## РАЗРАБОТКА ИНТЕРАКТИВНОГО ПРИЛОЖЕНИЯ ПО ИЗУЧЕНИЮ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ КОМПЬЮТЕРА

Дьяконов И.И. Волкова Т.И., к.п.н., доцент

г. Бирск, ФГБОУ ВО Бирский филиал БашГУ

Современная электронная вычислительная машина - это сложнейший комплекс устройств, восхищающий своим технологическим совершенством и разнообразием физических принципов работы. Различные устройства машины представляют собой комбинацию большого числа элементовкирпичиков. Элементы компьютера выполняют хранение, преобразование и передачу двоичных переменных, а также ряд вспомогательных функций. Какой бы сложной ни была структура конкретной системы, ее можно изучать и описывать по частям, составляющим стандартные узлы и блоки. Разнообразие их, однако, не столь велико. Каждая машина состоит из ограниченного количества типов элементов, которые выполняют определенные логические функции, выражающие зависимость сигналов на выходах от сигналов на входах этих элементов. Для изучения логических элементов компьютера нами была разработана интерактивная программа, основные разделы которой представлена на рисунке 1.



Рис 1. Основные разделы программы

Имеется два основных модуля: «Теория» и «Тренажер». «Теория» включает в себя три раздела с теорией по темам и один раздел - контроль знаний:

«Базовые логические элементы» «Сумматор двоичных чисел» «RS-триггер как элемент памяти»

«Тестирование»

Раздел «Базовые логические элементы» включает в себя теоретический материал по темам логические элементы «И», «ИЛИ», «НЕ». Раздел «Сумматор двоичных чисел» содержит теоретический материал по темам: полусумматор, полный одноразрядный сумматор. Раздел «RS-триггер как элемент памяти» содержит теоретический материал по теме «RS-триггер». Теоретический материал разработан на основе пособий [1,2].

Раздел «Тестирование» несет функцию контроля знаний. Тест содержит вопросы, которые затрагивают все темы обучающей программы.

Раздел «Тренажер» включает в себя два тренажера:

«Синтез логических схем»

«Создание таблиц истинности»

Тренажер «Синтез логических схем» предоставляет пользователю изображение некоторой логической схемы, которую он должен проанализировать и записать ее структурную формулу.

Тренажер «Создание таблиц истинности» представляет собой таблицу истинности, в строках которой заданы значения аргументов А и В, а в столбцах - функции «не», «или», «и», «импликация», «эквивалентность». Пользователю необходимо верно выставить значения функции.

На рисунке 2 представлена структурная схема программы, раскрывающая взаимосвязь основных разделов и навигационную схему работы с программой:



Рис. 2 Навигационная схема программы

Интерфейс программы включает 7 форм. Рассмотрим их.

Form1 - главная форма, она имеет три контейнера Panel, которые выполняют роль меню (рис. 3). Каждая из панелей имеет кнопки, перенаправляющие в нужный раздел программы.



Рис. 3. Главная форма в режиме конструирования

1- panel1, 2 - panel2, 3 - panel3, 4 – buttonexit (выход из программы), 5 – buttonmenu (возвращение в главное меню)

FormOsnovy – это форма, которая содержит теоретический материал по теме «Базовые логические элементы». Все элементы формы находятся в контейнере tabControl с двумя вкладками. Для вывода текстовой информации использованы textbox (рис. 4).



Рис.4 FormSummator в режиме конструирования

1 - TabControl, 2 - textbox, 3 - groupbox, 4 - picturebox, 5 - button

Формы FormSummator и FormTrigger устроены идентично, поэтому мы не будем их рассматривать.

FormTest (рис. 5) – это форма, на которой проводится тестирование. Форма имеет 2 контейнера panel. На них отображается стартовое окно теста, возвращение в меню и результата тестирования. Результат выводится в textbox. В тесте использованы три типа вопросов: закрытого типа с одним верным ответом, закрытого типа с несколькими верными ответами и вопрос открытого типа. Элементы для каждого типа вопроса располагаются на отдельной панели.



Рис. 5. FormTest в режиме конструктора

1 -panelStart, 2-panelFinish, 3 - textBox, 4 - pictureBox, 5-button, 6-radiobutton

FormSintez (рис. 6) – это форма, используемая для тренажера «Синтез логических схем». Форма имеет два контейнера Panel. Panel1 – это начальное окно тренажера, оно имеет 2 кнопки button «Назад» и «Продолжить». Panel2 используется для вывода результата, на ней находится элемент textbox, который выводит результат тестирования, а так же кнопка button «Выход». Для ввода ответа используется textbox.



Рис. 6.FormSintez в режиме конструктора 1 - picturebox, 2 - panel2, 3,4- textbox, 5 - panel1

FormTableTrue – это форма (рис. 7) для тренажера «Создание таблиц истинности». Форма имеет 1 контейнер panel, выполняющий функцию стартового окна. На нем так же имеются 2 кнопки button «Назад» и «Продолжить». Основное окно тренажера имеет 40 элементов label для реализации тренажера.



Рис.7 FormTableTrue в режиме конструктора 1 panel1, 2 - label

Программа имеет три основных модуля: теория, тестирование и тренажер.

Основной проблемой реализации модуля теории было создание интерактивных схем. Принципы реализации схем дизъюнктора, конъюнктора и инвертора идентичны, поэтому рассмотрим только схему конъюнктора.

Для реализации движения электрического тока был использован таймер, выполняющий метод, который задавал движение точек по координатам.

```
private void timerAnimKon_Tick(object sender, EventArgs e)
```

```
{
    if (buttonKfalsetrue1.Text == "1" && buttonKfalsetrue2.Text == "1")
    {
        AnimKon();
        panelK.Refresh();
    }
}
```

Таймер запускается тогда, когда значение текста на кнопках управления становится равным 1. Метод AnimKon() задает движение точек с помощью условий, ниже код с условиями для одной точки.

int n = 10; if (xD1 == 30 && yD1 >= 50) yD1 += n; if (xD1 >= 30 && yD1 == 200) xD1 += n; if (xD1 == 230 && yD1 <= 200) yD1 -= n; if (xD1 <= 230 && yD1 == 50) xD1 -= n;</pre>

n – это колличество пикселей, на которое точка будет изменять значение с каждым событием Tick. xD1 и yD1 это глобальные переменные, они используются как текущие координаты точки.

Рассмотрим интерактивную схему работы полусумматора. Схема имеет 4 кнопки управления и соответственно 4 анимации. Анимация выполняется пошагово, используя элемент таймер. При загрузке формы выполняется метод RestartAnimation(), он выполняет начальную прорисовку схемы и вызывает принудительную перерисовку, так же он используется как первый шаг анимации для того, чтобы сбросить предыдущие значения. Для сохранения текущего шага анимации была использована глобальная переменная k с начальным значением - 1. Значения упраляющих кнопок так же сохраняются в глобальные переменные secondbit и firstbit типа bool. Ниже представлен фрагмент использования их в коде, на 3 шаге анимации.

```
if (k == 3)
{
    if (secondBit)
    {
        labelPolSummIN2.ForeColor = Color.Red;
        labelPolSummIN2.Text = "1";
    }
    else
    {
        labelPolSummIN2.ForeColor = Color.Gray;
        labelPolSummIN2.Text = "0";
    }
}
```

Рассмотрим схему RS-триггера. При загрузке формы происходит перерисовка схемы в начальное положение, используя метод StartImageTriggerAnimation. Для изображения электрической цепи были созданы две глобальные переменные типа Bitmap и Graphics. Для отображения imagetrigger на форме, был использован picturebox.

Bitmap imagetrigger; Graphics gr; imagetrigger = new Bitmap(groupBox1.Width, groupBox1.Height); gr = Graphics.FromImage(imagetrigger);

Данная схема выполняется в реальном времени, и перерисовка происходит сразу после изменения значения на входах S и R схемы. Для хранения сигналов с входов были созданы глобальные переменные типа bool set и reset, а так же для хранения выходящего значения триггера переменная result. Для работы анимации был сконструирован метод AnimationTrigger(). Он обрабатывает значения set и reset и перерисовывает форму. Работает он по событию Click.

Модуль тестирования был реализован следующим образом. Для удобного хранения вопросов, а так же их записи был создан новый класс QuestionforTest. В данном классе имеются свойства: Typequestions Textquest Picturequest Answer TrueAnswer. Эти свойства используются для хранения текста вопроса, хранения изображения, варианта ответа и верного ответа.

При запуске создается динамический массив, в который записываются все вопросы тестирования.

List<QuestionforTest>listquestions = new List<QuestionforTest>();

Во время тестирования вопросы выводятся случайным порядком. Реализация случайного порядка выполнялась так. Создан массив sp типа int размерностью, равной количеству вопросов в listquestions, в каждый элемент массива в цикле записывался его индекс. Создана глобальная переменная k, количеству элементов массива sp, переменная nq типа int, равная соответствующая номеру текущего вопроса в массиве listquestions. Именно из этого массива вопрос выводится пользователю, поэтому случайный порядок вывода вопросов зависит от переменной nq. Для задания значения этой переменной использовался метод RandomFunc(). В нем создана переменная r типа Random, nq принимает случайное значение элемента массива sp, индекс которого выбирается случайно в промежутке от 0 до k. Далее массив sp перезаписывается так, что все элементы массива sp, начиная с элемента под индексом равным nq, принимают значение, равное следующему элементу под индексом на 1 больше. После перезаписи массива sp переменная k уменьнается на 1. Ниже приведен код этого метода.

```
int[] sp;
int k;
private void RandomFunc()
{
    if (k != 0)
    {
        Random r = new Random();
        nq = sp[r.Next(0, k)];
        for (int j = nq; j < sp.Length - 1; j++)
            sp[j] = sp[j + 1];
    }
    else nq = sp[0];
    k--;
}
```

Модуль тренажера состоит из двух тренажеров «Синтез логических схем» и «Создание таблиц истинности».

Рассмотрим реализацию тренажера «Синтез логических схем». При запуске тренажера запускается метод ScaningImage() который загружает в массив imagelist типа List<Bitmap> изображения для вопросов, используя цикл for. Так же имеется массив adminotvety типа string для хранения верных ответов и массив userotvety типа string для хранения ответов пользователя. Создана глобальная переменная k типа int, значение которой используется для вывода изображения из массива imagelist, записи ответа пользователя в

массив userotvety. По окончании работы тренажера выводятся результаты, для это написан метод Results(). Он выполняет проверку ответов путем сравнения строк в массивах userotvety и adminotvety, выводит текст результата.

Рассмотрим реализацию тренажера «Создание таблиц истинности». При запуске тренажера запускается метод RandomLocation(), который выводит столбцы и строки таблицы в случаном порядке. Для изменения значения ячеек были созданы 24 обработчика события Click для label. Для простого доступа и хранения информации об элементах label были созданы 4 массива label1, label2, label3 и label4 типа Label. Для записи элементов в массив имеется метод ZZZ(), который запускается при открытии тренажера. При остановке работы тренажера результаты его работы в методе StopTrenager(). Ответы пользователя сравниваются с верными ответами, и количество верных ответов записывается в глобальную переменную schet типа int.

Проведенная экспериментальная апробация показала, что разработанное приложение устойчиво выполняет свои функции, что делает его применимым в процессе обучения.

## Литература

- 1. Информатика. Базовый курс. 2-е изд. [Текст]: Учебник/ Под ред. С.В. Симоновича. Спб.: Питер, 2007.
- 2. Н.Д.Угринович Информатика и ИКТ. 6-е изд. Москва: БИНОМ. Лаборатория знаний, 2012.