

## 2. Организация обмена информацией: Функции устройств магистрали

### 2.4. Функции устройств магистрали

Рассмотрим теперь, как взаимодействуют на магистрали основные устройства микропроцессорной системы: процессор, память (оперативная и постоянная), устройства ввода/вывода.

#### 2.4.1. Функции процессора

Процессор (рис. 23.1) обычно представляет собой отдельную микросхему или же часть микросхемы (в случае микроконтроллера). В прежние годы процессор иногда выполнялся на комплектах из нескольких микросхем, но сейчас от такого подхода уже практически отказались. Микросхема процессора обязательно имеет выводы трех шин: шины адреса, шины данных и шины управления. Иногда некоторые сигналы и шины мультиплексируются, чтобы уменьшить количество выводов микросхемы процессора.

*Важнейшие характеристики процессора — это количество разрядов его шины данных, количество разрядов его шины адреса и количество управляющих сигналов в шине управления. Разрядность шины данных определяет скорость работы системы. Разрядность шины адреса определяет допустимую сложность системы. Количество линий управления определяет разнообразие режимов обмена и эффективность обмена процессора с другими устройствами системы.*

Кроме выводов для сигналов трех основных шин процессор всегда имеет вывод (или два вывода) для подключения внешнего тактового сигнала или кварцевого резонатора (CLK), так как процессор всегда представляет собой тактируемое устройство. Чем больше тактовая частота процессора, тем он быстрее работает, то есть тем быстрее выполняет команды. Впрочем, быстродействие процессора определяется не только тактовой частотой, но и особенностями его структуры. Современные процессоры выполняют большинство команд за один такт и имеют средства для параллельного выполнения нескольких команд. Тактовая частота процессора не связана прямо и жестко со скоростью обмена по магистрали, так как скорость обмена по магистрали ограничена задержками распространения сигналов и искажениями сигналов на магистрали. То есть тактовая частота процессора определяет только его внутреннее быстродействие, а не внешнее. Иногда тактовая частота процессора имеет нижний и верхний пределы. При превышении верхнего предела частоты возможно перегревание процессора, а также сбои, причем, что самое неприятное, возникающие не всегда и нерегулярно. Так что с изменением этой частоты надо быть очень осторожным.

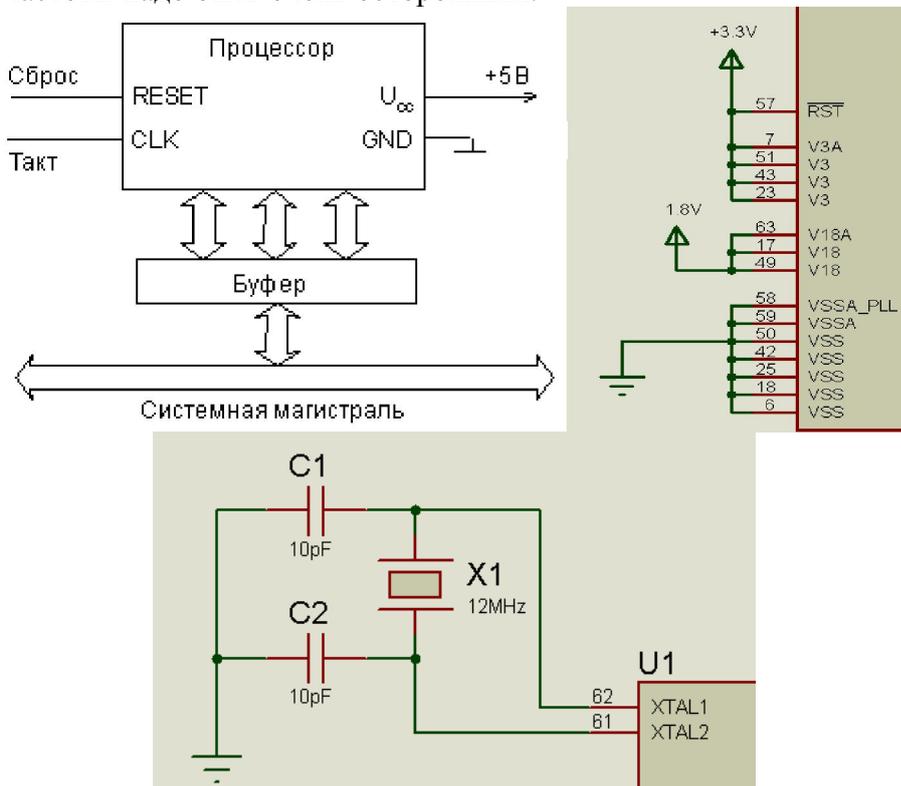


Рис. 23.1. Схема включения процессора

Еще один важный сигнал, который имеется в каждом процессоре, — это сигнал начального сброса RESET. При включении питания, при аварийной ситуации или зависании процессора подача этого сигнала приводит к инициализации процессора, заставляет его приступить к выполнению программы начального запуска (рис 23.2). Аварийная ситуация может быть вызвана помехами по цепям питания и «земли», сбоями в работе памяти, внешними ионизирующими излучениями и еще множеством причин. В результате процессор может потерять контроль над выполняемой программой и остановиться в каком-то адресе. Для выхода из этого состояния как раз и используется сигнал начального сброса. Этот же вход начального сброса может использоваться для оповещения процессора о том, что напряжение питания стало ниже установленного предела. В таком случае процессор переходит к выполнению программы сохранения важных данных. По сути, этот вход представляет собой особую разновидность радиального прерывания.

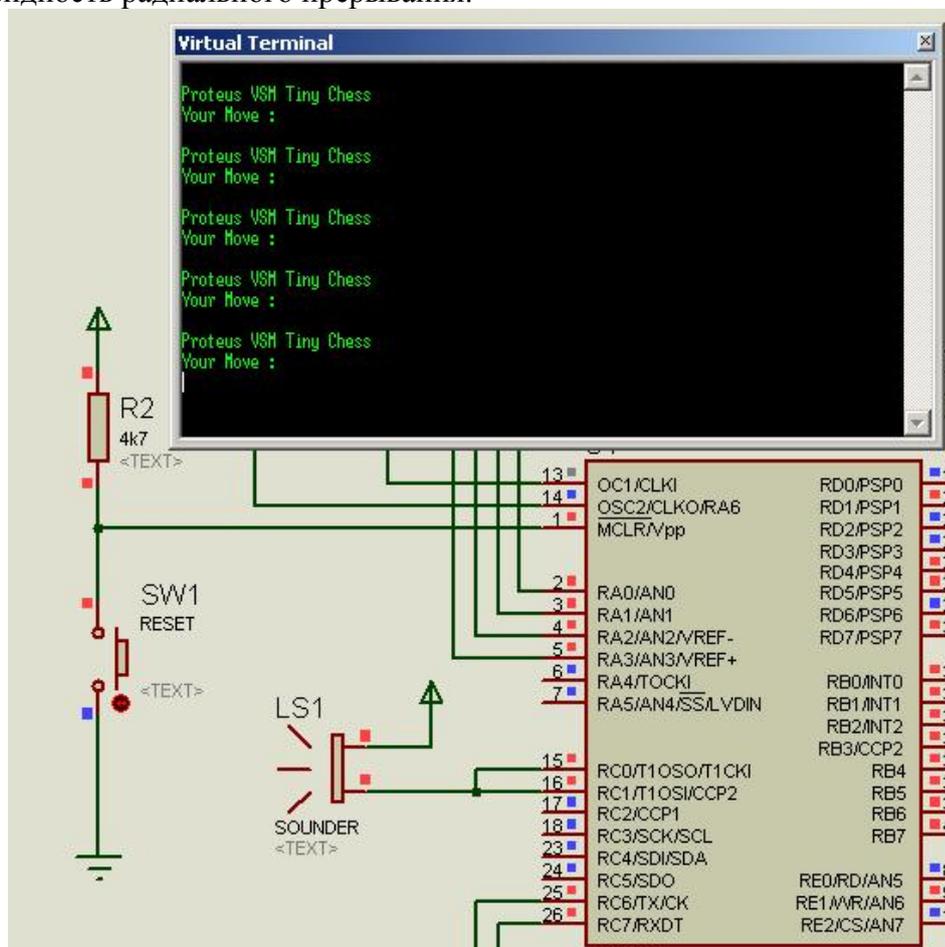


Рис 23.2 Вариант реализации сброса

Иногда у микросхемы процессора имеется еще один-два входа радиальных прерываний для обработки особых ситуаций (например, для прерывания от внешнего таймера).

Шина питания современного процессора обычно имеет одно напряжение питания (+5В или +3,3В) и общий провод («землю»). Первые процессоры нередко требовали нескольких напряжений питания. В некоторых процессорах предусмотрен режим пониженного энергопотребления. Вообще, современные микросхемы процессоров, особенно с высокими тактовыми частотами, потребляют довольно большую мощность. В результате для поддержания нормальной рабочей температуры корпуса на них нередко приходится устанавливать радиаторы, вентиляторы или даже специальные микрохолодильники.

Для подключения процессора к магистрали используются буферные микросхемы, обеспечивающие, если необходимо, демультиплексирование сигналов и электрическое буферирование сигналов магистрали. Иногда протоколы обмена по системной магистрали и по шинам процессора не совпадают между собой, тогда буферные микросхемы еще и согласуют эти протоколы друг с другом. Иногда в микропроцессорной системе используется несколько магистралей (системных и локальных), тогда для каждой из магистралей применяется свой буферный узел. Такая структура характерна, например, для персональных компьютеров.

После включения питания процессор переходит в первый адрес программы начального пуска и выполняет эту программу. Данная программа предварительно записана в постоянную

(энергонезависимую) память. После завершения программы начального пуска процессор начинает выполнять основную программу, находящуюся в постоянной или оперативной памяти, для чего выбирает по очереди все команды. От этой программы процессор могут отвлекать внешние прерывания или запросы на ПДП. Команды из памяти процессор выбирает с помощью циклов чтения по магистрали. При необходимости процессор записывает данные в память или в устройства ввода/вывода с помощью циклов записи или же читает данные из памяти или из устройств ввода/вывода с помощью циклов чтения.

Таким образом, основные функции любого процессора следующие:

- выборка (чтение) выполняемых команд;
- ввод (чтение) данных из памяти или устройства ввода/вывода;
- вывод (запись) данных в память или в устройства ввода/вывода;
- обработка данных (операндов), в том числе арифметические операции над ними;
- адресация памяти, то есть задание адреса памяти, с которым будет производиться обмен;
- обработка прерываний и режима прямого доступа.

Упрощенно структуру микропроцессора можно представить в следующем виде (рис. 23.3).

Основные функции показанных узлов следующие.

**Схема управления выборкой команд** выполняет чтение команд из памяти и их дешифрацию. В первых микропроцессорах было невозможно одновременное выполнение предыдущей команды и выборка следующей команды, так как процессор не мог совмещать эти операции. Но уже в 16-разрядных процессорах появляется так называемый конвейер (очередь) команд, позволяющий выбирать несколько следующих команд, пока выполняется предыдущая. Два процесса идут параллельно, что ускоряет работу процессора. Конвейер представляет собой небольшую внутреннюю память процессора, в которую при малейшей возможности (при освобождении внешней шины) записывается несколько команд, следующих за исполняемой. Читаются эти команды процессором в том же порядке, что и записываются в конвейер (это память типа FIFO, First In — First Out, первый вошел — первый вышел). Правда, если выполняемая команда предполагает переход не на следующую ячейку памяти, а на удаленную (с меньшим или большим адресом), конвейер не помогает, и его приходится сбрасывать. Но такие команды встречаются в программах сравнительно редко.

Развитием идеи конвейера стало использование внутренней кэш-памяти процессора, которая заполняется командами, пока процессор занят выполнением предыдущих команд. Чем больше объем кэш-памяти, тем меньше вероятность того, что ее содержимое придется сбросить при команде перехода. Понятно, что обрабатывать команды, находящиеся во внутренней памяти, процессор может гораздо быстрее, чем те, которые расположены во внешней памяти. В кэш-памяти могут храниться и данные, которые обрабатываются в данный момент, это также ускоряет работу. Для большего ускорения выборки команд в современных процессорах применяют совмещение выборки и дешифрации, одновременную дешифрацию нескольких команд, несколько параллельных конвейеров команд, предсказание команд переходов и некоторые другие методы.

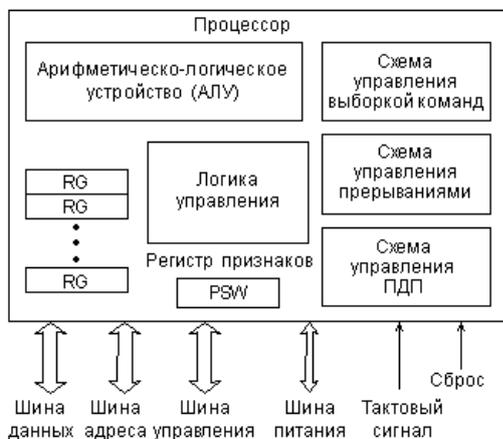


Рис. 23.3. Внутренняя структура микропроцессора.

**Арифметико-логическое устройство** (или *АЛУ*, *ALU*) предназначено для обработки информации в соответствии с полученной процессором командой. Примерами обработки могут служить логические операции (типа логического «И», «ИЛИ», «Исключающего ИЛИ» и т.д.) то есть

побитные операции над операндами, а также арифметические операции (типа сложения, вычитания, умножения, деления и т.д.). Над какими кодами производится операция, куда помещается ее результат — определяется выполняемой командой. Если команда сводится всего лишь к пересылке данных без их обработки, то *АЛУ* не участвует в ее выполнении.

Быстродействие *АЛУ* во многом определяет производительность процессора. Причем важна не только частота тактового сигнала, которым тактируется *АЛУ*, но и количество тактов, необходимое для выполнения той или иной команды. Для повышения производительности разработчики стремятся довести время выполнения команды до одного такта, а также обеспечить работу *АЛУ* на возможно более высокой частоте. Один из путей решения этой задачи состоит в уменьшении количества выполняемых *АЛУ* команд, создание процессоров с уменьшенным набором команд (так называемые RISC-процессоры). Другой путь повышения производительности процессора — использование нескольких параллельно работающих *АЛУ*.

Что касается операций над числами с плавающей точкой и других специальных сложных операций, то в системах на базе первых процессоров их реализовали последовательностью более простых команд, специальными подпрограммами, однако затем были разработаны специальные вычислители — математические сопроцессоры, которые заменяли основной процессор на время выполнения таких команд. В современных микропроцессорах математические сопроцессоры входят в структуру как составная часть.

**Регистры** процессора представляют собой по сути ячейки очень быстрой памяти и служат для временного хранения различных кодов: данных, адресов, служебных кодов. Операции с этими кодами выполняются предельно быстро, поэтому, в общем случае, чем больше внутренних *регистров*, тем лучше. Кроме того, на быстродействие процессора сильно влияет разрядность *регистров*. Именно разрядность *регистров* и *АЛУ* называется **внутренней разрядностью процессора**, которая может не совпадать с **внешней разрядностью**.

По отношению к назначению внутренних *регистров* существует два основных подхода. Первого придерживается, например, компания Intel, которая каждому *регистру* отводит строго определенную функцию. С одной стороны, это упрощает организацию процессора и уменьшает время выполнения команды, но с другой — снижает гибкость, а иногда и замедляет работу программы. Например, некоторые арифметические операции и обмен с устройствами ввода/вывода проводятся только через один *регистр* — **аккумулятор**, в результате чего при выполнении некоторых процедур может потребоваться несколько дополнительных пересылок между *регистрами*. Второй подход состоит в том, чтобы все (или почти все) *регистры* сделать равноправными, как, например, в 16-разрядных процессорах T-11 фирмы DEC. При этом достигается высокая гибкость, но необходимо усложнение структуры процессора. Существуют и промежуточные решения, в частности, в процессоре MC68000 фирмы Motorola половина *регистров* использовалась для данных, и они были взаимозаменяемы, а другая половина — для адресов, и они также взаимозаменяемы.

*Регистр* признаков (*регистр* состояния) занимает особое место, хотя он также является внутренним *регистром* процессора. Содержащаяся в нем информация — это не данные, не адрес, а слово состояния процессора (ССП, PSW — Processor Status Word). Каждый бит этого слова (флаг) содержит информацию о результате предыдущей команды. Например, есть бит нулевого результата, который устанавливается в том случае, когда результат выполнения предыдущей команды — нуль, и очищается в том случае, когда результат выполнения команды отличен от нуля. Эти биты (флаги) используются командами условных переходов, например, командой перехода в случае нулевого результата. В этом же *регистре* иногда содержатся флаги управления, определяющие режим выполнения некоторых команд.

**Схема управления прерываниями** обрабатывает поступающий на процессор запрос прерывания, определяет адрес начала программы обработки прерывания (адрес вектора прерывания), обеспечивает переход к этой программе после выполнения текущей команды и сохранения в памяти (в *стеке*) текущего состояния *регистров* процессора. По окончании программы обработки прерывания процессор возвращается к прерванной программе с восстановленными из памяти (из *стека*) значениями внутренних *регистров*.

**Схема управления прямым доступом к памяти** служит для временного отключения процессора от внешних шин и приостановки работы процессора на время предоставления прямого доступа запросившему его устройству.

**Логика управления** организует взаимодействие всех узлов процессора, перенаправляет данные, синхронизирует работу процессора с внешними сигналами, а также реализует процедуры ввода и вывода информации.

Таким образом, в ходе работы процессора схема выборки команд выбирает последовательно команды из памяти, затем эти команды выполняются, причем в случае необходимости обработки данных подключается *АЛУ*. На входы *АЛУ* могут подаваться обрабатываемые данные из памяти или из внутренних *регистров*. Во внутренних *регистрах* хранятся также коды адресов обрабатываемых данных, расположенных в памяти. Результат обработки в *АЛУ* изменяет состояние *регистра* признаков и записывается во внутренний *регистр* или в память (как источник, так и приемник данных указывается в составе кода команды). При необходимости информация может переписываться из памяти (или из устройства ввода/вывода) во внутренний *регистр* или из внутреннего *регистра* в память (или в устройство ввода/вывода).

Внутренние *регистры* любого микропроцессора обязательно выполняют две служебные функции:

- определяют адрес в памяти, где находится выполняемая в данный момент команда (функция **счетчика команд** или **указателя команд**);
- определяют текущий адрес *стека* (функция указателя *стека*).

В разных процессорах для каждой из этих функций может отводиться один или два внутренних *регистра*. Эти два указателя отличаются от других не только своим специфическим, служебным, системным назначением, но и особым способом изменения содержимого. Их содержимое программы могут менять только в случае крайней необходимости, так как любая ошибка при этом грозит нарушением работы компьютера, зависанием и порчей содержимого памяти.

Содержимое указателя (счетчика) команд изменяется следующим образом. В начале работы системы (при включении питания) в него заносится раз и навсегда установленное значение. Это первый адрес программы начального запуска. Затем после выборки из памяти каждой следующей команды значение указателя команд автоматически увеличивается (инкрементируется) на единицу (или на два в зависимости от формата команд и типа процессора). То есть следующая команда будет выбираться из следующего по порядку адреса памяти. При выполнении команд перехода, нарушающих последовательный перебор адресов памяти, в указатель команд принудительно записывается новое значение — новый адрес в памяти, начиная с которого адресов команд опять же будут перебираться последовательно. Такая же смена содержимого указателя команд производится при вызове подпрограммы и возврате из нее или при начале обработки прерывания и после его окончания.