

6. Система прерываний

Прерывания служат для приостановки прямого выполнения программы, с тем, чтобы процессор смог отреагировать на определенный запрос, сформированный периферийным устройством в зависимости от какого-либо обстоятельства. Реакция ЦП на запрос выражается в переходе к выполнению некоторой программы, которая называется программой обработки прерывания (*Interrupt – Service Routine — ISR*). После её окончания продолжается выполнение прерванной программы¹⁾. Путём присвоения приоритетов запросам можно добиться, чтобы ЦП реагировал, прежде всего, на наиболее важное событие, игнорируя запросы либо прерывая ISR менее важных.

В отличие от циклического опроса, при котором программно проверяется выполнение определенного условия, и для которого требуется время выполнения и место в памяти, прерывание обеспечивает практически мгновенную реакцию, т.к. не зависит от периода опроса, отсутствие программы опроса, как таковой, выражается в экономии памяти.

Различные варианты применения прерываний являются эффективным средством повышения производительности микропроцессорной системы.

6.1. Разрешение и запрещение прерываний

Для приёма запросов на прерывания микропроцессор Z80 имеет два входа: INT и NMI.

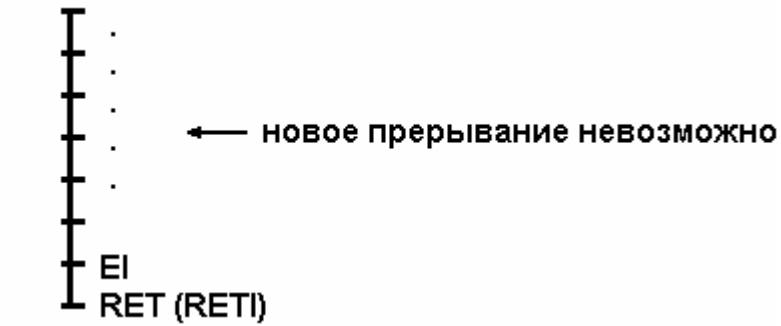
Различие между ними — в приоритете и маскируемости.

INT (маскируемое прерывание) может быть программно запрещено или разрешено. Необходимость в запрещении (маскировании) прерываний возникает, например, когда условия работы в реальном масштабе времени делают нежелательным прерывание данного участка программы.

Состояние «запретить прерывания» либо «разрешить прерывания» запоминается программно-доступным триггером IFF1 (*Interrupt Flip Flop*) внутри ЦП. Посредством команды разрешения прерываний EI (*Enable Interrupt*) либо запрещения прерываний DI (*Disable Interrupt*) этот триггер соответственно устанавливается либо сбрасывается. В некоторых случаях для сохранения его текущего состояния требуется промежуточная память, в качестве которой предусмотрен триггер IFF2.

При сбросе ЦП сигналом /RESET оба триггера сбрасываются, блокируя требования на маскируемые прерывания. Посредством команды EI их можно разблокировать. Когда прерывание принимается, IFF1 и IFF2 также автоматически сбрасываются, чтобы предотвратить дальнейшие прерывания до тех пор, пока они не будут разрешены новой командой EI.

Применение EI внутри ISR дает следующие две возможности:

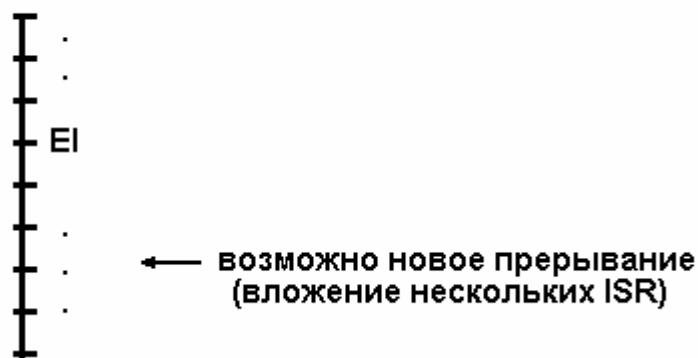
- а)


Начало ISR

новое прерывание невозможно

EI
RET (RETI)
- б)

Начало ISR



Важное обстоятельство: при выполнении команды EI поступивший в это время запрос на прерывание будет блокирован до тех пор, пока не выполнится следующая за EI команда. Цель такой задержки состоит в том, чтобы гарантировать (после команды EI) возможность выполнения команды возврата из ISR (RET или RETI).

NMI (немаскируемое прерывание) имеет более высокий по отношению к INT приоритет и не может быть запрещено программным способом. Таким образом, если оно затребовано каким-либо периферийным устройством, то ЦП, безусловно, прерывает текущую программу. Этот вид прерываний предназначен, как правило, для очень важных событий (отказ, спад напряжения питания и др.).

В отличие от INT, когда состояния IFF1 и IFF2 совпадают, при подтверждении немаскируемого прерывания IFF1 сбрасывается, а IFF2 остается без изменений. Это делается для того, чтобы на время обработки NMI сохранить состояние IFF1, которое имелося до приема NMI. После окончания NMI-SR по команде возврата RETN состояние IFF1 восстанавливается из IFF2.

Команды LD A,I и LD A,R предоставляют возможность тестирования IFF2: они пересылают содержимое IFF2 в P/V-флаг. Т. е. значение IFF2 может использоваться для ветвления программы.

В [таблице 6.1](#) в компактном виде представлены все действия, влияющие на состояние триггеров разрешения прерываний.

Таблица 6.1. Состояние триггеров разрешения прерываний

Операция	IFF1	IFF2	Примечание
Сброс ЦП сигналом /RESET	0	0	Запрещение INT при пуске системы
Команда EI	1	1	Разрешение INT (задерживается на одну команду)
Команда DI	0	0	Запрещение INT
Прием INT	0	0	Прием INT и обработка прерывания
Команда RETI	■	■	Выход из ISR
Прием NMI	0	■	Прием NMI и обработка прерывания
Команда RETN	IFF1←IFF2	■	Выход из NMI-SR
Команда LD A,I	■	■	P/V ← IFF2
Команда LD A,R	■	■	P/V ← IFF2

■ — состояние триггера не изменяется

6.2. Приём запросов в ЦП

Реакция микропроцессора на поступивший запрос прерывания зависит от точного времени поступления этого запроса, а также запросов с более высокими приоритетами.

Кроме двух входов прерывания у микропроцессора Z80 есть ещё вход запроса шины для ПДП (/BUSRQ), который имеет более высокий приоритет и, следовательно, тоже влияет на разрешение/запрещение прерывания.

На рис.6.1 в упрощенном виде показаны основные аппаратные средства микропроцессора, предназначенные для приёма запросов /BUSRQ, /NMI, /INT. Каждая линия запроса снабжена соответствующим триггером приёма (Plip-Flop): BUSRQ-FF, NMI-FF, INT-FF, в которые по нарастающему фронту последнего такта (TL) машинного цикла вводится информация о запросах. Установка какого-либо из этих триггеров означает приём соответствующего запроса в ЦП. Далее принятые запросы обрабатываются устройством управления, где в соответствии с указанными приоритетами подтверждается в первую очередь тот или иной запрос, и вырабатываются соответствующие внутренние и внешние сигналы управления.

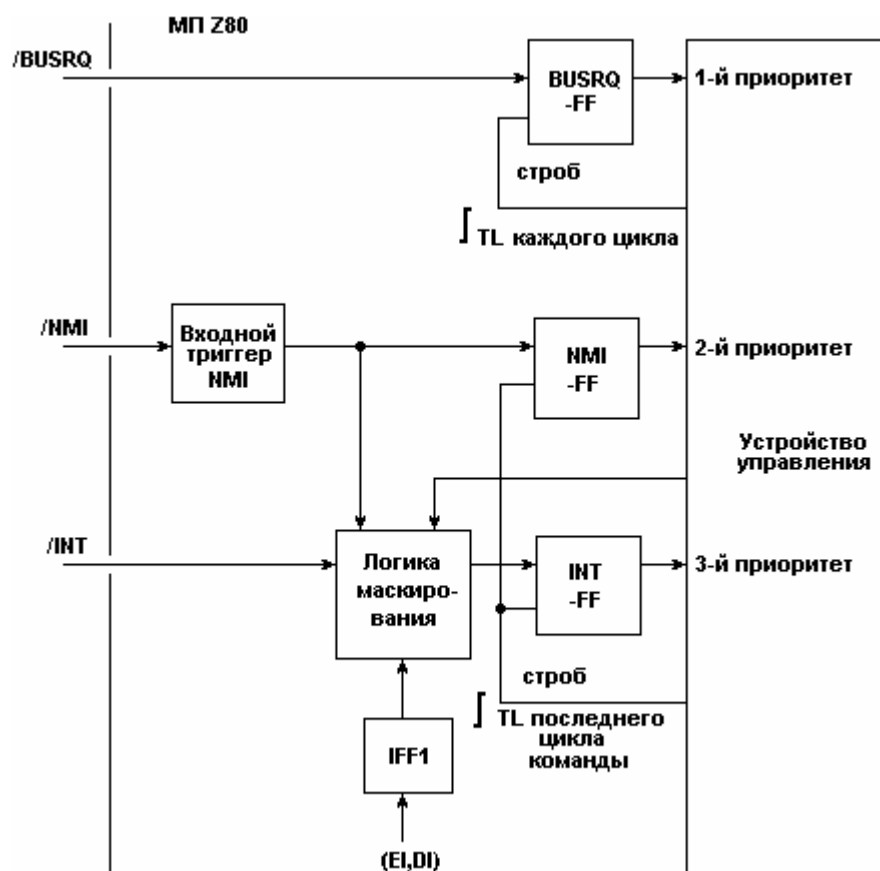


Рис. 6.1. Прием запросов /BUSRQ, /NMI, /INT

Линии приема запросов имеют существенные различия.

Триггером BUSRQ-FF анализируется уровень сигнала /BUSRQ по нарастающему фронту последнего такта (TL) каждого машинного цикла ²⁾.

Активный уровень сигнала /NMI сразу фиксируется входным триггером независимо от времени его поступления. Его Состояние (а не уровень /NMI) анализируется триггером NMI-FF по нарастающему фронту TL последнего цикла команды.

Уровень сигнала /INT до приёма в INT-FF анализируется схемой логики маскирования, где он может быть блокирован поступающими сюда же сигналами:

- от триггера разрешения/запрещения IFF1,
- от входного триггера NMI,
- от устройства управления, который сигнализирует об обработке запросов с высшими приоритетами.

Уровень результирующего сигнала анализируется триггером INT-FF по нарастающему фронту TL последнего цикла команды.

Приоритет /BUSRQ по отношению к /NMI и /INT состоит в следующем. При одновременном поступлении запроса шины и одного из запросов прерывания подтверждается запрос /BUSRQ, и шина предоставляется для ПДП. Запрос шины может прервать обработку NMI-SR или ISR на любой цикле. Во время ПДП прерывания от входов /NMI и /INT не обслуживаются.

Приоритет /NMI по отношению к /INT состоит в том, что если оба запроса поступили до нараставшего фронта такта TL последнего цикла команды, то /INT не воспринимается внутренним триггером INT-F/F и, следовательно, игнорируется до конца NMI-SR. Если же в течение одной команды запрос INT поступил до нарастающего фронта TL, а NMI — после,

то выполняется первая команда программы обработки INT, и только потом — переход к программе обработки NMI.

При подтверждении одного из запросов прерывания (NMI либо INT) процессор вырабатывает соответствующий цикл подтверждения (см. в конце [таблицы 5.3](#)).

Все реакция ЦП на сигналы управления /BUSRQ, /NMI и /INT с учётом граничных условий представлены в [таблице 6.2](#), а также в виде алгоритма на [рисунке 6.2](#).

Таблица 6.2. Реакция ЦП на запросы /BUSRQ, /NMI, /INT

Сигнал	Поступление сигнала	Приём сигнала	Действие	Примечание
/BUSRQ	До нарастающего фронта такта TL	При нарастающем фронте такта TL любого машинного цикла	Подтверждение (/BUSAK) в следующем машинном цикле	/BUSRQ имеет высокий приоритет, чем /NMI и /INT
/NMI	Внутри цикла до нарастающего фронта такта TL	Сразу фиксируется во входном триггере NMI (мин. ширина импульса 80 мс)	Непосредственно после обработки этой команды – переход к программе обработки NMI (так же непосредственно после EI)	Для повторного выполнения программы обработки NMI сигнала /NMI должен быть выключен на короткое время
	Внутри цикла после нарастающего фронта TL		После обработки этой команды выполняется следующая, потом переход к программе обработки NMI	
/INT	До нарастающего фронта такта TL	Только при нарастающем фронте последнего такта команды (TL)	После обработки команды, если прерывания были разрешены, переход к ISR. Если прерывания запрещены — /INT игнорируется	При выполнении команды EI /INT не прерывается, независимо от состояния IFF1 и IFF2. Он может быть принят при выполнении следующей за EI команды (если будет активен до нарастающего фронта TL этой команды)
	После нарастающего фронта такта TL		Блокируется независимо от состояния IFF1 и IFF2	Может быть принят при выполнении следующей команды, если будет активен до нарастающего фронта её последнего такта TL

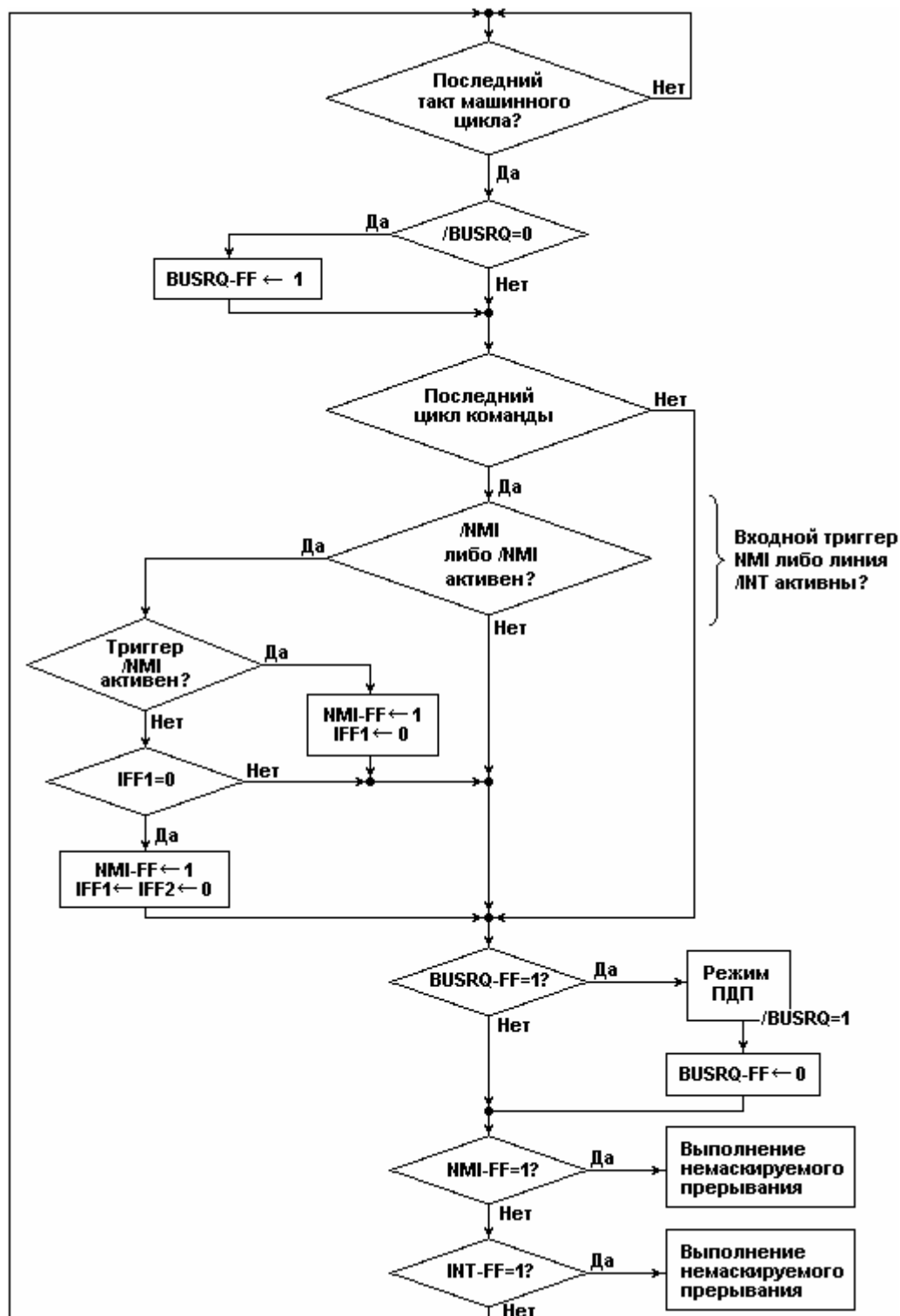


Рис. 6.2. Алгоритм обработки запросов /BUSRQ, /NMI, /INT

6.3. Обработка прерываний

6.3.1. Обработка немаскируемого прерывания

Если принят запрос NMI, то по завершении текущей команды состояние счётчика команд PC автоматически сохраняется в стеке, и в PC заносится адрес 0066H. Т.е. происходит перезапуск процессора с адреса 66H. С этого места в памяти должна начинаться подпрограмма обработки NMI; завершаться она должна командой возврата RETN. Алгоритм обработки NMI изображен на [рисунке 6.3](#), временные диаграммы вызова подпрограммы и возврата из неё

— на рисунке 6.4 и рисунке 6.5 соответственно.

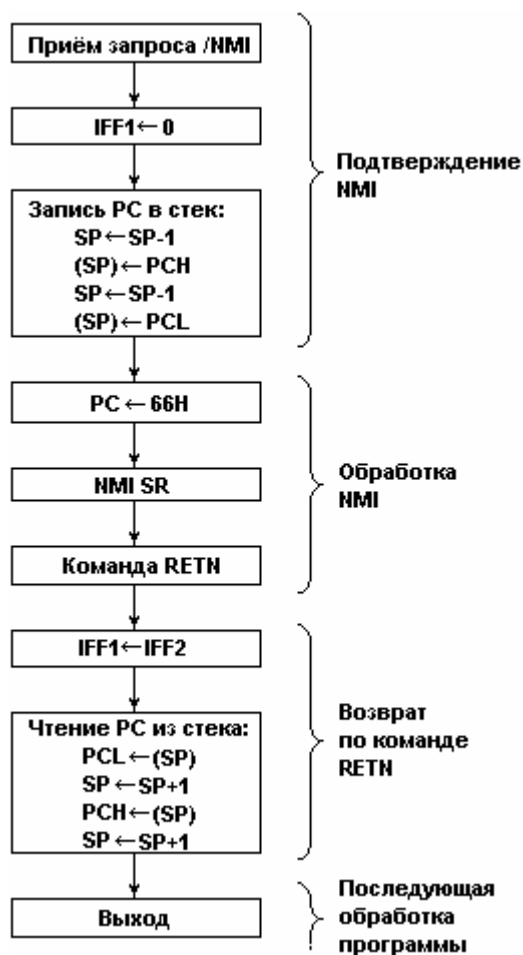


Рис. 6.3. Алгоритм обработки немаскируемого прерывания

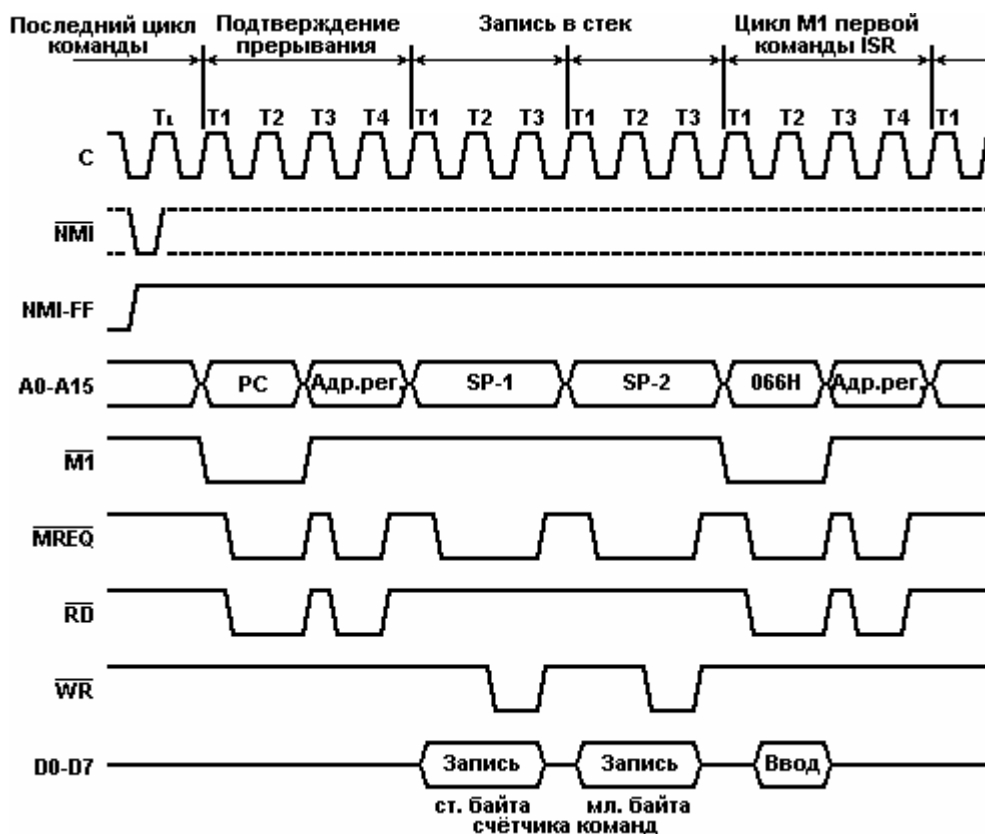


Рис. 6.4. Вызов программы обработки немаскируемого прерывания

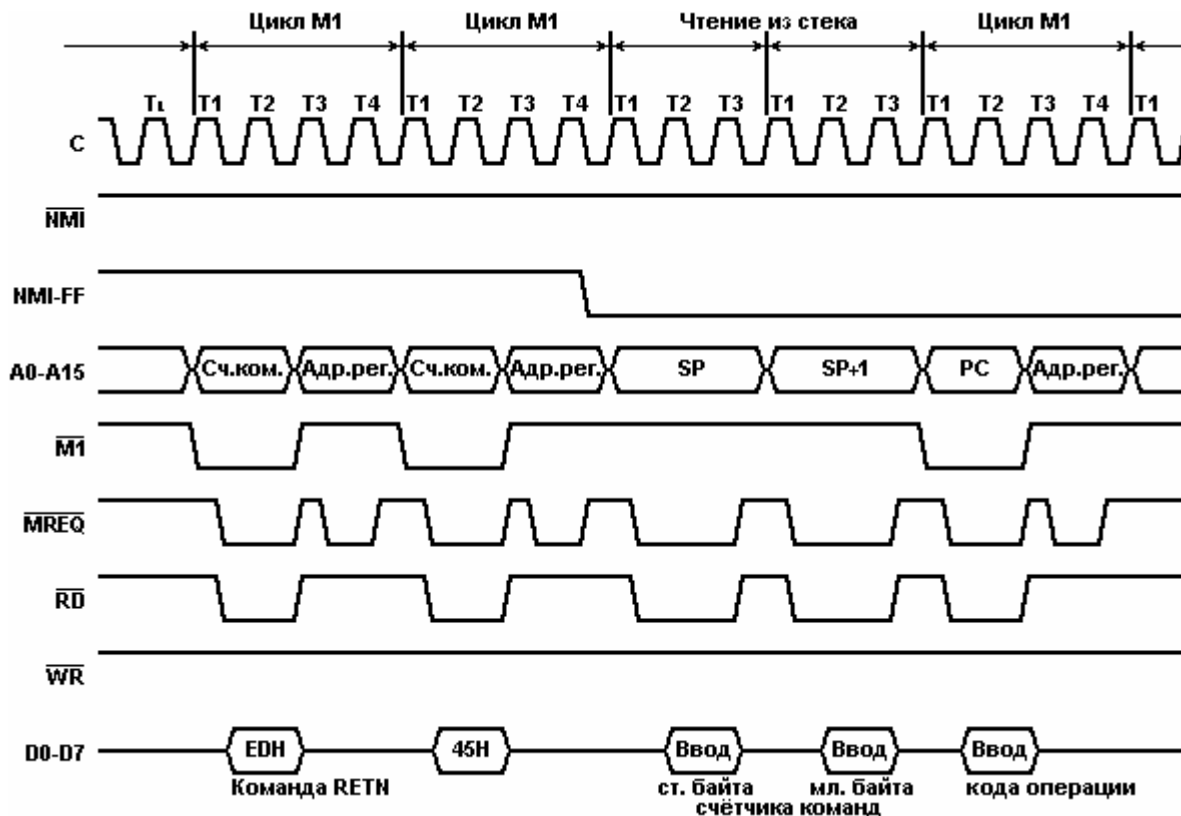


Рис. 6.5. Возврат из подпрограммы обработки немаскируемого прерывания

6.3.2. Маскируемое прерывание. Режим 0

Этот режим аналогичен реализованному в МП Intel 8080. Для него необходима внешняя вспомогательная логика, которая в случае прерывания подает команду на шину данных ЦП. Передача этой команды происходит в цикле подтверждения INTA, когда одновременно активизируются сигналы /M1 и /IORQ (см. рисунки 4.9 и 6.6).

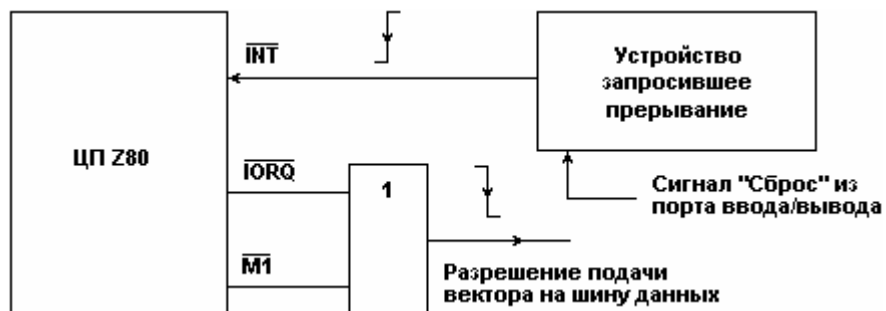


Рис. 6.6. Подача запроса на прерывание и формирование сигнала подтверждения прерывания

Т.е. происходит чтение кода операции, но не из памяти, а из прерывающего устройства. Это может быть любая команда, однако, наиболее эффективно в этом случае использование одной из восьми команд повторного запуска RST r, где r — адрес рестарта. Для использования этих команд на ШД должны быть поданы следующие комбинации битов:

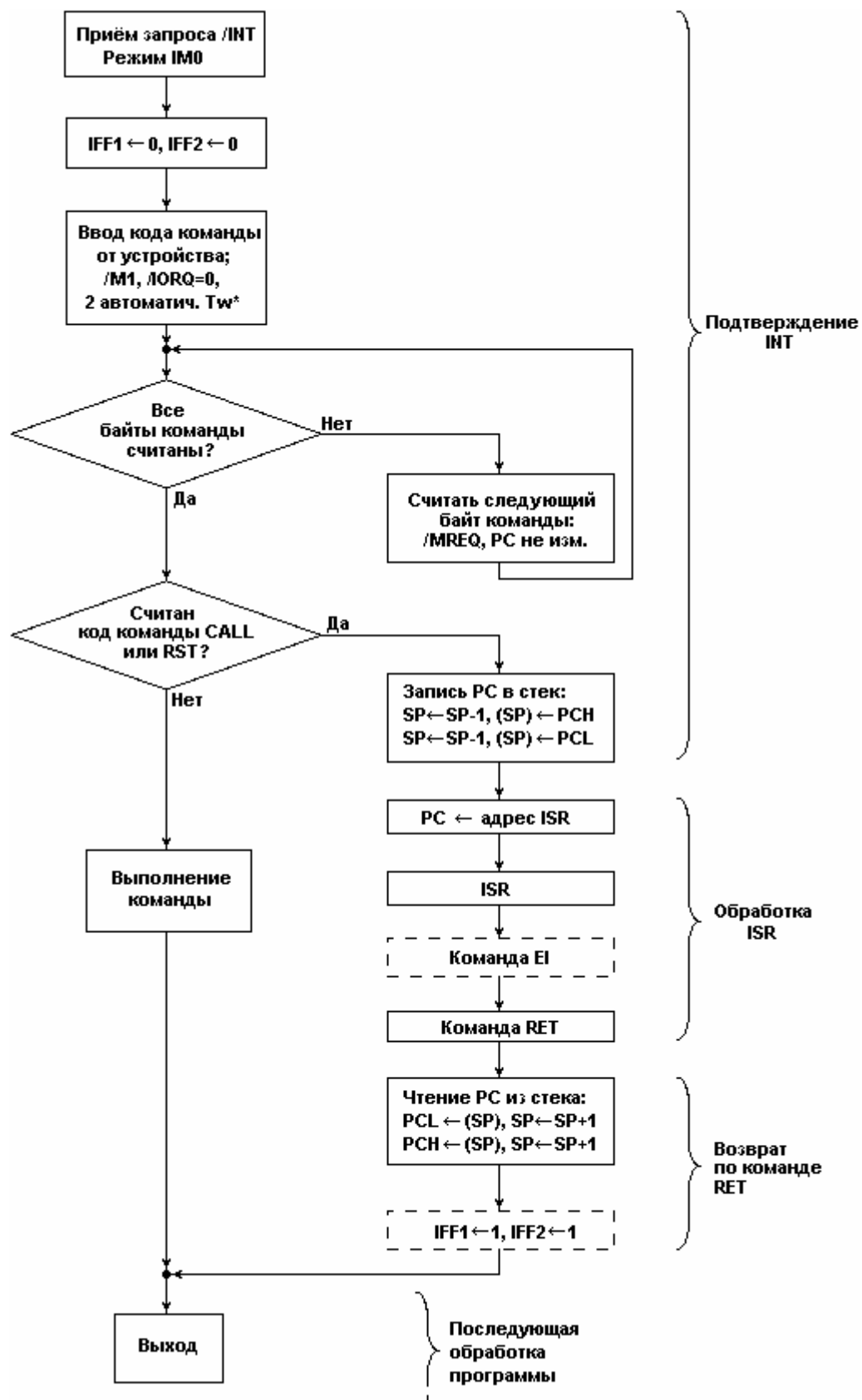
Команда	D7 ... D0
RST 00H	11 000 111
RST 08H	11 001 111
RST 10H	11 010 111
RST 18H	11 011 111
RST 20H	11 100 111
RST 28H	11 101 111

Команда	D7 ... D0
RST 30H	11 <u>110</u> 111
RST 38H	11 <u>111</u> 111

Алгоритм обработки изображен на [рис. 6.7](#). Если в подпрограмме присутствует команда EI, то после выполнения следующей за ней команды происходит установка IFF1, IFF2. Эти блоки отмечены пунктиром.

Как и для МП I8080, в этом режиме возможно использование контроллера прерываний 8259 (580BH59). Этот вариант подробно рассмотрен в книге 7.

Режим 0 автоматически устанавливается при выполнении общего сброса, а также соответствующей командой IM0 (Interrupt Mode 0).



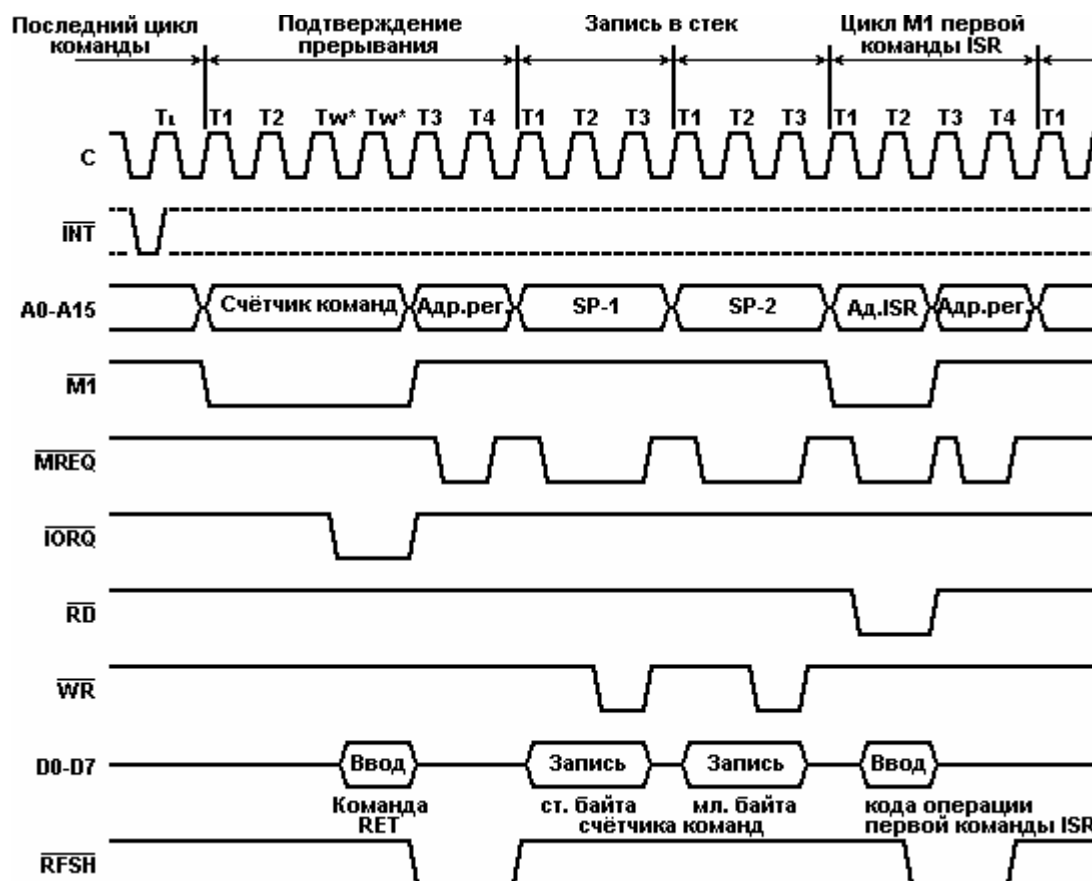


Рис. 6.8. Вызов программы обработки маскируемого прерывания ISR в режиме 0 (IM 0)

6.3.3. Маскируемое прерывание. Режим 1

Режим прерываний 1 устанавливается командой IM 1. Он применяется в системах с минимальной конфигурацией, где нежелательно наращивание аппаратной части введением вспомогательной логики. Обработка сходка с NMI, за исключением того, что вместо адреса 0066H происходит рестарт к адресу 0038H и цикл подтверждения продлевается на два такта ожидания. Возврат — по команде RET.

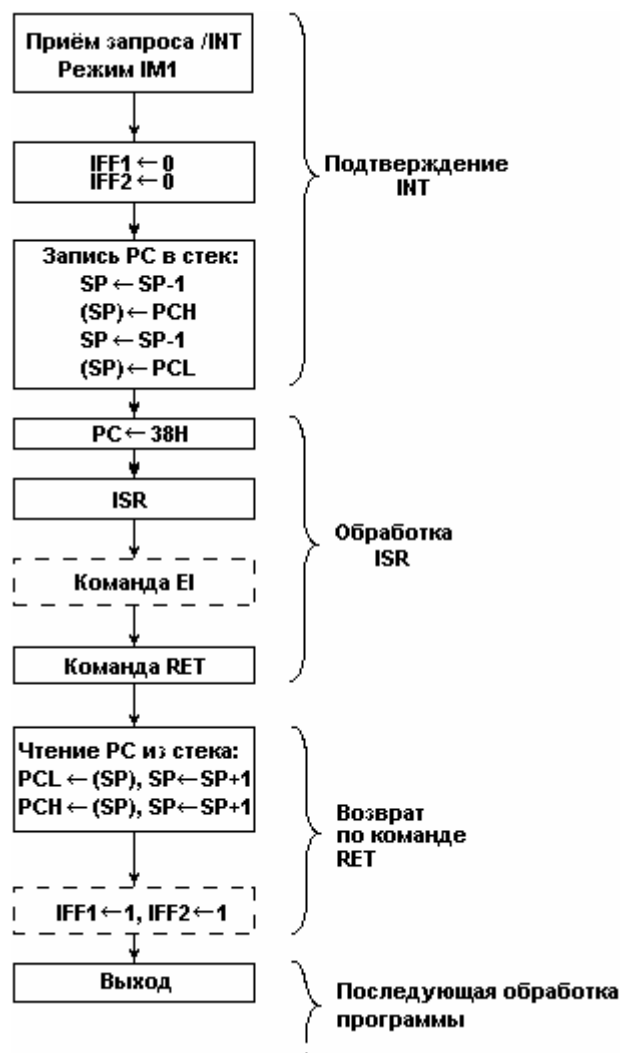
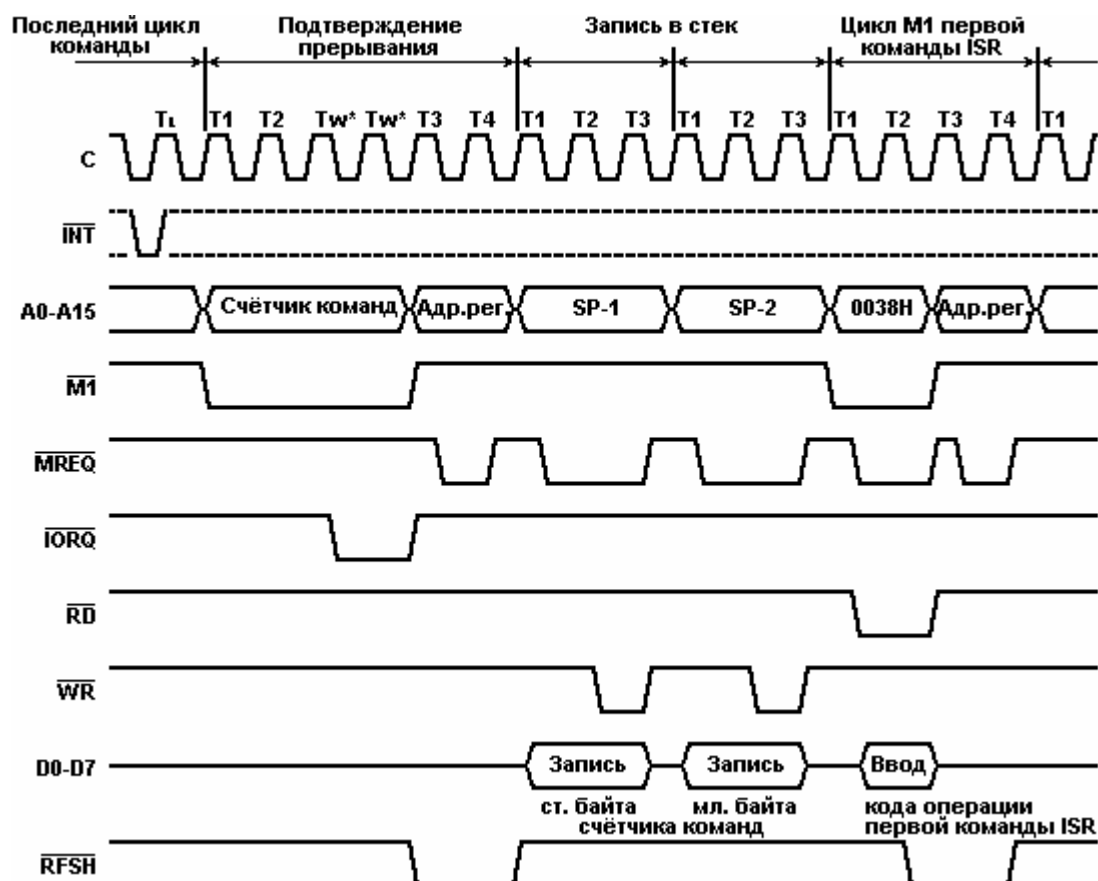


Рис. 6.9. Алгоритм обработки маскируемого прерывания. Режим 1



6.3.4. Маскируемое прерывание. Режим 2

Режим прерываний 2 устанавливается командой IM 2. Это самый производительный режим прерываний микропроцессора Z80. Он применяется в сложных системах с развитой сетью периферийных элементов. При этом в памяти программируется таблица стартовых адресов каждой ISR (см. [рис. 6.11](#)). Таблица может быть размещена в любой области памяти с соблюдением лишь правила записи стартовых адресов: младший байт заносится в ячейку с чётным адресом ($A0=0$), старший байт в следующую (по возрастанию) ячейку.

В случае приёма прерывания ЦП формирует 16-разрядный указатель IP (Interrupt Pointer) для выборки стартового адреса нужной программы ISR из таблицы. Для этого прерывавшим устройством в виде 8 разрядного вектора поставляется младший байт, а старший извлекается из регистра I микропроцессора. Сформированный таким образом IP указывает на ячейку памяти (в таблице), где хранится стартовый адрес ISR. Считав этот адрес из двух смежных ячеек, процессор переходит к обработке программы ISR. Возврат из неё должен происходить по команде RETI. В целом алгоритм обработки изображен на [рис. 6.12](#).

В микропроцессорном комплекте Z80 предусмотрено, что периферийные БИС могут выдавать в качестве вектора прерывания только чётные байты ($D0=0$) — для однозначности адресации ячеек таблицы. Значит, даже при неизменном состоянии регистра I центрального процессора возможно обслуживание 128-ми устройств в режиме прерываний 2, что во многом превышает потребности микропроцессорной системы среднего класса.

Для перехода к ISR в режиме 2 требуется 19 тактовых периодов (см. временные диаграммы на [рис. 6.13](#)): 7 периодов для ввода 8 разрядного вектора от прерывающего устройства, следующие 6 — для сохранения в стеке текущего состояния PC (адреса возврата), и ещё 6 — для считывания стартового адреса ISR.

Подробное описание прерываний в системе с полной конфигурацией, особенно, в отношении периферийных элементов комплекта Z80, а также организации приоритетного обслуживания см. в книге 7 «Z80/Построение систем. Программирование. Отладка».

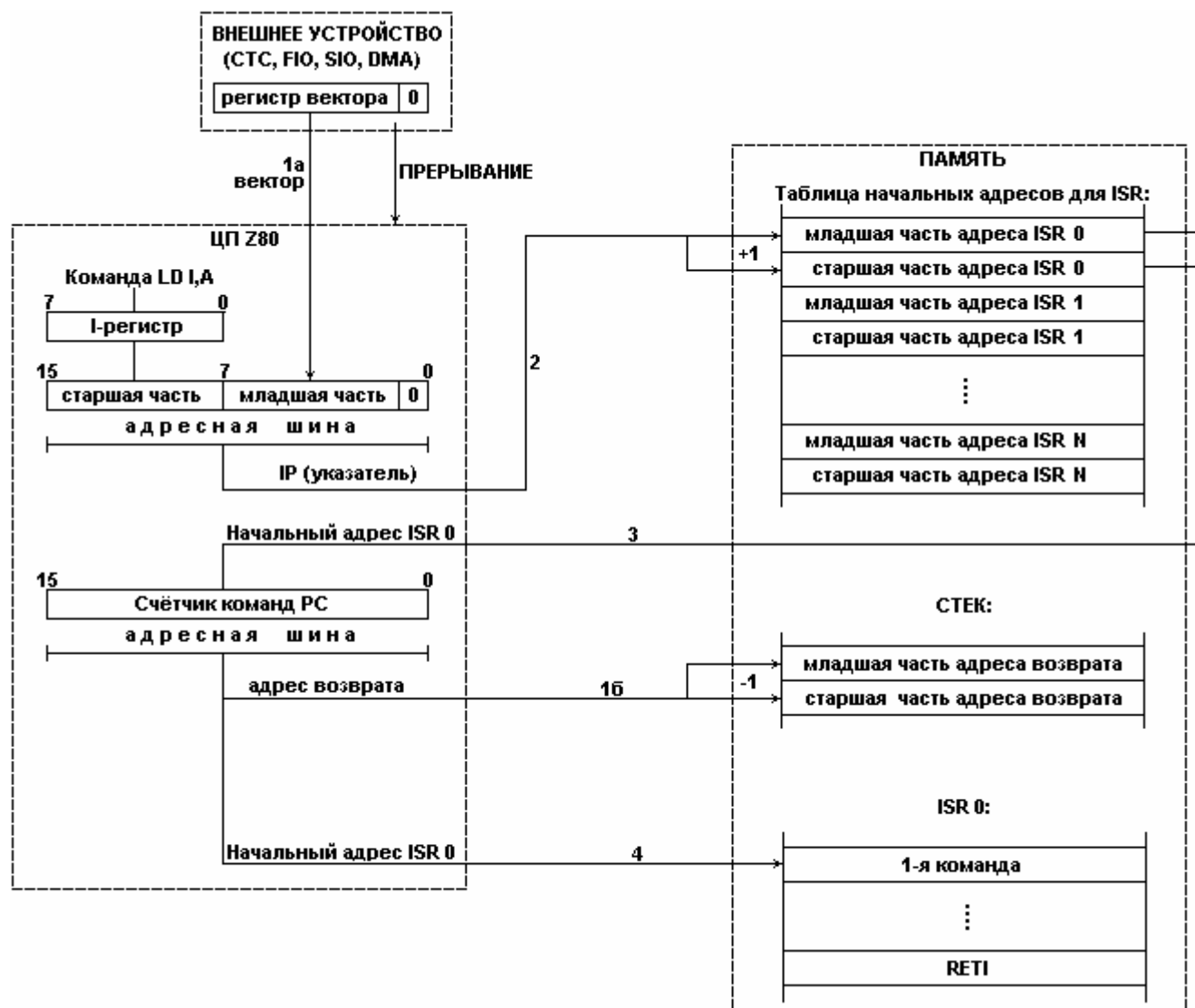


Рис. 6.11. Переход к ISR в режиме прерывания 2:

- 1a — приём вектора прерывания устройства в ЦП,
- 16 — адрес возврата сохраняется в стеке,
- 2 — сформированный 16-разрядный IP указывает на начальный адрес ISR,
- 3 — начальный адрес ISR загружается в PC,
- 4 — считывается 1-я команда программы ISR.

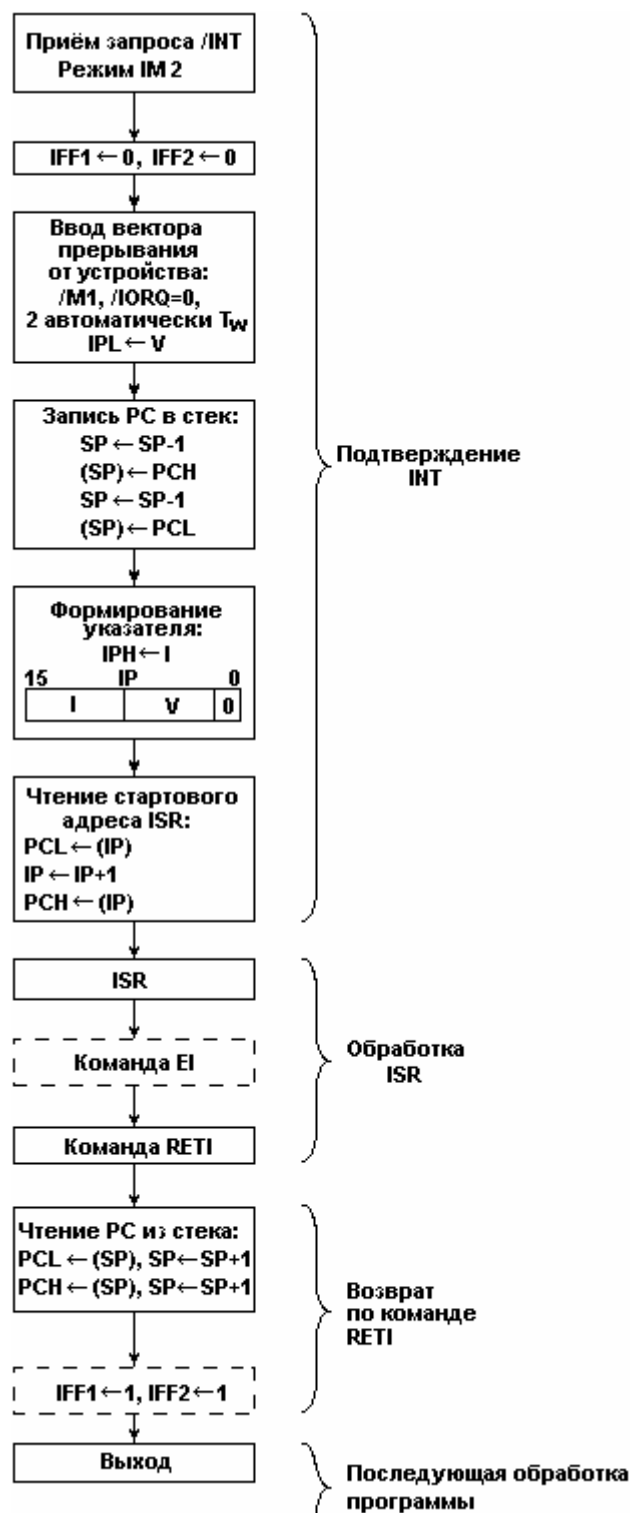


Рис. 6.12. Алгоритм обработки маскируемого прерывания. Режим 2.

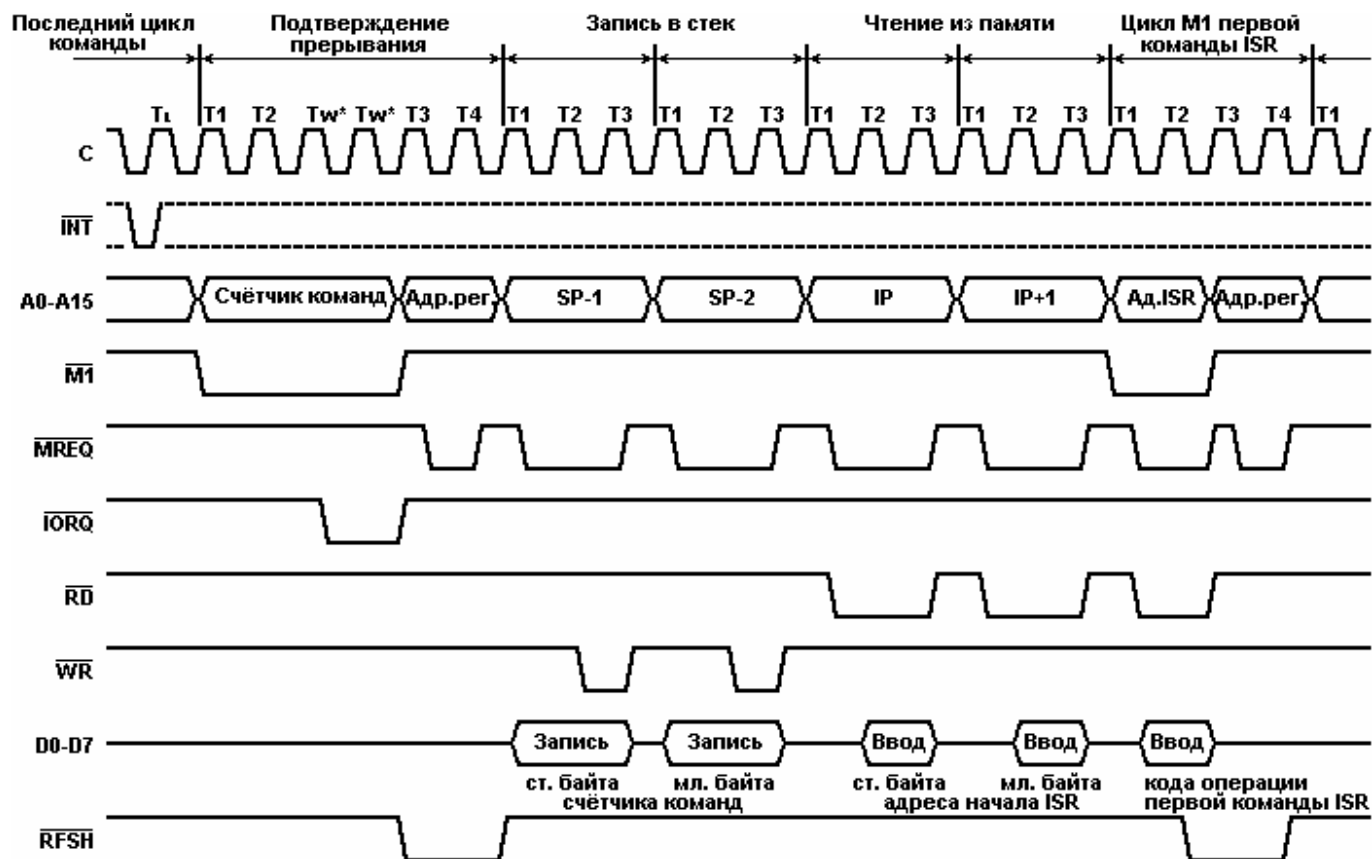


Рис. 6.13. Вызов программы обработки маскируемого прерывания ISR в режиме 2 (IM 2).

1)

Для корректного возврата в прерванную программу предусмотрено следующее:

- при входе в ISR адрес возврата автоматически сохраняет я в стеке;
- внутри ISR программно предпринимаются меры для сохранения содержимого регистров ЦП двумя способами:
 - а) смена блока регистров командами EXX, EX AF, AF'
 - б) пересылка в стек командой PUSH

2)

Временные диаграммы процессов при запросе шины изображены на [рисунке 4.8](#).

Описание режима ПДП для полной конфигурации системы содержится в книге 6 «Контроллер ПДП Z80DMA» и книге 7 «Z80/Построение систем. Программирование. Отладка».

https://sysadminmosaic.ru/microprocessor_kit_z80-book1/06

2021-05-18 13:58

