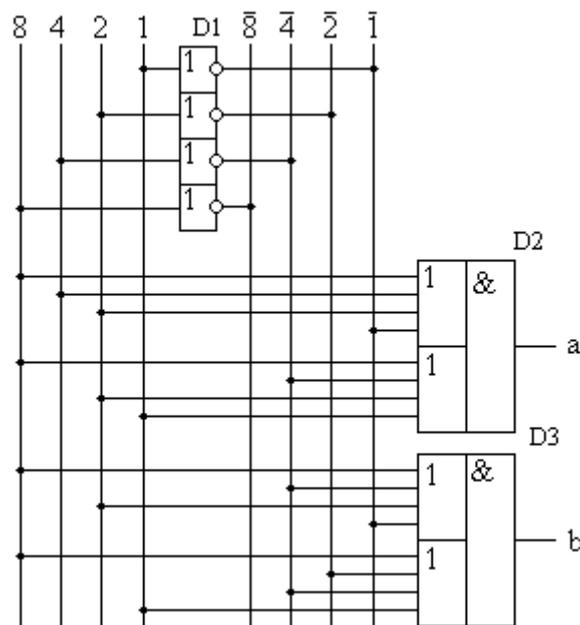
сигнале логический "0", реализуется схемой логического элемента "ИЛИ" с количеством входов, совпадающим с количеством входных сигналов в таблице истинности.

Для построения схемы, реализующей сигнал Out0, достаточно рассмотреть строки, выделенные курсивом. В рассматриваемой таблице истинности имеются всего две строки, содержащие логический ноль в выходном сигнале a, поэтому в формуле СКНФ будет содержаться две суммы входных сигналов:

$$a = (8 + 4 + 2 + \bar{1}) \cdot (8 + \bar{4} + 2 + 1)$$

Входные сигналы, описанные в таблице истинности логическим нулём, подаются на вход этой схемы непосредственно, а входные сигналы, описанные в таблице истинности логической единицей, подаются на логического элемента "ИЛИ" через инверторы. Объединение сигналов с выходов схем "ИЛИ", реализующих отдельные строки таблицы истинности, производится при помощи схемы логического элемента "И". Количество входов в схеме "И" определяется количеством строк в таблице истинности, в которых в выходном сигнале присутствует логическая единица.

Полученная формула в схеме на рисунке 9 реализуются микросхемой D2.



Для построения схемы, реализующей сигнал b, достаточно рассмотреть строки, выделенные жирным шрифтом. Эти строки в схеме на рисунке 9 реализуются микросхемой D3. Принцип построения этой схемы не отличается от примера, рассмотренного выше. В таблице истинности присутствуют всего две

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.32/72

строки, содержащие ноль в выходном сигнале b , поэтому в формуле СКНФ выхода b будет содержаться две суммы входных сигналов:

$$a = (8 + \bar{4} + 2 + \bar{1}) \cdot (8 + \bar{4} + \bar{2} + 1)$$

Вопросы для самопроверки:

1. Какие устройства относятся к цифровым комбинационным схемам?
2. Дайте понятие таблицы истинности.
3. Какие логические элементы используются для реализации цифровых логических схем?
4. Какие существуют способы синтеза цифровых схем, реализующих произвольную таблицу истинности?
5. В чём заключается способ реализации произвольной таблицы истинности, основанный на СДНФ?
6. В чём заключается способ реализации произвольной таблицы истинности, основанный на СДНФ?
7. Как определяется количество входов в логическом элементе "ИЛИ" в способе реализации произвольной таблицы истинности, основанном на СДНФ?
8. В чём заключается способ реализации произвольной таблицы истинности, основанный на СКНФ?
9. Как определяется количество входов в схеме "И" в способе реализации произвольной таблицы истинности, основанном на СДНФ?

Самостоятельная работа №5: Проработка конспектов занятий по теме 1.2. Подготовка к защите ЛР и ПР.

Время выполнения 6ч

Цель занятия: Повторение и закрепление темы 1.2

При самостоятельной работе с конспектом необходимо:

1. Внимательно изучить изложенный в конспекте материал, если этого недостаточно, повторить его по учебнику;
2. Повторить способы представления и передачи двоичных чисел в ЭВМ.
3. Повторить основные понятия алгебры логики. Элементарные логические функции.
4. Представление переключательных (логических) функций.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.33/72

5. Повторить нормальные и совершенные нормальные логические функции.
6. Повторить функционально полные системы переключательных функций алгебры логики.
7. Повторить условные обозначения логических элементов.
8. Повторить особенности построения схем логических устройств.
9. Повторить формы представления чисел в цифровых устройствах.
10. При самостоятельной подготовке к выполнению практического задания и лабораторной работы необходимо самостоятельно изучить методические указания к выполнению работы, в тетради для выполнения практических работ подготовить требуемые рисунки, схемы, таблицы. Продумать ход работы и ожидаемый результат

Контрольные вопросы:

1. Изобразить символическое обозначение логических элементов: «НЕ», «И», «ИЛИ», «И-НЕ», «ИЛИ-НЕ». Какие операции они выполняют?
2. Составить схему, реализующую логическое выражение:

$$Y = X_1 X_2 X_3 X_4 X_5 X_6$$
3. Назвать способы синтеза цифровых схем. Дать понятие СДНФ и СДНФ.
4. По какой форме логического выражения обычно строятся современные микросхемы большой интеграции - программируемые логические интегральные схемы (ПЛИС), такие как программируемые логические матрицы (ПЛМ), программируемые матрицы логики ПМЛ?
5. Какие цифровые устройства относятся к устройствам комбинационного типа?
6. Какие цифровые устройства относятся к устройствам последовательностного типа?

Самостоятельная работа №6: Проработка конспектов занятий по теме 2.1. Подготовка к защите ЛР и ПР

Время выполнения 4 ч

Цель занятия: Повторение и закрепление темы 2.1

При самостоятельной работе с конспектом необходимо:

1. Внимательно изучить изложенный в конспекте материал, если этого недостаточно, повторить его по учебнику;
2. Какие устройства относятся к устройствам комбинационного типа?

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.34/72

3. Повторить назначение, классификацию, символическое изображение, принцип построения схем, применение шифраторов.
4. Повторить назначение, классификацию, символическое изображение, принцип построения схем, применение дешифраторов. Принцип построения схем линейного и прямоугольного дешифраторов.
5. Повторить схемы преобразователей кодов. Применение преобразователей кодов.
6. Повторить назначение, символическое изображение, принцип построения схем, работа мультиплексоров. Мультиплексорное дерево. Демультимплексоры
7. При самостоятельной подготовке к выполнению практического задания и лабораторной работы необходимо самостоятельно изучить методические указания к выполнению работы, в тетради для выполнения практических работ подготовить требуемые рисунки, схемы, таблицы. Продумать ход работы и ожидаемый результат

Контрольные вопросы:

1. Какое устройство называется шифратором? дешифратором?
2. Из каких основных элементов строятся шифраторы и дешифраторы?
3. В каком случае можно построить линейный дешифратор?
4. Область применения шифраторов и дешифраторов.
5. В чём назначение мультиплексора?
6. Как изображается мультиплексор в принципиальных схемах?
7. В чём назначение стробирующего сигнала?
8. Как задаётся адрес выбираемого входа мультиплексора?
9. Как связаны между собой число информационных и адресных входов?
10. В каких случаях используется параллельное включение мультиплексоров?

Самостоятельная работа 7. Построение схем делителей частоты с заданным коэффициентом деления. Построение схем счётчиков и временных диаграмм, объясняющих их работу.

Цель работы: Закрепление темы 2.2. *Цифровые устройства последовательностного типа*

1. Электронные счетчики и делители частоты

На базе счетных триггеров можно построить цифровое устройство, получившее название **электронного счетчика**. Электронные счетчики (далее, просто счетчики) позволяют вести подсчет электрических импульсов, количество которых (поступивших на вход счетчика) представляется, обычно, в параллельном коде. Счетчики могут отличаться модулем счета и типом счетной последовательности, которая, в частности, может быть двоичной, двоично-десятичной, в коде Грея и т.п. Цифровые последовательностные устройства, выполненные по схеме счетчика, но имеющие один счетный вход и один выход называются **делителями** частоты. Таким образом, любой счетчик может служить в качестве делителя частоты, если используется информация только одного из его

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.35/72

выходов. Так как счетчики и делители имеют единую структуру, основное внимание будет уделено синтезу счетчиков.

Счетчики и делители подразделяются на **асинхронные** и **синхронные**. У синхронных счетчиков все разрядные триггеры синхронизируются параллельно одними и теми же синхроимпульсами, поступающими из источника этих импульсов. Асинхронные счетчики имеют последовательную синхронизацию, т.е. каждый последующий разрядный триггер синхронизируется выходными импульсами триггера предыдущего разряда. Асинхронные счетчики иногда называют **последовательными**, а синхронные счетчики - **параллельными**.

Синхронные счетчики, в свою очередь, подразделяются на параллельно-синхронные и последовательно-синхронные. Параллельные счетчики имеют более высокую скорость счета, чем асинхронные.

Счетчики, независимо от способа синхронизации, подразделяются на счетчики прямого счета (суммирующие) и на счетчики обратного счета (вычитающие). В интегральном исполнении выпускаются также реверсивные счетчики, в которых имеется специальный вход для переключения режима работы, т.е. направления счета. Многие типы счетчиков, выпускаемые промышленностью в интегральном исполнении, имеют дополнительные входы предустановки, позволяющие использовать эти счетчики в режиме регистра памяти.

В качестве разрядных триггеров счетчиков и делителей могут быть использованы двухступенчатые D-триггеры, T- и JK-триггеры.

Счетчики относятся к последовательностным устройствам с циклически повторяющейся последовательностью состояний. Число, соответствующее количеству импульсов (поступивших на вход счетчика), при котором счетчик “возвращается” в исходное состояние, называется **модулем** или **коэффициентом** счета. Модуль счета, обычно, обозначают буквой **M** (или $K_{сч}$). Например, максимальный модуль счета счетчика из двух триггеров равен $M = 2^2 = 4$, трех триггеров - $M = 2^3 = 8$ и т.д. В общем случае для n - разрядного счетчика - $M = 2^n$. Модуль счета счетчика численно совпадает с модулем деления делителя частоты. Счетчик по модулю 8 позволяет реализовать (без дополнительных схемных затрат) делитель частоты на 8. Это значит, что данный делитель делит частоту входной импульсной последовательности на 8.

2. Асинхронный двоичный счетчик.

Асинхронный двоичный счетчик представляет собой совокупность последовательно соединенных триггеров (D - или JK), каждый из которых ассоциируется с битом в двоичном представлении числа. Если в счетчике m триггеров, то число возможных состояний счетчика равно 2^m , и, следовательно, модуль счета M также равен 2^m . Счетная последовательность в двоичном суммирующем счетчике начинается с нуля и доходит до максимального числа $2^m - 1$, после чего снова проходит через нуль и повторяется. В вычитающем двоичном счетчике последовательные двоичные числа перебираются в обратном порядке, и при повторении последовательности максимальное число следует за нулем.

Рассмотрим устройство двоичного суммирующего счетчика по модулю $M=16$, выполненного на базе JK-триггеров (рис. 6.1, а).

Как видно из рис. 6.1, (а), синхронизирующие входы всех триггеров, кроме крайнего левого (Т1), соединены с выходами предыдущих триггеров. Поэтому

состояние триггера меняется в ответ на изменение состояния предыдущего триггера.

Из таблицы состояния счетчика (рис. 6.1, б) легко заметить, что значение разряда в выбранной позиции меняется тогда, когда в соседней справа позиции состояние переходит из "1" в "0", управление триггерами осуществляется задним фронтом синхроимпульсов (отрицательным перепадом напряжения импульса синхронизации).

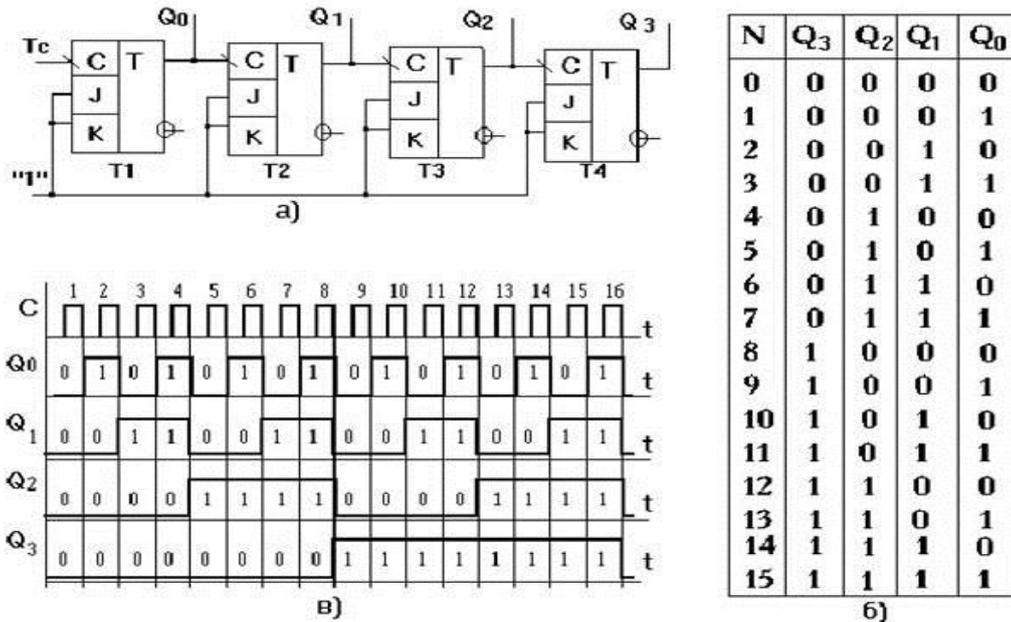


Рис. 6.1. Схема а), таблица состояний триггеров б) и временные диаграммы, поясняющие работу в) последовательного четырехразрядного счетчика на JK - триггерах

Временные диаграммы, поясняющие работу асинхронного суммирующего счетчика приведены на рис. 6.1, в.

3. Счетчики обратного счета (вычитающие счетчики).

На рис. 6.2 приведена схема асинхронного трехразрядного двоичного вычитающего счетчика, построенного на базе D-триггеров. Отметим, что условия для изменения состояний триггеров вычитающих счетчиков аналогичны условиям для суммирующих счетчиков с той лишь разницей, что они должны "опираться" на значения инверсных, а не прямых выходов триггеров. Следовательно, рассмотренный выше счетчик можно превратить в вычитающий, просто переключив входы "С" триггеров с выходов Q на выходы \bar{Q} . Когда в качестве разрядных триггеров используются D-триггеры, синхронизируемые передним фронтом синхроимпульсов, для получения вычитающего счетчика (асинхронного) входы "С" последующих триггеров соединяются с прямыми выходами предыдущих, также как в счетчике прямого счета, построенного на JK-триггерах.

Работа вычитающего счетчика на D-триггерах наглядно иллюстрирована на рис. 6.2, (б). Из рис. 6.2 следует, что после нулевого состояния всех триггеров, с приходом первого синхроимпульса они

устанавливаются в состояние "1". Поступление второго синхриимпульса приводит к уменьшению этого числа на одну единицу и т.д. После поступления восьмого импульса, снова, все триггеры обнуляются и цикл счета повторяется, что соответствует модулю $M=8$.

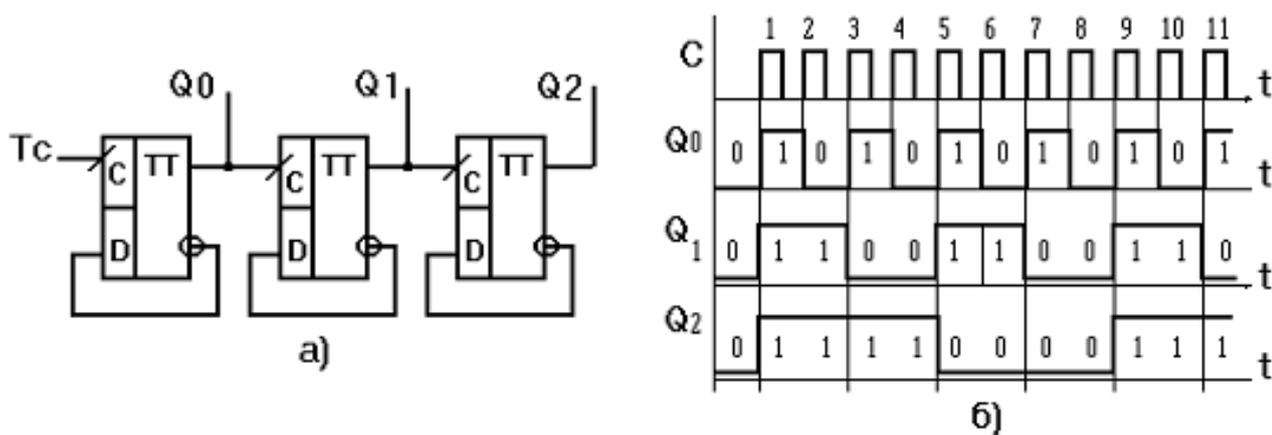


Рис. 3.34. Схема а) и временные диаграммы вычитающего трехразрядного счетчика на D - триггерах

В некоторых случаях необходимо, чтобы счетчик мог работать как в прямом, так и в обратном направлении счета. Такие **счетчики** называются **реверсивными**. Реверсивные счетчики могут быть как асинхронного, так и синхронного типа. Они строятся путем применения логических коммутаторов (мультиплексоров) в цепях связи между триггерами. Так, например, асинхронный реверсивный двоичный счетчик можно построить, если обеспечить подачу сигналов с прямого (при суммировании) или с инверсного (при вычитании) выхода предыдущего JK- или T-триггера на счетный вход последующего. В случае, когда реверсивный счетчик строится на базе D-триггеров, управляемых передним фронтом, для получения режима прямого счета следует соединить инверсный выход предыдущего с счетным входом последующего триггера.

Все рассмотренные типы счетчиков могут быть использованы в цифровых устройствах "умеренного" быстродействия, когда частота следования синхриимпульсов не превышает критического значения, при котором время задержки установки триггеров последних (старших) разрядов счетчика становится соизмеримым с длительностью периода входных тактовых импульсов. В связи с этим, асинхронные счетчики строятся на относительно небольшое количество разрядов, так как при большем количестве разрядов выходные сигналы триггеров старших разрядов появляются позднее, чем управляющие фронты синхриимпульсов (поступающих на вход первого триггера).

4. Параллельные счетчики (синхронные счетчики).

Как было уже сказано выше, параллельные счетчики бывают двух типов: **синхронные параллельные** и **синхронные последовательные**.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.39/72

Поскольку счетчик имеет одну общую линию синхронизации, состояние триггеров меняется синхронно, т.е. те триггеры, которые по синхроимпульсу должны изменить свое состояние, делают это одновременно, что существенно повышает быстродействие синхронных счетчиков.

6. Счетчики с произвольным коэффициентом счета.

Принцип построения подобного класса счетных устройств состоит в исключении нескольких состояний обычного двоичного счетчика, являющихся избыточными для счетчиков с коэффициентом пересчета, отличающимися от двоичных. При этом избыточные состояния исключаются с помощью обратных связей внутри счетчика.

Число избыточных состояний для любого счетчика определяется из следующего выражения:

$$M = 2^m - K_{сч},$$

где M - число запрещенных состояний, $K_{сч}$ - требуемый коэффициент счета; 2^m - число устойчивых состояний двоичного счетчика.

Задача синтеза счетчика с произвольным коэффициентом счета заключается в определении необходимых обратных связей и минимизации их числа. Требуемое количество триггеров определяется из выражения

$$n = \lceil \log_2 K_{сч} \rceil,$$

где $\lceil \log_2 K_{сч} \rceil$ - двоичный логарифм заданного коэффициента пересчета $K_{сч}$, округленный до ближайшего целого числа.

В каждом отдельном случае приходится применять какие-то конкретные методы получения требуемого коэффициента пересчета. Существует несколько методов получения счетчиков с заданным коэффициентом пересчета $K_{сч}$. Один из этих методов заключается в немедленном сбросе в "0" счетчика, установившегося в комбинацию, соответствующему числу $K_{сч}$. Его называют также **методом автосброса**. Рассмотрим пример реализации счетчика с $K_{сч}=10$ методом автосброса. Очевидно, что "сбрасывая" двоичный четырехразрядный счетчик на нуль каждый раз, когда он будет принимать состояние 1010, можно обеспечить "возврат" счетчика в исходное состояние после каждых десяти импульсов. Подобный прием удобно применять при использовании счетчиков в интегральном исполнении, имеющих ячейки конъюнкции (**И**) на входах установки в нуль, как это сделано в микросхеме К1533ИЕ5. В данном примере (рис. 3.37) организованы соединения, обеспечивающие коэффициент пересчета $K_{сч} = 10$.

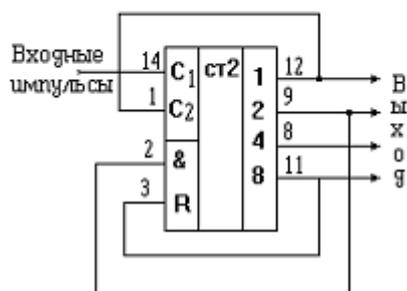


Рис. 3.37. Пример реализации счетчика с $K_{сч}=10$

Таблица 3.1

K1533IE5	Коэффициенты пересчета					
	3	5	6	9	10	12
Вход	14	14	14	14	14	14
Выход	9,12	8,9,12	8,9,12	все	все	все
Соединения выводов	1-12	1-12	1-12	1-12	1-12	1-12
	2-12	2-12	2-9	2-12	2-9	2-8
	3-9	3-8	3-8	3-11	3-11	3-11

Как следует из рис. 3.37, роль ячейки, выявляющей факт достижения кодовой комбинации 1010 на выходах счетчика, играет ячейка **И**, уже имеющаяся на входе сброса ИМС K1533IE5.

В таблице 3.1 поясняются конфигурации соединений для получения различных коэффициентов пересчета с помощью счетчика K1533IE5. Наиболее очевидные варианты получения коэффициентов (2, 4, 8, 16) в таблице не указаны. В графе “Соединения” таблицы указано, какие выводы микросхемы должны быть соединены между собой: например, указание 1-12 означает, что нужно соединить вывод 1 с выводом 12. В строках “Вход” и “Выход” таблицы указаны номера выводов микросхемы, на которые следует подавать входные импульсы и с которых надлежит снимать выходные, соответственно. Следует отметить, что ИМС K1533IE5 состоит из четырех счетных триггеров, один из которых имеет отдельные выходы входа и выхода, а остальные три триггера соединены последовательно по схеме асинхронного счетчика.

- Внимательно прочитав изложенный в пп1 – 6 материал, письменно ответить на вопросы для самопроверки

Вопросы для самопроверки:

- Что общего и в чём различие между построением счётчика и делителя частоты?
- От чего зависит коэффициент счёта?
- Изобразите схему ДЧ и временные диаграммы с коэффициентом деления 2.
- Каков принцип построения схем ДЧ с коэффициентом деления 2 в степени n ?
- Изобразите схему ДЧ. и временные диаграммы ДЧ с коэффициентом деления 8.
- Сравните принцип построения схем, изображенных в п.п. 3-4.
- Изобразите схему ДЧ и временные диаграммы ДЧ с коэффициентом деления 3. Отметьте её особенности по сравнению с предыдущими.
- Изобразите схему ДЧ и временные диаграммы ДЧ с коэффициентом деления 5. Отметьте её особенности по сравнению с предыдущими.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.41/72

9. Изобразите схему ДЧ и временные диаграммы ДЧ с коэффициентом деления 7. Отметьте её особенности по сравнению с предыдущими.

Самостоятельная работа №8. Десятичные сумматоры.

Цель работы:

1. Изучить принцип построения схем и работу десятичных сумматоров

План работы:

1. Назначение и принцип построения десятичных сумматоров
2. Сумматор для кода 8421.
3. Схема формирования обратного кода.

Назначение и принцип построения десятичных сумматоров

Для построения многоразрядных двоичных сумматоров необходимы одноразрядные двоичные сумматоры.

Многоразрядные десятичные сумматоры строятся с использованием одноразрядных десятичных сумматоров.

Одноразрядные десятичные сумматоры выполняют операцию суммирования десятичных цифр a_i , b_i и переноса P_i , поступающих в разряд, и формируют на выходах десятичную цифру суммы S_i и перенос P_{i+1} для передачи в следующий десятичный разряд.

При использовании десятичной системы счисления цифры разрядов десятичного числа представляются в некоторой двоичной форме. В связи с этим одна из особенностей одноразрядных десятичных сумматоров связана с тем, что суммируемые десятичные цифры a_i и b_i представляются многоразрядными двоичными числами (переносы P_i независимо от используемой системы счисления могут иметь лишь значения 0 либо 1). Рассмотрим построение одноразрядного сумматора десятичных цифр, представляемых в коде 8421.

Сумматор для кода 8421.

В работе сумматора этого типа имеются особенности в формировании переноса и суммы, отличающие его от работы двоичного сумматора.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.42/72

Правила сложения в десятичной системе счисления с использованием кода 8421 рассмотрены ранее. Построенная в соответствии с этими правилами схема одноразрядного десятичного сумматора (рис 6.1) включает в себя четырехразрядный двоичный сумматор(1), схему формирования переноса P_{i+1} в следующий десятичный разряд (2) и схему коррекции суммы (3). Последняя представляет собой трехразрядный сумматор, в котором при $P_{i+1} = 1$ производится прибавление единицы в разрядах нескорректированной суммы с весовыми коэффициентами 2 и 4.

Операция суммирования в случае, когда слагаемые (одно либо оба) имеют отрицательные значения, может производиться с представлением таких слагаемых в обратном коде.

Схема формирования обратного кода.

В десятичной системе счисления обратный код образуется путем преобразования каждой цифры числа в дополнение до 9. В табл. 6.1 приведены для десятичных цифр 0, 1,..., 9 прямые коды и соответствующие им обратные.

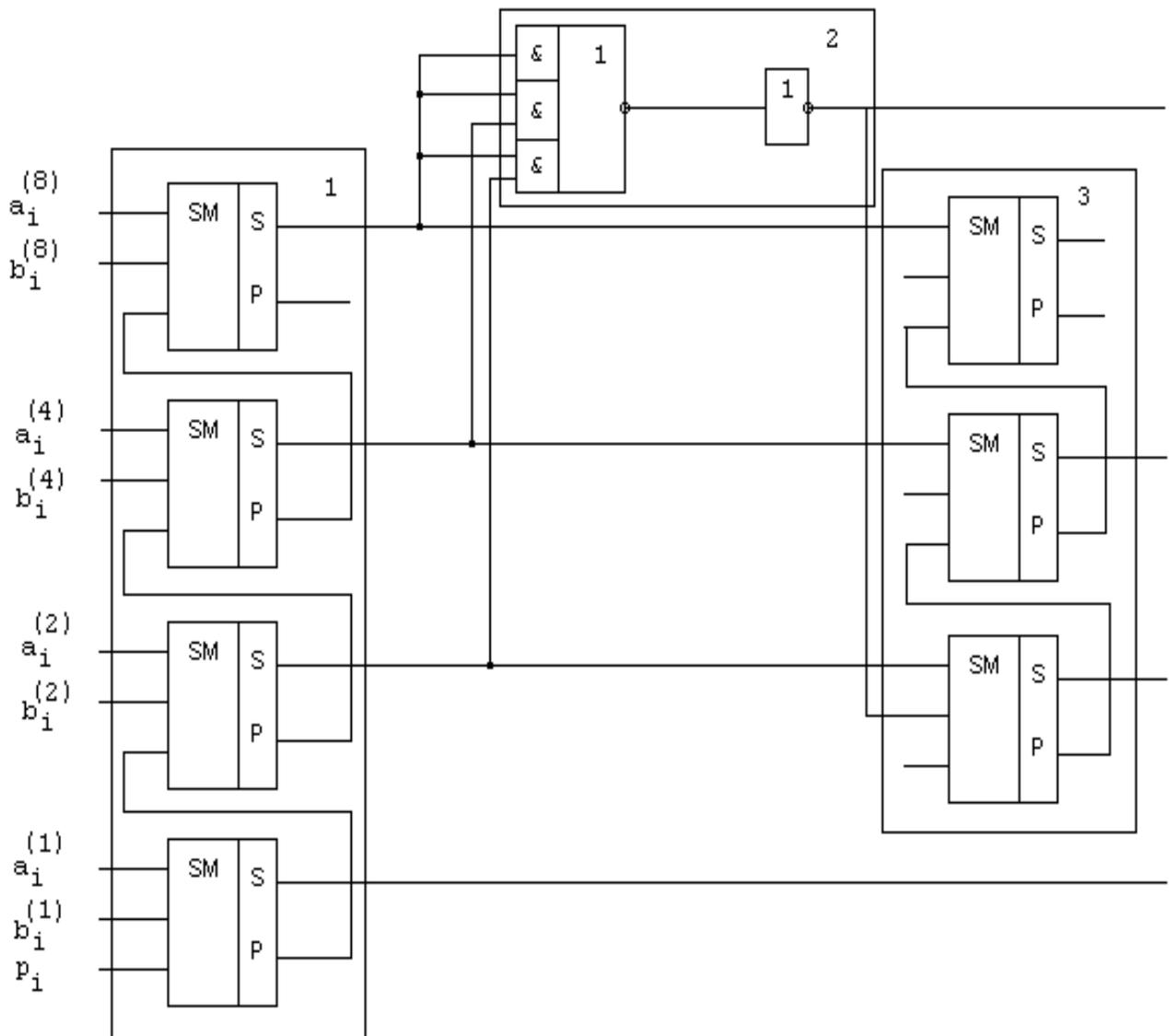


Рис 6.1 - Схема одноразрядного десятичного сумматора

Из сопоставления приведенных в таблице значений $\alpha_i^{(2)}, \alpha_i^{(1)}$ и соответствующих им $b_i^{(2)}, b_i^{(1)}$ нетрудно заключить, что $b_i^{(1)} = \bar{\alpha}_i^{(1)}, b_i^{(2)} = \bar{\alpha}_i^{(2)}$.

Таблица 6.1 Прямые и обратные коды десятичных цифр 0 - 9			
Десятичная ячейка	^ Прямой код 8421		

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.44/72

	$a_i^{(8)}$	$a_i^{(4)}$	$a_i^{(2)}$	$a_i^{(1)}$	
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	0
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	

Логические выражения для $b_i^{(4)}$ и $b_i^{(4)}$ можно получить из карт Вейча (табл. 6.2):

$$b_i^{(4)} = a_i^{(4)} \cdot a_i^{-(2)} \vee a_i^{-(4)} \cdot a_i^{(2)}, \quad b_i^{(8)} = a_i^{-(8)} \cdot a_i^{-(4)} \cdot a_i^{-(2)}$$

На рисунке 6.2 приведена схема, формирующая обратный код по полученным выше логическим выражениям,

Таблица 6.2 Логические выражения для $b_i^{(4)}$ и $b_i^{(4)}$

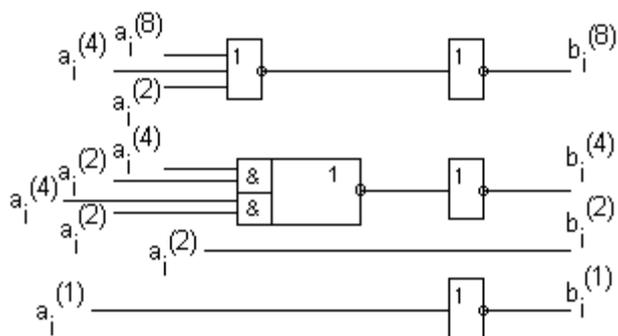
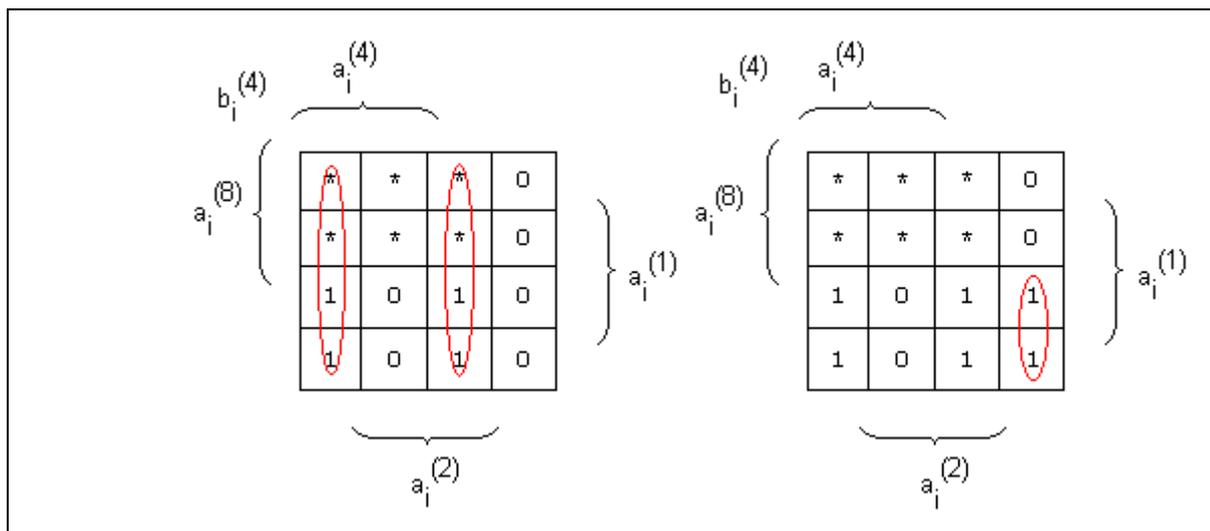


Рисунок 6.2 - Схема формирования обратного кода

Вопросы для самопроверки:

1. Принцип построения двоичного сумматора?
2. Принцип построения схемы полусумматора?
3. Принцип построения многоразрядного двоичного сумматора?
4. Доказать, что схема, представленная на рисунке 6.2, преобразует прямой код в обратный

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.46/72

Самостоятельная работа №9: Проработка конспектов занятий по теме 2.2. Подготовка к защите ЛР и ПР.

Время выполнения 6ч

Цель занятия: Повторение и закрепление темы 2.2

При самостоятельной работе с конспектом необходимо:

1. Внимательно изучить изложенный в конспекте материал, если этого недостаточно, повторить его по учебнику;

2. Повторить типы триггеров, символическое изображение.

3. Повторить разновидности схем, работу асинхронных RS – триггеров.

4. Повторить разновидности схем, работу синхронных RS – триггеров.

5. Триггеры с двухступенчатым запоминанием информации. Т- триггер. JK – триггеры.

8. Регистры - назначение, символическое изображение, классификация, применение.

9. Принципы построения и работы регистров параллельного и последовательного действия. Применение.

10.Регистры сдвига. Применение.

11.Назначение и типы счетчиков.

10. При самостоятельной подготовке к выполнению практического задания и лабораторной работы необходимо самостоятельно изучить методические указания к выполнению работы, в тетради для выполнения практических работ подготовить требуемые рисунки, схемы, таблицы. Продумать ход работы и ожидаемый результат

Контрольные вопросы:

1. Какие устройства относятся к устройствам последовательного типа?

2. Какое устройство называется триггером.

3. Построить схемы асинхронных RS-триггеров на элементах И-НЕ, ИЛИ-НЕ.

4. Основное отличие синхронных и асинхронных триггеров.

5. Характеристика RS, D, T, JK- триггеров. Применение.

6. Привести условные обозначения триггеров в интегральном исполнении.

7. Чем отличается триггер со статическим входом синхронизации от триггера с динамическим входом синхронизации.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.47/72

8. Условное обозначение динамических входов синхронизации триггеров.
9. Условные обозначения различных типов триггеров.
10. Какое устройство называется регистром? Применение регистров.
11. Какие схемы регистров Вам известны, их принципиальное отличие, достоинства и недостатки?
12. Объяснить, почему регистр может быть использован не только для хранения информации, но для выполнения арифметических действий. Какие это действия?
13. Какие функции выполняет делитель частоты?
14. На каких устройствах выполняется делитель частоты?
15. Сравните схемы и принцип работы счётчика и делителя частоты. В чём их отличие? Можно ли счётчики использовать в качестве делителя частоты?
16. Свяжите период импульсной последовательности с частотой повторения импульсов.
17. Свяжите виды межтриггерных связей с коэффициентом деления.
18. Изобразите временные диаграммы входных импульсов и импульсов на выходе каждого триггера делителя частоты.
19. Как определяется коэффициентом деления каскадного делителя частоты?
20. Какое устройство называется счетчиком импульсов?
21. Какие типы счетчиков импульсов Вам известны?
22. Чем отличаются синхронные счетчики от асинхронных, достоинства и недостатки?
23. Какие способы получения счетчика с заданным К сч. Вам известны?
24. В чем заключается принципиальная возможность получения реверсивных счетчиков?
25. В каком случае счетчик является двоичным, в каком нет?

Самостоятельная работа №10. Типовые узлы цифровых устройств, выполненные на программируемых логических устройствах с матричной структурой.

Цель работы: Закрепление темы 2,3

1. Программируемые логические устройства с матричной структурой

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.48/72

Программируемые логические матрицы (ПЛМ), выпускаемые отечественной промышленностью (K556PT1 и K556PT2), относятся к однократно программируемым большим интегральным схемам (БИС) с пережиганием плавких перемычек. Преимущество ПЛМ перед программируемым постоянным запоминающим устройством (ППЗУ) с произвольным доступом к памяти заключается в более экономном использовании логических элементов. Это объясняется тем, что ППЗУ с произвольным доступом к памяти реализует логическую функцию в совершенной дизъюнктивной нормальной форме (СДНФ), которая обладает избыточностью, в то время как ПЛМ реализует логический функционал в дизъюнктивной нормальной форме (ДНФ), минимальная форма которой и дает безызыточность. Отсюда следует, что перед программированием ПЛМ необходимо произвести минимизацию логического функционала, что целесообразно делать с помощью ЭВМ, сопряженной с программатором ПЛМ.

Программируемая логическая матрица серии K556PT1 относится к технологии транзисторно-транзисторная логика с диодами Шотки (ТТЛШ) и рассчитана на 16 входных переменных, 48 конъюнкций и 8 выходных функций (рис. 7.1). Кроме того, она имеет входы разрешения программирования W и разрешения выборки V . Время задержки составляет 50 нс. Потребляемая мощность 1 Вт. Обозначается PLA (programmable logic array).

унок 7.1 - 7.1

2. Программируемые логические интегральные схемы

Программируемые логические интегральные схемы (ПЛИС) появились в результате совершенствования во всех направлениях программируемых логических матриц, когда для многих разработчиков аппаратуры цифровой обработки сигналов стало ясно, что программируемая логическая интегральная схема – удобная в освоении и применении элементная база, альтернативы которой зачастую не найти.

Характеристика пакета	Назначение
Поддерживаемые устройства	EPF10K10, EPF10K10A, EPF10K20, EPF10K30, EPF10K30A, EPF10K30E (до 30000 эквивалентных вентилях), EPM9320, EPM9320A, EPF8452A, EPF8282A, MAX7000, FLEX6000, MAX5000, MAX3000A, Classic
Средства описания проекта	Схемный ввод, поддержка AHDL, средства интерфейса с САПР третьих фирм, топологический редактор, иерархическая структура проекта, наличие библиотеки

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.49/72

Средства компиляции проекта параметризуемых модулей
 Средства компиляции проекта Логический синтез и трассировка, автоматическое обнаружение ошибок, поддержка мегафункций по программам MegaCore и AMPP

Средства верификации проекта Временной анализ, функциональное и временное моделирование, анализ сигналов, возможность использования программ моделирования (симуляторов) третьих фирм

унок 7.1 -

3. Программирование ПЛМ

Программирование ПЛМ, выполняемое пользователем, проводится с помощью специальных программаторов и сведения для них должны иметь определенную форму. Имеются программаторы, которые принимают в качестве информации о ПЛМ таблицу функционирования (истинности), однако удобнее задавать сведения о самих перемычках. Символы, используемые при задании сведений для программирования ПЛМ:

– Н – переменная входит в терм в прямом виде, т.е. нужно оставить целой перемычку прямого входа и пережечь перемычку инверсного входа;

– «.» – указывает на то, что данная схема И не подключается к выходу и должна иметь переж

Вопросы для самопроверки:

1. Что общего и в чём различие между построением счётчика и делителя частоты?
2. От чего зависит коэффициент счёта?
Изобразите схему ДЧ и временные диаграммы с коэффициентом деления 2.
3. Каков принцип построения схем ДЧ с коэффициентом деления 2 в степени n?
4. Изобразите схему ДЧ. и временные диаграммы ДЧ с коэффициентом деления 8.
5. Сравните принцип построения схем, изображенных в п.п. 3-4.
6. Изобразите схему ДЧ и временные диаграммы ДЧ с коэффициентом деления 3. Отметьте её особенности по сравнению с предыдущими.
7. Изобразите схему ДЧ и временные диаграммы ДЧ с коэффициентом деления 5. Отметьте её особенности по сравнению с предыдущими.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.50/72

8. Изобразите схему ДЧ и временные диаграммы ДЧ с коэффициентом деления 7. Отметьте её особенности по сравнению с предыдущими

Самостоятельная работа №11: Проработка конспектов занятий по теме 2.4

Время выполнения 2 ч.

Цель занятия: Повторение и закрепление темы 2.4

При самостоятельной работе с конспектом необходимо:

1. Внимательно изучить изложенный в конспекте материал, если этого недостаточно, повторить его по учебнику;
2. Повторить принцип аналого-цифрового преобразования информации, процессы преобразования: дискретизация, квантование, кодирование. Погрешности квантования.
3. Ответить на контрольные вопросы.

Контрольные вопросы:

1. Что входит в процесс дискретизации?
2. Дайте понятие квантования?
3. Дайте определение частоты дискретизации.
4. В чём заключается процесс кодирования?
5. В чём причина шумов квантования? Как они проявляются?
6. Объясните понятие разрядности звуковой карты.
7. Единицы измерения разрядности звуковой карты. Назовите её величину.

Самостоятельная работа: Проработка конспектов занятий по теме 2.5. Подготовка к защите ПР.

Время выполнения 2 ч.

Цель занятия: Повторение и закрепление темы 2.5

При самостоятельной работе с конспектом необходимо:

1. Внимательно изучить изложенный в конспекте материал, если этого недостаточно, повторить его по учебнику.
2. Повторить типы полупроводниковых запоминающих устройства. Применение.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.51/72

3. Повторить принцип построения флэш памяти. Применеие.
- 4 Оперативные запоминающие устройства (ОЗУ). Организация оперативной памяти.
- 5.Постоянные запоминающие устройства (ПЗУ). Применение.
- 6.Перепрограммируемые ПЗУ. Применение.
- 7.Основные параметры полупроводниковых запоминающих устройств.
8. При самостоятельной подготовке к выполнению практического задания работы необходимо самостоятельно изучить методические указания к выполнению работы, в тетради для выполнения практических работ подготовить требуемые рисунки, схемы, таблицы. Продумать ход работы и ожидаемый результат
9. Ответить на контрольные вопросы.

Контрольные вопросы:

- 1.К какому типу памяти относится ОЗУ (оперативное запоминающее устройство)? В чём её особенности?
- 2.Дайте понятие регенерации. В чём её необходимость?
- 3.Назовите виды микросхем ОЗУ. Сравните их между собой.
- 4.Сравните надёжность микросхем ОЗУ в зависимости от степени интеграции.
5. Что обозначает разрядность шины памяти?
6. Дайте понятие банка памяти.
- 7.За счёт чего компьютер распознаёт модули памяти?
- 8.Что обозначают тактовая частота и пропускная способность модуля памяти? Как влияют на ЭВМ их величины?
- 11.Объясните назначение КЭШ в модулях памяти. Какие требования предъявляются к КЭШ? Чем достигается их выполнение?
- 12.Назовите преимущества и недостатки DirectRambus DRAM.
- 13.Из чего состоит ячейка памяти? Как обеспечивается доступ к ней?

Раздел 3 Основы микропроцессорных систем

Тема 3.1 Микропроцессорные системы

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.52/72

Самостоятельная работа: Проработка конспектов занятий Подготовка к защите ПР по разделу 3.

Время выполнения 8 ч.

Цель занятия: Повторение и закрепление темы 3.1 Раздела 3 Основы микропроцессорных систем

При самостоятельной работе с конспектом необходимо:

1. Внимательно изучить изложенный в конспекте материал, если этого недостаточно, повторить его по учебнику.
2. Повторить структуру базового микропроцессора
3. Повторить систему команд микропроцессора.
4. Повторить общие принципы построения процессора.
5. Повторить два подхода к построению процессора: использование принципа схемной логики и принципа программируемой логики.
9. Повторить способы адресации. Циклы и такты микропроцессора. Систему команд микропроцессора.
10. Повторить основные понятия об интерфейсе. Принципы организации интерфейсов. Классификация интерфейсов.
11. Повторить интерфейсы ввода-вывода.
12. Повторить периферийные устройства ЭВМ.
13. При самостоятельной подготовке к выполнению практических работ необходимо самостоятельно изучить методические указания к выполнению работ, в тетради для выполнения практических работ подготовить требуемые рисунки, схемы, таблицы. Продумать ход работы и ожидаемый результат
14. Ответить на контрольные вопросы.

Контрольные вопросы:

1. Дайте понятие драйвера в ПО компьютера. Чем является драйвер для подключаемого к ПК устройству?
2. Перечислите устройства ввода информации.
3. Перечислите устройства вывода информации.
4. Какие типы видеоплат используются в современных компьютерах?

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.53/72

5. Назовите главные компоненты и основные управляющие клавиши клавиатуры.
6. Перечислите основные компоненты видеосистемы компьютера.
7. Как формируется изображение на экране цветного монитора?
8. Как устроены жидкокристаллические мониторы? Проведите сравнение таких мониторов с мониторами, построенными на основе ЭЛТ.
9. Опишите работу матричных, лазерных и струйных принтеров.
10. Чем работа плоттера отличается от работы принтера?
11. Опишите способ передачи информации посредством модема.
12. Перечислите основные виды манипуляторов и опишите принципы их работы

Самостоятельная работа №12. Изучение набора микросхем системной платы компьютера

Цель работы: Закрепление темы 2.5. Полупроводниковые запоминающие устройства.

План работы: 1. Динамическая память с произвольным доступом – DRAM

2. История создания DRAM

3. Принцип действия DRAM

4. Характеристики памяти DRAM

5. Типы DRAM

6. Страничная память

7. SDR SDRAM — синхронная DRAM

8. Конструктивные исполнения памяти DRAM

9. Магниторезистивная оперативная память

Динамическая память с произвольным доступом – DRAM

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.54/72

DRAM (Dynamic random access memory, Динамическая память с произвольным доступом) — тип энергозависимой полупроводниковой памяти с произвольным доступом; DRAM широко используемая в качестве оперативной памяти современных компьютеров, а также в качестве постоянного хранилища информации в системах, требовательных к задержкам.

Физически DRAM состоит из ячеек, созданных в полупроводниковом материале, в каждой из которых можно хранить определённый объём данных, строку от 1 до 4 бит. Совокупность ячеек такой памяти образуют условный «прямоугольник», состоящий из определённого количества строк и столбцов. Один такой «прямоугольник» называется страницей, а совокупность страниц называется банком. Весь набор ячеек условно делится на несколько областей.

Как запоминающее устройство, DRAM представляет собой модуль памяти различных конструктивов, состоящий из электрической платы, на которой расположены микросхемы памяти и разъёма, необходимого для подключения модуля к материнской плате.

HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \\
".D0.98.D1.81.D1.82.D0.BE.D1.80.D0.B8.D1.8F" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \\
".D0.9F.D1.80.D0.B8.D0.BD.D1.86.D0.B8.D0.BF_.D0.B4.D0.B5.D0.B9.D1.81.D1.82.D0.B2.
D0.B8.D1.8F" HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \\
".D0.A0.D0.B5.D0.B3.D0.B5.D0.BD.D0.B5.D1.80.D0.B0.D1.86.D0.B8.D1.8F"
HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \\
".D0.A5.D0.B0.D1.80.D0.B0.D0.BA.D1.82.D0.B5.D1.80.D0.B8.D1.81.D1.82.D0.B8.D0.BA.D
0.B8_.D0.BF.D0.B0.D0.BC.D1.8F.D1.82.D0.B8_DRAM" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \ " ".D0.A2.D0.B8.D0.BF.D1.8B_DRAM" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \\
".D0.A1.D1.82.D1.80.D0.B0.D0.BD.D0.B8.D1.87.D0.BD.D0.B0.D1.8F_.D0.BF.D0.B0
.D0.BC.D1.8F.D1.82.D1.8C" HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \\
".D0.91.D1.8B.D1.81.D1.82.D1.80.D0.B0.D1.8F_.D1.81.D1.82.D1.80.D0.B0.D0.BD.
D0.B8.D1.87.D0.BD.D0.B0.D1.8F_.D0.BF.D0.B0.D0.BC.D1.8F.D1.82.D1.8C"
HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \\
"EDO_DRAM.C2.A0.E2.80.94_.D0.BF.D0.B0.D0.BC.D1.8F.D1.82.D1.8C_.D1.81_.D
1.83.D1.81.D0.BE.D0.B2.D0.B5.D1.80.D1.88.D0.B5.D0.BD.D1.81.D1.82.D0.B2.D0.
BE.D0.B2.D0.B0.D0.BD.D0.BD.D1.8B.D0.BC_.D0.B2.D1.8B.D1.85.D0.BE.D0.B4.D0
.BE.D0.BC" HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \\
"SDR_SDRAM.C2.A0.E2.80.94_.D1.81.D0.B8.D0.BD.D1.85.D1.80.D0.BE.D0.BD.D0
.BD.D0.B0.D1.8F_DRAM" HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \\
"Enhanced_SDRAM_.28ESDRAM.29" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \\
".D0.9F.D0.B0.D0.BA.D0.B5.D1.82.D0.BD.D0.B0.D1.8F_EDO_RAM" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \ "Video_RAM" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \ "DDR_SDRAM" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \\
"Direct_RDRAM_.D0.B8.D0.BB.D0.B8_Direct_Rambus_DRAM" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \ "DDR2_SDRAM" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \ "DDR3_SDRAM" HYPERLINK
"https://ru.wikipedia.org/wiki/DRAM" \\
".D0.9A.D0.BE.D0.BD.D1.81.D1.82.D1.80.D1.83.D0.BA.D1.82.D0.B8.D0.B2.D0.BD.D1.8B.

МО-11 02 03-ОП.04.СР	КМПК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.55/72

[D0.B5_.D0.B8.D1.81.D0.BF.D0.BE.D0.BB.D0.BD.D0.B5.D0.BD.D0.B8.D1.8F_.D0.BF.D0.B0.D0.BC.D1.8F.D1.82.D0.B8_DRAM](https://ru.wikipedia.org/wiki/DRAM)" HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \
[".D0.9C.D0.BE.D0.B4.D1.83.D0.BB.D0.B8_SIPP"](https://ru.wikipedia.org/wiki/DRAM) HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \
[".D0.9C.D0.BE.D0.B4.D1.83.D0.BB.D0.B8_SIMM"](https://ru.wikipedia.org/wiki/DRAM) HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \
[".D0.9C.D0.BE.D0.B4.D1.83.D0.BB.D0.B8_DIMM"](https://ru.wikipedia.org/wiki/DRAM) HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \
[".D0.9C.D0.BE.D0.B4.D1.83.D0.BB.D0.B8_SO-DIMM"](https://ru.wikipedia.org/wiki/DRAM) HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \
[".D0.9C.D0.BE.D0.B4.D1.83.D0.BB.D0.B8_RIMM"](https://ru.wikipedia.org/wiki/DRAM) HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \
[".D0.9F.D1.80.D0.BE.D0.B8.D0.B7.D0.B2.D0.BE.D0.B4.D0.B8.D1.82.D0.B5.D0.BB.D0.B8_.D0.BC.D0.B8.D0.BA.D1.80.D0.BE.D1.81.D1.85.D0.B5.D0.BC_.D0.B8_.D1.81.D0.B1.D0.BE.D1.80.D1.89.D0.B8.D0.BA.D0.B8_.D0.BC.D0.BE.D0.B4.D1.83.D0.BB.D0.B5.D0.B9"](https://ru.wikipedia.org/wiki/DRAM) HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \
[".D0.A1.D0.BC._.D1.82.D0.B0.D0.BA.D0.B6.D0.B5"](https://ru.wikipedia.org/wiki/DRAM) HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \
[".D0.9F.D1.80.D0.B8.D0.BC.D0.B5.D1.87.D0.B0.D0.BD.D0.B8.D1.8F"](https://ru.wikipedia.org/wiki/DRAM) HYPERLINK "https://ru.wikipedia.org/wiki/DRAM" \ ".D0.A1.D1.81.D1.8B.D0.BB.D0.BA.D0.B8"

История

Впервые динамическая память была реализована в дешифровальной машине «Aquarius», использовавшейся во время второй мировой войны в правительственной школе кодов и шифров в Блетчли-парк. Считываемые с бумажной ленты символы «запоминались в динамическом хранилище. ... Хранилище представляло собой блок конденсаторов, которые были либо заряжены либо разряжены, заряженный конденсатор соответствовал символу «X» (логической единице), разряженный — символу «.» (логическому нулю). Поскольку конденсаторы теряли заряд из-за утечки, на них периодически подавался импульс для подзарядки (отсюда термин динамическая)».

В 1966 году ученым Робертом Деннардом из исследовательского центра имени Томаса Уотсона компании IBM была изобретена современная DRAM память. В 1968 году Деннарду был выдан патент США под номером 3387286. Конденсаторы использовались в более ранних конструкциях памяти, таких как барабан компьютера Атанасова — Берри, трубках Уильямса и селектронах.

Принцип действия

Физически DRAM-память представляет собой набор запоминающих ячеек, которые состоят из конденсаторов и транзисторов, расположенных внутри полупроводниковых микросхем памяти.^[2]

При отсутствии подачи электроэнергии к памяти этого типа происходит разряд конденсаторов, и память опустошается (обнуляется). Для поддержания необходимого напряжения на обкладках конденсаторов ячеек и сохранения их содержимого, их необходимо периодически подзаряжать, прилагая к ним напряжения через коммутирующие транзисторные ключи. Такое динамическое

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.56/72

поддержание заряда конденсатора является основополагающим принципом работы памяти типа DRAM. Конденсаторы заряжают в случае, когда в «ячейку» записывается единичный бит, и разряжают в случае, когда в «ячейку» необходимо записать нулевой бит.

Важным элементом памяти этого типа является чувствительный усилитель-компаратор (англ. *sense amp*), подключенный к каждому из столбцов «прямоугольника». Он, реагируя на слабый поток электронов, устремившихся через открытые транзисторы с обкладок конденсаторов, считывает всю строку целиком. Именно строка является минимальной порцией обмена с динамической памятью, поэтому обмен данными с отдельно взятой ячейкой невозможен.

В отличие от быстрой, но дорогой статической памяти типа SRAM (англ. *static random access memory*), которая является конструктивно более сложным и более дорогим типом памяти и используется в основном в кэш-памяти, медленная, но дешёвая память DRAM изготавливается на основе конденсаторов небольшой ёмкости, которые быстро теряют заряд, поэтому информацию приходится обновлять через определённые промежутки времени во избежание потерь данных. Этот процесс называется регенерацией памяти. Он реализуется специальным контроллером, установленным на материнской плате или же на кристалле центрального процессора. На протяжении времени, называемого шагом регенерации, в DRAM перезаписывается целая строка ячеек, и через 8-64 мс обновляются все строки памяти.

Процесс регенерации памяти в классическом варианте существенно тормозит работу системы, поскольку в это время обмен данными с памятью невозможен. Регенерация, основанная на обычном переборе строк, в современных типах DRAM не применяется. Существует несколько более экономичных вариантов этого процесса — расширенный, пакетный, распределённый; наиболее экономичной является скрытая (тенивая) регенерация.

Среди новых технологий регенерации — PASR (англ. *Partial Array Self Refresh*), применяемый компанией Samsung в чипах памяти SDRAM с низким уровнем энергопотребления. Регенерация ячеек выполняется только в период ожидания в тех банках памяти, в которых имеются данные.

Параллельно с этой технологией реализуется метод TCSR (англ. *Temperature Compensated Self Refresh*), который предназначен для регулировки скорости процесса регенерации в зависимости от рабочей температуры.

Характеристики памяти DRAM

Основными характеристиками DRAM являются рабочая частота и тайминги.

При обращении к ячейке памяти контроллер памяти задаёт номер банка, номер страницы в нём, номер строки и номер столбца и на все эти запросы тратится

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.57/72

время, помимо этого довольно большой период уходит на открытие и закрытие банка после самой операции. На каждое действие требуется время, называемое таймингом.

Основными таймингами DRAM являются: задержка между подачей номера строки и номера столбца, называемая временем полного доступа (англ. RAS to CAS delay), задержка между подачей номера столбца и получением содержимого ячейки, называемая временем рабочего цикла (англ. CAS delay), задержка между чтением последней ячейки и подачей номера новой строки (англ. RAS precharge). Тайминги измеряются в наносекундах или тактах, и чем меньше величина этих таймингов, тем быстрее работает оперативная память.

Типы DRAM

На протяжении долгого времени разработчиками создавались различные типы памяти. Они обладали разными характеристиками, в них были использованы разные технические решения. Основной движущей силой развития памяти было развитие компьютеров и центральных процессоров. Постоянно требовалось увеличение быстродействия и объёма оперативной памяти.

Страничная память

Страничная память (англ. page mode DRAM, PM DRAM) являлась одним из первых типов выпускаемой компьютерной оперативной памяти. Память такого типа выпускалась в начале 1990-х годов, но с ростом производительности процессоров и ресурсоёмкости приложений требовалось увеличивать не только объём памяти, но и скорость её работы.

SDR SDRAM — синхронная DRAM

В связи с выпуском новых процессоров и постепенным увеличением частоты системной шины, стабильность работы памяти типа EDO DRAM стала заметно падать. Ей на смену пришла синхронная память — Single Data Rate Synchronous Dynamic Random Access Memory (SDR SDRAM). Новыми особенностями этого типа памяти являлись использование тактового генератора для синхронизации всех сигналов и использование конвейерной обработки информации. Также память надёжно работала на более высоких частотах системной шины (100 МГц и выше).

Если для FPM и EDO памяти указывается время чтения первой ячейки в цепочке (время доступа), то для SDRAM указывается время считывания последующих ячеек. Цепочка — несколько последовательных ячеек. На считывание первой ячейки уходит довольно много времени (60-70 нс) независимо от типа памяти, а вот время чтения последующих сильно зависит от типа. Рабочие частоты этого типа памяти могли равняться 66, 100 или 133 МГц, время полного доступа — 40 и 30 нс, а время рабочего цикла — 10 и 7,5 нс.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.58/72

С этим типом памяти применялась технология *Virtual Channel Memory (VCM)*. VCM использует архитектуру виртуального канала, позволяющую более гибко и эффективно передавать данные с использованием каналов регистра на чипе. Данная архитектура интегрирована в SDRAM. VCM, помимо высокой скорости передачи данных, была совместима с существующими SDRAM, что позволяло делать апгрейд системы без значительных затрат и модификаций. Это решение нашло поддержку у некоторых производителей чипсетов.

Enhanced SDRAM (ESDRAM)

Для преодоления некоторых проблем с задержкой сигнала, присущих стандартной DRAM-памяти, было решено встроить небольшое количество SRAM в чип, то есть создать на чипе кеш.

ESDRAM — это, по существу, SDRAM с небольшим количеством SRAM. При малой задержке и пакетной работе достигается частота до 200 МГц. Как и в случае внешней кеш-памяти, SRAM-кеш предназначен для хранения и выборки наиболее часто используемых данных. Отсюда и уменьшение времени доступа к данным медленной DRAM.

Одним из таких решений являлась ESDRAM от Ramtron International Corporation.

Пакетная EDO RAM

Пакетная память EDO RAM (англ. burst extended data output DRAM, BEDO DRAM) стала дешёвой альтернативой памяти типа SDRAM. Основанная на памяти EDO DRAM, её ключевой особенностью являлась технология поблочного чтения данных (блок данных читался за один такт), что сделало её работу быстрее, чем у памяти типа SDRAM. Однако невозможность работать на частоте системной шины более 66 МГц не позволила данному типу памяти стать популярным.

Video RAM

Специальный тип оперативной памяти — Video RAM (VRAM) — был разработан на основе памяти типа SDRAM для использования в видеоплатах. Он позволял обеспечить непрерывный поток данных в процессе обновления изображения, что было необходимо для реализации изображений высокого качества. На основе памяти типа VRAM, появилась спецификация памяти типа Windows RAM (WRAM), иногда её ошибочно связывают с операционными системами семейства Windows. Её производительность стала на 25 % выше, чем у оригинальной памяти типа SDRAM, благодаря некоторым техническим изменениям.

DDR SDRAM

По сравнению с обычной памятью типа SDR SDRAM, в памяти SDRAM с удвоенной скоростью передачи данных (англ. double data rate SDRAM, DDR SDRAM или SDRAM II) была вдвое увеличена пропускная способность. Первоначально память такого

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.59/72

типа применялась в видеоплатах, но позднее появилась поддержка DDR SDRAM со стороны чипсетов.

У всех предыдущих DRAM были разделены линии адреса, данных и управления, которые накладывают ограничения на скорость работы устройств. Для преодоления этого ограничения в некоторых технологических решениях все сигналы стали выполняться на одной шине. Двумя из таких решений являются технологии DRDRAM и SLDRAM. Они получили наибольшую популярность и заслуживают внимания. Стандарт SLDRAM является открытым и, подобно предыдущей технологии, SLDRAM использует оба перепада тактового сигнала. Что касается интерфейса, то SLDRAM перенимает протокол, названный SynchLink Interface и стремится работать на частоте 400 МГц.

Память DDR SDRAM работает на частотах в 100, 133, 166 и 200 МГц, её время полного доступа — 30 и 22,5 нс, а время рабочего цикла — 5, 3,75, 3 и 2,5 нс.

Так как частота синхронизации лежит в пределах от 100 до 200 МГц, а данные передаются по 2 бита на один синхроимпульс, как по фронту, так и по спаду тактового импульса, то эффективная частота передачи данных лежит в пределах от 200 до 400 МГц. Такие модули памяти обозначаются DDR200, DDR266, DDR333, DDR400.

Direct RDRAM или Direct Rambus DRAM

Тип памяти RDRAM является разработкой компании Rambus. Высокое быстродействие этой памяти достигается рядом особенностей, не встречающихся в других типах памяти. Первоначальная очень высокая стоимость памяти RDRAM привела к тому, что производители мощных компьютеров предпочли менее производительную, зато более дешёвую память DDR SDRAM. Рабочие частоты памяти — 400, 600 и 800 МГц, время полного доступа — до 30 нс, время рабочего цикла — до 2,5 нс.

DDR2 SDRAM

Конструктивно новый тип оперативной памяти DDR2 SDRAM был выпущен в 2004 году. Основываясь на технологии DDR SDRAM, этот тип памяти за счёт технических изменений показывает более высокое быстродействие и предназначен для использования на современных компьютерах. Память может работать с тактовой частотой шины 200, 266, 333, 337, 400, 533, 575 и 600 МГц. При этом эффективная частота передачи данных соответственно будет 400, 533, 667, 675, 800, 1066, 1150 и 1200 МГц. Некоторые производители модулей памяти помимо стандартных частот выпускают и образцы, работающие на нестандартных (промежуточных) частотах. Они предназначены для использования в разогнанных системах, где требуется запас по частоте. Время

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.60/72

полного доступа — 25, 11,25, 9, 7,5 нс и менее. Время рабочего цикла — от 5 до 1,67 нс.

DDR3 SDR

Этот тип памяти основан на технологиях DDR2 SDRAM со вдвое увеличенной частотой передачи данных по шине памяти. Отличается пониженным энергопотреблением по сравнению с предшественниками. Частота полосы пропускания лежит в пределах от 800 до 2400 МГц (рекорд частоты — более 3000 МГц), что обеспечивает большую пропускную способность по сравнению со всеми предшественниками.

Конструктивные исполнения памяти DRAM

Память типа DRAM конструктивно выполняют и в виде отдельных микросхем в корпусах типа DIP, SOIC, BGA и в виде модулей памяти типа SIPP, SIMM, DIMM, RIMM.

Первоначально микросхемы памяти выпускались в корпусах типа DIP (к примеру, серия K565PYxx), далее они стали производиться в более технологичных для применения в модулях корпусах.

На многих модулях SIMM и подавляющем числе DIMM устанавливалась SPD (Serial Presence Detect) — небольшая микросхема памяти EEPROM, хранящая параметры модуля (ёмкость, тип, рабочее напряжение, число банков, время доступа и т. п.), которые программно были доступны как оборудованию, в котором модуль был установлен (применялось для автонастройки параметров), так и пользователям и производителям.

Модули SIPP

Модули типа SIPP (Single In-line Pin Package) представляют собой прямоугольные платы с контактами в виде ряда маленьких штырьков. Этот тип конструктивного исполнения уже практически не используется, так как он далее был вытеснен модулями типа SIMM.

Модули SIMM

Модули типа SIMM (Single In-line Memory Module) представляют собой длинные прямоугольные платы с рядом контактных площадок вдоль одной из её сторон. Модули фиксируются в разъёме (сожете) подключения с помощью защёлок, путём установки платы под некоторым углом и нажатия на неё до приведения в вертикальное положение. Выпускались модули на 4, 8, 16, 32, 64, 128 Мбайт.

Наиболее распространены 30- и 72-контактные модули SIMM.

Модули DIMM

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.61/72

Модули типа DIMM (*Dual In-line Memory Module*) представляют собой длинные прямоугольные платы с рядами контактных площадок вдоль обеих её сторон, устанавливаемые в разъём подключения вертикально и фиксируемые по обоим торцам защёлками. Микросхемы памяти на них могут быть размещены как с одной, так и с обеих сторон платы.

Модули памяти типа SDRAM наиболее распространены в виде 168-контактных DIMM-модулей, памяти типа DDR SDRAM — в виде 184-контактных, а модули типа DDR2, DDR3 и FB-DIMM SDRAM — 240-контактных модулей.

Модули SO-DIMM

Для портативных и компактных устройств (материнских плат форм-фактора Mini-ITX, ноутбуков, планшетов и т. п.), а также принтеров, сетевой и телекоммуникационной техники и пр. широко применяются конструктивно уменьшенные модули DRAM (как SDRAM, так и DDR SDRAM) — SO-DIMM (*Small outline DIMM*) — аналоги модулей DIMM в компактном исполнении для экономии места.

Модули SO-DIMM существуют в 72-, 100-, 144-, 200- и 204-контактном исполнении.

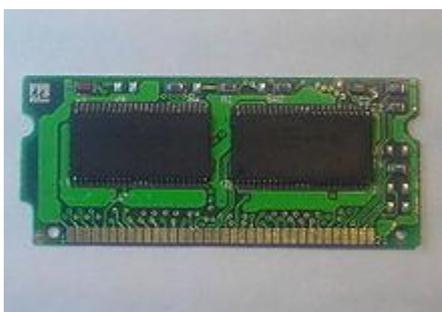
Модули RIMM

Модули типа RIMM (*Rambus In-line Memory Module*) менее распространены, в них выпускается память типа RDRAM. Они представлены 168- и 184-контактными разновидностями, причём на материнской плате такие модули обязательно должны устанавливаться только в парах, в противном случае в пустые разъёмы устанавливаются специальные модули-заглушки (это связано с особенностями конструкции таких модулей). Также существуют 242-контактные PC1066 RDRAM модули RIMM 4200, не совместимы с 184-контактными разъёмами, и уменьшенная версия RIMM — SO-RIMM, которые применяются в портативных устройствах.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.62/72



Различные корпуса DRAM. Сверху вниз: DIP, SIPP, SIMM (30-контактный), SIMM (72-контактный), DIMM (168-контактный), DIMM (184-контактный, DDR)



модуль SDRAM в 72-контактном корпусе SO-DIMM



модуль DDR2 в 204-контактном

Магниторезистивная оперативная память

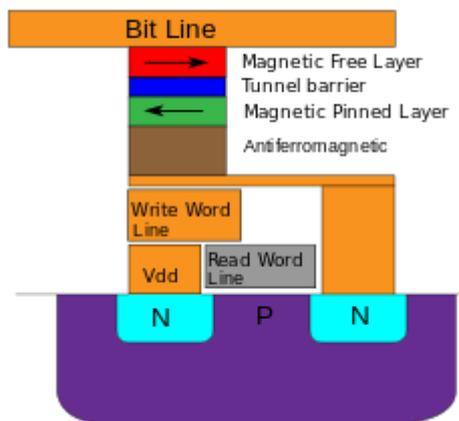
Магниторезистивная оперативная память (MRAM, а не электрических зарядов. — англ. *magnetoresistive random-access memory*) — запоминающее

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.63/72

устройство с произвольным доступом, которое хранит информацию при помощи магнитных моментов

Важнейшее преимущество этого типа памяти — энергонезависимость, то есть способность сохранять записанную информацию (например, программные контексты задач в системе и состояние всей системы) при отсутствии внешнего питания.

Технология магниторезистивной памяти разрабатывается с 1990-х годов. В сравнении с растущим объемом производства других типов компьютерной памяти, особенно флэш-памятью и памятью типа DRAM, она пока широко не представлена на рынке. Однако её сторонники верят, что благодаря ряду преимуществ, она в конечном счёте заменит все типы компьютерной памяти, и станет по-настоящему «универсальной» компьютерной памятью.



Упрощенная структура ячейки MRAM памяти

HYPERLINK

"

"

"

"

МО-11 02 03-ОП.04.СР	КМПК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.64/72

%BC%D1%8F%D1%82%D1%8C" \\
 ".D0.9F.D0.BB.D0.BE.D1.82.D0.BD.D0.BE.D1.81.D1.82.D1.8C_.D1.80.D0.B0.D0.B7
 .D0.BC.D0.B5.D1.89.D0.B5.D0.BD.D0.B8.D1.8F_.D1.8D.D0.BB.D0.B5.D0.BC.D0.B5
 .D0.BD.D1.82.D0.BE.D0.B2_.D0.B2_.D0.BC.D0.B8.D0.BA.D1.80.D0.BE.D1.81.D1.8
 5.D0.B5.D0.BC.D0.B5" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B3%D0%BD%D0%B8%D1%
 82%D0%BE%D1%80%D0%B5%D0%B7%D0%B8%D1%81%D1%82%D0%B8%D0
 %B2%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0
 %D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0
 %BC%D1%8F%D1%82%D1%8C" \\
 ".D0.AD.D0.BD.D0.B5.D1.80.D0.B3.D0.BE.D0.BF.D0.BE.D1.82.D1.80.D0.B5.D0.B1.
 D0.BB.D0.B5.D0.BD.D0.B8.D0.B5" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B3%D0%BD%D0%B8%D1%
 82%D0%BE%D1%80%D0%B5%D0%B7%D0%B8%D1%81%D1%82%D0%B8%D0
 %B2%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0
 %D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0
 %BC%D1%8F%D1%82%D1%8C" \\
 ".D0.91.D1.8B.D1.81.D1.82.D1.80.D0.BE.D0.B4.D0.B5.D0.B9.D1.81.D1.82.D0.B2.D
 0.B8.D0.B5" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B3%D0%BD%D0%B8%D1%
 82%D0%BE%D1%80%D0%B5%D0%B7%D0%B8%D1%81%D1%82%D0%B8%D0
 %B2%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0
 %D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0
 %BC%D1%8F%D1%82%D1%8C" \\
 ".D0.9E.D0.B1.D1.89.D0.B5.D0.B5_.D1.81.D1.80.D0.B0.D0.B2.D0.BD.D0.B5.D0.BD
 .D0.B8.D0.B5" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B3%D0%BD%D0%B8%D1%82%D0
 %BE%D1%80%D0%B5%D0%B7%D0%B8%D1%81%D1%82%D0%B8%D0%B2%D0%BD
 %D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D
 0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%
 8C" \\
 ".D0.98.D1.81.D1.82.D0.BE.D1.80.D0.B8.D1.8F" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B3%D0%BD%D0%B8%D1%82%D0
 %BE%D1%80%D0%B5%D0%B7%D0%B8%D1%81%D1%82%D0%B8%D0%B2%D0%BD
 %D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D
 0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%
 8C" \\
 ".D0.9F.D1.80.D0.B8.D0.BC.D0.B5.D0.BD.D0.B5.D0.BD.D0.B8.D0.B5" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B3%D0%BD%D0%B8%D1%82%D0
 %BE%D1%80%D0%B5%D0%B7%D0%B8%D1%81%D1%82%D0%B8%D0%B2%D0%BD
 %D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D
 0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%
 8C" \\
 ".D0.A1.D0.BC_.D1.82.D0.B0.D0.BA.D0.B6.D0.B5" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B3%D0%BD%D0%B8%D1%82%D0
 %BE%D1%80%D0%B5%D0%B7%D0%B8%D1%81%D1%82%D0%B8%D0%B2%D0%BD
 %D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D
 0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%
 8C" \\
 ".D0.A1.D1.81.D1.8B.D0.BB.D0.BA.D0.B8" **Описание**

В отличие от других типов запоминающих устройств, информация в магниторезистивной памяти хранится не в виде электрических зарядов или токов, а в магнитных элементах памяти. Магнитные элементы сформированы из двух ферромагнитных слоёв, разделенных тонким слоем диэлектрика. Один из

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.65/72

слоёв представляет собой постоянный магнит, намагниченный в определённом направлении, а намагниченность другого слоя изменяется

под действием внешнего поля. Устройство памяти организовано по принципу сетки, состоящей из отдельных «ячеек», содержащих элемент памяти и транзистор.

Считывание информации осуществляется измерением электрического сопротивления ячейки. Отдельная ячейка (обычно) выбирается подачей питания на соответствующий ей транзистор, который подаёт ток от источника питания через ячейку памяти на общую землю микросхемы. Вследствие эффекта туннельного магнитосопротивления, электрическое сопротивление ячейки изменяется в зависимости от взаимной ориентации намагниченностей в слоях. По величине протекающего тока, можно определить сопротивление данной ячейки, и как следствие, полярность перезаписываемого слоя. Обычно одинаковая ориентация намагниченности в слоях элемента интерпретируется как «0», в то время как противоположное направление намагниченности слоёв, характеризующееся более высоким сопротивлением — как «1».

Информацию можно записывать в ячейки, используя множество способов. В простейшем случае, каждая ячейка лежит между двумя линиями записи, размещёнными под прямым углом друг к другу, одна над, а другая под ячейкой. Когда ток проходит через них, в точке пересечения линий записи наводится магнитное поле, которое воздействует на перезаписываемый слой. Такой же способ записи использовался в памяти на магнитных сердечниках, которая использовалась в 1960х годах. Этот способ требует достаточно большого тока, необходимого для создания поля, и это делает их не очень подходящими для применения в портативных устройствах для которых важна малое потребление энергии, это один из основных недостатков MRAM. Кроме того, с уменьшением размера микросхем, придёт время, когда индуцированное поле перекроет соседние ячейки на маленькой площади, что приведёт к возможным ошибкам записи. Из-за этого в памяти MRAM данного типа необходимо использовать ячейки достаточно большого размера. Одним из экспериментальных решений этой проблемы было использование круглых доменов, читаемых и записываемых с помощью эффекта гигантского магнитного сопротивления, но исследования в этом направлении более не проводятся.

Другой подход, переключения режимов, использует многоступенчатую запись с модифицированной многослойной ячейкой. Ячейка модифицирована содержит в себе искусственный антиферромагнетик, где магнитная ориентация чередуется назад и вперёд через поверхность, с обоими прикрепленным и свободным слоями, составленными из многослойных стеков изолированных тонким «соединяющим слоем». Результирующие слои имеют только два стабильных состояния, которые могут быть переключены из одного в другое выбором времени тока записи в двух

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.66/72

линиях так одна немного задерживается, таким образом «поворачивая» поле. Любое напряжение меньшее, чем полный уровень записи фактически увеличивает его сопротивление для переключения. Это значит, что ячейки расположенные вдоль одной из линий записи не будут подвержены эффекту непреднамеренного перемагничивания, позволяя использовать меньшие размеры ячеек.

Новая технология, переноса спинового момента (*spin-torque-transfer-STT*) или переключение с помощью переноса спина, использует электроны с заданным состоянием спина («поляризованные»), прямо вращая области. Особенно, если электроны текут внутрь слоя, должно измениться их вращение, это будет способствовать вращению, будет перенесено на ближайший слой. Это уменьшает величину тока, необходимую для записи информации в ячейку памяти, и потребление только при чтении и записи становится примерно одинаковым. Технология STT должна решить проблемы с которыми «классическая» технология MRAM будет сталкиваться при увеличении плотности размещения ячеек памяти и соответствующего увеличения тока необходимого для записи. Поэтому технология STT будет актуальна при использовании технологического процесса 65 нм и менее. Нижняя сторона такая, в настоящее время, STT необходимо переключать больше тока через управляющий транзистор, чем обычной MRAM, требующей большой транзистор, и необходимо поддерживать когерентность вращения. В целом, несмотря на это, STT требует намного меньшего тока записи, чем обычная или переключательная MRAM.

Другими возможными путями развития технологии магниторезистивной памяти являются технология термического переключения (*TAS-Thermal Assisted Switching*) при которой во время процесса записи магнитный туннельный переход быстро нагревается (подобно PRAM) и в остальное время остается стабильным при более низкой температуре, а также технология вертикального транспорта (*VMRAM-vertical transport MRAM*) в которой ток проходящий через вертикальный столбцы меняет магнитную ориентацию, и такое геометрическое расположение ячеек памяти уменьшает проблему случайного перемагничивания и соответственно может увеличить возможную плотность размещения ячеек.

Сравнение с другими типами памяти

Плотность размещения элементов в микросхеме

Главным фактором, от которого зависит себестоимость производства микросхем памяти, это плотность размещения в ней отдельных ячеек. Чем меньше размер одной ячейки, тем большее их количество может быть размещено на одной микросхеме, и соответственно большее число микросхем может быть произведено за один раз из одной кремниевой пластины. Это улучшает выход годных изделий, и снижает стоимость производства микросхем.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.67/72

В памяти типа DRAM в качестве элементов памяти используются конденсаторы, проводники переносят ток к ним и от них, и управляющий транзистор — так называемая ячейка «1Т/1С». Конденсатор представляет собой две маленькие металлические пластинки, разделённые тонким слоем диэлектрика, он может быть изготовлен таким маленьким, как это позволяет сделать текущее развитие технологического процесса. Память DRAM имеет наивысшую плотность ячеек из всех доступных на сегодняшний день типов памяти. Это делает её наиболее дешёвой, и она используется в качестве основной оперативной памяти компьютеров.

Своей конструкцией ячейка памяти MRAM похожа на ячейку DRAM, хотя иногда в ней не используется транзистор для записи информации. Однако как упоминалось выше, память MRAM испытывает проблему полувыбора, из-за которой размер ячейки при использовании обычной технологии MRAM ограничен размером 180 нм и более. Используя технологию MRAM с переключением режимов можно достичь гораздо меньшего размера ячейки до того как эффект полувыбора станет проблемой — по видимому около 90 нм. Большинство современных микросхем DRAM памяти имеют размер ячейки 32 и 20 нм. Хотя это достаточно хорошие характеристики для внедрения в производство, есть перспективы в достижении магниторезистивной памятью размеров 65 нм, аналогично самым передовым устройствам памяти, для этого требуются использовать технологию STT.

Энергопотребление

Так как конденсаторы, используемые в микросхемах DRAM, со временем теряют свой заряд, микросхемы памяти, использующие их, должны периодически обновлять содержимое всех ячеек, считывая каждую ячейку и перезаписывая её содержимое. Это требует наличия постоянного источника питания, поэтому, как только питание компьютера отключается, память типа DRAM теряет всю хранимую информацию. Чем меньше размеры ячейки памяти, тем чаще необходимы циклы обновления, и в связи с этим энергопотребление растёт.

В отличие от DRAM, MRAM не требует постоянного обновления. Это означает не только то, что память сохраняет записанную в неё информацию при отключенном питании, но и то что при отсутствии операций чтения или записи, энергия вообще не потребляется. Хотя теоретически при чтении информации память MRAM должна потреблять больше энергии, чем DRAM, на

практике энергоёмкость чтения у них почти одинаковая. Тем не менее, процесс записи требует от 3—8 раз большей энергии чем при чтении, — эта энергия расходуется на изменение магнитного поля. Хотя точное количество сберегаемой энергии зависит от характера работы, — более частая запись потребует больше энергии, — в целом ожидается более низкое энергопотребление (до 99% меньше) в сравнении с DRAM. При применении

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.68/72

технологии STT MRAM потребление энергии при записи и чтении примерно одинаковое, и общее энергопотребление еще меньше.

Можно сравнить магниторезистивную память с еще одним конкурирующим типом памяти, с флэш-памятью. Как и магнито-резистивная память, флэш-память энергонезависима. Флэш-память не теряет информацию при отключении питания, что делает её очень удобной для замены жёстких дисков в портативных устройствах, таких как цифровые плееры или цифровые камеры. При чтении информации, флэш-память и MRAM почти одинаковы по уровню энергопотребления. Однако для записи информации в микросхемах флэш-памяти, необходим мощный импульс напряжения (около 10 В), который накапливается определенное время при накачке заряда, — для этого требуется много энергии и времени. Кроме этого импульс тока физически разрушает ячейки флэш-памяти, и информация в флэш-памяти может быть записана ограниченное число раз, прежде чем ячейка памяти выйдет из строя.

В отличие от флэш-памяти, микросхемам MRAM для записи энергии требуется ненамного больше, чем для чтения. Но при этом не надо увеличивать напряжение и не требуется накачка заряда. Это ведёт к более быстрым операциям, меньшему энергопотреблению, и к отсутствию ограничения срока службы.

Быстродействие

Быстродействие памяти типа DRAM ограничено скоростью, с которой заряд, хранящийся в ячейках, может быть слит (для чтения) или накоплен (для записи). Работа MRAM основана на измерении напряжений, что предпочтительнее, чем работа с токами, так как переходные процессы более быстрые. Исследователи IBM продемонстрировали устройства MRAM с временем доступа порядка 2 нс, что заметно лучше чем даже у самых совершенных DRAM, построенных на самых новых технологических процессах. Преимущества

по сравнению с Flash-памятью более значительные, — длительность чтения у них почти одинаковая, но длительность записи в MRAM в тысячи раз меньше.

Только одна современная технология памяти может конкурировать в быстродействии с магниторезистивной памятью. Это статическая память или SRAM. Ячейками SRAM памяти являются триггеры, которые хранят одно из двух состояний так долго, как долго поступает энергия. Каждый триггер состоит из нескольких транзисторов. Так как для транзисторов характерно очень низкое энергопотребление, длительность их переключения очень мала. Но поскольку ячейка памяти SRAM состоит из нескольких транзисторов, — обычно четырёх или шести, — её площадь больше, чем у ячейки памяти типа DRAM. Это делает память SRAM более дорогостоящей, поэтому она используется только в малых объемах, — в качестве особо быстродействующей памяти, как например кэш-память и процессорные регистры в большинстве современных моделей

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.69/72

центральных процессоров. Не следует забывать также, что и сейчас у процессоров делают несколько уровней кэш-памяти, имеющих разную скорость и объем.

Хотя магниторезистивная память не такая быстрая, как SRAM-память, она достаточно интересна и в этом качестве. Она обладает более высокой плотностью, и разработчики центральных процессоров могли бы в будущем выбирать для использования в качестве кэш-памяти между большим объемом менее быстрой MRAM-памяти и меньшим объемом более быстрой SRAM-памяти. Остается увидеть, как она продаётся, как сыграет в будущем.

Запоминающее устройство с произвольным доступом

Запоминающее устройство с произвольным доступом (сокращённо **ЗУПД**; также **Запоминающее устройство с произвольной выборкой**, сокращённо **ЗУПВ**; англ. *Random Access Memory*) — один из видов памяти компьютера, позволяющий единовременно получить доступ к любой ячейке (всегда за одно и то же время, вне зависимости от расположения) по её адресу на чтение или запись.

Это отличает данный вид памяти от устройств памяти первых компьютеров (последовательных компьютеров), созданных в конце 1940-х — начале 1950-х годов (EDSAC, EDVAC, UNIVAC), которые для хранения программы использовали разрядно-последовательную память- на ртутных линиях задержки, при которой разряды слова для последующей обработки в АЛУ поступали последовательно один за другим.

HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B5%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D1%81_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC" \

".D0.98.D1.81.D1.82.D0.BE.D1.80.D0.B8.D1.8F" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B5%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D1%81_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC" \

".D0.92.D0.B8.D0.B4.D1.8B_.D0.97.D0.A3.D0.9F.D0.92" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B5%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D1%81_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC" \

".D0.9D.D0.B0_.D0.BF.D0.BE.D0.BB.D1.83.D0.BF.D1.80.D0.BE.D0.B2.D0.BE.D0.B4.D0.BD.D0.B8.D0.BA.D0.B0.D1.85" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B5%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D1%81_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC" \

МО-11 02 03-ОП.04.СР	КМПК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.70/72

1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D1%81_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC" \l

".D0.9D.D0.B0_.D1.84.D0.B5.D1.80.D1.80.D0.BE.D0.BC.D0.B0.D0.B3.D0.BD.D0.B5.D1.82.D0.B8.D0.BA.D0.B0.D1.85" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B5%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D1%81_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC" \l

".D0.A1.D0.BC_.D1.82.D0.B0.D0.BA.D0.B6.D0.B5" HYPERLINK

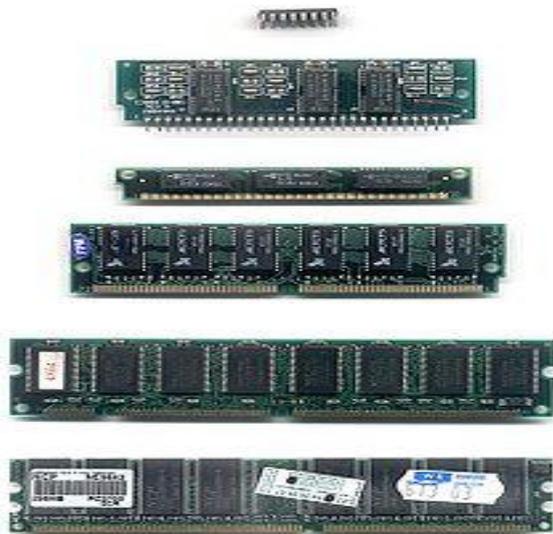
"https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B5%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D1%81_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC" \l

".D0.9F.D1.80.D0.B8.D0.BC.D0.B5.D1.87.D0.B0.D0.BD.D0.B8.D1.8F" HYPERLINK

"https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B5%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D1%81_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC" \l

".D0.A1.D1.81.D1.8B.D0.BB.D0.BA.D0.B8" **Виды ЗУПВ:**

На полупроводниках



Эволюционное развитие конструкции модулей памяти, используемых в качестве ОЗУ компьютеров. Сверху вниз: DIP, SIPP, SIMM 30 pin, SIMM 72 pin, DIMM, DDR DIMM

Полупроводниковая статическая (англ. Static Random Access Memory, SRAM) — ячейки **T-RAM**

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.71/72

T-RAM (англ. Thyristor RAM) — тиристорная память с произвольным доступом, новый вид оперативной памяти, сочетающий в себе сильные стороны DRAM и SRAM: высокую скорость работы и большой объём. Данная технология использует ячейки памяти, основанные на NDR, которые называются *Thin-Capacitively-Coupled-Thyristor*. T-RAM уходит от привычных дизайнов ячеек памяти: 1Т и 6Т, применяемых в DRAM и SRAM памяти. Благодаря этому, данная память является хорошо масштабируемой, и уже имеет плотность хранения данных в несколько раз превышающую её у SRAM памяти.

Вопросы для самопроверки:

1. Назвать типы запоминающих устройств.
2. Объяснить принцип действия DRAM
3. Перечислить основные характеристики памяти DRAM, дать им оценку.
4. Дать оценку страничной памяти
5. Дать оценку SDR SDRAM — синхронной D
6. Конструктивные исполнения памяти D
7. Понятие магниторезистивной оперативная память.

Самостоятельная работа №13. Проработка конспектов занятий Подготовка к защите ПР по разделу 3.

САМОСТОЯТЕЛЬНАЯ ВНЕАУДИТОРНАЯ РАБОТА С КОНСПЕКТОМ

Раздел 1 Арифметические и логические основы ЭВМ

Тема 1.1. Арифметические основы ЭВМ

Самостоятельная работа:

Проработка конспектов занятий по теме 1.1. Подготовка к защите ЛР и ПР

Время выполнения 4ч.

Цель занятия: Повторение и закрепление темы 1.1

При самостоятельной работе с конспектом необходимо:

1. Внимательно изучить изложенный в конспекте материал, если этого недостаточно, повторить его по учебнику;
2. Повторить формы представления чисел в цифровых устройствах.
3. Повторить кодирование чисел.

МО-11 02 03-ОП.04.СР	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА	С.72/72

4. Повторить выполнение арифметических операций над двоичными числами.

4. Ответить на контрольные вопросы:

5. При самостоятельной подготовке к выполнению практического задания и лабораторной работы необходимо самостоятельно изучить методические указания к выполнению работы, в тетради для выполнения практических работ подготовить требуемые рисунки, схемы, таблицы. Продумать ход работы и ожидаемый результат

Контрольные вопросы:

1. Почему в цифровой технике основной системой счисления является двоичная?
2. Как при кодировании чисел машинными кодами обозначается знак числа?
3. Чем обычный код числа отличается от модифицированного кода этого же числа?
4. Для чего используются модифицированные коды чисел?
5. Как получить прямой код числа?
6. Как получить обратный код числа?
7. Как получить дополнительный код числа?
8. Для чего в цифровой технике операция «вычитание» заменяется операцией «сложение»? Как это можно выполнить?
9. Как представить число в форме с плавающей запятой?
10. Как получить двоично-десятичный код десятичного числа?

Используемые источники литературы

Виды источников	Наименование рекомендуемых учебных изданий
Основные	Соколова, В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений : учебное пособие для вузов / В. В. Соколова. - Москва : Юрайт, 2022. - 175 on-line. - (Высшее образование).
	Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации [Электронный ресурс] : учебное пособие / А. П. Пятибратов ; ред. Л. П. Гудыно. - М. : КноРус, 2017
	Шевченко, В. П. Вычислительные системы, сети и телекоммуникации [Электронный ресурс] : учебное пособие / В. П. Шевченко. - М. : КноРус, 2017.
Дополнительные,	Литвинская, О. С. Основы теории передачи информации [Электронный ресурс] : учебное пособие / О. С. Литвинская, Н. И. Чернышев. - Москва : КноРус, 2017
	Т.Л.Партыка, И.И.Попов «Вычислительная техника» Москва: Форум, 2010г
Электронные образовательные ресурсы	1. ЭБС «Book.ru», https://www.book.ru 2. ЭБС «ЮРАЙТ» https://www.biblio-online.ru 3. ЭБС «Академия», https://www.academia-moscow.ru 4. Издательство «Лань», https://e.lanbook.com 5.Электронно-библиотечная система «Университетская библиотека онлайн», https://www.biblioclub.ru
Периодические издания	Журнал «Радио»; Журнал «Эксплуатация морского транспорта»; Журнал «Морские вести России»; Журнал «Морской Флот»; Журнал «Стандарты и качество». Научно-технический сборник российского морского регистра судоходства.