

Лабораторная работа № 1

Изучение принципов построения комбинационных логических схем в системе автоматизированного проектирования

1. Цель и задачи лабораторной работы

Цель: изучить принципы построения схем из цифровых логических элементов; научиться моделировать работу таких схем в программе система автоматизированного проектирования (САПР).

Задания:

1. Реализовать предложенную преподавателем таблицу истинности булевой функции. Проверить ее работоспособность и продемонстрировать преподавателю.
2. Построить сдвиговый регистр на 8 разрядов на триггерах.
3. Построить и изучить работу RS- и D-триггера.

Отчет должен содержать: таблицу истинности, булеву функцию этой таблицы, схему её реализации.

2. Краткие теоретические сведения

CircuitMaker 5 – мощная и удобная программа схематического сбора данных и инструмент схемотехнического моделирования, разработанная MicroCode Engineering Inc. Его возможности позволяют вывести любую электронную схему и создать модель работы инструментальных средств конструкции блоком управления процессором. Это позволяет произвести моделирование работы цифровой части введенной электросхемы. CircuitMaker 5 – это 32 разрядная программа, предназначенная для использования в среде Windows 95.

CircuitMaker 5 позволяет создавать множество проектов пользователей, при этом он позволяет обеспечить работу нескольких пользователей независимых друг от друга и не взаимодействующих друг с другом. Для этого каждый пользователь должен работать со своим собственным каталогом, куда будут помещены его библиотеки и т.д. В каталоге каждого пользователя необходимо разместить копию файла `cirmaker.dat`, этот файл содержит основную базу и настройки определяемые пользователем. Если пользователи не должны взаимодействовать друг с другом вообще, то необходимо поместить в их каталоги копию файлов `hotkeydb.dat` (этот файл влияет на зарегистрированные «горячие клавиши»), `user.lib` и `device.dat` (эти файлы содержат пользовательские библиотеки). Каталог `Models`, который содержит модели схем, тоже необходимо поместить в каталог пользователя. Для того чтобы CircuitMaker 5 мог обеспечить работу нескольких пользователей необходимо в свойствах ярлыка CircuitMaker 5 в панели Пуск дописать проект в пути его местонахождения «`c:\...\cirmaker.exe [проект]`», это допустит пользователя к диалоговому окну, где он сможет выбрать свой проект.

Для сохранения индивидуальных настроек каждого пользователя необходимо выбрать в меню Circuit Maker вкладку «File» в ней выбрать «Preferences», в открывшемся окне зайти в «Directories and Files...», появятся четыре адреса, которые определяют местонахождение файлов и каталога данного пользователя.

Описание функций основных иконок Circuit Maker



Создание нового файла.



Открытие имеющегося файла.



Сохранение текущего файла.



Печать текущего файла.



Используется, чтобы выбирать и перемещать элементы, щелкать выключателями, выбирать инструментальные средства в панели инструментальных средств Горячая клавиша «Alt+A», Путь Options\Cursor Tools\Arrow



Используется для соединения элементов электрическими цепями. Горячая клавиша «Alt+W», Путь Options\Cursor Tools\Wire



Используется для размещения текста в схемах. Горячая клавиша «Alt+T», Путь Options\Cursor Tools\Text



Используется для удаления элементов схемы, «+Shift» = Разделение объекта. Горячая клавиша «Alt+D», Путь Options\Cursor Tools\Delete



Используется для редактирования названия входов, выходов шины схемы. Горячая клавиша «Alt+N», Путь Options\Cursor Tools\Name Item



Используется для увеличения («+Shift» = уменьшения) масштаба схемы на экране. Горячая клавиша «Alt+Z», Путь Options\Cursor Tools\Zoom



Используется для поворота элемента на 90°. Горячая клавиша «Alt+R», Путь Edit\Rotate 90



Используется для зеркального отображения: при нажатии на элемент курсором он будет зеркально отображен относительно вертикальной оси элемента. Можно также нажать на клавишу «M» для зеркального отображения элемента выбранного в библиотеке. Горячая клавиша «Alt+M», Путь Edit\Mirror.



Кнопка выбора типа моделирования схемы: аналоговый/цифровой.



Кнопка возврата: в цифровом режиме моделирования схемы все сигналы будут установлены в исходное состояние.



Используется только при цифровом режиме моделирования схемы для проработки модели одного шага работы шага. Необходима для отладки схемы. Горячая клавиша «F9», Путь Options\Step Simulation.



Функционирует как в цифровом, так и в аналоговом режиме. Предназначена для моделирования работы схемы. Необходимо нажать на эту кнопку для начала моделирования. Затем нажать на стоп для конца моделирования, при этом, если не нажать на стоп, моделирование работы останавливается само



по завершении анализа схемы. Горячая клавиша «F9», Путь Options\Run simulation.



Используется для редактирования любого узла в схеме, а также для просмотра состояния схемы. Используется также для переключения переключателей с показанием их состояния во время моделирования работы схемы. Горячая клавиша «Alt+P», Путь Options\Cursor Tools\Probe.



Функционирует только в цифровом режиме, используется только в цифровом режиме, используется для определения состояния проводов: 1-красный, 0-синий, неопределен - зеленый. Обеспечивает удобства работы со схемой. Горячая клавиша «F11», Путь Options\Trace On.



Используется в цифровом режиме для просмотра окна цифрового вида волн. В аналоговом режиме: для просмотра графиков для данной схемы без повторного запуска моделирования для этой схемы. Горячая клавиша «Ctrl+W» Путь: Options\Arrow\Wire.



Brows Используется для входа в библиотеку элементов и выбора элемента.



Используется для создания нового элемента.



Вызов справки.

Описание основных элементов использующихся в лабораторной работе Полусумматор

Таблица 2.1

Таблица истинности

| В | А | Е | Со |
|---|---|---|----|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

полусумматора

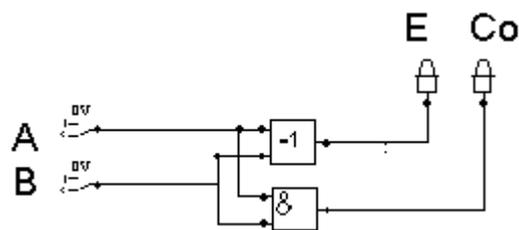


Рис. 2.1. Схема полусумматора

Состояние выхода Е полусумматора описывается булевым выражением $E = NA \cdot B + A \cdot NB$. Для реализации такой логической функции можно использовать 2 логических элемента И и 1 логический элемент ИЛИ. Для обеспечения требуемого (в соответствии с таблицей истинности) логического уровня на выходе суммы нам нужен только один 2-входовый логический элемент исключающее ИЛИ.

Логическая схема полусумматора, составленного из 2-входового логического элемента И и 2-входового логического элемента исключающее ИЛИ, показана на рисунке. Полусумматор осуществляет сложение только в разряде единиц. Для двоичного сложения в разрядах двоек, четверок, восьмерок и т.д. нужно использовать устройство, называемое полным сумматором.

Полный сумматор

Таблица 2.2

Таблица истинности полного сумматора

| C | B | A | E | C ₀ |
|---|---|---|---|----------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

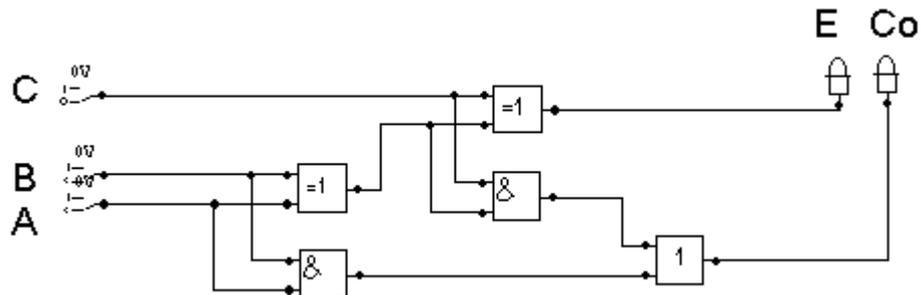


Рис. 2.2. Схема полного сумматора

Полный сумматор – трехвходовая схема. Сигналы на его выходах E и C₀ получаются в результате сложения трех входных сигналов (на входах A, B и C).

В одном из простейших способов формирования комбинационной логической структуры полного сумматора используются 2 полусумматора и логический элемент ИЛИ.

Полусумматоры и полные сумматоры обычно используются вместе: нужны 1 полусумматор (для сложения в разряде единиц) и 2 полных сумматора (для сложения в разрядах двоек и четверок). Полусумматоры и полные сумматоры - сравнительно простые схемы, однако при необходимости сложения многоразрядных двоичных чисел таких схем требуется довольно много.

RS-триггер

Таблица 2.3

Таблица истинности RS -триггера

| | S | R | C | Q | NQ |
|-----------------------|---|---|---|---|----|
| Запрещенное состояние | 0 | 0 | X | X | 1 |
| Установка 1 | 0 | 1 | 1 | 1 | 0 |
| Установка 0 | 1 | 0 | 1 | 0 | 1 |
| Хранение | 1 | 1 | 1 | Q | NQ |

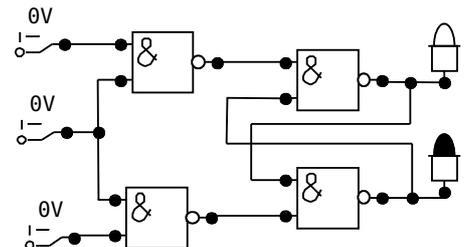


Рис. 2.3. Схема RS-триггера

Этот триггер имеет два выхода: Q и $\text{не-}Q$, каждый из которых является дополнением другого. Это означает что если $Q = 1$, то $\text{не-}Q = 0$ и наоборот. При $C = 1$ схема функционирует как обычный RS-триггер.

Предположим, что на S -вход триггера подан сигнал низкого уровня (0), а R -вход установлен в состояние высокого уровня (1). Предположим далее, что сигнал на выходе Q высокого уровня (1). Тогда вентиль 1 И-НЕ будет «включен», поскольку на обоих входах имеется сигнал низкого уровня. Сигнал с выхода Q поступает на один из входов вентиля И-НЕ, в связи с чем сигнал на его выходе будет иметь высокий уровень (логическую 1). Поскольку на другом входе вентиля 2 И-НЕ — также сигнал высокого уровня, выходной сигнал вентиля 2 И-НЕ будет низкого уровня (0). Это состояние, при котором $Q = 1$ и $\text{не-}Q = 0$, является стабильным. Если на R -входе сигнал переходит в 0, а на S -входе — в 1, то выходной сигнал вентиля 2 (Q) должен перейти в состояние высокого уровня. В свою очередь поступление этого сигнала на вход вентиля 2 изменит состояние его выхода на 0 (низкий уровень). Так как выход каждого вентиля соединен с входом другого, наличие одинаковых логических состояний на двух выходах триггера одновременно невозможно. В результате получим, что если S -вход установлен в состояние низкого уровня, то RS-триггер имеет выходное состояние $Q = 1$, $\text{не-}Q = 0$. Если R -вход установлен в 0, то выходы будут находиться в состоянии $Q = 0$, $\text{не-}Q = 1$. При работе с триггером нужно следить за тем, чтобы на обоих входах одновременно не было сигналов низкого уровня, поскольку в этом случае триггер будет неуправляемым. При $C = 0$ триггер отключён от управляющих S - и R -входов и находится в режиме хранения ранее полученной информации.

D-триггер

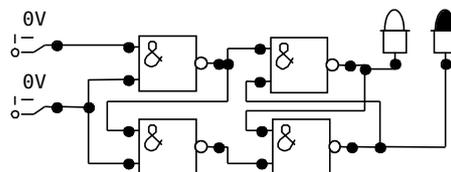


Рис. 2.4. Схема D-триггера

Триггер D-типа имеет ряд отличительных особенностей по сравнению с RS-триггером. D-триггер представляет собой «синхронизируемое» устройство. Это означает, что изменение выходных сигналов Q и $\text{не-}Q$ управляется входным сигналом синхронизации прямоугольной формы некоторой частоты. Логический сигнал,

поступающий на D-вход, передается на выход Q с некоторой задержкой после изменения сигнала синхронизации из 0 в 1 (от низкого к высокому уровню), который часто называют «передним фронтом». В некоторых случаях для управления устройством используется задний фронт, который определяет переход сигнала синхронизации из 1 в 0. Триггер D-типа применяют в тех случаях, когда необходимо либо передать разряд данных от D к Q, либо сохранить его в памяти.

D-триггеры позволяют спроектировать регистр, носящий название регистра типа «зашелка». Выходные сигналы какого-либо цифрового устройства постоянно поступают на входы регистра-«зашелки», но заносятся в него лишь при подаче импульса синхронизации одновременно на все триггеры регистра. Дальнейшее изменение сигналов на входах регистра-«зашелки» не оказывает влияния на его состояние до прихода следующего сигнала синхронизации. При определении количества триггеров регистра-«зашелки» исходят из разрядности фиксируемых слов. Так, для «зашелкивания» восьмиразрядных слов регистр-«зашелка» должен состоять из восьми триггеров.

Сдвиговые регистры

8-разрядный сдвиговый регистр, построенный на D-триггерах

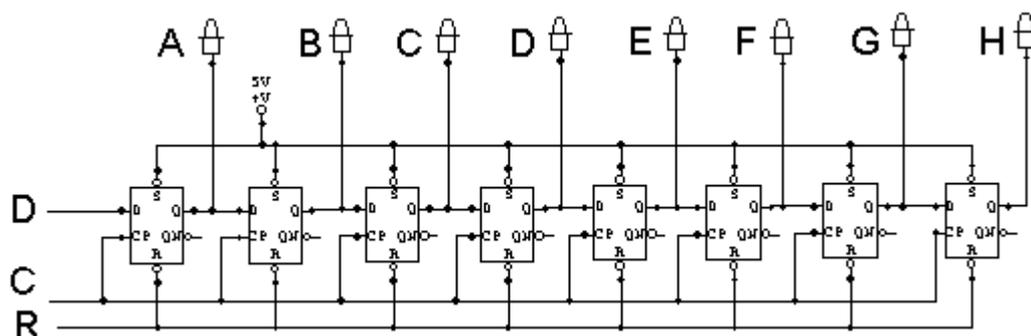


Рис. 2.5. Схема 8-разрядного сдвигового регистра, построенного на D-триггерах

Проверка правильности работы сдвигового регистра, построенного на D-триггерах

1. Операция. Устанавливаем 0 на входе очистки и затем возвращаем этот вход к 1.

Результат. На выходном индикаторе: 0000.

Вывод. Цепь очистки регистра исправна.

2. Операция. Устанавливаем 1 на информационном входе ($D = 1$). Подаем одиночный импульс на синхронизирующий вход регистра от генератора цифровых импульсов.

Результат. На выходном индикаторе: 1000.

Вывод. Загрузка 1 в триггер ТА осуществляется должным образом.

3. Операция. $D = 1$

Подаем одиночный импульс на синхронизирующий вход регистра от генератора цифровых импульсов.

Результат. На выходном индикаторе: 1100.

Вывод. Загрузка «единиц» в триггеры ТА и ТВ осуществляется должным образом.

4. Операция. $D = 1$.

Подаем одиночный импульс на синхронизирующий вход регистра от генератора цифровых импульсов.

Результат. На выходном индикаторе: 1110.

Вывод. Загрузка «единиц» в триггеры ТА, ТВ и ТС осуществляется должным образом.

5. Операция. $D = 1$

Подаем одиночный импульс на синхронизирующий вход регистра от генератора цифровых импульсов.

Результат. На выходном индикаторе: 1110.

Вывод. Неисправность следует искать вблизи или в самом триггере TD, так как он не загружается «единицей».

6. Операция. Логическим пробником проверяем вход D триггера TD (действительно ли $D = 1$).

Результат. $D = 1$ для триггера TD.

Вывод. На информационном входе D триггера TD действует требуемый **ВЫСОКИЙ** уровень сигнала.

7. Операция. Подаем одиночный импульс от генератора цифровых импульсов на синхронизирующий вход триггера TD.

Результат. На выходном индикаторе: 1110.

Вывод. Информационный сигнал с входа D триггера TD не переносится на его выход Q с приходом тактового импульса.

8. Операция. Логическим пробником проверяется выход Q триггера TD.

Результат. Отсутствует свечение как индикатора **ВЫСОКОГО**, так и индикатора **НИЗКОГО** уровня сигнала.

Вывод. На выходе Q триггера TD имеет место плавающий потенциал в неопределенной области между **ВЫСОКИМ** и **НИЗКИМ** уровнями сигнала. По-видимому, неисправен триггер TD.

9. Операция. Повторяем всю последовательность проверок, начиная с операции 1.

Результат. Все триггеры загружаются «единицами» и «нулями».

Вывод. Регистр сдвига теперь исправен.

Изучение принципов булевой алгебры на примере построения простейшей комбинационной схемы

Таблица 2.4

Таблица истинности

| A1 | A2 | A3 | F1 |
|----|----|----|----|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |

Рассмотрим комбинации двоичных сигналов, которые на выходе дают высокий сигнал (логическую 1). Составим выражения, которые дают эти комбинации. Для

этого сигналы на входе, представленные низким уровнем (логическим 0), инвертируем в противоположные (логическую 1) и связываем входные инвертированные и не инвертированные сигналы уже высокого уровня (логическая 1) логическим элементом И. Затем полученные выражения связываем логическим элементом ИЛИ.

Иными словами, процедуру построения логического выражения по таблице истинности можно представить в виде следующего алгоритма:

- 1.выбираем строки, в которых значение функции равно 1;
- 2.для них составляем термы, т.е. объединённые с помощью логического умножения исходные переменные, причём переменные, имеющие для рассматриваемой строки значение логического нуля, записываем в инверсной форме, а логической 1 – в прямой;
- 3.объединяем полученные термы с помощью логического сложения.

$$F^1 = \overline{A^1} * \overline{A^2} * A^3 + A^1 * \overline{A^2} * \overline{A^3} = A^2 * (\overline{A^1} * A^3 + A^1 * \overline{A^3})$$

На основе булевых выражений строим простейшую комбинационную схему. Для этого в программе CircuitMaker 5 выбираем «browse» в «hotkeys», появится диалоговое окно «Device Selection», где по принципу класс/подкласс расположены виды устройств, а также горячие клавиши их вызова. Для инверторов, логических элементов И и ИЛИ легче выбрать «hotkeys», далее inverter, AND или OR соответственно. Для входного ключа следует выбрать в «hotkeys» ..., а для лампы индикатора выходного сигнала – в «hotkeys» «browse», далее... .Нужные элементы размещаем. Далее необходимо их соединить. Для этого обычно используется кнопка «Wire» или иконка «+» на панели инструментов.

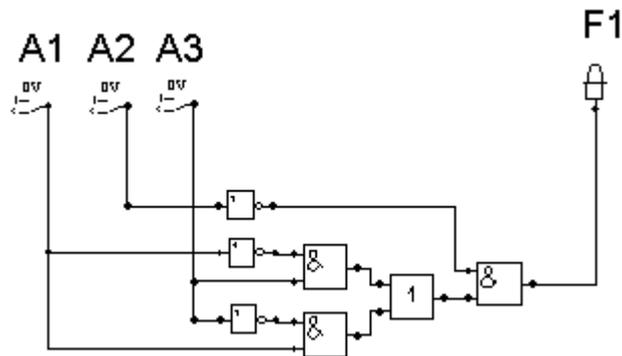


Рис. 2.6. Комбинационная схема к таблице истинности

3. Ход работы

1. Написать формулы и построить комбинационную схему, реализующую таблицу истинности.

Варианты заданий

Вариант № 1

| A 1 | A 2 | A 3 | F 1 | F 2 | F 3 | F 4 |
|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 |

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

Вариант № 2

| A 1 | A 2 | A 3 | F 1 | F 2 | F 3 | F 4 |
|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

Вариант № 3

| A 1 | A 2 | A 3 | F 1 | F 2 | F 3 | F 4 |
|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |

Вариант № 4

| A 1 | A 2 | A 3 | F 1 | F 2 | F 3 | F 4 |
|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 |

Вариант № 5

| A 1 | A 2 | A 3 | F 1 | F 2 | F 3 | F 4 |
|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |

2. Построить RS-триггер и изучить принцип его работы.

3. Построить сдвиговый регистр и ввести в него указанное преподавателем число.

4. Контрольные вопросы

1. Для чего используется полусумматор и какая его таблица истинности?
2. Какие элементы используются для построения полного сумматора?
3. Опишите принцип работы RS-триггера.
4. Какими функциональными возможностями обладает сдвиговый регистр?
5. Каким образом по таблице истинности строятся логические выражения?
6. Как по логическим выражениям построить комбинационную схему?
7. С помощью каких элементов можно задать входные логические уровни, а также как они могут быть проиндицированы?

