

Архитектура современных вычислительных устройств

Лекции 7–8. События, многозадачность и устройства ввода-вывода

Макаров П. А.

ноябрь 2024 г.

Содержание

1	События — прерывания и исключения	1
2	Переключение задач и виртуальные машины	4
2.1	Защищённый режим и виртуальная память	5
3	Режимы работы УВВ	6
3.1	Режим программного опроса (по флагу или по ожиданию)	6
3.2	Режим работы по прерываниям	6
3.3	Режим прямого доступа к памяти	8

1 События — прерывания и исключения

Прерывания и исключения — это события, которые указывают на возникновение в системе или в выполняемой в данный момент задаче определённых условий, требующих вмешательства процессора. Возникновение таких событий вынуждает процессор прервать выполнение текущей задачи и передать управление специальной процедуре либо задаче, называемой обработчиком прерывания или исключения. Различные синхронные и асинхронные события в системе можно классифицировать как показано на рис. 1

Все прерывания и исключения идентифицируется номером (неотрицательным целым числом в диапазоне от 0 до 255), который называется вектором прерывания.



Рис. 1: Классификация событий

При возникновении событий процессор сохраняет в стеке содержимое своих регистров и указатель $CS:IP$ на ту инструкцию, которую он должен будет выполнять после обработки события. Далее процессор переходит к исполнению кода обработчика данного события, определяя точку входа в него через номер по таблице прерываний. Номер элементов в таблице прерываний является вектором прерывания. Прерывания всегда обрабатываются на “границе инструкций”, т. е. при появлении сигнала прерывания процессор сначала завершит выполняемую в данный момент инструкцию (или итерацию при наличии префикса повторения), и только потом начнёт обрабатывать прерывание. Помещаемый в стек обработчика адрес очередной инструкции позволяет корректно возобновить выполнение прерванной программы.

Обработчик события должен заканчиваться специальной инструкцией возврата $IRET$, по которой из стека восстанавливаются значения регистров процессора и указатель инструкций.

Прерывания обычно возникают в произвольный момент времени и бывают аппаратными (внешними) и программными.

Внешние прерывания генерируются по аппаратному сигналу, поступающему от периферийного оборудования, когда оно требует обслуживания. Процессор определяет необходимость обработки внешнего прерывания по наличию сигнала на линии запроса прерывания $INTR$ ($INTerrupt Request$). В ответ на поступивший запрос процессор вырабатывает сигнал $INTA$ ($INTerrupt Acknowledgment$) — подтверждение обработки прерывания, поступающий на внешний контроллер прерываний (в ранних моделях персональных компьютеров семейства $IBM PC$ — это микросхема Intel 8259 (PIC — $Programmable Interrupt Controller$), в более поздних — её “продвинутый” вариант $APIC$). После получения этого подтверждения

контроллер прерываний должен предоставить процессору вектор прерывания.

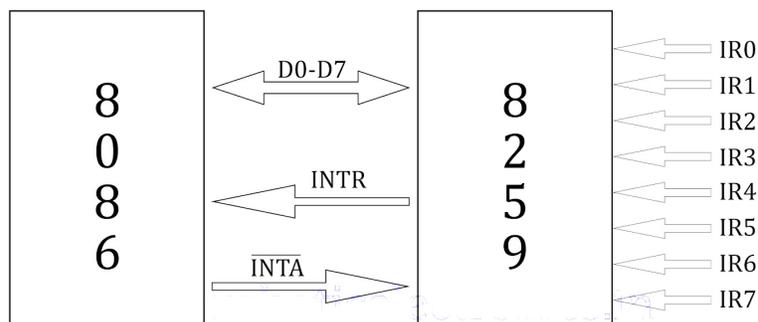


Рис. 2: Механизм взаимодействия центрального процессора и контроллера прерываний

При обработке любого события процессор автоматически сбрасывает флаг разрешения прерываний IF , т. к. возможны вложенные прерывания, то есть те, которые центральный процессор обрабатывает до завершения предыдущих прерываний. Они опасны переполнением стека и, как следствие, зависанием компьютера.

Маскируемые внешние прерывания обрабатываются процессором по сигналу на входе $INTR$ только при установленном флаге разрешения прерываний IF . Немаскируемые прерывания обрабатываются процессором независимо от состояния флага разрешения прерывания IF .

Исключения являются для процессора внутренними событиями и сигнализируют о каких-либо ошибочных условиях при выполнении той или иной инструкции. Источниками исключений являются три типа событий:

- генерируемые программой исключения, позволяющие программе контролировать определённые условия в заданных точках программы (проверка на переполнение, контрольная точка, проверка границ массива);
- исключения машинного контроля, возникающие в процессе контроля операций внутри микросхем и транзакций на шине процессора;
- обнаруженные процессором ошибки в программе (деление на ноль, отсутствие страницы памяти и т. п.).

Первый из перечисленных выше типов исключений называют **отказом** — исключением, возникающим до исполнения инструкции, когда

процессор обнаруживает невозможность её исполнения. Состояние вычислительной машины при этом не меняется.

Третий тип исключения называют **ловушкой** — это исключение, возникающее в результате исполнения ошибочной инструкции. Состояние вычислительной машины при этом меняется.

Отдельно можно выделить вариант исключения, называемый **аварией**. Это исключение, для которого невозможно точно определить породившую его инструкцию, и невозможны корректные действия для продолжения работы.

2 Переключение задач и виртуальные машины

Процессор машины фон Неймана может выполнять только один процесс, передавая управление от инструкции к инструкции согласно исполняемой программе. При этом могут обрабатываться переходы, ветвления и вызовы процедур, но вся эта цепочка запрограммирована разработчиком программы.

В многозадачных системах для переключения задач используют прерывания. Пусть, например, имеются два потока инструкций, которые должны выполняться одновременно. Можно передать на исполнение центральному процессору первый поток инструкций, а через некоторое время (при поступлении аппаратного прерывания от таймера) сохранить в памяти текущее состояние регистров, и передать управление второму потоку инструкций. Через некоторое время необходимо произвести обратное переключение. Эти переключения задач необходимо выполнять в течение исполнения обеих программ с частотой, создающей иллюзию непрерывности и одновременности.

Процессоры, начиная с i80386 имеют встроенные средства поддержки многозадачности в защищённом режиме. Переключение задач производится по сигналу прерывания совершенно “прозрачно” для процессов, работающих псевдопараллельно. Благодаря этой прозрачности программисту в большинстве случаев не надо заботиться об обеспечении многозадачности. В распоряжении его программы предоставляется виртуальная машина, в которой управление передаётся этой программе так, — как будто она единственная, которую выполняет процессор.

2.1 Защищённый режим и виртуальная память

Для того, чтобы потоки (задачи) “не мешали друг другу”, необходимы меры принудительной защиты критических ресурсов. В современных ОС используется защищённый режим процессора, в котором эти меры реализуются на аппаратном уровне. При этом каждой виртуальной машине предоставляется своя виртуальная память. В защищённом режиме существует разделение привилегий между пользовательскими процессами и операционной системой.

Операционной системе предоставляются неограниченные права управления всеми процессами, их виртуальной памятью, прерываниями и вводом-выводом. В процессорах класса x86 заложена четырёхуровневая система привилегий (Protection rings — кольца защиты) от supervisor 0 (уровень неограниченных возможностей) до user 3 (уровень самых жёстких ограничений).

Для защиты и виртуализации памяти предусмотрены сегментация и страничная трансляция адресов. Сегменты — области памяти различного назначения с разными правами (и разного размера), выделяемые операционной системой. Каждому процессу для программного кода выделяются специальные сегменты. Инструкции могут выбираться и исполняться только из них. Это является искусственным ограничением для машины фон Неймана, однако позволяет снизить вероятность ошибок, так как процессор трактует содержимое таких ячеек памяти в первую очередь как коды инструкции.

При страничной трансляции адресов виртуальная логическая память (для каждой виртуальной машины) делится на страницы одинакового размера. Любая страница виртуальной логической памяти может отображаться на любую область физической памяти. Отображение поддерживается с помощью специальных таблиц страничной трансляции адресов, в которых помимо записей, описывающих связи адресов есть указание на присутствие/отсутствие страницы в физической памяти. Благодаря этому страница памяти, ненужная процессору в данный момент, может быть выгружена на устройство хранения. Таким образом в распоряжении всех процессов, исполняемых на персональном компьютере псевдопараллельно, предоставляется виртуальная оперативная память, размер которой ограничен суммой объёма физической оперативной памяти и области дисковой памяти, выделенной для подкачки страниц.

Изначально страницы не делились на страницы кода и данных, чем и могли пользоваться вирусы. В новых процессорах появилась возможность запрета исполнения кода из страниц памяти, предназначенных для данных (в процессорах Intel эта технология называется Execute Disable Bit, у моделей AMD — Enhanced Virus Protection).

3 Режимы работы УВВ

Как правило, у большинства периферийных устройств есть регистры (или порты) данных (data), управления (control) и состояния (state). На уровне архитектуры системы команд вся работа с внешней аппаратурой компьютера выполняется как чтение и запись этих портов ввода/вывода (IO ports) с помощью следующих инструкций:

```
in al, dx ; read byte to AL from IO port, addressed by DX
out dx, al ; write byte from AL to IO port
```

Периферийные устройства (или устройства ввода/вывода — УВВ) могут работать в нескольких режимах.

3.1 Режим программного опроса (по флагу или по ожиданию)

В таком режиме центральный процессор под управлением программного обеспечения опрашивает УВВ (как правило — регистр состояния), выбирая его адрес, и контролируя состояние определённого разряда (как правило — старшего). Состояние флага читается, если флаг поднят (то есть в состоянии «1») данные считываются из УВВ, считаются истинными и обрабатываются. Если флаг опущен, то процессор выполняет цикл холостого чтения до тех пор, пока не будет поднят флаг. В некоторых устройствах в качестве флага готовности может использоваться старший разряд регистра данных (например, готовность АЦП записывается в старший разряд порта данных).

Режим программного опроса может быть реализован во всех устройствах, но он самый неэкономичный, т. к. центральный процессор большую часть машинного времени тратит на холостые чтения в ожидании флага готовности устройства.

3.2 Режим работы по прерываниям

Режим работы по прерываниям — основной, т. к. процессор может выполнять некую фоновую задачу, обращаясь к устройству, только получив соответствующий запрос. После обработки прерывания процессор возобновляет выполнение фоновой задачи. Данный режим сложнее в реализации, чем предыдущий, но гораздо более эффективный.

УВВ готовое к обмену информацией, вырабатывает сигнал **IRQ** — запрос аппаратного прерывания. Этот сигнал поступает на контроллер прерываний, а с него на контакт процессора **INTR**. В ответ на поступивший

запрос, процессор формирует сигнал подтверждения обработки прерывания $INTA$, запрашивая при этом от контроллера прерывания вектор прерывания.

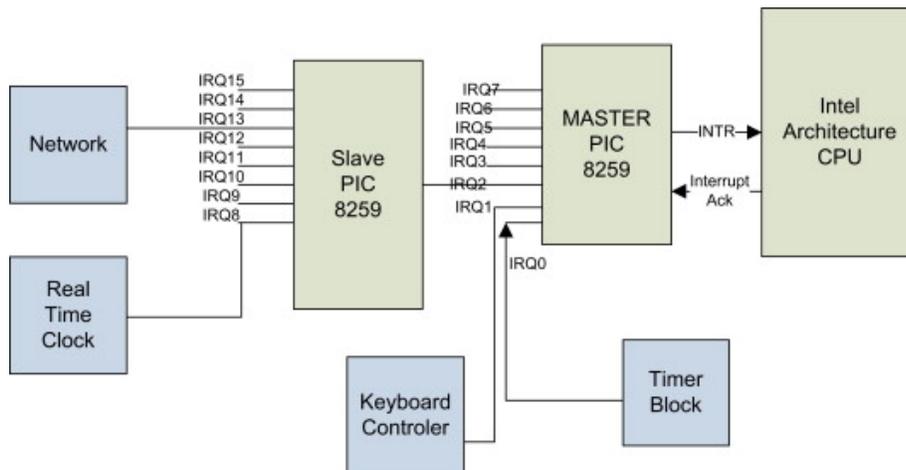


Рис. 3: Цепочка из двух контроллеров прерываний 8259 PIC, стандартная для компьютеров архитектуры IBM PC

К линии $INTR$ процессора подключается ведущий контроллер прерываний. Его входы $IRQ0$ - $IRQ7$ соединяются с непосредственными устройствами — источниками аппаратных прерываний (чем меньше номер, тем больше приоритет). Если на контроллер прерываний поступает одновременно несколько запросов, то на выход контроллера $INTR$, подключённый к одноимённой линии процессора пропускается сигнал с наибольшим приоритетом. Менее приоритетные сигналы блокируются, и ждут своей очереди. К выходу $IRQ2$ ведущего контроллера подключается ведомый контроллер, на входы которого подаются запросы на аппаратные прерывания $IRQ8$ - $IRQ15$. Для каждого устройства в персональном компьютере отведён свой номер запроса аппаратного прерывания $IRQx$. У современных компьютеров один номер $IRQx$ может использоваться несколькими устройствами.

Не перечисленные номера $IRQx$ — свободны, и могут использоваться дополнительными устройствами. Наивысший приоритет у таймера, так как с его помощью осуществляется регенерация динамической памяти.

Для обработки аппаратных прерываний в многозадачных системах традиционные системы (8259 PIC) уже непригодны, так как прежняя схема подачи запроса и передачи вектора прерывания ориентированы на единственность процессора. Для решения этой задачи начиная с процессоров Pentium Pro введён усовершенствованный программируемый контроллер прерываний APIC.

IRQ 0 - Timer	}	IRQ 8 - RTC (часы)
IRQ 1 - KB		IRQ 9 - Video
IRQ 2		IRQ 13 - HDD
IRQ 3 - COM 2		IRQ 14 - HDD
IRQ 4 - COM 1		IRQ 15 - резерв
IRQ 5 - LPT2, резерв		
IRQ 6 - FDD		
IRQ 7 - LPT1		

Рис. 4: Стандартная для IBM PC схема распределения номеров IRQx между УВВ

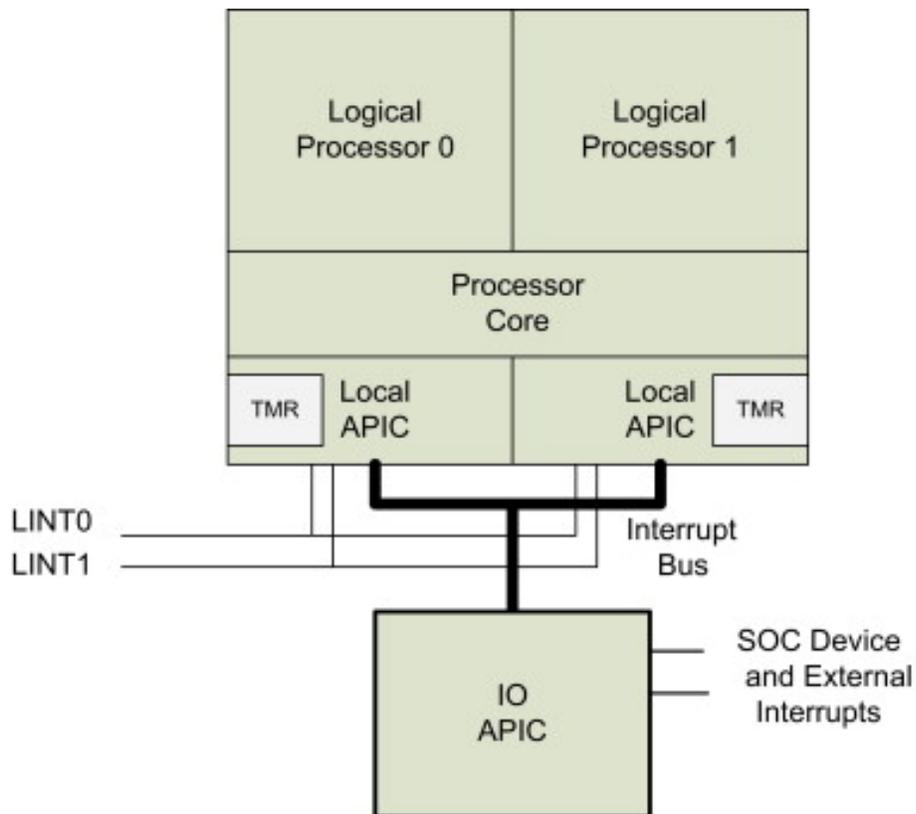


Рис. 5: Схема обработки аппаратных прерываний с помощью контроллера APIC

3.3 Режим прямого доступа к памяти

В этом режиме устройство, готовое к обмену информацией, подаёт на контроллер DMA (Direct Memory Access) сигнал запроса некоторой памяти DRQx. Контроллер формирует ответный сигнал DACK — подтвер-

ждение режима DMA. После этого контроллер блокирует центральный процессор, и идёт прямая передача данных из УВВ в память (или обратно). Передача осуществляется не сплошным массивом, а отдельными блоками (страницами памяти определённого размера). По окончании передачи вырабатывается сигнал «Конец цикла» и центральный процессор включается обратно в работу.

Это один из самых скоростных режимов (в настоящее время используется режим Ultra DMA). Передача данных в этом режиме идёт постранично, так как в ПК используется динамическая память, требующая регенерации. В машинах IBM PC/XT для регенерации использовался канал DMA0 и регенерация памяти происходила в промежутках между передачей страниц. В режиме DMA может использоваться сигнал Bus Mastering, который позволяет организовать режим DMA, не используя стандартный контроллер DMA (при этом контроллер должен быть в УВВ).