

«СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ НА БАЗЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ».

Цель работы. Получение практических навыков разработки функциональных узлов комбинационного типа.

Методические указания.

а) Работа выполняется по индивидуальному заданию. Варианты заданий приведены в табл.1, номер конкретного задания указывается преподавателем.

Заданием предусматривается проектирование двух схем устройства - для полностью определенной таблицей истинности (не обращая внимания на звездочки, которыми помечены некоторые значения выходного сигнала) и для недоопределенной таблицей истинности (значения функции, помеченные звездочками, могут быть скорректированы по желанию разработчика).

б) Выполнение работы заключается в разработке логической схемы с последующей ее проверкой путем электронного моделирования с использованием программы Electronic Workbench. Проверка спроектированной схемы осуществляется путем тестирования ее электронной модели.

Варианты заданий.

N ²	X ₁	X ₂	X ₃	X ₄	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	Y ₈	Y ₉	Y ₁₀	Y ₁₁	Y ₁₂	Y ₁₃	Y ₁₄	Y ₁₅	Y ₁₆	Y ₁₇	Y ₁₈	Y ₁₉	Y ₂₀	Y ₂₁	Y ₂₂	Y ₂₃	Y ₂₄	Y ₂₅	
0	0	0	0	0	1	1	1	0	1	1	1	1	1	0	0	1	0	0	1	1	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	1	1	0	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	1	0	0	0	1	0	1	1	1	1	0
3	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1	1	1	0	0	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	0	1	0	0	1	1	1	0	0	1	1	1	1	1	1	1	0	0	0
5	0	1	0	1	0	1	1	0	0	0	0	1	1	0	0	1	0	1	1	0	1	0	0	1	1	1	1	0	1	1
6	0	1	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	0	1
7	0	1	1	1	1	1	1	0	1	0	0	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1
8	1	0	0	0	1	1	1	1	1	0	0	1	0	1	0	1	1	1	1	0	0	1	0	1	0	1	1	0	1	0
9	1	0	0	1	0	1	1	1	1	0	1	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	0	1
10	1	0	1	0	1	0	1	1	0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1	1	0	0	0
11	1	0	1	1	0	1	1	1	0	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1	0
12	1	1	0	0	1	1	0	1	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	0	1
13	1	1	0	1	1	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	0	1
14	1	1	1	0	1	0	1	1	0	1	1	1	1	1	1	0	1	0	0	1	0	1	0	0	0	1	1	0	0	0
15	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	1	0	1	1	0	1	0	1	0	0	0	0	0	0

Краткие сведения из теории

Проектирование логических устройств

Логические устройства – устройства, входные и выходные сигналы которых являются логическими, т.е. предназначенными для передачи информации, исчерпываемой только двух возможными значениями «да» и «нет» («логический 0 или «логическая 1»). Примером подобного устройства является мажоритарный элемент «2 из 3» - устройство, значение выходного сигнала которого соответствует большинству значений из трех его входных логических сигналов. Простейшим видом логических устройств являются устройства *комбинационного типа* – устройства без памяти, т.е. устройства, выходной сигнал которых определяется только сигналами на его входе в данный момент, но не зависит от предыдущего состояния самого устройства. Примером подобного устройства и является мажоритарный элемент.

Проектирование логического устройства осуществляется в несколько этапов, на первом из которых создается логическая модель, отражающая выбранный алгоритм для построения требуемого устройства. Этот этап основан только на теоретических положениях булевой алгебры.

При описании логических устройств используют буквенное обозначение сигналов (например, X_i - для входных сигналов, Y – для выходного). Тогда работа логического устройства комбинационного типа может быть представлена в виде $Y=F(X_1, X_2, \dots, X_n)$, где n - число входных сигналов (для мажоритарного элемента $n=3$). Функция $F()$, устанавливающая связь между логическими сигналами на вхо-

X_1	X_2	X_3	F	
0	0	0		де и выходе, называется логической функцией (переключа-
0	0	0		тельной функцией), при анализе записывается в виде струк-
0	1	0		турной формулы. В общем случае, отыскание структурной
0	1	1		формулы осуществляется на основе <i>таблицы истинности</i>
1	0	0		(<i>таблицы функционирования</i> , полностью определяющей тре-
1	0	1		буемое значение выходного сигнала для каждой возможной
1	1	1		комбинации входных сигналов (набора). Например, для ма-
1	1	1		жоритарного элемента таблица истинности имеет вид.

Существует две формы записи структурной формулы дизъюнктивная нормальная (ДНФ) и конъюнктивная нормальная (КНФ) формы.

ДНФ – форма представления логической функции в виде дизъюнкции (логического сложения) элементарных конъюнкций (логических произведений) аргументов или их инверсий, при которых выходной сигнал принимает единичное значение. Причем в конъюнкцию каждый аргумент или его инверсия могут входить только один раз. Если каждая конъюнкция содержит все входные переменные или их инверсию, то такая форма записи называется совершенной (СДНФ) и является единственной для конкретной логической функции. В то же время логическая функция допускает существование большого числа ДНФ, получаемых путем упрощения СДНФ, одна из которых и используется при построении логических устройств. Для

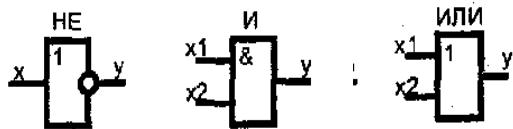
мажоритарного элемента СДНФ имеет вид

$$F = \overline{X_1} \wedge X_2 \wedge X_3 \vee X_1 \wedge \overline{X_2} \wedge X_3 \vee X_1 \wedge X_2 \wedge \overline{X_3} \vee X_1 \wedge X_2 \wedge X_3$$

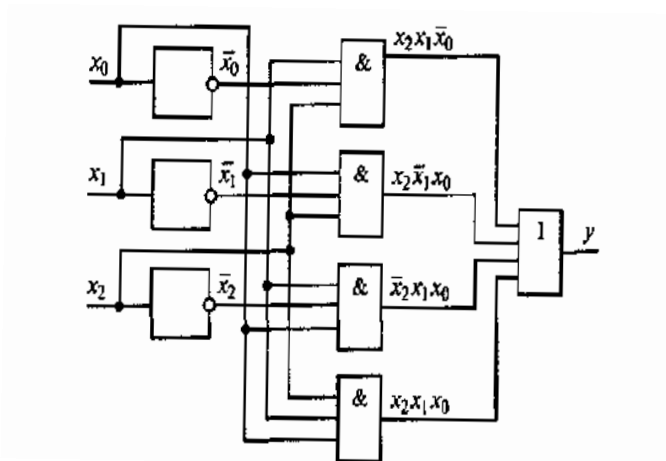
КНФ – форма записи логической функции в виде конъюнкции элементарных дизъюнкций и тоже имеет единственную СДНФ и допускает большое число представлений.

СКНФ получается из таблицы истинности в виде логического произведения и содержит столько сомножителей, сколько нулей имеется среди значений логической функции в таблице истинности. Каждая из элементарных дизъюнкций при этом составляется с учетом значений входных переменных в соответствующем наборе, обеспечивающим нулевое значение на выходе. Для мажоритарного элемента СКНФ имеет вид $F = (X_1 \vee X_2 \vee X_3) \wedge (X_1 \vee X_2 \vee \overline{X_3}) \wedge (X_1 \vee \overline{X_2} \vee X_3) \wedge (\overline{X_1} \vee X_2 \vee X_3)$

Полученная структурная формула позволяет непосредственно построить логическую модель проектируемого устройства. Причем в силу множества возможных логических функций при их реализации используются элементарные логические функции. Вводится понятие *функционально полного набора* логических элементов- простейших логических функций, позволяющих реализовать любую сколь угодно сложную функцию. Наиболее известным подобным набором является набор элементов «И», «ИЛИ», «НЕ». Ниже приведены их графические изображения.



В результате, представляя логическое устройство как совокупность элементарных устройств, осуществляющих последовательное преобразование над входными переменными, оказывается возможным реализовать любое логическое устройство. В частности, пример устройства, реализующего СДНФ мажоритарного элемента имеет вид



Минимизация логических функций

Логическое устройство проектируется как устройство, реализующее выбранную структурную формулу. В силу ряда причин наиболее распространенной является ее представление в виде ДНФ, допускающее многовариантность. Очевидно, что в результате и логическое устройство допускает множество вариантов его реализации. Отыскание оптимального варианта построения логического устройства, таким образом, решающим образом определяется выбранным вариантом представления ДНФ

В большинстве случаев задача сводится к отысканию наиболее простой формы записи структурной формулы - ее *минимизации*, предполагающей в дальнейшем наиболее простую реализацию устройства. Существует ряд методов решения этой задачи. Наиболее естественной является упрощение на основе аналитического анализа с использованием правил булевой алгебры.

В сложных логических выражениях устанавливается определенный порядок выполнения операций *операции в скобках*, «НЕ», «И», «ИЛИ». Во вложенных скобках сначала выполняются операции во внутренних скобках.

Над логическими выражениями производят тождественные преобразования с использованием законов булевой алгебры.

Функции являются эквивалентными, если они принимают одинаковые значения при одних и тех же наборах входных сигналов.

Две эквивалентные функции, приравненные друг к другу, называются тождеством.

Булева алгебра, оперирующая лишь с логическими сигналами и операциями, характеризуется *принципом двойственности*.

Если в правилах выполнения логического сложения все переменные и саму функцию заменить на инверсные значения, а операцию сложения заменить на операцию умножения, то равенство в правилах выполнения логических операций сохраняется

$$f = \overline{xy} \vee x\overline{y} \qquad \overline{f} = (\overline{\overline{x} \vee \overline{y}}) \vee \overline{y}$$

Осуществление тождественных преобразований логических функций базируется на специфических теоремах булевой алгебры.

Первые девять теорем относятся к функциям одной переменной, остальные к функциям двух и большего числа переменных. Все теоремы легко доказать путем перебора всех возможных комбинаций переменных.

$$\begin{array}{llll} 1 & x \vee 0 = x & 2 & x \vee 1 = 1 \\ 3 & x \vee x = x & 4 & x + \overline{x} = 1 \\ 5 & x \wedge 0 = 0 & 6 & x \wedge 1 = x \\ 7 & x \wedge x = x & 8 & x \wedge \overline{x} = 0 \\ 9 & \overline{\overline{x}} = x \end{array}$$

Следующие теоремы, содержащие две и большее число переменных, имеет каждая два варианта: для логического сложения и для логического умножения.

$$10. \quad x \vee y = y \vee x \qquad x \wedge y = y \wedge x \quad (\text{«переместительный закон»}).$$

$$11. \quad (\overline{x} \vee \overline{y}) \vee z = \overline{x \vee y} \vee z \qquad (\overline{x} \wedge \overline{y}) \wedge z = \overline{x \wedge y} \wedge z \quad (\text{«сочетательный закон»}).$$

$$12. \quad x \wedge (\overline{x} \vee z) = x \wedge z \qquad x \vee (\overline{x} \wedge z) = (x \vee \overline{x}) \wedge z$$

$$13. \quad x \vee x \wedge y = x \qquad x \wedge (\overline{x} \vee y) = \overline{x} \vee y \quad (\text{«закон поглощения»}).$$

14. $x \vee \bar{x} \wedge y = x \wedge y$ $x \wedge (\vee y) \stackrel{=}{=} x \wedge y$ («закон свертки»)
15. $x \wedge y \vee x \wedge \bar{y} = x$ $(\wedge y) \wedge (\vee \bar{y}) \stackrel{=}{=} x$ («закон склеивания»)
16. $\overline{x \vee y \vee z \vee \dots} = \bar{x} \wedge \bar{y} \wedge \bar{z} \wedge \dots$ $\overline{x \wedge y \wedge z \wedge \dots} = \bar{x} \vee \bar{y} \vee \bar{z} \vee \dots$ (теорема де Моргана)

Для переменной и функции справедлива теорема

17. $x \vee F(\langle y, z, \dots \rangle) \stackrel{=}{=} x \vee F(\langle y, z, \dots \rangle)$ $x \wedge F(\langle y, z, \dots \rangle) \stackrel{=}{=} x \wedge F(\langle y, z, \dots \rangle)$

Справедливость всех приведенных законов может быть доказана прямой подстановкой.

Минимизацию логических выражений для упрощения структурной формулы можно проводить непосредственно с использованием законов булевой алгебры.

Предварительно введем следующие определения:

1. Два произведения логических переменных называются соседними, если они отличаются значением только одной переменной. Например, произведения $AB\bar{C}$ и ABC являются соседними, т.к. переменные А и В входят в оба произведения без знаков инверсии, а переменная С входит в одно произведение с инверсией, в другое – без инверсии.
2. Количество переменных в произведении называют его рангом.

С использованием этих определений эти правила формулируются следующим образом.

1. Два соседних слагаемых К-го ранга можно заменить одним слагаемым (К-1)-го ранга. Действительно, $A \wedge B \wedge \bar{C} \vee A \wedge B \wedge C = A \wedge B \wedge (C \vee \bar{C}) = A \wedge B$.

2. Поскольку сумма любого числа одинаковых слагаемых равна этому слагаемому, то из m одинаковых слагаемых (m-1) слагаемое являются лишними (избыточными) и их можно отбросить. И, наоборот, что к логической сумме можно добавить еще сколько угодно слагаемых, повторяющих уже в этой сумме содержащихся.

3. Любой несовершенный дизъюнктивный вид записи структурной формулы можно расширить до совершенного вида путем умножения слагаемых на выражение $(\vee \bar{x})$. Например,

$$F = \overline{A}BC + \overline{A}B\overline{C} + AB = \overline{A}BC + \overline{A}B\overline{C} + AB(C + \overline{C}) = \overline{A}BC + \overline{A}B\overline{C} + ABC + AB\overline{C}$$

Если функция, представленная в СНДФ содержит 2^n слагаемых, где n - количество переменных, то она тождественно равна единице. И обратное: если функция тождественно равна единице $F=1$, то она имеет все 2^n слагаемых и на каждом из 2^n наборов функция принимает значение 1.

4. Если функция F , записанная в СНДФ содержит m слагаемых, то ее отрицание \overline{F} должно содержать остальные $2^n - m$ слагаемых.

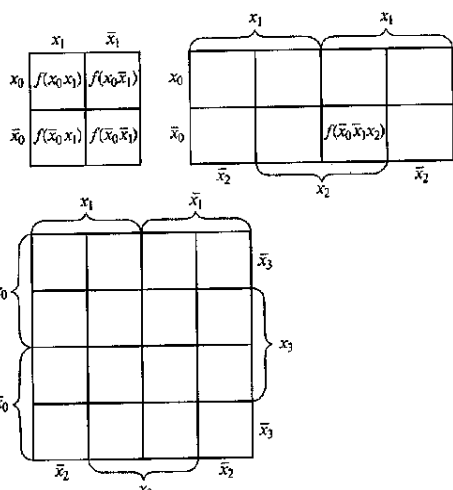
Для мажоритарного элемента, например, это позволяет представить его структурную формулу в виде
$$F = \overline{x_1} \wedge x_2 \wedge x_3 \vee x_1 \wedge \overline{x_2} \wedge x_3 \vee x_1 \wedge x_2 \wedge \overline{x_3} \vee x_1 \wedge x_2 \wedge x_3 \vee \underbrace{x_1 \wedge x_2 \wedge x_3 \vee x_1 \wedge x_2 \wedge x_3}_{\text{повторение}}$$

где выражение в скобках представляет собой повторение одного из логических слагаемых. Последовательно производя операцию логического сложения с учетом еще одного из свойств булевой алгебры $x + \overline{x} = 1$, можем получить $F = x_2 \wedge x_3 \vee x_1 \wedge x_3 \vee x_1 \wedge x_2$, что, в свою очередь, предполагает достаточно простую его схемную реализацию.

Существует ряд алгоритмов минимизации структурной формулы непосредственно в аналитическом виде, но наиболее наглядным является метод, использующий графическое представление таблицы истинности. Наиболее известными методами для этих целей являются карты Карно и диаграммы Вейча, различие между которыми сводится лишь к различию в графическом представлении таблицы истинности. Эти методы широко используются для ручной, без применения ЭВМ, минимизации логических функций, число аргументов которой не превышает 4...5.

Картой Вейча (Карно) называется таблица, число клеток которой для функции n переменных равно 2^n , причем каждому минтерму соответствует своя клетка карты.

Из приведенных рисунков видно, что минтерм представляется минимальным участком площади - одной клеткой на картах Карно (картах минтермов). При этом в смежных клетках наборы



оказываются соседними. Соседними же являются наборы, находящиеся в верхней и нижней строках и в правом и левом столбце. Линиями за границами прямоугольника указаны столбцы и строки, в которых соответствующие переменные встречаются в наборах без инверсии. В других столбцах и строках эти переменные входят в наборы с инверсией. Остальные клетки остаются незаполненными или заполняются нулями. В результате логическая функция представляется совокупностью клеток, заполненных единицами, а инверсия функции – совокупностью нулей или пустых клеток.

Алгоритм минимизации сводится к следующему:

1. На карте выделяют прямоугольные области, объединяющие клетки, в которых функция принимает единичное значение. Каждая область объединения (контур) может содержать только 2^k – клеток (1, 2, 4, 8, 16 и т.д.). При этом контура могут неоднократно пересекаться – каждая клетка может входить в несколько контуров.
2. Каждая из выделенных областей является самостоятельным произведением переменных, значения которых в рамках выделенной области остаются постоянными.
3. Из полученного множества выделенных областей выбирают минимальное количество максимально больших областей. В этом случае сумма полученных произведений образует минимальную структурную формулу.
4. Клетки, содержащие единицы, в которых в диаграмме находились соседние наборы, объединяются контурами по следующим правилам
 - а) Если единицы занимают две соседние строки или два соседних столбца, в том числе расположенные по краям диаграммы и разделенные внешними границами, то они могут быть объединены контурами и т.к. такой контур содержит восемь клеток, слагаемые четвертого ранга, находящиеся в этих клетках могут быть заменены одним слагаемым первого ранга, т.е. той переменной, которая является для них общей.
 - б) Если объединяются четыре квадрата, то четыре слагаемых четвертого ранга заменяются одним слагаемым второго ранга, т.е. переменными, которые яв-

ляются для них общими. Упрощение получается за счет того, что общие переменные можно вынести за скобки, а в скобках остается полный набор из оставшихся переменных, который всегда равен 1, но чтобы это произошло, объединять можно только 2^n клеток. В ином случае в скобках не получится полного набора, который можно заменить 1.

- с) Если объединяются два квадрата, то слагаемые четвертого ранга заменяются одним слагаемым третьего ранга. Одна и та же единица может входить во сколько угодно контуров. Всегда нужно стремиться объединить большее число клеток с единицами и получить при этом наименьшее число контуров. Все единицы должны быть объединены контурами, даже если какая-либо единица не может быть объединена с другими. В этом случае контуром обводится одна клетка и упрощения слагаемого не получается. Далее из каждого контура выписываются одинаковые переменные. Количество слагаемых в получившейся минимизированной формуле должно совпасть с количеством контуров.

В качестве примера рассмотрим задание №25. Первая часть задания требует строгой реализации этой логической функции, т.е. необходимо обеспечить реализацию заданной логической функции для всех комбинаций входных сигналов.

Введем эту функцию в диаграмму Вейча. Все введенные единицы могут быть объединены в 3 контура – 2 контура по 2 единицы и 1 из 4 единиц (каждая из единиц может входить во сколько угодно большое число контуров). Тогда СНФ заданной логической функции может быть представлена дизъюнкцией (логической суммой) трех составляющих, учитывающих размер соответствующих контуров $F = Y_1 \vee Y_2 \vee Y_3$.

Наборы, принадлежащие контуру из четырех клеток, могут быть объединены и их результирующее представление в СНДФ может быть заменено одним членом $Y_1 = \overline{X_1} \wedge X_4$.

Каждый из двух других контуров объединяет единицы лишь двух соседних наборов, что позволяет снизить их общий ранг представления в НДФ лишь на

единицу. В результате можем записать соответственно $Y_2 = X_1 \wedge X_2 \wedge \overline{X_3}$ и $Y_3 = \overline{X_1} \wedge \overline{X_2} \wedge \overline{X_3}$.

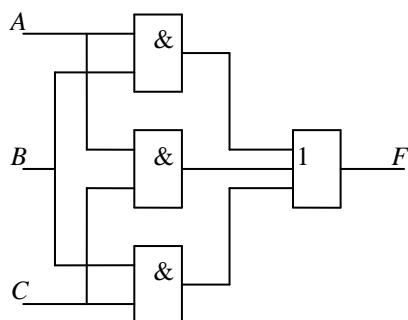
Наиболее простое описание заданной логической функции в ДНФ, таким образом, имеет вид $F = \overline{X_1} \wedge X_4 \vee X_1 \wedge X_2 \wedge \overline{X_3} \vee \overline{X_1} \wedge \overline{X_2} \wedge \overline{X_3}$.

При использовании диаграммы Вейча или карты Карно можно добиться более существенной минимизации, если использовать условие избыточности. Это условие означает, что некоторые наборы переменных на входах проектируемого устройства никогда не встречаются. Например, бывает известно, что цифровое устройство работает в пределах первого десятка. Для этого необходимо четыре двоичных разряда, из четырех переменных можно составить шестнадцать различных наборов, но наборы, соответствующие десятичным числам от 10 до 15 никогда не будут присутствовать на входах устройства и поэтому совершенно безразлично, что будет на выходе устройства при таких наборах на входах.

Это условие позволяет поставить в клетки диаграммы, где находятся эти наборы, как 0, так и 1 в зависимости от того, что обеспечит лучшую минимизацию, т.е. меньшее количество контуров и большее количество клеток в каждом из них (вторая часть задания).

Иногда бывает выгоднее минимизировать не собственно функцию F , а ее инверсию. В этом случае объединяются в контуры нули, что обеспечивает минимизацию функции \overline{F} . Для получения искомой функции достаточно включить дополнительный инвертор, реализующий путем дополнительной операции отрицания получение требуемого результата.

Часто требуется реализовать логическое устройство в другом функционально полном наборе элементов (базисе). Это имеет место, например, в случае, когда разрабатываемое устройство является частью более сложного устройства, при проектировании которого оказались свободные логические элементы. Наиболее часто встречаются базисы «И-НЕ» или «ИЛИ-НЕ». Для приведения структурной формулы к подобному базису используются теоремы булевой алгебры, в частности, законы де Моргана. Для данного примера соответствующее приведение к базису «И-НЕ» имеет вид $F = \overline{\overline{X_1} \wedge X_4 \wedge \overline{X_1} \wedge X_2 \wedge \overline{X_3} \wedge \overline{X_1} \wedge \overline{X_2} \wedge \overline{X_3}}$



Содержание лабораторной работы

1. Разработать логическую модель устройства согласно заданию (указывается преподавателем).
2. Привести ее к базису «И-НЕ».
3. Убедиться в работоспособности устройства путем ее электронного моделирования с использованием программы Electronic Workbench.
4. Оценить возможности упрощения устройства с использованием условия избыточности.

Форма отчетности

Работа сдается путем предъявления разработанной электронной модели.

Литература

- 1) Угрюмов Е.П. Цифровая схемотехника. СПб: «БХВ – Петербург», 2002г.
- 2) Схемотехника электронных систем. Цифровые устройства : Учебник/ В.И. Бойко, А.Н. Гуржий, В.Я. Жуйков и др.. -СПб.: БХВ-Петербург, 2004.-506 с.
- 3) Ушаков В.Н. Основы аналоговой и импульсной техники. -М.: РадиоСофт, 2004.
- 4) Новиков, Юрий Витальевич. Основы цифровой схемотехники. Базовые элементы и схемы. Методы проектирования. -М.: Мир, 2001.

Лабораторная работа №407

«СИНТЕЗ СЧЕТЧИКОВ».

Цель работы. Получение практических навыков разработки функциональных узлов комбинационного типа.

Методические указания.

а) Работа выполняется по индивидуальному заданию, номер конкретного задания указывается преподавателем.

б) Выполнение работы заключается в разработке логической схемы с последующей ее проверкой путем электронного моделирования с использованием программы Electronic Workbench.

Проверка спроектированной схемы осуществляется путем ее моделирования. Образец электронного моделирования содержится в той же папке.

Варианты заданий.

Номер зада-		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Тип счетчика		A	A	C	C	A	A	A	A	A	C	A	C	A	A	A	A	A	A	A	A
Модуль счета		10	10	10	10	10	10	10	10	12	10	10	10	10	10	10	10	10	10	10	10
№ п/п	Код	5-3-2-1	5-3-2-1	5-4-2-1	5-3-2-1	2-4-2-1	6-4-2-1	5-4-2-1	5-2-2-1	8-4-2-1	5-2-2-1	5-2-2-1	6-4-2-1	5-2-2-1	5-2-1-1	4-4-2-1	4-2-2-1	4-4-2-1	2-4-2-1	4-4-2-1	4-2-2-1
0	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0001	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0010	2	2	2	2	2	2	2		2		2	2	2		2	2	2	2	2	
3	0011		3	3	3	3	3	3		3			3	3	2	3	3	3	3	3	
4	0100	3		4			4	4	2	4	2		4			4		4			2
5	0101	4	4		4		5		3	5	3	3	5			5		5			3
6	0110						6		4	6	4	4	6	4	3	6	4	6			4
7	0111						7			7			7		4	7	5	7			5
8	1000	5	5	5	5			5	5	8	5	5		5	5					4	
9	1001	6	6	6	6			6	6	9	6	6		6	6					5	
10	1010	7	7	7	7	4	8	7		10		7	8	7			6		4	6	
II	1011		8	8	8	5	9	8		11			9	8	7		7		5	7	
12	1100	8		9		6		9	7		7					8		8	6	8	6
13	1101	9	9		9	7			8		8	8				9		9	7	9	7
14	1110					8			9		9	9		9	8		8		8		8
15	1111					9									9		9		9		9

Примечание. А - асинхронный, С - синхронный счетчики.

Краткие сведения из теории

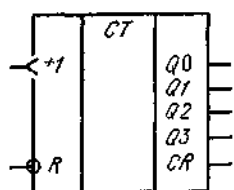
Счетчики

Счетчиком называется последовательное устройство, предназначенное для счета входных импульсов и фиксации их числа. Счет импульсов является одной из наиболее распространенных операций в устройствах дискретной обработки информации.

Основное функциональное назначение счетчиков:

- ♦ регистрация числа импульсов, поступивших на вход;
- ♦ деление частоты (используется однозначная связь между частотами следования периодической последовательности импульсов на входе счетчика и его выходе $F_{\text{вых}} = F_{\text{вх}} / K$.

По мере поступления входных сигналов счетчик последовательно перебирает свои состояния в определенном для данной схемы порядке. Длину списка используемых состояний (параметр K) называют *модулем пересчета* или *емкостью* счетчика. Одно из возможных состояний счетчика принимается за *начальное (нулевое)*. Если счетчик начал считать с начального состояния, то через каждые K сигналов в нем снова устанавливается начальное состояние, а на выходе счетчика при этом появляется сигнал *K -ичного переноса* CR .

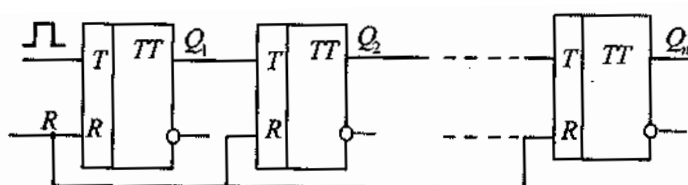


❖ $Q0—Q3$ — выходы счетчика, комбинация значений

которых определяет номер его состояния; CR — выход переноса, R - вход общего сброса. Часто счетчик снабжен и входами данных Dj для *параллельной загрузки* произвольного кода. Загрузка

осуществляется при подаче сигнала на еще один вход — вход параллельной загрузки L .

Из определения и логики работы счетчиков следует, что их текущее состояние зависит не только от нового пришедшего импульса, но также и от количества предыдущих импульсов, т.е. счетчики относятся к устройствам с памятью.



Строятся счетчики, как и регистры, на основе однотипных связанных между собой триггеров.

Наиболее часто используются T - и JK -триггеры. (T -триггеры так и называются — счетные триггеры). В JK -триггерах счетный режим обеспечивается при $J=K=1$. Комбинационные элементы в счетчиках используются для управления работой триггеров. Число триггеров определяет максимальное количество импульсов, которое может быть подсчитано счетчиком.

В счетчиках выполняются следующие логические операции:

- ◆ установка в нулевое состояние (сброс);
- ◆ запись входной информации в параллельной форме — начального кода, с которого начинается счет;
- ◆ хранение записанной информации;
- выдача хранимой информации в параллельной форме;
- ◆ инкремент — увеличение хранящегося числа на единицу;
- ◆ декремент — уменьшение хранящегося числа на единицу.

По значению модуля счета K различают:

- ◆ двоичные счетчики ($K=2^n$, где n — число используемых триггеров);
- ◆ недвоичные ($K < 2^n$).

Различные схемы счетчиков могут перебирать свои состояния в самом различном порядке. В двоичных счетчиках порядок смены состояний триггеров соответствует последовательности двоичных чисел. Кроме того, применяют *одинарное* кодирование, когда состояние счетчика представлено местом расположения единственной единицы (например, сдвигающий регистр с однойдвигающейся единицей — *кольцевой счетчик, распределитель тактов*), возможно *унитарное* кодирование, когда состояние представлено числом единиц и более сложные виды кодирования.

Различают *суммирующие, вычитающие и реверсивные* счетчики. Суммирующий счетчик перебирает свои состояния в возрастающем порядке, *вычитающий* — в убывающем. В *реверсивном счетчике* предусматривается вход для управляющего сигнала, устанавливающего направление счета.

Счетчики, которые в процессе работы для переключения требуют подачи синхросигналов, называют *синхронными (тактируемыми)*, а счетчики, у которых для переключения достаточно подавать лишь входные сигналы, — *асинхронными*.

Суммирующий вход счетчика обозначается «+1», вычитающий — «-1». Это счетные входы. У асинхронных счетчиков они помечаются таким же символом, как и счетный вход Т-триггера, указывающим полярность перепада входного сигнала, по которому счетчик меняет состояние своего выхода.

Временные характеристики счетчиков оцениваются их разрешающей способностью, быстродействием и временем установления кода.

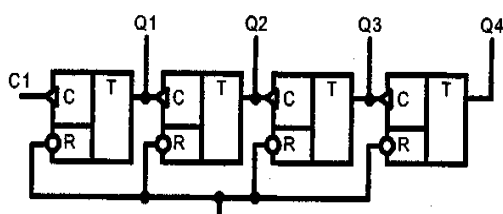
Разрешающая способность t_{pc} определяется минимальным интервалом времени между двумя входными импульсами, при котором еще сохраняется работоспособность счетчика. Очевидно, что этот параметр определяется характеристиками первого триггера счетчика $t_{pc} = t_T$.

Быстродействие характеризуется максимальной частотой входных импульсов в режиме деления частоты $F_{max} = 1/t_T$.

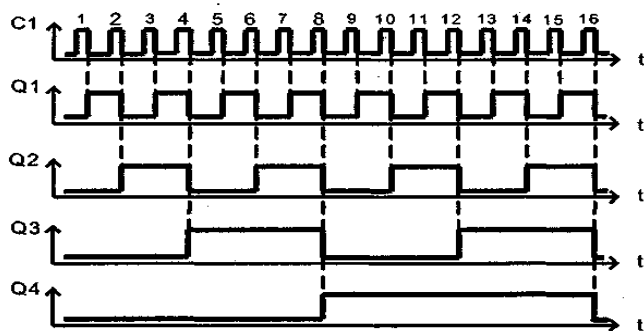
Время установления кода $t_{уст}$ отсчитывается от начала входного импульса до момента получения нового состояния на выходе счетчика.

Двоичные счетчики

Суммирующий двоичный асинхронный счетчик с последовательным переносом



На рис. приведена схема простейшего двоичного 4-разрядного суммирующего асинхронного счетчика.



Временные диаграммы показывают состояния каждого из триггеров при поступлении на вход периодической последовательности импульсов.

В таблице истинности приведено состояние триггеров, соответствующее числу поступивших на вход импульсов.

Таблица истинности четырехразрядного счетчика

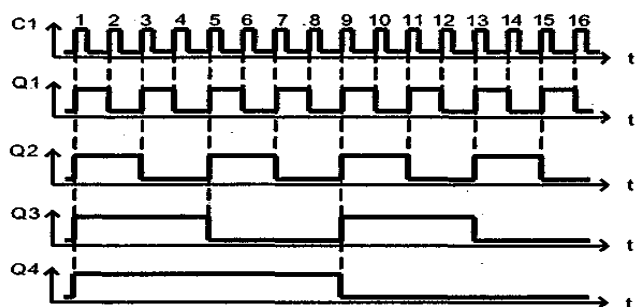
Число поступивших импульсов		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Выходы	Q1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	Q2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	Q3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	Q4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0

Как можно заметить, для срабатывания по заднему фронту (срезу) входных импульсов триггеры имеют инверсные динамические входы. Состояние счетчика в двоичном коде по

приходу на вход каждого нового импульса увеличивается на единицу - осуществляется операция инкремента.

Так как счетный триггер делит частоту входных импульсов на два, то цепочка из четырех последовательно соединенных триггеров делит частоту на 16. По приходу каждого шестнадцатого импульса счетчик обнуляется и цикл счета начинается сначала.

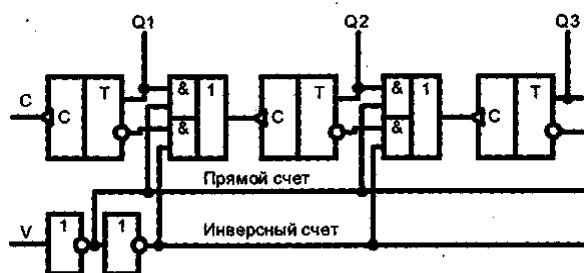
Вычитающий двоичный асинхронный счетчик с последовательным переносом



Если исходные асинхронные T -триггеры имеют прямые динамические входы (срабатывают по переднему фронту импульса при переходе из 0 в 1), то счетчик превращается в вычитающий. Он выполняет операцию декремента.

мента.

Того же эффекта можно достичь, если использовать не прямые, а инверсные выходы предыдущих триггеров счетчика. Это и используется при построении реверсивного счетчика.



Действительно, управляя выбором прямого или инверсного выхода предыдущего триггера, можно реализовать как режим суммирования входных импульсов, так и режим вычитания.

Комбинационная схема для обеспечения режима реверсивного счетчика представляет собой группу их соответствующего числа мультиплексоров $2 \rightarrow 1$, причем их схемная может быть различной. Кроме готовых ИС мультиплексоров, они могут

быть реализованы на логических элементах. Еще один способ — использование элемента «ИСКЛЮЧАЮЩЕЕ ИЛИ», который является управляемым инвертором, что и требуется для данного случая.

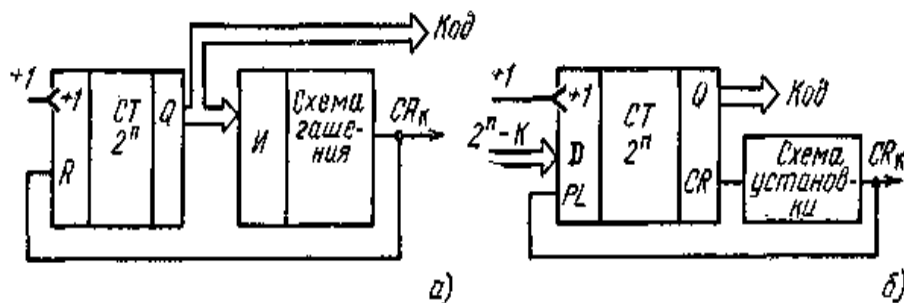
Счетчики с модулем счета $K \neq 2^n$

Различные области применения требуют счетчиков с модулями пересчета (основаниями), не только кратными целой степени двойки, но и другими. Например для работы в десятичной системе — 10, для схем часов и календарей — 60, 24, 7... В общем случае требуется строить счетчики по любому заданному основанию K . Иногда пересчет выгоднее реализовать на единственном счетчике, иногда — разложить основание на два сомножителя: целую степень двойки, реализуемую на обычном двоичном счетчике, и оставшееся нечетное число, являющееся основанием счетчика уже меньшего размера, чем он получится, если его строить непосредственно для заданного основания.

На базе ИС счетчиков требуемый счетчик по произвольному основанию можно построить двумя основными способами.

1. Двоичный счетчик разрядности n , такой, чтобы 2^n было больше K , дополняется элементом И, который по состояниям выходов Q_i обнаруживает код (обычно $K-1$), после чего по цепи R сбрасывает счетчик в 0.

Структурная схема, реализующая этот способ, показана на рис. а. Сигнал, сбрасывающий счетчик, одновременно является и сигналом K -ичного переноса CR_K .



Достоинства способа: а) естественная двоичная последовательность кодов от 0 до $K-1$,

б) использование обычно имеющегося в ИС счетчиков входа R . В случае суммирующего счетчика достаточно собрать на элементе

И лишь прямые выходы тех триггеров, которые при коде конца счета равны 1.

Число входов элемента И, таким образом, зависит от кода конца счета.

2. Второй способ иллюстрирует рис. 6. Двоичный счетчик перед началом счета по тракту параллельной загрузки загружается кодом дополнения числа K до 2^n . Кодом конца счета в этом случае является естественное переполнение счетчика, т. е. код *Все единицы*, обнаруживаемый штатным трактом переноса, в результате чего вырабатывается сигнал CR . Сигнал CR , воздействуя на вход L , управляющий параллельной загрузкой, снова устанавливает в счетчике дополнение K до 2^n .

Достоинства способа — а) использование штатного тракта переноса и имеющихся во многих счетчиках входов параллельной загрузки,

б) легкая смена основания пересчета. Для этого входы Di нужно подключить не к константам 1 и 0 (питание и общий провод), а к выходу специального регистра начальных состояний.

Недостатком способа является неестественная последовательность получаемых кодов, требующая в случае их использования перекодировки.

Данный способ применяется, когда показания счетчика не важны, а используется лишь сигнал его выходного переноса. Это типично для задачи деления частоты входных сигналов на некоторое число K . Счетчики, выполняющие эту функцию, называют *делителями*.

Счетчики с недвоичными весами разрядов

Многообразие задач, решаемых средствами цифровой электроники, использует счетчики, в которых состояния могут меняться по несколько иным законам. В автоматике, например, широко используется код Грея, характеризующийся тем, что соседние состояния счетчика различаются значениями только одного из разрядов. Такое представление состояний в устройствах силовой автоматики позволяет свести до минимума влияние переходных процессов при управлении мощными установками. Одним из возможных способов реализации подобных счетчиков является их синтез на базе универсальных JK – триггеров.

Синтез счетчика с недвоичными весами разрядов

Основные принципы проектирования рассмотрим на конкретном примере. Пусть требуется синтезировать асинхронный двоично-десятичный счетчик, работающий в коде

5-2-1-1. В соответствии с заданным кодом заполняем левую часть табл. 1 функционирования счётчика (столбцы Q_1, Q_2, Q_3, Q_4 .

В таблице n - номер состояния счетчика, меняющийся на единицу при подаче каждого счетного импульса; Q_1, Q_2, Q_3, Q_4 - логические переменные на выходе четырех триггеров, первый триггер с выходом Q_1 соответствует первому младшему разряду; J и K - значения необходимых сигналов на соответствующих входах JK - триггеров.

Указанный в задании режим асинхронного счетчика означает, что каждый из универсальных триггеров проектируемого счетчика может переключаться как при подаче на его вход C положительного («одиначного») импульса, так и при подаче сигнала на вход C с выхода другого триггера., тогда как для синхронного допускается только первый вариант. При этом переключение последующего триггера происходит тогда, когда предыдущий переключается из «1» в «0». Учитывая это, находим необходимое место подключения входов C всех четырех триггеров. Первый триггер должен переключаться при подаче первого, пятого и нулевого (десятого) импульса. Ни один из последующих триггеров не может обеспечить всех трех переключений. Поэтому на вход C первого триггера нужно подавать счетные импульсы. Они обеспечивают максимальную частоту переключений, а выборка нужных моментов переключения обеспечивается подачей сигналов на вход J и K .

На вход C второго триггера также нужно подать счетные импульсы, так как выходные сигналы ни одного из триггеров не обеспечат его переключения в нужный момент.

Вход C третьего триггера нужно соединить с выходом второго, так как его переключение при подаче нулевого, третьего, пятого и восьмого импульсов на вход счетчика из «1» в «0» может быть использовано для переключения третьего триггера.

По тем же соображениям выход третьего триггера нужно соединить со входом четвертого.

Таким образом, мы определили принцип построения счетчика - выбрали его базовые элементы -- JK – триггеры и определили сигналы, которые способны изменить состояние отдельных триггеров в соответствующие моменты времени. Однако наличие этого сигнала на входе C отдельного триггера является лишь необходимым, но

не достаточным условием для правильного поведения этого триггера в этот момент. Эта задача решается с помощью формирования управляющих сигналов на его JK - входах.

Для определения требуемых сигналов на входах J и K заполняют правую часть

Таблица 2

$Q^n \rightarrow Q^{n+1}$	J^n	K^n
0 0	0	-
0 1	1	-
1 0	-	1
1 1	-	0

Таблица 3

		X_1		X_2	
X_3	X_4	X_1	X_2	X_1	X_2
		0101	0111	0110	0100
		1101	1111	1110	1100
		1001	1011	1010	1000
		0001	0011	0010	0000

табл. 1, используя таблицу функционирования JK - триггера (табл. 2).
Прочерк в таблице пока-

зывает, что значение сигнала на данном входе не вызывает изменения состояния соответствующего триггера. Например, для переключения первого триггера из «0» в «1» при подаче первого счетного импульса требуется, чтобы $J_1 = 1$ в позиции $n = 0$, а значения K_1 может быть любым, что означает прочерк. Если при очередном такте работы на какой-либо триггер сигнал на его вход C не подается, т.е. триггер, с которого он может прийти, не сбрасывается в «0», то в соответствующих клетках J и K можно ставить прочерки, так как при любых значениях J и K триггер не переключается. Таким способом заполняется вся таблица.

Перенесем теперь данные правой части табл. 1 на диаграммы Вейча.

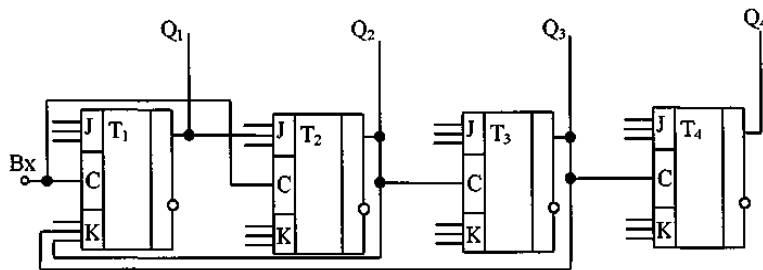
В табл. 3 приведена диаграмма Вейча для четырех логических переменных. Принимаем за логические переменные значения сигналов на выходах триггеров и заполним восемь диаграмм Вейча (табл. 4), по которым определим сигналы на входах J и K четырех триггеров. При заполнении диаграмм единицы, нули или прочерк ставятся в тех клетках, в которых находятся соответствующие комбинации выходных переменных Q .

Таблица 4

J_1				K_1				J_2				K_2			
-	-	-	-	0	1	-	-	1	-	-	-	-	1	-	-
-	-	-	-	0	1	-	-	1	-	-	-	-	1	-	-
-	-	-	1	0	0	-	-	1	-	-	0	-	1	-	-
-	-	-	1	0	0	-	-	1	-	-	0	-	1	-	-
J_3				K_3				J_4				K_4			
-	-	-	-	-	1	-	-	-	1	-	-	-	-	-	-
-	-	-	-	-	1	-	-	-	-	-	-	-	1	-	-
-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-

В клетках, в которых функция не определена или её значение не играет роли (т.е. ставится прочерк), можно помещать любые значения пе-

ременных, чтобы объединить контуром наибольшее количество клеток. Так, в табл. 4 для J_1 во всех клетках можно по-ставить единицы и объединить одним контуром. Это означает, что $J_1 = 1$. из всех других диаграмм следует, что $K_1 = Q_2 Q_3$, $J_2 = Q_1$, $K_2 = 1$, $J_3 = 1$, $K_3 = 1$, $J_4 = 1$, $K_4 = 1$. Следовательно, на входы J_1 , K_2 , J_3 , K_3 , J_4 , K_4 надо подать единицы, на вход



K_1 - конъюнкцию сигналов с прямых выходов второго и третьего триггеров, а вход J_2 соединить с прямым выходом

первого триггера. Если какой либо вход элементов ТТЛ никуда не подключен, это эквивалентно подаче на этот вход единицы. Таким образом, счетчик синтезирован.

Содержание лабораторной работы

5. Разработать логическую модель счетчика с недвоичным основанием согласно индивидуальному заданию (указывается преподавателем).
6. Убедиться в работоспособности устройства путем ее электронного моделирования с использованием программы Electronic Workbench.

Форма отчетности

Работа сдается путем предъявления разработанной электронной модели.

Контрольные вопросы

- a) Назначение счетчиков импульсов.
- b) Особенности применения счетчиков в режиме делителей частоты.
- c) Принципы построения счетчиков с произвольным модулем счета.
- d) Принципы построения недвоичных счетчиков.
- e) Сравнительная характеристика синхронных и асинхронных счетчиков.

Литература

- a) Угрюмов Е.П. Цифровая схемотехника. СПб: «БХВ – Петербург», 2002г.
- b) Схемотехника электронных систем. Цифровые устройства : Учебник/ В.И. Бойко, А.Н. Гуржий, В.Я. Жуйков и др.. -СПб.: БХВ-Петербург, 2004.-506 с.
- c) Ушаков В.Н. Основы аналоговой и импульсной техники. -М.: Радио-Софт, 2004.
- d) Новиков, Юрий Витальевич. Основы цифровой схемотехники. Базовые элементы и схемы. Методы проектирования. -М.: Мир, 2001.