Вопрос № 1

Функциональные узлы последовательного типа: регистры, триггеры, счетчики

Последовательными логическими схемами называют полные цифровые автоматы, выходные сигналы которых зависят не только от состояний входных сигналов в текущий момент времени, но и от состояния схемы в предыдущий момент времени

Триггер – это элемент цифрового устройства с двумя устойчивыми состояниями. Под воздействием входного сигнала триггер может переключаться из одного положения в другое, при этом напряжение на его выходе скачкообразно изменяется.

Как правило, триггер имеет два выхода прямой и инверсный(Q, $\overline{Q}$). Число входов зависит от структуры и функций, выполняемых триггером. Входы, как и сигналы, подаваемые на них делятся на информационные и вспомогательные. Информационные сигналы через соответствующие входы управляют состоянием триггера. Сигналы на вспомогательных входах служат для предварительной установки триггера в заданное состояние и его синхронизации. Вспомогательные входы могут при необходимости выполнять роль информационного.

Входы и выходы триггера, как и соответствующие им сигналы, обозначают буквами:

S – раздельный вход установки в единичное состояние (напряжение высокого уровня на прямом входе Q);

R – раздельный вход установки в нулевое состояние (напряжение низкого уровня на прямом входе Q);

Д – информационный вход (на него передается информация, предназначенная для занесения в триггер);

C – вход синхронизации;

T – счетный вход.

Триггеры классифицируют по ряду признаков. По функциональным возможностям выделяют триггеры: с раздельной установкой "0" и "1" (RS – триггеры); с приемом информации по одному входу (Д – триггеры); счетный T – триггер; универсальный IK – триггер.

По способу приема информации триггеры подразделяют на асинхронные и синхронные. Асинхронные триггеры реагируют на информационные сигналы в момент их появления на входе. Синхронные – при наличии разрешающего сигнала специально предусмотренном входе С.

Счетчиком называют устройство, сигналы, на входе которого в определенном коде отображают число импульсов, поступивших на счетный вход. Триггер Т-типа может служить примером простейшего счетчика. Такой счетчик считает до двух. Счетчик, образованный цепочкой из m-триггеров, сможет посчитать в двоичном коде 2^m импульсов. Каждый из триггеров цепочки называют разрядом счетчика. Число m определяет количество разрядов двоичного числа, которое может быть записано в счетчик. Число $К_а=2^m$ называют коэффициентом (модулем) счета.

Информация снимается с прямых и (или) инверсных выходов всех триггеров. В паузах между входными импульсами триггеры сохраняют свое состояние, т. е. счетчик запоминает число сосчитанных импульсов.

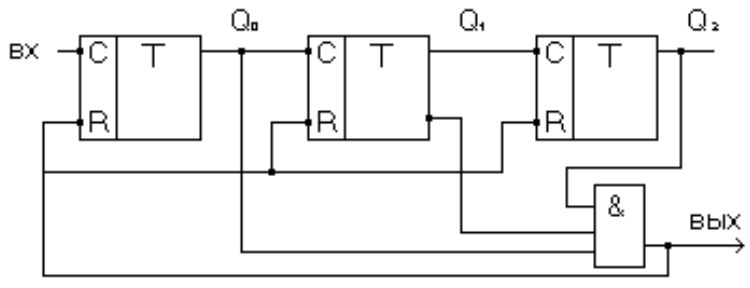
Нулевое состояние всех триггеров принимается за нулевое состояние счетчика в целом.

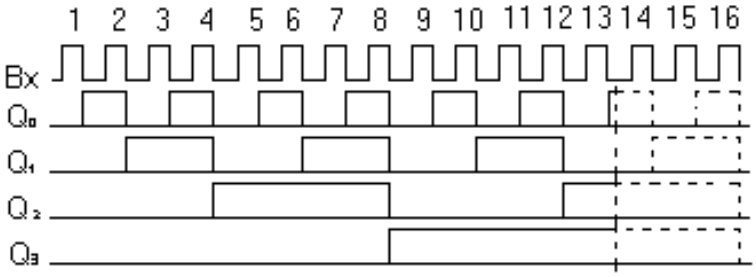
Классификация счетчиков.

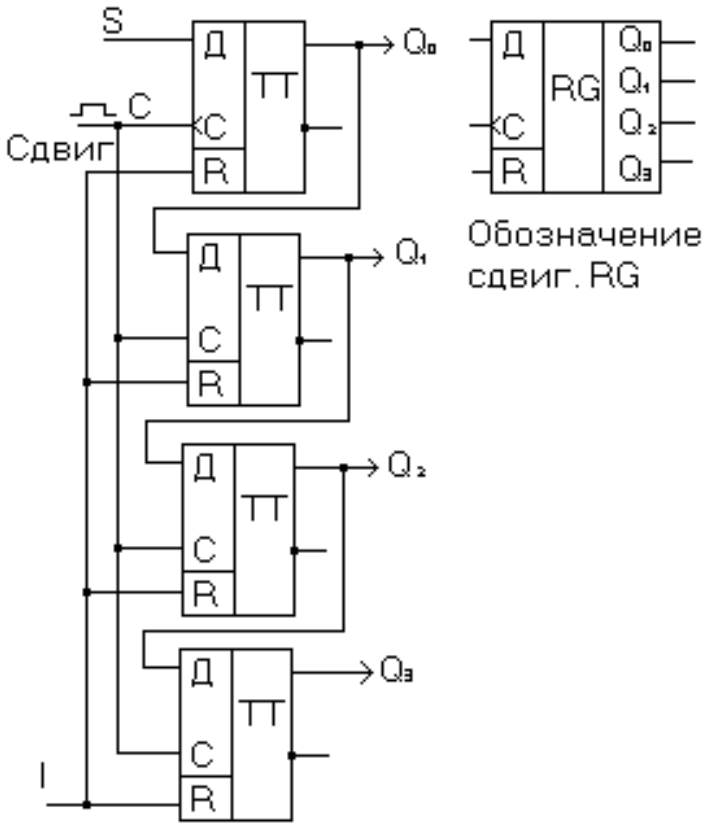
По коэффициенту счета: двоичные; двоично-десятичные (декадные) или с другим основанием счета; с произвольным постоянным модулем; с переменным модулем.

По направлению счета: суммарные; вычитающие; реверсные.

По способу организации внутренних связей: с последовательным переносом; с параллельным переносом; с комбинированным переносом; кольцевые.







Сдвиговый регистр

Счетчик с параллельным переносом строят из синхронного триггера. Счетные импульсы подаются одновременно на все тактовые входы, а каждый из триггеров цепочки служит по отношению к последующим только источником информации. Срабатывание триггеров параллельного счетчика происходит синхронно и задержка переключения всего счетчика равна задержке переключения для одного триггера. Счетчики с параллельным переносом применяют в быстродействующих устройствах.

Регистры

Назначение регистров – хранение и преобразование много разрядных двоичных чисел. Регистры наряду со счетчиками и запоминающими устройствами являются наиболее распространенными последовательными устройствами цифровой техники. Регистры обладают большими функциональными возможностями. Они используются в качестве управляющих и запоминающих устройств, генераторов и преобразователей кодов, счетчиков, делителей частоты, узлов временной задержки.

Элементами структуры регистров являются синхронные триггеры Д-типа, либо RS(IK)-типа с динамическим или статическим управлением.

Единичный триггер – простейший регистр (RG) используют в RG цепочке триггеров. Понятие "весовой коэффициент" к разрядам регистра неприменимо в отличие от счетчика, поэтому на условных изображениях нумерация входов и выходов идет подряд (Д1, Д2,Д3 и т.д., а не Д1, Д2, Д4, Д8).

RG в зависимости от функциональных свойств делятся на: накопительные и сдвигающие.

По способу ввода, вывода информации – параллельные, последовательные и комбинированные (//-последовательные и последовательно-//).

По направлению передачи (сдвига) информации – однонаправленные и реверсные.

Регистры памяти – простейший вид регистров – хранят двоичную информацию.

Это набор синхронных триггеров, каждый из которых хранит один разряд двоичного числа. Ввод (запись) и вывод (считывание) производится одновременно во всех разрядах. С приходом очередного тактового импульса происходит обновление информации.

Считывание информации в прямом или обратном (с $\overline{Q}$) коде.

Вопрос № 2

Назначение, классификация математических моделей и методы их построения. Проверка адекватности математических моделей

Задача проектирования на современном этапе заключается в разработке нового объекта, исключающего ошибки не только работоспособного, но и оптимального с точки зрения заданного критерия.

Для решения данной задачи необходимо наличие

1. Варьируемых переменных, т.е. таких, которые можно произвольно менять в некоторых пределах.
2. Критерия, т.е. показателя, с использованием которого сравниваются различные варианты проектных решения, полученных при различных значениях варьируемых переменных
3. Мат. модель, связывающая варьируемые переменные и критерий.

Мат. модель – это система:

- Булевых
- Алгебраических
- Дифференциальных уравнений
- Интегральных уравнений

Назначение мат. модели:

- Управление объектом
- Оптимизация действующих объектов
- Проектирование новых объектов

Классификация мат. моделей:

1. По режиму работы объекта (статики и динамики)
2. По свойствам объекта (сосредоточенные координаты (переменные объекта одинаковы во всех точках) и распределенные координаты(наоборот ☺))
3. Линейные (удовл. принципу суперпозиции)/нелинейные (наоборот ☺)

Методы построения мат. моделей

1. Аналитический
2. Экспериментальный
3. Экспериментально-аналитический (Некоторые параметры аналитически построенной модели уточняются с помощью экспериментов)

Вопрос № 3

Алгоритмы сжатия графических данных.

Алгоритмы сжатия изображений бурно развивающаяся область машинной графики. Основной объект приложения – изображение, характеризуется тремя особенностями:

1. изображение требует для хранения гораздо больше памяти, чем текст. Это определяет актуальность алгоритмов архивации графики.
2. человеческое зрение при анализе изображения оперирует контурами и общими переходами цветов и сравнительно нечувствительно к малым изменениям в изображении. Это позволяет создать специальные алгоритмы сжатия ориентированные только на изображения.
3. изображения обладают избыточностью в двух измерениях (по горизонтали и вертикали). Т.о. при создании алгоритма компрессии графики мы используем особенности структуры изображения.

На данный момент известно минимум три семейства алгоритмов, разработанных исключительно для сжатия изображений.

Классы изображений:

1. изображения с небольшим количеством цветов (4-16) и большими областями, заполненными одним цветом, плавные переходы цветов отсутствуют.
2. изображения с плавными переходами цветов.
3. фотореалистические изображения.
4. фотореалистические изображения с наложением деловой графики.

Алгоритмы архивации без потерь**.**

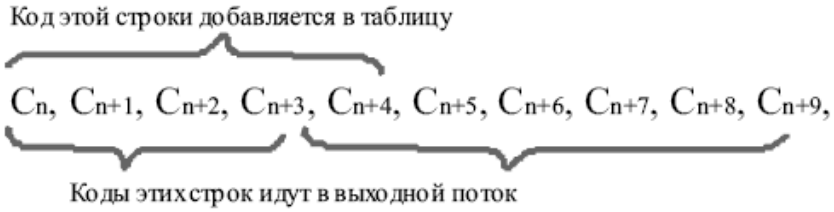
RLE (Run Length Encoding)

Групповое кодирование – самый старый и самый простой алгоритм. Изображение в нем вытягивается в цепочку байт по строкам растра. Цепочки одинаковых байт в изображении заменяются на пары <счетчик повторений, значение>.

LZW (Lempel, Ziv, Welch)

Сжатие происходит за счет одинаковых цепочек байт. Процесс сжатия выглядит достаточно просто. Мы считываем последовательно символы входного потока и проверяем, есть ли в созданной нами таблице строк такая строка. Если строка есть, то считываем следующий символ, а если строки нет, то мы заносим в поток код для предыдущей найденной строки, заносим строку в таблицу и начинаем поиск снова.

Особенность алгоритма заключается в том, что для декомпрессии нам не надо сохранять таблицу строк в файл для распаковки. Алгоритм построен таким образом, что мы в состоянии восстановить таблицу строк, пользуясь только потоком кодов. Мы знаем, что для каждого кода надо добавлять в таблицу строку, состоящую из уже присутствующей там строки и символа, с которого начинается следующая строка в потоке.



Пример:

Пусть мы сжимаем последовательность 45,55,55,151,55,55,55. Тогда мы имеем в выходной строке сначала код очистки <256>. Потом добавим к изначально пустой строке 45 и проверим, есть ли строка 45 в таблице. Поскольку мы при инициализации занесли в таблицу все строки из одного символа, то 45 есть в таблице. Далее читаем следующий символ из входного потока 55 и проверяем есть ли строка 45,55 в таблице. Такой строки в таблице пока нет. Мы заносим в таблицу строку 45,55 с первым свободным кодом 258 и записываем в поток код 45. Можно коротко представить архивацию так:

45 – есть в таблице

45,55 – нет, добавляем в таблицу 258 - 45,55; в поток – 45.

55,55 – нет, в таблицу 259 – 55, 55; в поток – 55.

55,151 – нет, в таблицу 260 – 55,151; в поток – 55.

151,55 – нет, в таблицу 261 – 151,55; в поток – 151.

55,55 – есть в таблице.

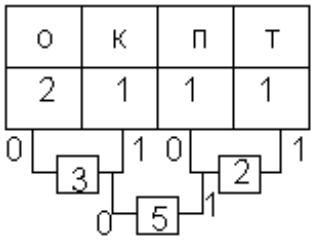
55,55,55 – нет, в таблицу 262 - 55,55,55; в поток 259.

Алгоритм Хаффмана

Классический алгоритм Хаффмана практически не применяется к изображениям в чистом виде, а используется как один из этапов компрессии в более сложных схемах. Близкая модификация алгоритма используется при сжатии черно-белых изображений. Последовательность подряд идущих черных и белых точек заменяется числом, равным их количеству. А этот ряд сжимается по Хаффману с фиксированной таблицей. Каждая строка изображения сжимается независимо.

Пример:

Сжать по методу Хаффмана *поток*



О – 00

К – 01

П – 10

Т – 11

1000110001

Алгоритмы архивации с потерями

Алгоритм JPEG

Один из самых новых и мощных алгоритмов. Он является стандартом де-факто для полноцветных изображений. Алгоритм оперирует областями 8*8, на которых яркость и цвет меняются достаточно плавно. Вследствие этого, при разложении такой матрицы в двойной ряд по косинусам значимыми являются только первые коэффициенты. Т.о. сжатие осуществляется за счет плавности изменения цветов в изображении. В целом алгоритм основан на применении дискретно-косинусоидального преобразования (ДКП) к матрице изображения для получения новой матрицы коэффициентов. Для получения исходного изображения применяется обратное преобразование.

Алгоритм:

1. переводим изображения из RGB в модель YCrCb (яркость, хроматический красный, хроматический синий). За счет того, что человеческий глаз менее чувствителен к цвету, чем к яркости, появляется возможность архивировать массивы Cr, Cb с большими потерями.

$$\begin{vmatrix} Y \\ Cb \\ Cr \end{vmatrix} = \begin{vmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{vmatrix} * \begin{vmatrix} R \\ G \\ B \end{vmatrix} + \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix}$$

$$\begin{vmatrix} R \\ G \\ B \end{vmatrix} = \begin{vmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{vmatrix} * \left(\begin{vmatrix} Y \\ Cb \\ Cr \end{vmatrix} - \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix} \right)$$

2. разбиваем исходное изображение на матрицы 8*8

3. применяем ДКП к матрицам

$$Y[u,v] = \frac{1}{4} \sum_{i=0}^{i} \sum_{j=0}^{j} C(i,u) \times C(j,v) \times \quad y[i,j]$$

где $C(i,u) = A(u) \times \quad \cos\left(\frac{(2 \times i+1) \times u \times \pi}{2 \cdot n}\right)$

$$A(u) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for} \quad u \equiv 0 \\ 1, & \text{for} \quad u \neq 0 \end{cases}$$

При этом получаем матрицу, в которой коэффициенты в левом верхнем углу соответствуют низкочастотной составляющей изображения, а в правом нижнем – высокочастотной. Плавное изменение цвета соответствует низкочастотной составляющей, а резкие скачки – высокочастотной.

4. квантование – деление рабочей матрицы на матрицу квантования

$$Yq[u,v] = \text{IntegerRound}\left(\frac{Y[u,v]}{q[u,v]}\right)$$

На этом шаге осуществляется управление степенью сжатия и происходят самые большие потери.

5. переводим матрицу 8*8 в 64-элементный вектор при помощи зигзаг-сканирования. Т.о. получаем в начале вектора коэффициенты, соответствующие низким частотам, а в конце – высоким.

6. свертываем вектор с помощью алгоритма группового кодирования

7. свертываем получившиеся пары кодированием по Хаффману.