1. Типы цифровых устройств. Определение принадлежности устройства к первому или второму типу. Примеры устройств относящихся к первому и второму типу.

Эти устройства работают на 2х уровнях напряжения 0 и 1. Все устройства должны иметь схемы (логические элементы) где информация хранится, преобразуется и направляется. 2 типа схем: а) преобразования без учета предыдущего состояния (сумматоры, шифраторы и дешифраторы, мультиплексоры и демультиплексоры), б)запоминающие, преобразующие с учетом предыдущего состояния (триггеры, счетчики, регистры) Все цифровые устройства делятся на 2 больших класса: КЦУ и конечные автоматы.

Все цифровые устройства делятся на 2 основных класса :

комбинационные цифровые устройства (КЦУ)

Б . конечные автоматы

КЦУ-устройства не содержащие в своей структуре обратной связи. В каждый момент времени состояние выхода зависит только от входного воздействия.

Устройства этого класса служат для преобразования информации и коммутации.

К таким устройствам относятся:

Сумматоры

Шифраторы и дешифраторы

А также различные кодопреобразователи

Коммутационные КЦУ- мультиплексоры и демультиплексоры

Конечные автоматы- устройства содержащие в своей структуре обратную связь, в каждый момент времени состояние выходов зависит и от входных воздействий, и от предыдущих состояний выходов.

КА служат для хранения информации и её преобразования с учетом предыдущих состояний.

К устройствам этого класса относят: Триггеры и все устройства, построенные на основе триггерной системы (счетчики, регистры)

2. Комбинационные цифровые устройства. Шифратор и дешифратор.

КЦУ это устройства не содержащие в своей структуре обратной связи, т.о. в каждый момент времени состояние выходов такого устройства зависит только от входного воздействия. КЦУ служит для преобразования информации и коммутации.

К кодопреобразующим КЦУ относятся шифратор, дешифратор, различные типы кодопреобразователей, коммутационные КЦУ: (де)мультиплексор. Синтез КЦУ начинается с технического задания (определение функционирования), по заданию записываются таблицы функционирования и при необходимости логические уравнения, описывающие уравнения устройства. Для кодопреобразователя синтез производится аналогично рассмотренным устройствам(составляется таблица переключений, для каждого выхода записывается логическое уравнение относительно входов).

Дешифратор – устройство определяющее направление по поступившему на входы адресы. В каждый момент времени активный уровень может появиться только на одном выходе устройства, индекс выхода при этом совпадает с двоичным кодом поступившим на вход адреса.

Шифратор –устройство, определяющее адрес направления, по которому поступил запрос. В каждый момент времени в таком устройстве активный сигнал может быть только на одном входе. В противном случае состояние выходов не определено. Для шифратора с количеством входов направлений N и количеством адресных выходов M соотношение $M = \log_2 N$

3. Комбинационные цифровые устройства. Мультиплексор и демультиплексор.

Мультиплексор – это устройство, коммутирующее на единственный выход тот из входов данных, адрес которого указан на адресных входах.

Для n адресных входов и m входов данных соотношение $m=2n$ ($m<2n$).

Функция выхода:

Do = nA1 & nA0 & Di0 \/ nA1 & A0 & Di1 \/ A1 & nA0 & Di2 \/ A1 & A0 & Di3;

Демультиплексор - это устройство, коммутирующее единственный вход данных с тем из выходов данных, адрес которого указан на адресных входах.

Соотношение между m и n аналогично соотношению для мультиплексора.

Универсальный коммутатор. Устройство, связывающее единственный регистр с одним из нескольких, имеющих адреса. Может функционировать как мультиплексор или как демультиплексор в зависимости от ситуации. Адресные входы включены в дешифратор, управляющий ключами направлений.

4. Комбинационные цифровые устройства. Сумматор.

Сумматор – комбинационное цифровое устройство, предназначенное для получения арифметической суммы двух чисел, представленных в двоичном коде.

Замечаем, что все разрядные сумматоры, кроме нулевого разряда, имеют дополнительный вход переноса.

Итак, для реализации N-рядного сумматора необходим один модуль полусумматора, реализующий 0-й разряд (2 входа и 2 выхода), и N-1 модуль полного одноразрядного сумматора (3 входа и 2 выхода).

5. Конечные автоматы. Принцип функционирования КА. Типы КА.

Конечные автоматы- устройства содержащие в своей структуре обратную связь, в каждый момент времени состояние выходов зависит и от входных воздействий, и от предыдущих состояний выходов.

КА служат для хранения информации и её преобразования с учетом предыдущих состояний.

К устройствам этого класса относят: Триггеры и все устройства, построенные на основе триггерной системы (счетчики, регистры)

6. Простейший конечный автомат. Принцип функционирования, описание, таблица истинности асинхронного RS- триггера.

Простейшим конечным автоматом является триггер. Триггер – устройство, имеющее 2 устойчивых состояния (уровень 0 и уровень 1). Различают триггеры переключательного типа и триггеры установочного типа. Триггер переключательного типа изменяет свое состояние на противоположное после каждого поступившего на вход импульса.

Пример – Т-триггер. Этот триггер имеет тактовый вход Т и единственный выход Q.

Триггеры переключательного типа имеют ограниченную область применения. В основном устройства вычислительной техники строятся на основе триггеров установочного типа.

Простейшей ячейкой установочного типа является асинхронный RS-триггер. Такой триггер имеет два выхода - прямой и инверсный (устойчивым состоянием триггера всегда считается состояние прямого выхода Q, подтвержденное своей инверсией), и два входа S, Set – вход установки 1 и R, Reset – вход установки 0. Запишем таблицу функционирования триггера. Управляющим уровнем для такой структуры является 0.

При отсутствии управления триггер находится в режиме хранения информации, подача управления на R-вход приводит к установке триггера в 0, подача управления на S-вход к установке 1, подача управления на оба входа одновременно приводит к неопределенности (на прямом и инверсном выходах уровень 1). Выход из такого состояния в режим хранения не определяется. Поэтому подача управления на оба входа называется запрещенной комбинацией.

7. Функционирование D- триггера с динамическим синхровходом.

Асинхронные входы S и R. По информационным входам в триггер поступает информация, которая записывается в момент прихода фронта синхроимпульса (динамический синхровход) и сохраняется в триггере до следующего момента записи). По установочным входам триггер устанавливается в предписанное входом состояние независимо от состояния синхровхода и информации. На время действия установочного сигнала синхровход блокируется.

Асинхронная RS-ячейка – основа построения всех триггеров установочного типа.

вход синхронизации – C

информационный вход - D

Эта схема называется схемой 3-х триггеров. Действительно, триггер, образованный элементами 1 и 2, позволяет записать 0 в момент прихода синхроимпульса и удерживать его до следующей записи, триггер, образованный элементами 3 и 4, позволяет записать и удерживать 1, триггер на элементах 5 и 6 – основная ячейка. Из схемы очевидно функционирование управляющих входов.

8. Типы триггеров. Назначения входов, описание функционирования.

К основным типам триггеров относятся:

- триггер с раздельной установкой состояний (RS-триггер),
- триггер "защелка" (D - триггер),
- универсальный триггер (JK - триггер),
- триггер со счетным входом (T - триггер).

По способу записи информации триггеры подразделяются на асинхронные и синхронные или тактируемые, а по способу управления - на триггеры со статическим управлением (единичным или реже нулевым уровнем тактового сигнала) и триггеры с динамическим управлением (положительным - из 0 в 1, или отрицательным - из 1 в 0 фронтом тактового сигнала). В последнем случае говорят о триггерах с прямым или инверсным динамическим входом управления.

Входы называются *статическими*, если они имеют непосредственную связь с источником входных сигналов. Сигналом для управления статическим триггером с прямыми статическими входами является уровень лог. «1», а для управления триггером с инверсными входами — уровень лог «0».

Входы называются *динамическими,* если они соединены с источником входных сигналов через развязывающие цепи: магнитные, электронные или RC-цепи. Они реагируют только на перепады входных сигналов. Если срабатывание триггера происходит при изменении входного сигнала от «0» к «1», то входы называются прямыми, а если при изменении сигнала от «1» к «0», то — инверсными.

Входы Ś и Ŕ называются входами асинхронной установки триггера. Они предназначены для подачи приоритетных сигналов установки триггера в исходное состояние (0 или 1) в начале цикла работы независимо от воздействия информационных сигналов, то есть в обход схемы управления.

9. Счетчики. Классификация по порядку счета.

Счетчики – конечные автоматы, каждое последующее состояние которых на 1 отличается от предыдущего.
Устройство на выходах которого получается двоичный код, определяемый числом поступивших импульсов. Счетчики могут строиться на D, T и JK триггерах.
Счетчики удобнее строить на основе JK-триггеров
Счетчики могут классифицироваться по порядку счета (суммирующие, вычитающие и реверсивные).
В суммирующем счетчике, где изменение каждого последующего разряда возможно лишь при установке предыдущих в 1, уравнение связей записывается как
JKi = Q0 & Q1&…& Qi-1 ;
Для вычитающего счетчика, где разряды должны устанавливаться в 0, эта связь будет записываться как
JKi = nQ0 & nQ1&…& nQi-1 ;
В реверсивном счетчике, где вводится дополнительный сигнал – реверс (REV), позволяющий производить вычитание при значении REV=1, уравнение связи выглядит следующим образом:
JKi =(Q0 xor REV) & (Q1 xor REV) &…&(Qi-1 xor REV);
Очевидно, что сигнал REV позволяет переключать связи, определяющие порядок счета.

10. Счетчики. Классификация по способу синхронизации и коэффициенту счета.

Счетчики могут классифицироваться:
по способу подачи синхроимпульса
(асинхронные и синхронные) и
по количеству состояний в цикле счета (двоичные – K = 2n и недвоичные – K < 2n, где n – количество разрядных триггеров).
В асинхронных счетчиках порядок счета определяется связями между выходами разрядных триггеров и входами синхронизации. Связь с прямого выхода – вычитающий счетчик, с инверсного – суммирующий.
Асинхронные счетчики имеют большую задержку установки состояния, так как задержка счетчика накапливается из задержек триггеров. По этой причине используются, в основном, синхронные счетчики. В таких счетчиках порядок счета устанавливается связями между выходами триггеров и их информационными входами.
по модулю счёта:
двоично-десятичные (декада);
двоичные;
с произвольным постоянным модулем счёта;
с переменным модулем счёта;

11. Принцип построения синхронных счетчиков и конечных автоматов с произвольной сменой состояний.

Недвоичный счетчик легко получить из двоичного, используя установочные входы разрядных триггеров. Так, например, если необходимо построить суммирующий счетчик, имеющий состояния от 1 до 5, то его можно получить из 3- разрядного двоичного суммирующего счетчика с начальной установкой в 1 (по входам S0, R1, R2), которая производится сигналом, формируемым из состояния, следующего за конечным (6).

Более сложен синтез конечных автоматов с произвольной сменой состояний. Существуют два типа таких автоматов: автомат Мура и автомат Мили. В первом случае переход в следующее состояние зависит только от предыдущего, а во втором появляются также сигналы внешних воздействий. В любом случае для синтеза необходимо определить уравнения связей информационных входов с выходами разрядных триггеров. Для этого по имеющейся в задании таблице переключений строится таблица воздействий, с помощью которой возможна запись уравнений связей.

12. Регистры. Классификация.

Регистры – конечные автоматы, служащие для сдвига информации (последовательные регистры) и для хранения информации (параллельные регистры).
используемое для хранения n-разрядных двоичных чисел и выполнения преобразований над ними.
Последовательный регистр имеет вход данных и один выход данных.
Параллельный регистр имеет количество входов и выходов данных, обеспечивающих разрядность подключенной шины.

13. Последовательные регистры. Принцип построения и функционирования.

Последовательный регистр имеет вход данных и один выход данных.
Данные идут по одному проводу (однопроводная схема)
Запись информации – на фронте синхроимпульса. Если задержка триггера сравнима с длительностью фронта синхроимпульса, информация на одном такте может переписаться в несколько триггеров одновременно, во избежание такой ситуации между триггерами ставятся линии задержки или триггер строится по двухтактной схеме. По спаду синхроимпульса информация выходит из триггера.
Последовательные (сдвигающие) регистры представляют собою цепочку разрядных схем, связанных цепями переноса. Основной режим работы — сдвиг разрядов кода от одного триггера к другому на каждый импульс тактового сигнала.
В однотактных регистрах со сдвигом на один разряд вправо слово сдвигается при поступлении синхросигнала. Вход и выход последовательные

14. Параллельные регистры. Буферные регистры.

Параллельный регистр имеет количество входов и выходов данных, обеспечивающих разрядность подключенной шины.
Регистр, входы и выходы которого всегда подключены к шинам, называется регистр-защелка.
В момент поступления синхроимпульса, состояние шины D передается шине Q.
Синхроимпульсы в этом случае не следуют в общем потоке, а вырабатываются как отдельные управляющие сигналы. Пример, разделения общей внешней шины на шину адреса и шину данных внутри микро процессорной системы. Защелки используются для разделения шин.
Регистр, входы и выходы которого подключаются к шине по управляющему сигналу – буферный регистр.
Запись производится при поступлении фронта синхросигнала. C! – управление буфером записи. C2 – управление буфером считывания.
Если на одной шине подключено множество регистров, используются буферные регистры.
Всегда запись происходит в момент прихода синхроимпульса
C1=1+SYNC - условие записи
C2=1 – условие считывания

15. Структуры регистров с различными способами записи и считывания.

По способу записи-считывания различают также последовательно-параллельный и параллельно-последовательный регистры.
параллельные - запись и считывание информации происходит одновременно на все входы и со всех выходов;
последовательные - запись и считывание информации происходит в первый триггер, а та информация, которая была в этом триггере, перезаписывается в следующий - то же самое происходит и с остальными триггерами;
комбинирован
Параллельно-последовательные регистры
N – шина, по которой идет информация
В момент SYNC регистр записывает состояние N+ при условии что ena=1
D(i)=N(i)*ena - первый момент времени. После записи мы проталкиваем информацию, тогда D(i)=N(i)*ena V Q(i-1)* ena. Следующая запись будет через 13 тактов, подача сигналов “ena”=1 регулирует счетчик.
Последовательно-параллельные регистры
C2=1, тогда мы считываем информацию. МЫ читаем информацию с целью записать в парал. регистр когда подается синхроимпульс. Чтобы верно считать информацию мы ставим буфер чтения (он запрещает сдвиг информации), при C2=1 «ena” д.б. равна 0 (подача регулируется счетчиком.

16. Регистровые файлы. Принцип построения. Устройства в которых используются регистровые файлы.

Регистровый файл — запоминающее устройство процессора со сверхбыстрым доступом для хранения данных, участвующих в обработке, а также управляющей информации процессора.
Регистровый файл — модуль микропроцессора (CPU), содержащий в себе реализацию регистров процессора.
Любое оборудование, построенное на многоразрядных ячейках можно назвать регистровым файлом, если аппаратно устанавливается регламент обращения для записи и чтения любой ячейки. (ячейка – буферно-параллельный регистр).
Управление по записи и считыванию с ячеек строится с использованием счетчиков и управляющих КЦУ. Устройства построенные на основе регистровых файлов: LIFO – стэк память, это память временного хранения информации, кто последний пришел – первый вышел),
память FIFO. Должен быть REV-счетчик, определяющий число читаемых ячеек (запрет), в котором не было записи. Для стека подключаем буферы запись/чтение. Счетчик физически не связан со входами разрешающими запись/чтение, а только отмечает адрес текущий верхнего регистра.

17. Устройства памяти. Общая классификация.

Память в любой системе можно разделить на 2 типа: Внутреннюю и Внешнюю.

Внутренняя память микропроцессорной системы по способу доступа к ячейкам накопителя делится на 3 типа: адресная память, память с последовательным доступом и ассоциативная память.

В адресной памяти доступ к любой ячейке накопителя возможен по любому выставленному на шине адресу, независимо от предыдущего обращения.

В памяти с последовательным доступом порядок обращения к ячейкам задается счетчиком адресов, который невозможно переустановить в процессе работы с памятью. Таким образом, адрес обращения к каждой последующей ячейке отличается от предыдущего всегда на определенную величину.

В ассоциативной памяти ячейка накопителя имеет два адреса: первый – сохраненное значение адреса основной памяти, откуда копировалась информация, второй – адрес внутри организованного дополнительного пространства памяти. Обращение по второму адресу происходит только при совпадении адреса происходящего обращения и первого адреса.

Пример: КЭШ-память.

18. Память с последовательным доступом. Принцип построения стека.

Примеры памяти с последовательным доступом: память FIFO(first input, first output) и память LIFO(last input, first output), или stack.

LIFO (акроним **Last In, First Out**, «последним пришёл — первым ушёл») — способ организации и манипулирования данными относительно времени и приоритетов. В структурированном линейном списке, организованном по принципу LIFO, элементы могут добавляться и выбираться только с одного конца, называемого «вершиной списка». Структура LIFO может быть проиллюстрирована на примере стопки тарелок: чтобы взять вторую сверху, нужно снять верхнюю, а чтобы снять последнюю, нужно снять все, лежащие выше.

FIFO (акроним **First In, First Ou**t — «первым пришёл — первым ушёл») — способ организации и манипулирования данными относительно времени и приоритетов. Это выражение описывает принцип технической обработки очереди или обслуживания конфликтных требований путём упорядочения процесса по принципу: «первым пришёл — первым обслужен» (ППО). Тот, кто приходит первым, тот и обслуживается первым, пришедший следующим ждёт, пока обслуживание первого не будет закончено, и так далее.

Этот принцип аналогичен поведению лиц, стоящих в очереди, когда люди получают обслуживание в том порядке, в котором они занимали очередь.

Стек представляет собой регистр сдвига, ячейками которого являются параллельные регистры. Для просмотра доступен только первый регистр (нулевой разряд регистра сдвига), адрес которого на каждом такте меняется, обращение регламентируется с помощью реверсивного счетчика. При включении rev, идет запись. Информация продвигается от D(i) к D(i+1)

Структура n-разрядного стека глубиной k. Вершина стека – RG0.

19. Адресная память. Принцип построения статических ОЗУ.

Адресная память основана на размещении данных в памяти по адресам хранения, в качестве которых служат номера ячеек. Код адреса однозначно определяет номера ячеек, в которых находится требуемая информация.
Пример ROM, RAM: DRAM, SRAM (динамическая, статическая память)

Примеры адресной памяти: постоянные запоминающие устройства ROM и оперативные запоминающие устройства RAM.

Постоянные запоминающие устройства, имеющие диодные накопители, могут программироваться однократно (ROM, PROM). Постоянные запоминающие устройства, матрицы накопителей которых построены на полевых транзисторах, возможно многократно перепрограммировать. Это могут быть схемы с ультрафиолетовым стиранием информации (EPROM), или с электрическим стиранием (EEPROM). В матрице накопителя постоянного запоминающего устройства строки подключены к выходам дешифратора адреса, а столбцы – к шине данных.

Оперативные запоминающие устройства делятся на 2 класса по структуре матриц накопителя. Это динамические ОЗУ (DRAM), имеющие ячейки накопителя емкостного типа, и статические ОЗУ (SRAM), в которых накопители строятся на основе триггерных ячеек.

CS- вход выборки кристалла, W/R – запись/чтение, по N адресным линиям мы можем определить одну из двух ячеек матрицы. N=K/2

Все входы “S” и“R” объединены для одной матрицы. К одной матрице подходит одна линия данных. (т.к в каждый момент времени мы можем активировать только одну ячейку).

Подключение линии данных

«0» - вход активен по нулю

Если активен буфер чтения, то мы подключаем ячейку к внутренней линии и мы подкл. ее к внешней линии.

Вход CS – вход адресной области. Если выбор входа CS производится по старшим адресным линиям, это строничная организация, если по младшим – банковая организация.

Регенерация и восстановление не требуется. Высокое быстродействие.

20. Адресная память. Принципы функционирования динамических ОЗУ.

Ячейками накопителя являются увеличенная емкость затвор- сток полевых транзисторов. Емкость мгновенно отработать не может. Активизация ячейки по строке и столбцу производится в 2 этапа. Для этого шина адреса мультиплексируется. Четные линии несут адрес строки, нечетные адрес столбца. Ко времени поступления нуля на вход RAS, вход мультиплекс. ADR в нуле, на входах схемы адрес строки. По срезу (нулев. фронту) импульса на вход RAS адрес записывается в регистре защелки адреса строки. Выходы этого регистра соединены с дешифратором убирающем строку. Выбранная строка переписывается в регистр (при этом емкости разрешаются).

Такой памяти необходима постоянная регенерация и восстановление состояния ячеек после обращения. Такая память имеет низкое быстродействие.

21. Адресная память. ПЗУ. Общая классификация.

Постоянное запоминающее устройство (ПЗУ) — энергонезависимая память, используется для хранения массива неизменяемых данных.

В постоянную память часто записывают микропрограмму управления техническим устройством: телевизором, сотовым телефоном, различными контроллерами, или компьютером.
По структуре ячеек в матрице накопителя все ПЗУ можно разделить на 2 группы:

1. Ократнопрограммируемые на диодах.
2. Многократнопрограммируемые на транзисторах.

По разновидностям микросхем ПЗУ

По технологии изготовления кристалла:

ROM — (англ. read-only memory, постоянное запоминающее устройство), масочное ПЗУ, изготавливается фабричным методом. В дальнейшем нет возможности изменить записанные данные.

PROM — (англ. programmable read-only memory, программируемое ПЗУ (ППЗУ)) — ПЗУ, однократно «прошиваемое» пользователем.

EPROM — (англ. erasable programmable read-only memory, перепрограммируемое/репрограммируемое ПЗУ (ППППЗУ/РПЗУ)).

(Память со стиранием с помощью фототока). Например, содержимое микросхемы К537РФ1 стиралось при помощи ультрафиолетовой лампы. Для прохождения ультрафиолетовых лучей к кристаллу в корпусе микросхемы было предусмотрено окошко с кварцевым стеклом.

EEPROM — (англ. electrically erasable programmable read-only memory, электрически стираемое перепрограммируемое ПЗУ). Память такого типа может стираться и заполняться данными несколько десятков тысяч раз. Используется в твердотельных накопителях. Одной из разновидностей EEPROM является флеш-память (англ. flash memory

22. Адресная память. Принцип построения репрограммируемых ПЗУ.

По способу *программирования* микросхем (записи в них прошивки):

1. **Непрограммируемые ПЗУ;**
2. **ПЗУ, программируемые только с помощью специального устройства** — *программатора ПЗУ* (как однократно, так и многократно прошиваемые). Использование программатора необходимо, в частности, для подачи нестандартных и относительно высоких напряжений (до +/- 27 В) на специальные выводы.
3. **Внутрисхемно (пере)программируемые ПЗУ** (*ISP, in-system programming*) — такие микросхемы имеют внутри генератор всех необходимых высоких напряжений, и могут быть перепрошиты без программатора и даже без выпайки из печатной платы, программным способом

EPROM (англ. **Erasable Programmable Read Only Memory**) — класс полупроводниковых запоминающих устройств, постоянная память, для записи информации (программирования) в которую используется электронное устройство-программатор и которое допускает перезапись. Представляет собой матрицу транзисторов с плавающим затвором индивидуально запрограммированных с помощью электронного устройства, которое подаёт более высокое напряжение, чем обычно используется в цифровых схемах. В отличие от PROM, после программирования данные на EPROM можно стереть (сильным ультрафиолетовым светом от ртутного источника света). EPROM легко узнаваем по прозрачному окну из кварцевого стекла в верхней части корпуса, через которое виден кремниевый чип и через которое производится облучение ультрафиолетовым светом во время стирания.

Каждый бит памяти EPROM состоит из одного полевого транзистора. Каждый полевой транзистор состоит из канала в полупроводниковой подложке устройства. Контакты истока и стока подходят к зонам в конце канала. Изолирующий слой оксида выращивается поверх канала, затем наносится проводящий управляющий электрод (кремний или алюминий), и затем ещё толстый слой оксида осаждается на управляющем электроде. Плавающий затвор не имеет связи с другими частями интегральной схемы и полностью изолирован от окружающих слоёв оксида. На затвор наносится управляющий электрод, который затем покрывается оксидом.
Для извлечения данных из EPROM адрес, представляющий значение нужного контакта EPROM, декодируется и используется для подключения одного слова памяти (как правило, 8-битного байта) к усилителю выходного буфера. Каждый бит этого слова имеет значение 1 или 0, в зависимости от того, был включён или выключен транзистор, был он в проводящем состоянии или непроводящем.
Переключение состояния полевого транзистора управляется напряжением на управляющем затворе транзистора. Наличие напряжения на этом затворе создаёт проводящий канал в транзисторе, переключая его в состояние «включено». По сути накопленный заряд на плавающем затворе позволяет пороговому напряжению транзистора программировать его состояние.
Для запоминания данных требуется выбрать нужный адрес и подать более высокое напряжение на транзисторы. Это создаёт лавинный разряд электронов, которые получают достаточно энергии, чтобы пройти через изолирующий слой окисла и аккумулироваться на управляющем электроде. Когда высокое напряжение снимается, электроны оказываются запертыми на электроде. Из-за высокой изолирующей величины оксида кремния, окружающего затвор, накопленный заряд не может утечь, и данные в нём хранятся в течение десятилетий.

23. Адресная память. Принцип построения многократно программируемых ПЗУ.

EEPROM (англ. *Electrically Erasable Programmable Read-Only Memory*) — электрически стираемое перепрограммируемое ПЗУ (ЭСППЗУ), один из видов энергонезависимой памяти (таких как PROM и EPROM). Память такого типа может стираться и заполняться данными до миллиона раз.
На сегодняшний день классическая двухтранзисторная технология EEPROM практически полностью вытеснена флеш-памятью типа NOR. Однако название *EEPROM* прочно закрепилось за сегментом памяти малой ёмкости независимо от технологии.П
Принцип работы EEPROM основан на изменении и регистрации электрического заряда в изолированной области (кармане) полупроводниковой структуры.[1]
Изменение заряда («запись» и «стирание») производится приложением между затвором и истоком большого потенциала чтобы напряженность электрического поля в тонком диэлектрике между каналом транзистора и карманом оказалась достаточна для возникновения туннельного эффекта. Для усиления эффекта тунеллирования электронов в карман при записи применяется небольшое ускорение электронов путем пропускания тока через канал полевого транзистора (эффектHot carrier injection (англ.)русск.).
Чтение выполняется полевым транзистором, для которого карман выполняет роль затвора. Потенциал плавающего затвора изменяет пороговые характеристики транзистора что и регистрируется цепями чтения.
Основная особенность классической ячейки EEPROM — наличие второго транзистора, который помогает управлять режимами записи и стирания. Некоторые реализации выполнялись в виде одного трехзатворного полевого транзистора (один затвор плавающий и два обычных).
Эта конструкция снабжается элементами которые позволяют ей работать в большом массиве таких же ячеек. Соединение выполняется в виде двумерной матрицы, в которой на пересечении столбцов и строк находится одна ячейка. Поскольку ячейка EEPROM имеет третий затвор то помимо подложки к каждой ячейке подходят 3 проводника (один проводник столбцов и 2 проводника строк).

24. Общая классификация кристаллов программируемой логики.

Как уже отмечалось ранее, существует тип кристалла, в котором все связи могут быть запрограммированы пользователем. Рассмотрим структуры таких кристаллов. Структура любого кристалла базируется на p-n переходе, но, соответственно возможностям описания цифрового устройства, эти переходы могут группироваться для построения матриц логических элементов (И – ИЛИ), или же для построения матриц простейших таблиц
функционирования. Таким образом, ПЛИС делятся на два различных класса. Конструктивно ПЛИС состоит из внешней части, содержащей буферные и различные адаптирующие элементы, и внутренней части, состоящей из логических блоков, системы межсоединений этих блоков и элементов памяти конфигурации.
Кристалл- программи́руемая логи́ческая интегра́льная схе́ма, используемая для создания цифровых интегральных схем.CPLD – комплексные программируемые логические устройства FPGA – программируемые пользователем вентильные матрицы. Любая логическая схема имеет внутреннюю и внешнюю часть. Схема устройства строится во внутренней части. Во внешней части находятся контактные площадки и все адаптирующие элементы. Любая внутренняя часть для любого типа состоит из логических блоков, определенным образом соединенных между собой. При рассмотрении класса кристалла необходимо рассматривать структуру логического блока, систему межсоединений и строение памяти конфигураций.

CPLD (англ. complex programmable logic device — сложные программируемые логические устройства) содержат относительно крупные программируемые логические блоки — макроячейки, соединённые с внешними выводами и внутренними шинами.
FPGA (англ. field-programmable gate array) содержат блоки умножения-суммирования, которые широко применяются при обработке сигналов (DSP), а также логические элементы (как правило, на базе таблиц перекодировки — таблиц истинности) и их блоки коммутации. FPGA обычно используются для обработки сигналов, имеют больше логических элементов и более гибкую архитектуру, чем CPLD.

25. Принцип построения ПЛМ и ПМЛ.

Выполнение логических операций микроконтроллером или программируемой логической матрицей (ПЛМ) позволяет унифицировать логическую часть автоматических устройств любой сложности. Такая микросхема содержит инверторы, элементы И и ИЛИ, соединенные друг с другом определенным образом, в том числе и с помощью легкоплавких перемычек (проводников) внутри самой микросхемы. Путем электрической настройки (иначе программирования) ненужные связи между элементами удаляются (перемычки пережигаются), а требуемые оставляются.
Недостаток такой архитектуры - слабое использование ресурсов программируемой матрицы "ИЛИ".
Дальнейшее развитие получили микросхемы, построенные по архитектуре программируемой матричной логики (PAL - Programmable Array Logic) - это ПЛИС, имеющие программируемую матрицу "И" и фиксированную матрицу "ИЛИ". К этому классу относятся большинство современных ПЛИС небольшой степени интеграции.

26. Общая структура CPLD.

Первый класс, CPLD, имеет структуру логического блока, представленную устройством ПМЛ (программируемой матричной логики) с параметрами для кристаллов фирмы Altera 36x80x16. Т.е. блок содержит 36 входов, 80 термов (элементов И) и 16 выходов (элементов ИЛИ).
Матрица ИЛИ для ПМЛ связана, поэтому логический блок состоит из 16 макроячеек, в каждой из которых в элемент ИЛИ возможно подключение 5 термов. Для увеличения количества термов, включаемых в ИЛИ, в структуре блока содержится параллельный логический расширитель, для расширения состава терма служит разделяемый логический расширитель, подключающий инверсный выход 5 терма каждой макроячейки для доступа всем термам своего логического блока. Для сохранения информации предыдущего такта в состав макроячейки входит триггер.
Система межсоединений CPLD представлена программируемой матрицей соединений, позволяющей соединить любую макроячейку с кристалла с другой, в котором из логических блоков она бы ни находилась. Такая структура строится по принципу программируемой логической матрицы (ПЛМ), матрица ИЛИ в которой полнодоступна.
Система межсоединений, построенная на основе гибкой логики, позволяет предсказать задержки в схеме.

27. Структура макроячейки CPLD.

Каждая МЯ содержит 1 триггер (для хранения предыдущего состояния функции). Максимальное количество логических блоков для таких структур-16. ПМС соединенный имеет структуры ПЛМ, таким образом каждая МЯ м.б. соединена с другой. Каждому из 36 выходов ПМС могут быть подсоединены как глобальные линии (синхронизация, сброс, установка буфера), так и локальные, идущие от внешних входов и выходов МЯ, представляющие собой промежуточные входы блоков.

28. (26) Принципы построения вентильных матриц (GA). Общая классификация.

Вентиль – простейший базовый логический элемент И-НЕ или ИЛИ-НЕ. Эквивалентный вентиль – то количество переходов, из которых можно составить базовый логический элемент.
Внутренняя часть такого кристалла представляет собой матрицу логических блоков.
ЛБ для простейших схем вентильных матриц – это выделенное пространство переходов на основе которых строится узел схемы.
ЛБ объединяются между собой через линии каналов.
В первых вариантах вентильных матриц пытались внедрить бесканальную
структуру, т.е. соединение логических блоков через свободные переходы на границах
блоков. Однако такая структура не выносит работу на высоких частотах. Каналы
также служат для отвода тепла.

29. (28)Структура логических блоков FPGA.

Для FPGA первых двух поколений ЛБ разделились по структуре «зерна»: структура мелкого зерна повторяла структуру вентильной матрицы (ЛБ на основе цепочек логических элементов).
Структура среднего зерна строилась на основе мультиплексоров.
Структура среднего зерна базируется на запоминающих устройствах из 16-ти ячеек, которые носят название LUT (Look Up Tables).

Она выглядит так:

Основные недостатки первых 2х поколений это неопределенность задержки на каналах и жесткость конструкции ЛБ

30. (29) Система межсоединений FPGA. Принцип построения ПМС CPLD.

В первых структурах система соединений была безканальная, ЛБ соединялись через граничные переходы, в последующих поколениях появились каналы между ЛБ. Затем блоки стали делить на подблоки (макроячейки) которые объединялись между собой через локальную матрицу соединений, а ЛБ через глобальную. Локальная матрица соединений использует принцип ПМС в CPLD.
Объединение только с помощью соединительных линий снижает гибкость функционирования схемы. Все эти недостатки устранялись в FPGA следующих поколений. Так, в схемах 3 поколения, ЛБ составлялись из подблоков – микроячеек, объединяемых с помощью локальной матрицы соединений, построенной на основе комбинаторной логики. Сами ЛБ соединены с помощью локальной матрицы соединений - канала. Кроме того в состав ЛБ стал входить встроенный блок памяти (ВБП – это память типа SRAM, объемом 2к (т.е 2048 ячеек) с возможностью переконфигурации.)

31. (30) Память конфигурации FPGA. Распределенная и выделенная память. Память конфигурации CPLD.

Память конфигурации построена на триггерх. Память конфигурации на основе SRAM может быть распределенной и выделенной.
Встроенные блоки памяти – это выделенная память. Она служит для функционирования конфигураций (LUT-**выделенная память**, служащая для построения конфигураций)
Распределенная память- это триггеры внутри каждой ячейки и триггеры в точках соединений на каналах.
CPLD: Память конфигураций построена на основе EEPROM (память распределенная энергонезависимая).

32. (31) Основные различия кристаллов CPLD и FPGA. Алгоритмы программирования под CPLD и FPGA

Принципиальные различия заключаются в том, что в CPLD содержатся укрупненные ЛБ на основе элементов 2И-НЕ, 2ИЛИ-НЕ, а в FPGA в структуре содержатся более компактные ЛБ на основе таблицы истинности, количество ячеек FPGA превышает кол-во ячеек в CPLD на 1-2 порядка.
Алгоритмы программирования интегральных схем заключается в переносе конфигурации матрицы соединения на память интегральной схемы. CPLD содержит энергонезависимую память, поэтому при старте системы ее не надо каждый раз загружать, она сохраняется, а в FPGA при включении каждый раз надо заново загружать конфигурацию.

33. (32) Структуры кристаллов последних поколений.

В схемах 4го и 5го (FPGA) поколений ЛБ объединяются в мегаблоки. Кроме ЛБ в состав магеблока входят встроенные блоки памяти существенно больше емкости (не 2К, а 16К), схема быстрого умножителя 16на16 и блок управления частотой, в котором кроме изменения фазы основной частоты есть возможность умножения и деления основной частоты, а так же изменение фазы полученной частоты. ЛБ в такой системе условно разделился на 2 части «правую» и «левую», в левой таблицы (LUT), в правой комбинаторная логика объединенная с помощью каналов только для мегаблоков.

34. (33) протокол JTAG. Структура TAP.

Протокол нужен для проверки(тестирования) схем. //К группе для разработки решения проблем тестирования интегральных схем присоединились представители североамериканских компаний, и название было изменено на Joint Test Action Group (JTAG).//
TAP(порт тестирования, через который информация передается на плату(схему, которую мы тестируем)) состоит из 2х основных частей: контролера и сдвигового регистра, состоящий из ячеек граничного сканирования (BSC), расположенный во внешней области кристалла.
Контролер это КА на 16 состояний переключения из одного состояния в другое зависимо от выбора режима:
-1ый связь ячейки с внутренней частью,
-2ой-связь ячейки с контактной площадкой,
-3ий это и то и другое.

35. (34) Протокол JTAG. Режим работы ячейки граничного сканирования.

Протокол нужен для проверки схем.
Режим работы ячейки:
1ый режим-режим считывания конфигурации или программирования,
2ой-чтение соединений на плате(соединение с контактной площадкой),
3ий-проверка функционирования платы.
Ячейка состоит из 2х триггеров. Первая ячейка связана с контактной площадкой, запись всегда производится в первую ячейку, а затем сдвигается в следующие. С1 осуществляет сдвиг в n-ую ячейку, от с2

синхроимпульс позволяет фиксировать информацию в триггере Т2 и затем вывести на выход.

36. (35) Общая структура микропроцессорной системы.

Любая микропроцессорная система состоит из трех основных компонентов: собственно микропроцессора, производящего операции над данными, области памяти, где хранятся коды программ работы процессора и обрабатываемые массивы данных, и области устройств ввода-вывода, состоящую из схем, адаптирующих процессорную систему к внешним устройствам. Соединение блоков микропроцессорной системы производится по системной шине (СШ).
Основной блок : CPU – Блок центрального процессора

1) I/O – блок ввода-вывода
2) MEM – блок памяти (хранятся и коды и данные)

Системная шина состоит из трех групп шин:
-шины адреса, на которую процессор выставляет адрес устройства в пространстве памяти или пространстве ввода-вывода, с которым будет производится обмен информацией;
-шины данных, по которой производится обмен информацией;
-шины управления, по которой процессор посылает управляющие сигналы и получает запросы от устройств ввода-вывода.
Шина адреса однонаправлена, информация, следующая по ней, многоразрядна. Шина данных двунаправлена и информация, следующая по ней, также многоразрядна. Шина управления состоит из отдельных проводников, каждый из которых передает определенный сигнал управления. Формировать управляющие сигналы может процессорный блок и блок устройств ввода-вывода (запросы на процессорный блок).
Для полноценного обмена процессора по шине данных важны три момента: направление обмена (чтение/запись), объект обмена (память/устройства ввода-вывода) и, наконец, структура информации (данные/код).

37. (36) Производительность микропроцессоров. Типы микропроцессоров.

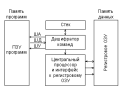
Производительность процессора.
Если процессор работает с тактовой частотой F, то время T=1/F называется тактом. Время выполнения тестовой задачи можно рассчитать через такт
T x C x I,
где C – количество тактов на инструкцию, а I – количество инструкций на задачу.
Соответственно, чем меньше времени затрачивается на решение тестовой задачи, тем производительность процессора выше. В указанном выше выражении уменьшение T ограничено свойствами структуры, поэтому изменение производительности можно достичь изменением I или C.
Рассмотрим две основные архитектуры процессорного ядра. RISC – процессоры (Reduced Instruction Set Computer) и CISC - процессоры (Complete Instruction Set Computer).
Любой тип процессора выполняет инструкции, непрерывным потоком поступающие из памяти по шине данных. Выполнение инструкции можно разбить на 5 этапов:
1 – выборка кода из памяти по выставленному на адресной шине адресу,
2 – дешифрация кода,
3 – исполнение,
4 – получение результата,
5 – обратная загрузка результата.

38. (37) Типы архитектур микропроцессорных систем.

В настоящее время существуют два типа архитектуры микропроцессорных систем – Принстонская, или архитектура фон-Неймана и Гарвардская.
В 1945 г. американский математик Джон фон Нейман сформулировал основные принципы работы современных компьютеров. Им была предложена архитектура, получившая его имя (von Neumann architecture) и предполагающая хранение программ и данных в общей памяти (1946 г.).
Сегодня такая архитектура наиболее характерна для микропроцессоров, ориентированных на использование в компьютерах. Примером могут служить микропроцессоры семейства x86. Эти микропроцессоры относятся к CISC-процессорам.
Имеется общая шина данных для общения с памятью, выделяется место и для кодов и для данных. Каждая команда должна быть дешифрована, разделение кодов и данных производится внутри процессорного блока(демультиплексирование на входе)
Плюсы: пространство памяти можно изменить
Минусы: если необходимо взять данные , получается, что на момент выдачи(записи) операнда, прекращается поток инструкции
Архитектура, предполагающая раздельное использование памяти программ и данных, носит название гарвардской (Harvard architecture). Гарвардская архитектура позволяет центральному процессору работать одновременно как с памятью программ, так и с памятью данных, что существенно увеличивает производительность.
Данная архитектура ориентирована на использование RISC-процессоров.
Инструкции записываются по одной линии, а данные по другой.
Плюсы: процессор не останавливает на время выдачи операнда,
минус: нет возможности распределять пространство памяти.

39. (38)Конвейер. Этапы выполнения операций.

Любой тип процессора выполняет инструкции, непрерывным потоком поступающие из памяти по шине данных. Выполнение инструкции можно разбить на 5 этапов:



1 – выборка кода из памяти по выставленному на адресной шине адресу,
2 – дешифрация кода,
3 – исполнение,
4 – получение результата,
5 – обратная загрузка результата.

Для ускорения процесса работа производится конвейерным способом, т.е. в каждый момент времени одновременно выполняются разные этапы следующих подряд команд. Рассмотренный выше случай – пятиступенчатый конвейер, но для разных процессоров возможно объединение 4 и 5 или 3, 4 и 5 этапов, в этих случаях мы имеем четырех- или трехступенчатый конвейер.
Для CISC – процессоров характерны сложные многотактовые инструкции, производители этих процессоров старались увеличить производительность за счет уменьшения I. Но это приводило к приостановке конвейера, а, следовательно, снова снижало производительность процессора.
RISC – процессоры выполняют простые однотактовые операции. Они, в отличие от CISC не могут выполнять сложные задачи, зато для них C = 1,а так как операции обмена с пространством памяти в RISC выделены в отдельную группу, конвейер работает практически безостановочно и производительность высока.

40. (39) RISC-процессоры. Особенности выполнения операций.

Особенности выполнения операций в RISC процессорах является наличие 2 источников и 1 приемника, поэтому при имеющихся 32-х регистрах невозможно описать все адреса непосредственно в полях команд. (Поле источник и поле приемник пишутся через thumb, кодирование сжатое) От способа кодирования зависит сложность блока дешифрации команд. Формат команд у такого процессора всегда одинаковый – 16 бит и 32 бита.
Структура ядра RISC-процессора предполагает наличие большой внутренней памяти, состоящей из регистров общего назначения, не имеющих дополнительных специальных функций.(Исключение составляет регистр нуля, присутствующий во многих модификациях таких процессоров. Этот регистр всегда хранит 0 и на запись не работает.) С помощью команд прямой и обратной загрузки происходит обмен между регистрами общего назначения и памятью данных. При работе команд, использующих АЛУ (арифметико-логическое устройство), операнды поступают только из внутренней памяти ядра, что экономит время обработки.

41. (40) CISC-процессоры. Особенности выполнения операций.

Для CISC процессора нет строго фиксированного формата, формат м.б. мин 8 бит, 16(слово) 32 (двойное слово) до 40, то есть до 5 байт. CISC свободно общается с памятью следовательно в CISC-процессоре появляется адресация операндов, позволяющая выполнить операцию над числами,1 из которых мб в памяти, а второе всегда в регистре.
За счет указания смещения или непосредственного операнда формат команды может доходить до 5 байт, однако основные сведения должны укладываться на 16 битах.
Код операции для CISC хранится в дополнительной памяти и в поле кода пишется указание на адрес этой памяти, то есть сокращается кол-во бит записи. В операции могут участвовать только 2 регистра (1 источник и 1 приемник), адрес регистра запишется прямо в поле источника и приемника. Для указания размерности источника и полного описания как источника так и приемника в младших разрядах используется 2 служебных бита.
Структура ядра CISC-процессора, напротив, предполагает наличие малого количества регистров общего назначения, к тому же имеющих строго определенные функции. Эти функции обусловлены наличием большого количества указателей и счетчиков, входящих в состав ядра и позволяющих выполнять циклические операции, записанные в одной инструкции. При выполнении операций в АЛУ процессор может пользоваться операндами как хранящимися в регистрах общего назначения, так и в пространстве памяти, выделенном под данные. Таким образом, для написания программ под

42. (41)Способы адресации данных и их особенности для различных типов процессоров.

Адресация: регистровая и непосредственная. При регистровой оба операнда в регистре процессора, при непосредственной источник задан числом (в регистр помещается какое-то число)
При обмене с памятью: прямая адресация: адрес ячейки указывается в тексте команды; косвенная (несколько видов): адрес ячейки записывается в базовом регистре. Различают следующие виды косвенной адресации: базовая, базовая индексная, базовая индексная со смещением.(Пример в базовый регистр BX записываем точку данных, откуда начинаем считать, либо до куда записываем)
Risc: способы адресации данных : для команды прямой и обратной загрузки всегда косвенная адресация (LD, r1,r2, disp) в r1 пишется базовая точка отсчета – начало области памяти данных, дальше дается смешение от этой точки. Если мы загружаем полуслово 16 бит, то берем через 2 ячейки, если грузим 32 бита, то берем через 4 ячейки. При работе внутри ядра регистровая, непосредственная и битовая адресации.
В битовой адресации открывается поле от 3 до 5 бит, на котором может указываться в зависимости от типа команды или код условие или номер изменяемого бита. (условие - флаги. Флаг – состояние триггера в регистре.)
Команды RISC-процессора однотактовые, все КЦУ, участвующие в процессе обработки, как коммутирующие, так и кодопреобразующие образуют цепочки и имеют общую задержку, не превышающую время такта. Команды пересылок между регистрами выполняются как арифметическая операция сложения с 0.
CISC-процессор используется большее количество адресаций данных, чем при программировании под RISC-процессор. Команды CISC-процессора выполняются за несколько тактов, что позволяет не выстраивать коммутационные КЦУ в цепочку, а использовать один коммутатор и служебный регистр (например, в командах пересылок)

43. (42) Структура команды ассемблера.

Формат команды делится на поля строго определённой длинны . Основное поле представляет код операции (COP), эта информация поступает на основной дешифратор. Следующее поле - адреса источников и приемников (пойдут на блок регистров)
Операнд модно положить в один регистр. Формат команд должен быть фиксирован. Единица информации в процессорной системе это 1 байт, поэтому формат команды всегда кратен байту.