

2. При малой сетевой скорости покажите, что вы не можете ввести новый кадр, пока не передадите предыдущий.

5.5. Содержание отчета

1. Блок-схемы программ СУ и ПУ с комментариями.
2. Листинги разработанных программ.

ЛАБОРАТОРНАЯ РАБОТА № 6 ПЕРЕДАЧА И ПРИЕМ КАДРОВ С АДРЕСАЦИЕЙ

Цель работы: получить навыки разработки и исследования алгоритмов адресации кадров.

6.1. Индивидуальное задание

Разработайте и выполните отладку программы передачи и приема кадров с адресацией узлов станций ЛВС. Настройка адаптера UART производится в соответствии с исходными данными к лабораторной работе № 1.

6.2. Методические указания

В данной лабораторной работе в формат кадра вводятся поля для адресации приемной (DID) и передающей (SID) станций [2]:

PRE, DID, SID, ..., данные ..., EOT

С точки зрения передатчика адресация кадра предполагает необходимость включения информации об адресе назначения кадра в сам кадр. С точки зрения приемника адресация кадра предполагает необходимость анализировать приходящие кадры и отвергать те кадры, адреса которых не соответствуют адресу приемного узла.

Заметим, что в формате кадра не определены байты CR и LF. Они все же используются как заключительные символы в сообщениях, выводимых с клавиатуры на экран, однако в действительности они не принадлежат полю данных и имеют значение только для прикладного уровня (ПУ) и для человека - пользователя сети.

На передающей станции необходимо поместить байт, представляющий адрес назначения DID, на второе место в кадре, а байт, представляющий идентификатор отправителя SID - на третье место в кадре, то есть во второй и третий байты буфера TBUF соответственно. В данной работе заполнение позиций байтов DID и SID в буфере TBUF осуществляется прикладным уровнем. Предполагается, что самый первый символ, вводимый с клавиатуры для каждого кадра, всегда представляет идентификатор узла назначения. Поэтому программу для ПУ из пятой

лабораторной работы необходимо модифицировать так , чтобы поместить SID в третью байтовую позицию в буфере TBUF . Отметим , что SID всегда равен собственному идентификатору узла , то есть MID .

Что касается сетевого уровня (СУ) , то здесь по сравнению с пятой работой должна быть модифицирована только приемная часть . Передающая часть остается той же самой , так как она построена для выдачи данных из TBUF после того , как он подготовлен прикладным уровнем . Приемная часть должна быть переработана так , чтобы она могла анализировать приходящие последовательности кадров с целью определения , какие кадры должны быть приняты , а какие отвергнуты .

Построение алгоритмов

Для конечного автомата сетевого уровня могут быть выделены следующие состояния :

- передача ;
- ожидание приема PRE (обозначается через R0) ;
- ожидание приема MID (обозначается через R1) ;
- активность приема (обозначается через R2) .

Конечный автомат сетевого уровня становится сложным . Поэтому его удобно представить в виде двухуровневого автомата : на первом уровне анализируются обобщенные состояния (прием или передача) , а на втором - рассматриваются детали изменения состояний в обобщенном состоянии " прием " (R0 , R1 , R2) . В табл . 6 . 1 представлено кодирование состояний автомата сетевого уровня с помощью флага BYS .

Таблица 6 . 1

Кодирование обобщенных состояний автомата СУ

Состояние	Флаг BYS
Прием	0
Передача	1

Отметим , что флаг BYR больше не используется . Вместо этого используется целочисленная переменная rstate , с помощью которой кодируются подсостояния " приема " (табл . 6 . 2) .

Каждый раз , когда некоторый узел передает байт , все узлы , сетевой уровень которых находится в состоянии " покой " , принимают этот байт и сравнивают его с кодовой комбинацией PRE . В состоянии " прием " можно выделить два подсостояния : состояние " покой приема " , когда

поступающие из сети байты сравниваются с кодом PRE , и состояние “ активность приема “, когда поступающие из сети байты принимаются и помещаются в приемный буфер RBUF.

Таблица 6.2

Кодирование подсостояний состояния “ прием “

Подсостояние	rstate
R0	0
R1	1
R2	2

Работа конечного автомата сетевого уровня описывается таблицами переходов табл. 6.3 и табл. 6.4.

Таблица 6.3

Таблица переходов автомата СУ с обобщенными состояниями

Тек. сост.	След. сост.	Условие	Действия
Прием	Передача	$SP = 1$ и $rstate = 0$	$BYS := 1$
Передача	Прием	Послан EOT	$BYS := 0$, $rstate := 0$

Таблица 6.4

Таблица переходов автомата СУ с подсостояниями “ прием “

Тек. сост.	След.сост.	Условия	Действия
Rx	R1	Принят байт = PRE	$rstate := 1$
R1	R0	Принят байт $\neq$ MID	$rstate := 0$
R1	R2	Принят MID , RBF = 0	$rstate := 2$
R2	R0	Принят байт = EOT	$rstate := 0$, $DF := RBF := 1$ $rbend := rbnxt$, $rbnxt := rborg$

Реализация алгоритмов

Основная часть

Main

* Clear _ flags (5)

Прикладной уровень

* Tisr

Input

(5)

Сетевой уровень

Nisr

Receive

* Initialize	(5)	* Echo_byte	(5)	Transmit
* Schedule	(5)	* Output	(5)	* Enable_receive
* Start_send	(5)			PRE_received
* Start_display	(5)			DID_received
Input_MID				Accept
Display				
Input_byte				
End_display				

Знаком (*) обозначены программные модули , определенные в лабораторной работе № 5 . Они могут быть использованы в данной работе без каких - либо существенных изменений . Поясним новые и модифицированные программные модули , используемые в данной работе .

В модуле Main оператором узла вводится собственный адрес станции MID . Далее инициализируются UART , логические переменные (флаги) , указатели буферов передачи и приема , а также разрешаются прерывания CPU . Наконец , программа входит в бесконечный цикл , в котором выполняется процедура планирования .

Модуль Input_MID отображает на экране терминала сообщение “ Введите MID “ . Введенный байт помещается в переменную MID и с этого момента представляет идентификатор ID узла . Далее курсор перемещается в начало следующей строки .

Программный модуль Display производит отображение некоторого сообщения , выбираемого указателем txtptr . С помощью модуля End_display курсор перемещается к началу следующей строки .

Программный модуль Input используется для ввода кадра данных с клавиатуры терминала в буфер TBUF . Предполагается , что первая нажатая клавиша представляет адрес назначения кадра DID . Всякий раз при нажатии первой клавиши (BYI = 0) в буфер TBUF записываются два байта : байт DID и байт SID , равный байту MID . Для каждой нажатой клавиши программа Echo_byte производит отображение кода клавиши на экране дисплея .

Последовательность ввода байт кадра данных продолжается до тех пор , пока программа Input не обнаружит байт CR . При нажатии на клавишу CR завершается формирование кадра данных добавлением байтов LF и EOT . Кроме того , на экран терминала посылается байт LF , что вызывает перемещение курсора на следующую строку .

Когда программа Input заканчивает ввод кадра данных , она также изменяет состояние ПУ на “ готов “ , запрашивает передачу кадра (SP = 1) и устанавливает флаг TBF . Поскольку следующей операцией , выполняемой

с буфером TBUF , является передача кадра уровнем СУ , производится подготовка указателей буфера tbnxt и tbend так , чтобы они определяли в памяти границы буфера .

Модуль Receive вызывается программой Nisr при возникновении прерывания сетевого порта . Программа Receive всегда вводит байт из сетевого порта . Даже если байт должен быть отвергнут , он все равно считывается из сетевого порта , чтобы сбросить его запрос на прерывание . Если имеются ошибки , байт игнорируется .

Модуль PRE _ receive вызывается при получении преамбулы . В этом случае переменная rstate полагается равной единице (СУ переводится в режим ожидания второго байта кадра) .

Программа чтения Ассерпт вызывается тогда , когда из сети поступает некоторый байт при rstate = 2 . Если этот байт является байтом EOT , это означает конец кадра данных . В этом случае установкой флага DP производится запрос к ПУ на отображение кадра . Если принятый байт не является байтом EOT , он добавляется в буфер приема RBUF .

6.3 . Контрольные вопросы

1. Как реализовать в алгоритме многоадресный режим передачи ?
2. Каким образом реализовать подтверждение приема кадра ?
3. Вместо EOT используется счетчик байт кадра . Где его следует поместить в кадре ? Разработайте соответствующие алгоритмы .

6.4 . Демонстрации в лаборатории

1. Произведите дуплексный обмен кадрами между узлами А и В .
2. Организуйте обмен кадрами при несовпадении адресов MID и DID .
3. Создайте окно , отражающее состояние флагов СУ и ПУ .

6.5 . Содержание отчета

Блок - схемы и листинги программ с комментариями .

ЛАБОРАТОРНАЯ РАБОТА № 7 МЕТОД ДОСТУПА К СРЕДЕ ТИПА МАРКЕРНОЙ ШИНЫ

Цель работы : получить навыки разработки и исследования алгоритмов и программ управления работой ЛВС на основе метода доступа типа маркерной шины .

7.1. Индивидуальное задание

Разработайте и выполните отладку программ функционирования узла ЛВС с маркерным методом доступа. Настройку адаптера UART произведите в соответствии с исходными данными к первой лабораторной работе.

7.2. Методические указания

В данной работе рассматривается ЛВС с методом доступа типа маркерной шины [2]. Формат кадра данных : PRE , DID , SID , , данные , , EOT . Формат маркера : PRE , DID , EOT .

Пассивное логическое кольцо первоначально создается вручную путем ввода в каждый узел собственного сетевого адреса MID и адреса следующего узла DID . Вначале ни один узел не владеет правом передачи . Чтобы активизировать кольцо (начать передачу маркера) , необходимо предоставить маркер одному из узлов .

Потеря маркера определяется измерением времени с момента последнего владения маркером . Время обращения маркера зависит от скорости передачи данных , используемой в сети . В данной работе пакет кадра данных ограничен 86 байтами (80 символов текста сообщения плюс CR и перевод строки LF) . При скорости 2400 байт / с для передачи пакета следующему узлу может потребоваться около 0 , 37 секунд .

Каждый узел , который удерживает маркер , может передавать пакет , а после этого должен передать маркер . Узел теряет маркер в тот момент , когда он заканчивает его передачу . Если в логическом кольце имеется N узлов , то время ожидания маркера составит $0,37 * (N - 1)$. Если за это время маркер не будет получен , он считается потерянным . Измерение времени обращения маркера узел начинает при получении маркера в первый раз .

Построение алгоритмов

Построение алгоритмов основывается на многоуровневом подходе с ПУ и межуровневом интерфейсе , взятыми из лабораторной работы № 5 . Кроме пакетов данных , СУ в данной работе передает и принимает кадры маркера . Кадр маркера хранится в буфере MBUF , из которого происходит его передача , когда узел передает право использовать среду следующему узлу .

В операциях сетевого уровня можно выделить три состояния : “ прием ” , “ посылка пакета ” и “ посылка маркера ” . В состоянии “ прием ” СУ контролирует кадры , поступающие из сетевой среды , и ожидает кадр маркера или пакета . Если получен пакет , об этом

информируется прикладной уровень и СУ остается в состоянии “прием”, будучи готовым принять кадр другого пакета или маркер. Если принят маркер, это означает, что узел получил право передачи. Если имеется пакет, ожидающий передачи, состояние СУ меняется на “посылка пакета” и начинается передача пакета. В межуровневом интерфейсе ПУ - СУ запрос на посылку пакета осуществляется с помощью флага $SP = 1$. После завершения передачи пакета состояние СУ меняется на “посылка маркера” и начинается передача маркера. Кроме того, СУ может изменить свое состояние на “посылка маркера” непосредственно из состояния “прием”, если в момент прибытия кадра маркера отсутствует запрос от ПУ на передачу пакета ($SP = 0$). Определение обобщенного конечного автомата СУ представлено в табл. 7.1.

Таблица 7.1

Таблица переходов обобщенных состояний автомата СУ

Тек. сост.	След. сост.	Условия	Действия
Прием	Посылка маркера	Получен маркер и $SP = 0$	$BST := 1$, остановить таймер
Прием	Посылка пакета	Получен маркер и $SP = 1$	$BYS := 1$, $SP := 0$, остановить таймер
Посылка маркера	Прием	Послан кадр маркера	$BST := 0$, запустить таймер
Посылка пакета	Посылка маркера	Послан кадр пакета	$BYS := 0$, $BST := 1$

Состояние “прием” разбито на несколько подсостояний, как показано в табл. 7.2. В подсостоянии R0 СУ ожидает первого байта кадра PRE. Когда это происходит, состояние автомата меняется на R1, находясь в котором СУ ожидает второй байт кадра DID. Если байт поступает, когда СУ находится в состоянии R1, он сравнивается с MID. Если они равны, это означает, что кадр адресован данному узлу и должен быть принят. Поэтому состояние меняется на R2 и будет ожидаться третий байт. Если же байт, принятый в состоянии R1, не совпадает с MID, кадр отвергается. Состояние меняется на R0 и снова будет ожидаться байт преамбулы PRE.

Если третий байт равен EOT, это означает, что был принят маркер. В этом случае завершается ожидание маркера, обобщенное состояние

меняется либо на “посылка пакета”, либо на “посылка маркера”, а подсостояние “прием” снова меняется на R0. Если третий байт не является байтом EOT, принимаемый кадр считается кадром пакета данных, а третий байт считается байтом SID. Затем подсостояние “прием” меняется на R3, в котором будет осуществляться прием байтов, поступающих из среды, и размещение их в ячейки буфера RBUF, пока не будет принят байт EOT. Прием EOT в состоянии R3 означает конец пакета данных. Уровню ПУ сообщается о поступлении пакета и подсостояние “прием” снова меняется на R0.

Таблица 7.2

Подсостояние “прием” автомата сетевого уровня

Тек. сост.	След. сост.	Условие	Действия
R0	R1	Принятый байт = PRE	rstate := 1
R1	R0	Принятый байт $\neq$ MID	rstate := 0
R1	R2	Принятый байт = MID	rstate := 2
R2	R0	Принятый байт = EOT	rstate := 0, таймер $\uparrow$ *)
R2	R3	Принятый байт $\neq$ EOT и DP = 0 и BYD = 0	rstate := 3
R3	R0	Принятый байт = EOT	rstate := 0, DP := 1
Rx	R1	Принятый байт = PRE	rstate := 1

*) Кадр маркера принят, если SP = 1, тогда BSP := 1, иначе BST := 1.

Реализация алгоритмов

Основная часть

Main
 * Input_MID (6)
 Input_NID
 Input_token
 * Display (6)
 * Input_byte (6)
 * End_display (6)
 Clrar_flags
 Initialize

Прикладной уровень

* Tisr (5)
 * Input (6)
 * Echo_byte (5)
 * Output (5)

Сетевой уровень

Nisr
 Send
 Send_token
 * Enable_receive (5)
 Receive
 PRE_received
 DID_received
 Token_received
 Start_transmit

Schedule

* Accept (6)

* Start _ display (5)

Timer _ isr

Disp _ token _ lost

Start _ timer _ to _ wait _ for _ token

(*) - программы , определенные в работах 5 и 6 .

В программе Main оператор с клавиатуры терминала вводит MID и DID . Эти два шага определяются программами Input _ MID и Input _ NID . Далее программа Clear _ flags производит сброс логических переменных , а программа Inizialize производит инициализацию буферов и портов .

Далее оператор решает , какой узел будет инициировать маркер . Для этого вызывается программа Input _ token . Затем иницируется передача маркера путем установки флага BYS и разрешения прерываний передачи в сетевом порте . Наконец , программа Main входит в цикл , в котором вызывается программа планирования Schedule . Цикл повторяется до тех пор , пока не будет обнаружена потеря маркера ($TL = 1$) . Когда это произойдет , программа Display _ token _ lost выведет на экран сообщение о потере маркера .

Программа Nisr вызывается прерыванием сетевого порта , когда порт готов для вывода (во время передачи) или когда в порт поступает байт из сетевой среды (во время приема) . Когда при $BYS = 1$ возникает прерывание , предполагается , что узел передает пакет данных и вызывается программа Send . Программа Send _ token вызывается программой Nisr , если СУ занят передачей маркера ($BST = 1$) . После того как переданы все байты кадра маркера , для измерения времени ожидания маркера запускается таймер .

Программа Receive вместе с программами PRE _ received , Token _ received и Ассерт отвечает за обработку всех байтов , поступающих из сетевой среды , пока сам узел не производит передачу . Программа Token _ received анализирует третий байт кадра . Если это байт EOT , значит маркер был получен . В этом случае производится остановка таймера и путем вызова программы Start _ transmit иницируется передача кадра пакета данных или маркера .

Байты пакета данных добавляются в буфер RBUF программой Ассерт до тех пор , пока не будет получен байт EOT , который означает конец пакета .

Всякий раз , когда СУ принимает маркер , вызывается программа Start _ transmit . В этом месте принимается решение : ожидает ли передачи ($SP = 1$) пакет данных , подготовленный уровнем ПУ ? Если это так , иницируется передача пакета ($BYS := 1$) и сбрасывается запрос на

передачу пакета ($SP := 0$). Если пакета данных нет , инициируется передача маркера ($BST := 1$).

В данной работе используются две программы , относящиеся к таймеру . Программа `Start_timer_to_wait_for_token` вызывается в конце передачи маркера из программы `Send_token` . В этот момент производится инициализация таймера , чтобы начать отсчет времени ожидания . Когда принимается кадр маркера , таймер может быть остановлен программой `Token_received` .

Программа `Timer_isr` вызывается прерыванием таймера при его переполнении . Это означает потерю маркера и останавливает работу узла . Программа `Timer_isr` устанавливает флаг `TF` , останавливает таймер и сбрасывает прерывание . Флаг `TF` опрашивается программой `Main` . Когда определяется , что $TF = 1$, выводится сообщение “ маркер потерян ” и программа узла заканчивается .

7.3 . Контрольные вопросы

1. Как определить время ожидания маркера для различных скоростей передачи данных , различного числа узлов в сети и различных длин пакетов ?
2. Каким образом реализовать передачу многопакетных сообщений ?
3. Реализуйте подтверждение приема маркера .
4. Реализуйте обнаружение ошибок в пакетах и подтверждение приема пакетов .

7.4 . Демонстрации в лаборатории

1. Создайте кольцо из двух узлов . Произведите пересылку сообщений между узлами (фамилия , имя , отчество) .
2. Попробуйте создать кольцо с двумя узлами , имеющими одинаковые идентификаторы .

7.5 . Содержание отчета

- 1 . Блок - схемы алгоритмов работы сетевых узлов .
- 2 . Листинги программ с подробными комментариями .

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИССЛЕДОВАНИЮ БЛОКОВ НА ОСНОВЕ АДАПТЕРА ПОСЛЕДОВАТЕЛЬНОЙ СВЯЗИ

Интегральная микросхема i8250 является основой для построения адаптеров последовательной связи в персональных компьютерах IBM PC. Микросхема является универсальным асинхронным приемопередатчиком (UART), предназначенным для асинхронного обмена в последовательном формате между процессором (CPU) и периферийными устройствами (УВВ) [1, 3, 6]. Все входные и выходные сигналы совмещены со схемами ТТЛ - логики. Микросхема функционально включает в себя несколько отдельных блоков (рис. П1.1). Рассмотрим назначение сигналов UART и состав каждого блока в отдельности:

- D7 - D0 - двунаправленная шина данных и команд с тремя состояниями;
- MR - входная линия системного сброса UART в исходное положение;
- CS0, CS1, $\overline{\text{CS2}}$ - входы выбора микросхемы при работе с CPU;
- A0, A1, A2 - входы выборки внутреннего регистра UART;
- $\overline{\text{ADS}}$ - входной строб адреса;
- DISTR, $\overline{\text{DISTR}}$, DOSTR, $\overline{\text{DOSTR}}$ - входные сигналы разрешения операции чтения (записи) данных или статуса (команды) в (из) CPU;
- SIN - вход последовательных данных;
- RCLK - сигнал синхронизации ввода последовательных данных;
- $\overline{\text{RLS0}}$ - вход обнаружения сигнала на линии приема;
- $\overline{\text{RI}}$ - входной сигнал индикации кольца;
- SOUT - последовательный выход данных;
- BAUDOUT - частота синхронизации последовательных выходных данных;
- INTRPT - запрос прерывания;
- CSOUT - выход индикации выбора микросхемы;
- DDIS - выход индикации чтения данных из UART;
- XTAL1, XTAL2 - входы подключения кварца или тактового генератора;

Блок селекции адреса

На вход этого блока поступают три сигнала выбора микросхемы (CS0, CS1 и $\overline{\text{CS2}}$), входной строб адреса $\overline{\text{ADS}}$ и три сигнала выбора одного из внутренних регистров (A0, A1 и A2). Таким образом, доступ к микросхеме осуществляется как к восьми портам ввода - вывода. Адресную шину можно использовать различными способами для выбора внутренних регистров UART. Один из вариантов предусматривает подсоединение трех младших разрядов адресной шины CPU ко входам A0, A1 и A2 микросхемы. Это обеспечит выбор i8250 по восьми непрерывным адресам.

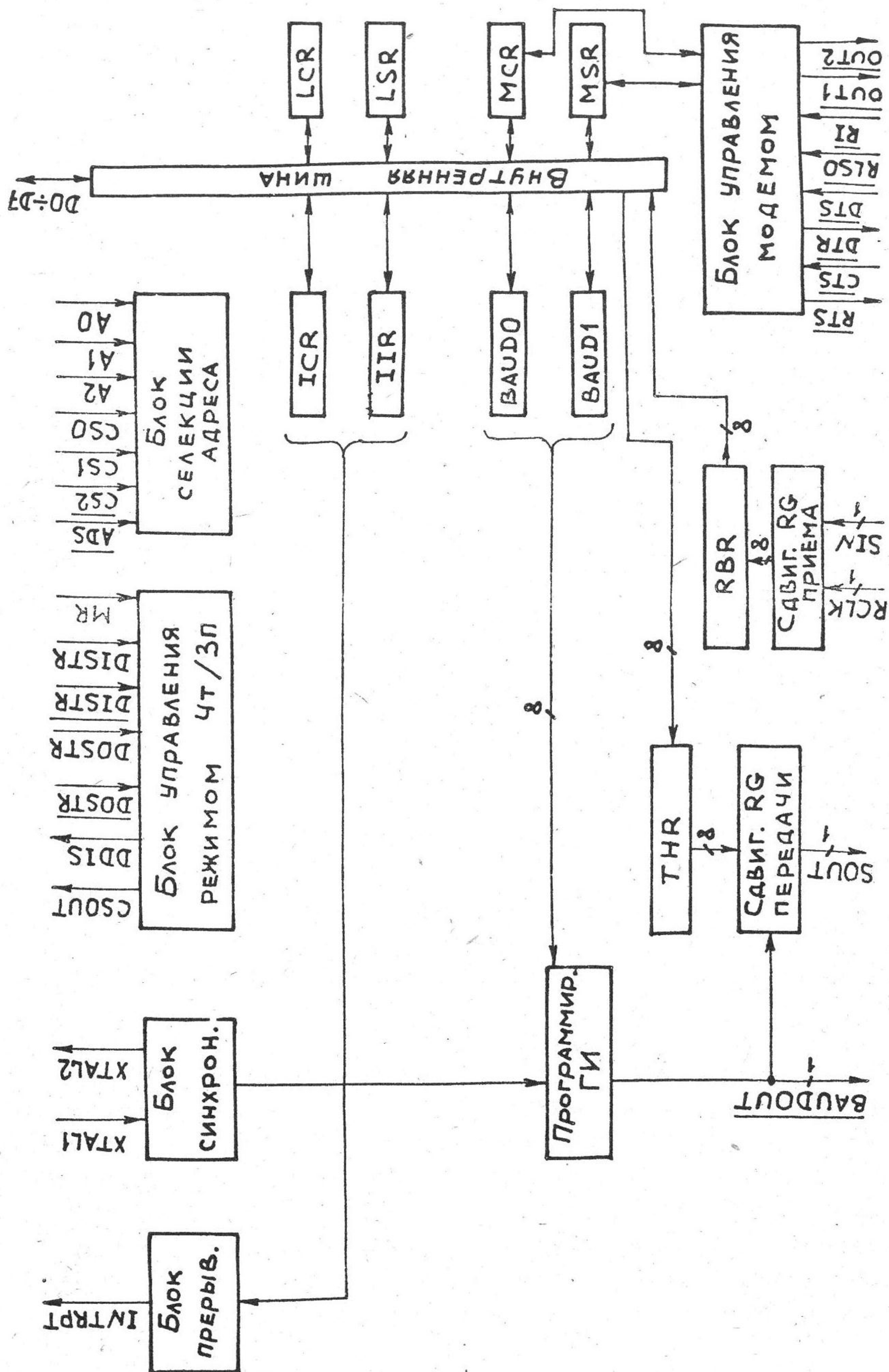


Рис. П1. 1. Структурная схема UART i8250

Схема генерации сигналов CS0 , CS1 и $\overline{CS2}$ удобна для систем со многими внешними устройствами . Один сигнал $\overline{CS2}$ может быть использован для выбора одного из четырех однотипных устройств . Идентификация устройств осуществляется с помощью сигналов CS0 и CS1 .

Блок управления режимами записи - чтения

В интерфейсе CPU и i8250 управляющие сигналы записи и чтения позволяют простым способом организовать связь с любым типом микропроцессора . Существуют по два управляющих сигнала для чтения (DISTR и $\overline{DISTR}$) и записи (DOSTR и $\overline{DOSTR}$) . Дублирующие управляющие входы различаются только уровнями истинности и позволяют сократить количество интерфейсных схем , которые должны быть установлены между системной шиной и i8250 .

Различные варианты управления чтением и записью приведены на рис . П1.2 . В случае а) управление чтением и записью применимо к памяти или УВВ . Раздельные селекторы $\overline{MEMSEL}$ (выбор RAM) и $\overline{IOSEL}$ (выбор I/O) определяют выбор между памятью и внешним устройством .

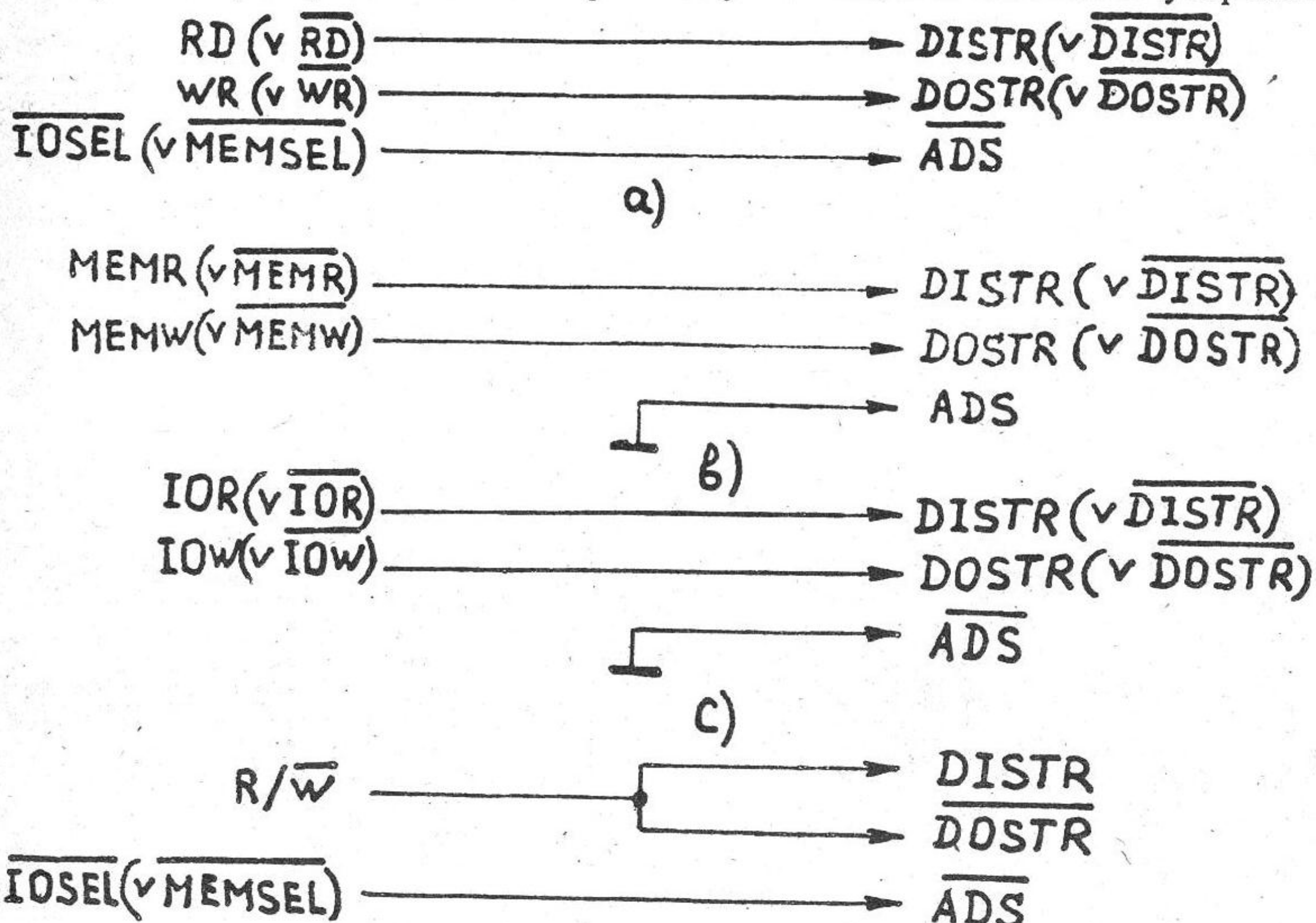


Рис . П1 .2. Варианты управления чтением - записью

Варианты b) и c) соответствуют микропроцессору , имеющему раздельное управление чтением из RAM или УВВ и записью в RAM или устройство УВВ . В этих двух случаях сигнал $\overline{ADS}$ не нужен и должен быть заземлен . В варианте d) показан случай , при котором CPU использует

один сигнал с двумя активными состояниями для различения операций чтения и записи.

Блок синхронизации

Схема тактирования i8250 необычна по сравнению с другими устройствами последовательного ввода. Обычно UART получает три тактовых сигнала (рис. П1.3): один - для тактирования входных последовательных данных (TxR), другой - для тактирования выходных последовательных данных (TxT), третий - для управления внутренними схемами устройства (CLK).

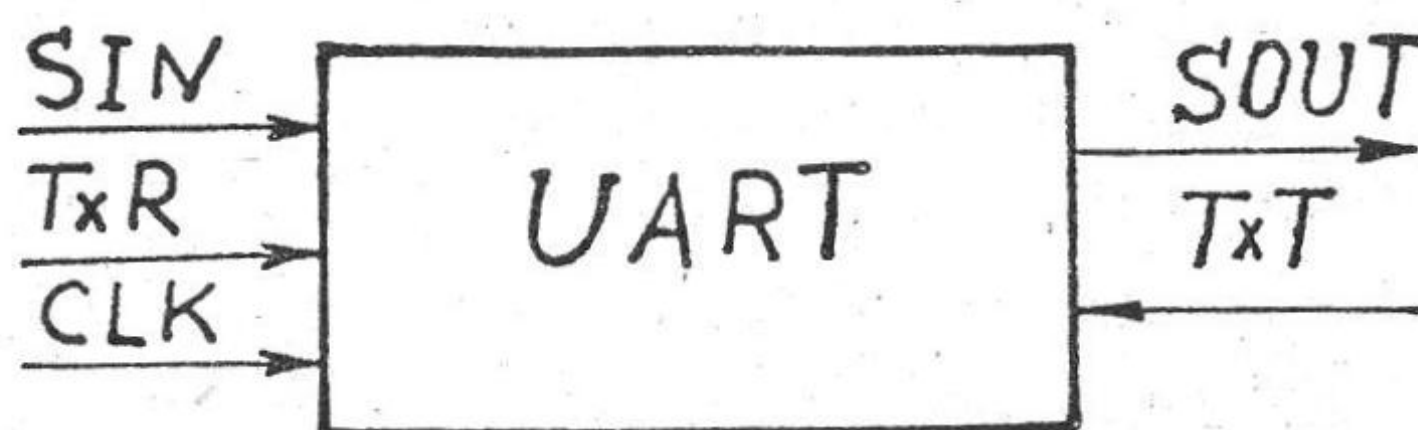


Рис. П1.3. Стандартная схема тактирования UART

В БИС i8250 объединены схема тактирования выходных последовательных данных (SOUT) и схема внутреннего тактирования. Можно использовать внешний кварц (BQ), либо вход тактового сигнала для генерации внутреннего тактового сигнала CLK и тактового сигнала выходных последовательных данных BAUDOUT. Эти два случая показаны на рис. П1.4. Здесь символ "+" обозначает схему деления тактового сигнала CLK для получения сигнала BAUDOUT. Этот сигнал идентифицирует частоту последовательного вывода данных на выходе SOUT.

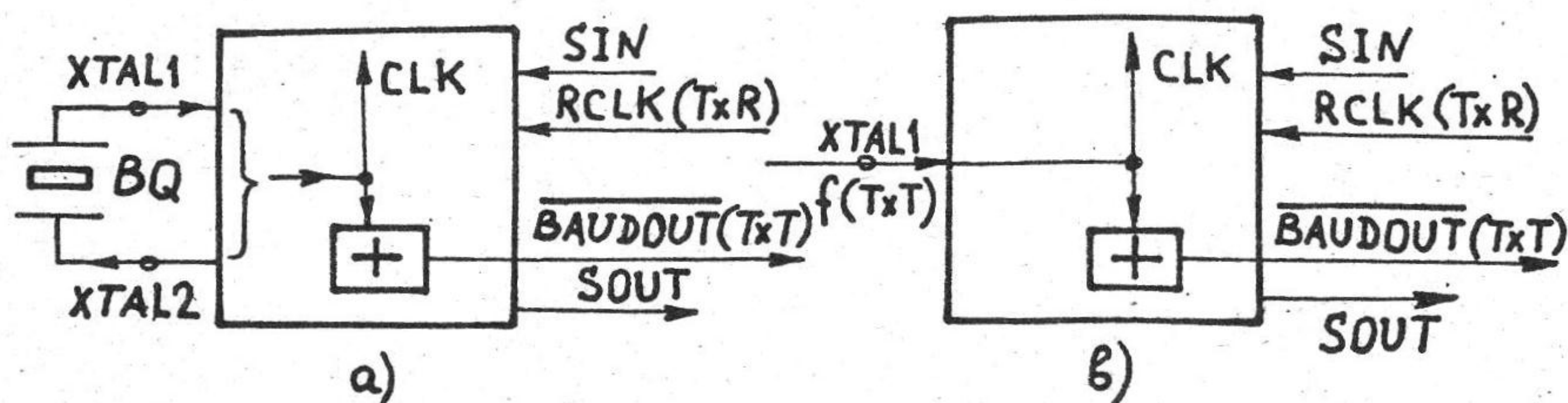


Рис. П1.4. Схема синхронизации UART i8250

$f(TxT)$ на рис. П1.4, б представляет некоторую функцию от сигнала TxT. Другими словами, если известна частота вывода последовательных данных SOUT, то можно ввести входной тактовый сигнал через вход

XTAL1 , частота которого меньше желаемой частоты вывода последовательных данных на некоторый фиксированный коэффициент N :

$$f (T_{xT}) = N * \overline{BAUDOUT}.$$

Если i8250 управляет частотой вывода последовательных данных , то частоту кварца (или тактового сигнала , выдаваемого CPU) можно разделить для получения любой требуемой частоты в бодах (бит в секунду) на выходе SOUT .

Для тактирования ввода последовательных данных через вход SIN можно использовать отдельный тактовый сигнал RCLK , либо использовать тактовый сигнал вывода последовательных данных , соединив контакты CLK и $\overline{BAUDOUT}$.

Адресуемые регистры БИС i8250

Микросхема UART i8250 имеет 10 адресуемых регистров . Одни и те же адреса портов могут соответствовать различным внутренним регистрам в зависимости от направления передачи информации и значения седьмого бита регистра управления линией . Этот бит обозначают DLAB .

Адресация отдельных регистров микросхемы осуществляется с помощью сигналов , поданных на входы A0 , A1 и A2 (табл . П1.1) .

Таблица П1.1

Адресация регистров UART

A0 A1 A2	Доступные регистры
0 0 0	При передаче или приеме данных (бит DLAB = 0) : регистр данных передатчика (THR) при записи в порт ; регистр данных приемника (RBR) при чтении из порта ; При задании формата слова делителя частоты (бит DLAB = 1) : делитель частоты (BAUD0) - младший байт .
0 0 1	При передаче или приеме данных (бит DLAB = 0) : регистр управления прерываниями (ICR) ; При задании формата слова делителя частоты (DLAB = 1) : делитель частоты (BAUD1) - старший байт .
0 1 0	Регистр идентификации прерывания (IIR)
0 1 1	Регистр управления линией (LCR)
1 0 0	Регистр управления модемом (MCR)
1 0 1	Регистр состояния линии (LSR)
1 1 0	Регистр состояния модема (MSR)

Рассмотрим более подробно описание отдельных регистров UART.

Регистры данных THR и RBR

Если UART передает или принимает информацию, бит DLAB должен быть равен нулю. В этом случае при записи в порт с номером 000 данные оказываются в регистре THR. Отсюда данные без дополнительных управляющих воздействий передаются в сдвиговый регистр передатчика, который формирует слово информации и бит за битом выдает его в интерфейс по линии SOUT (первым выдается младший бит слова D0). Чтение байта из внутреннего регистра с номером 000 (RBR) возвращает последний принятый UART байт данных.

Регистры делителя частоты BAUD0 и BAUD1

Если бит DLAB = 1, то порты с адресами 000 и 001 используются как защелки для делителя частоты передачи (приема). Значение делителя - это число, полученное делением частоты внутреннего источника синхронизации микросхемы i8250 CLK (равна 1,8432 МГц) на частоту передачи в бодах, умноженную на 16.

$$\text{BAUDOUT} = \text{CLK} / N = 16 * (\text{частота передачи в бодах}),$$

где N - это 16-битовое значение, полученное из регистров BAUD.

Делитель частоты, равный 1, соответствует максимально возможной скорости передачи в 115 200 бод. Это следует из того, что $1\,843\,200 / (115\,200 * 16) = 1$. Частоте передачи в 57 600 бод соответствует делитель частоты, равный 2, а частоте 38 400 бод - делитель, равный 3.

Регистр управления линией LCR

Этот регистр используется для задания формата слова, передаваемого по интерфейсу (рис. П1.5).

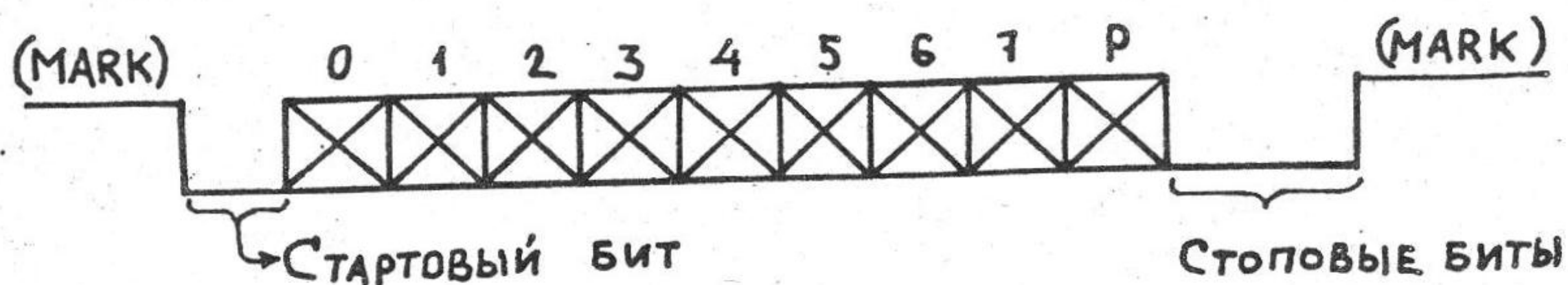


Рис. П1.5. Формат асинхронного символа

Назначение разрядов регистра LCR приведено в табл. П1.2.

Назначение разрядов регистра LCR

Бит	Описание
D7	Бит DLAB. Управляет доступом к регистрам данных, делителя частоты и управления прерываниями
D6	= 0 - обычное функционирование адаптера ; = 1 - вызывает посылку в интерфейс сигнала BREAK
D5 - D3	Выбор способа контроля по паритету : = XX0 - отсутствие бита контроля (P) ; = 001 - бит контроля формируется по четности ; = 011 - бит контроля формируется по нечетности ; = 101 - бит контроля равен 1 ; = 111 - бит контроля равен 0 .
D2	Число стоп - битов : 0 - 1 стоп - бит ; 1 - 2 стоп - бита.
D1 , D0	Длина слова в битах : 00 - 5 ; 01 - 6 ; 10 - 7 ; 11 - 8 бит .

Исходное состояние линии последовательной передачи данных - уровень логической единицы (MARK). Когда начинается передача данных, уровень линии переходит в 0. Если линия находится в этом положении больше положенного времени, считается, что произошел разрыв связи.

Стартовый бит сигнализирует о начале передачи данных. Далее передаются биты данных (с D0 по D7). Если используется бит четности, то передается и он. В самом конце передается 1 или 2 стоповых бита. Затем линия снова переходит в состояние MARK.

Регистр управления модемом MCR (табл. П1.3)

Таблица П1.3

Бит	Описание (если бит равен единице)
D7 - D5	Не используются
D4	Запуск автономного теста для UART (LOOP)
D3	Разрешение прерывания от адаптера UART (OUT2)
D2	Определяемый пользователем запрос прерывания (OUT1)
D1	Устанавливает сигнал на линии интерфейса RTS
D0	Устанавливает сигнал на линии интерфейса DTR

По четырем линиям управления модемом выдаются инверсии установок значений битов регистра MCR.

Регистр состояния линии LCR

Этот регистр сообщает компьютеру информацию о состоянии как модема, так и тракта приема (передачи) адаптера (табл. П1.4). Бит D0 сбрасывается в 0, как только CPU прочитает содержимое буфера приема. Биты D1 - D4 сбрасываются, когда CPU читает содержимое регистра статуса линии. Бит D5 сбрасывается в 0 как только CPU запишет в буфер очередной байт данных. Бит D6 устанавливается в 1, когда регистр передатчика пустой. В этом случае через выход SOUT выдается сигнал высокого уровня (MARK).

Таблица П1.4

Назначение разрядов регистра LSR

Бит	Описание (если бит равен единицы)
D7	Всегда 0
D6	Сдвиговый регистр передатчика пуст
D5	Буферный регистр передатчика пуст
D4	Принят BREAK
D3	Ошибка кадрирования (рассинхронизация)
D2	Ошибка паритета
D1	Ошибка переопределения данных
D0	Готовность принятых данных

Регистры, связанные с прерываниями

В микросхеме i8250 имеется два регистра, связанных с прерываниями: ICR - регистр управления прерываниями и IIR - регистр идентификации прерывания. В UART существует 10 условий, которые могут привести к возникновению запроса на прерывание по линии INTRPT. Эти 10 условий разбиты на четыре уровня приоритетов следующим образом:

- 1) наивысший приоритет - одно из четырех состояний ошибок в регистре статуса линии;

- 2) второй приоритет - доступны принятые данные ;
- 3) третий приоритет - пустой буфер передатчика ;
- 4) четвертый (самый низкий приоритет) - изменение на одном из входов управления модемом .

Запросы на прерывание обрабатываются в соответствии со схемой , показанной на рис . П1. 6 . Прерывание наивысшего и самого низкого уровня приоритета может быть вызвано одним из четырех условий . Поскольку прерывания разрешаются или запрещаются по уровням , можно разрешить или запретить все четыре условия , которые вызывают прерывания разных уровней приоритета .

Запись в регистр ICR байта 00h запрещает генерацию запроса прерывания для всех возможных ситуаций . Запись значения 0Fh напротив разрешает генерацию запроса прерывания для всех типов ситуаций . Определить причину прерывания можно чтением регистра IIR (табл . П1. 6) .

Таблица П1. 6

Назначение разрядов регистра IIR

Бит	Описание
D7 - D3	Не используются
D2 - D1	Идентификация причины прерывания : = 00 - изменение разрядов регистра MSR ; = 01 - пуст регистр данных передатчика THR ; = 10 - готовность данных в регистре RBR ; = 11 - выделена ошибка или сигнал BREAK
D0	= 0 - прерывание не обработано ; = 1 - нет активных прерываний .

Одному прерыванию может соответствовать несколько причин сразу и устранение одной причины тем не менее не вызовет установки в 1 бита D0 регистра IIR. Поэтому обработчик прерывания не должен завершать свою работу до тех пор , пока не устранил все одновременно возникшие прерывания.

Сброс микросхемы i8250 осуществляется подачей сигнала высокого уровня на вход MR. При этом происходит обнуление регистров ICR , IIR , LCR , MCR. Выходные сигналы имеют уровни : SOUT , DDIS , OUT1 , OUT2 , $\overline{RTS}$, $\overline{DTR}$ - высокий ; BAUDOUT , INTRPT - низкий.

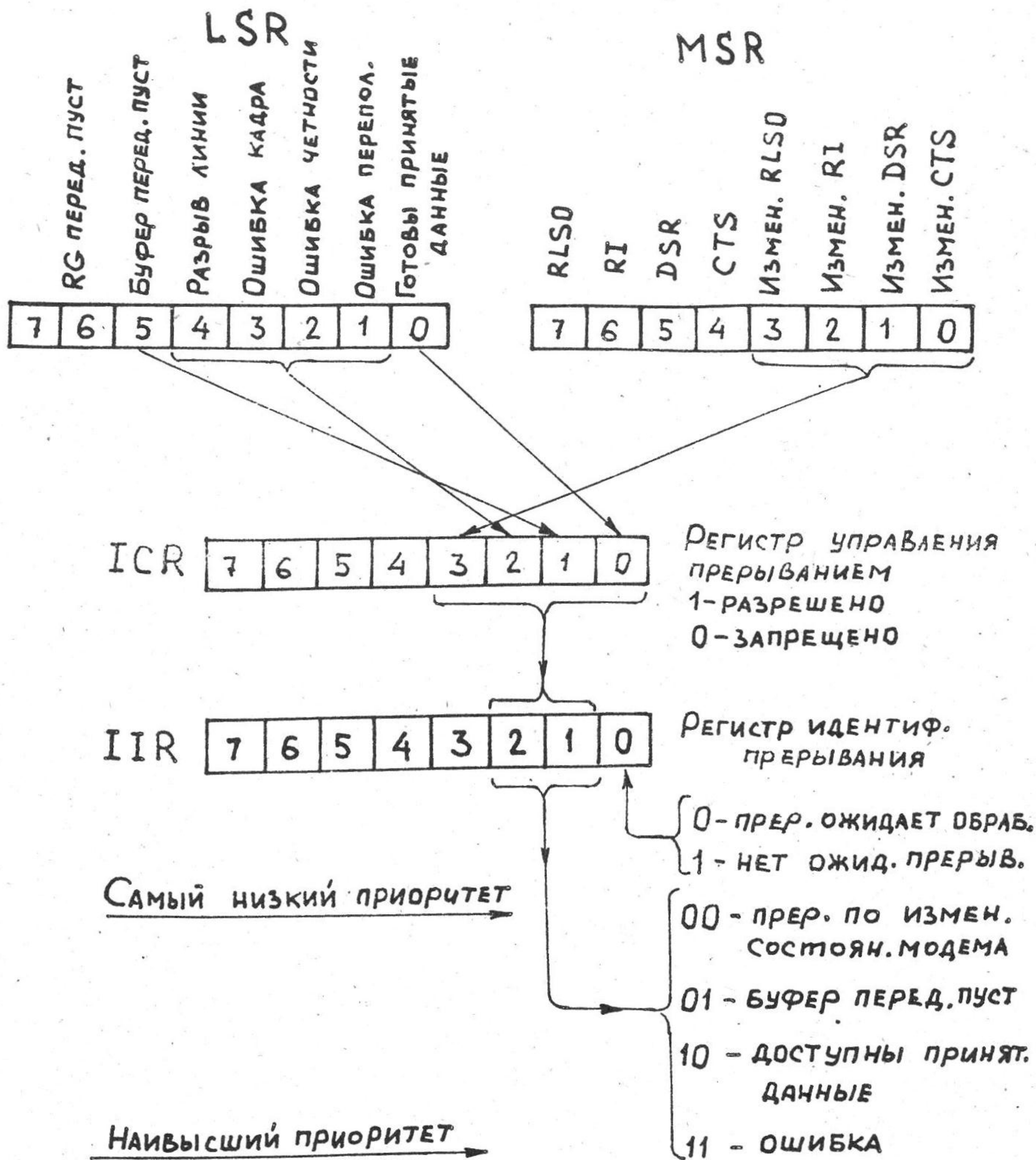


Рис. П1.5. Схема организации прерываний UART

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИССЛЕДОВАНИЮ БЛОКОВ НА ОСНОВЕ ИНТЕРВАЛЬНОГО ТАЙМЕРА

Микросхема i8253 (аналог КР580ВИ53) представляет собой программируемый таймер (PIT), предназначенный для формирования программно - управляемых временных интервалов. Микросхема может также использоваться как счетчик внешних событий, программируемый делитель частоты и одновибратор с программируемой длительностью импульса.

Программируемый таймер i8253 включает в себя три независимых канала (рис. П2.1), каждый из которых может быть запрограммирован на работу в одном из шести режимов для двоичного или двоично - десятичного счета.

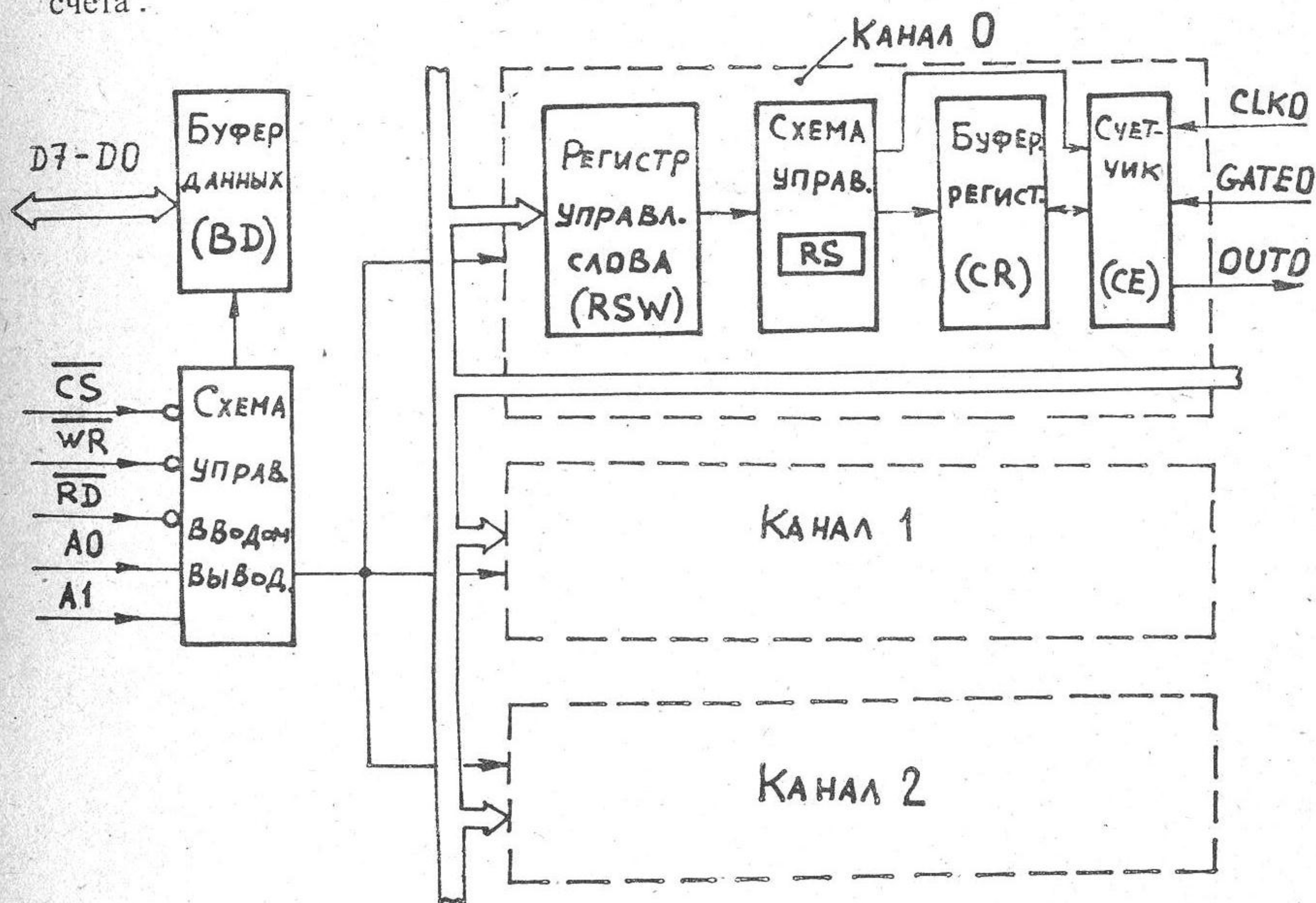


Рис. П2.1. Функциональная схема таймера

В состав функциональной схемы входят буфер шины данных (BD), схема управления вводом - выводом и три канала К2 - К0. Каждый из каналов содержит регистр управляющего слова (RSW), схему управления, 16 - разрядные буферный регистр (CR) и счетчик (CE). В состав схемы

управления входит 8 - разрядный регистр состояния канала (RS), содержимое которого можно сосчитать .

Для программирования работы каждого счетчика в регистр RSW необходимо командой OUT заслать из CPU по шине данных управляющее слово . В зависимости от комбинаций управляющих сигналов на входах $\overline{CS}$, $\overline{WR}$, $\overline{RD}$, A0 и A1 различные блоки таймера подключаются к шине данных в соответствии с табл . П2. 1.

Таблица П2.1

CS	WR	RD	A1	A2	Направление передачи информации
0	0	0	0	0	D7 - D0 → счетчик 0
0	0	1	0	1	D7 - D0 → счетчик 1
0	0	1	1	0	D7 - D0 → счетчик 2
0	0	1	1	1	D7 - D0 → регистр управляющего слова
0	1	0	0	0	Счетчик 0 → D7 - D0
0	1	0	0	1	Счетчик 1 → D7 - D0
0	1	0	1	0	Счетчик 2 → D7 - D0
0	*	0	1	1	D7 - D0 → в третьем состоянии
0	1	1	*	*	D7 - D0 → в третьем состоянии
1	0	0	0	0	D7 - D0 → в третьем состоянии

Формат управляющего слова (УС) таймера представлен на рис. П2.2.

D7 D6	D5 D4	D3 D2 D1	D0
Выбор канала	Чтение / запись	Режим работы	Счетчик
0 0 - канал 0	0 0 - код команды CLC	0 0 0 - режим 0	0 - 2 - й
0 1 - канал 1	0 1 - младший байт	0 0 1 - режим 1	1 - 2 / 10
1 0 - канал 2	1 0 - старший байт	x 1 0 - режим 2	
1 1 - запрет	1 1 - младший , затем старший байт	x 1 1 - режим 3 1 0 0 - режим 4 1 0 1 - режим 5	

Рис . П2. 2. Формат управляющего слова

Управляющее слово загружается в тот канал таймера , адрес которого указан в самом формате управляющего слова , и сохраняется там во все время работы или до следующего программирования .

После загрузки управляющего слова необходимо загрузить в каналы константы пересчета . Константа пересчета загружается в таймер по командам вывода (OUT), но с адресом , формирующим на входах A0 и A1 код соответствующего канала .

Порядок загрузки каналов управляющими словами и константами пересчета строго определен . Возможны два варианта . Первый предполагает загрузку в любой последовательности всех управляющих слов , а затем констант пересчета . Второй предполагает загрузку управляющего слова для любого канала , а затем константы пересчета для этого же канала .

Состояния счетчиков таймера может быть прочитано тремя способами :

- 1) чтение по обычным командам ввода (IN) позволяет прочесть состояние счетчика в любой момент времени . Код канала задается на адресных входах A0 и A1. Необходимым условием для выполнения этой операции является остановка счета перед выполнением команды чтения , т. е. $GATE = 0$. Сначала считывается младший байт , затем - старший .
- 2) чтение по команде CLC (чтение без остановки счета) . Для этого необходимо загрузить команду CLC (рис . П2.3) в определенный момент времени . При загрузке команды на входах $A0 = 1$ и $A1 = 1$. Разряды D7 и D6 задают адрес защелкиваемого канала . Разряды $D5 = 0$ и $D4 = 0$ указывают на код команды CLC . Разряды D3 - D0 не используются и могут принимать любые значения .
- 3) чтение состояния канала по команде RBC позволяет в любой момент времени прочитать слово состояния канала (рис.П2.4). Команда загружается в таймер также , как управляющее слово . Особенностью этой команды является возможность выполнения операции одновременно в нескольких каналах .

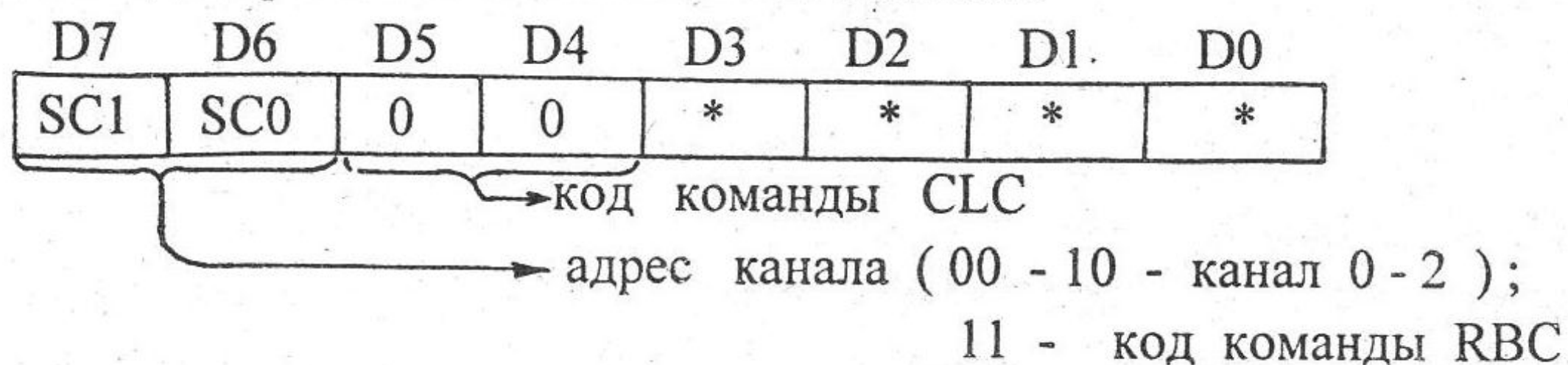


Рис . П2.3. Формат команды CLC

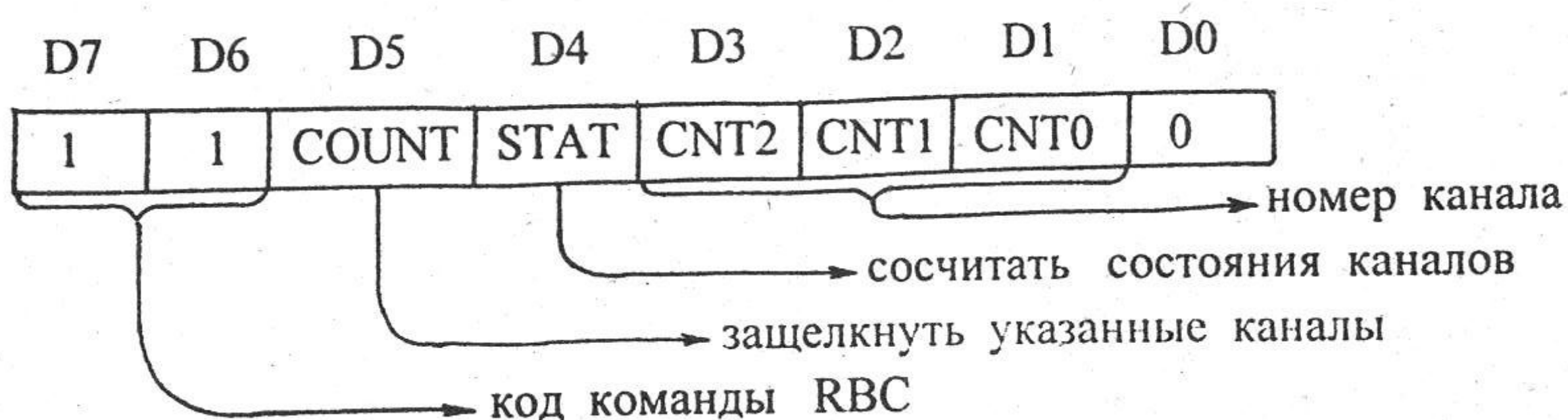


Рис . П2. 4. Формат команды RBC

Каждый канал имеет регистр слова состояния (RS). Формат слова состояния приведен на рис . П2 . 5. Разряд D7 слова состояния канала несет информацию о состоянии выхода OUT в момент выполнения команды RBC . Разряд D6 является флагом обновления (FN) регистра констант . Флаг позволяет определить , произошла ли загрузка константы из регистра констант в счетчик , или нет . Изменение состояния флага FN можно проиллюстрировать следующим примером :

- при записи управляющего слова флаг $FN = 1$;
- при записи констант в CR флаг $FN = 1$;
- при загрузке константы в CE флаг $FN = 0$.

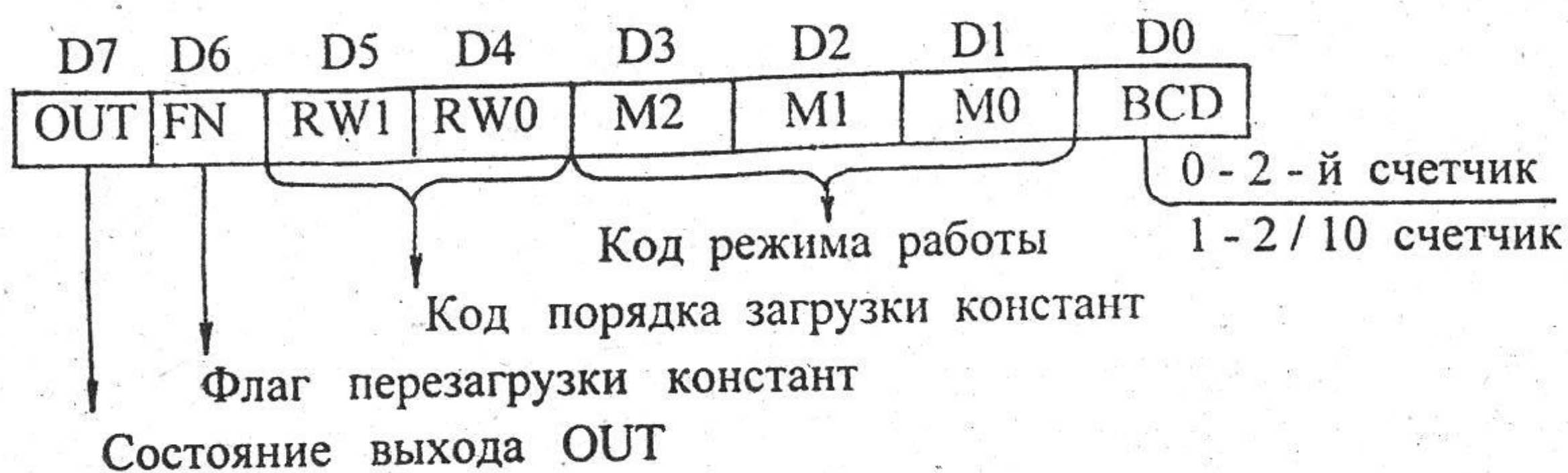


Рис . П2. 5. Формат слова состояния канала

В режиме 0 таймер работает как программируемый счетчик . С момента записи числа в счетчик , настроенный на режим 0 , до окончания счета на соответствующем выходе OUT удерживается сигнал низкого уровня . Уменьшение содержимого счетчика происходит под воздействием импульсов CLK при высоком уровне сигнала GATE. После окончания счета уровень сигнала OUT становится высоким и сохраняется до следующей загрузки счетчика (рис . П2. 6 , а) . Низкий уровень сигнала GATE запрещает счет .

В режиме 1 счетчик таймера используется в качестве программируемого одновибратора. На выходе OUT формируется сигнал низкого уровня длительностью $T * n$, где T - период тактовых импульсов на входах CLK, а n - число, загружаемое в счетчик. На выходе OUT по положительному фронту сигнала GATE удерживается низкий уровень сигнала, который изменяется после окончания счета. Режим 1 является режимом с перезапуском. По каждому фронту сигнала на входе GATE регистр CE перезагружается содержимым регистра CR. Если во время счета в таймер загружается новая константа, то она не влияет на текущий счет. Новый счет начнется только по фронту следующего сигнала GATE (рис. П2.6, b).

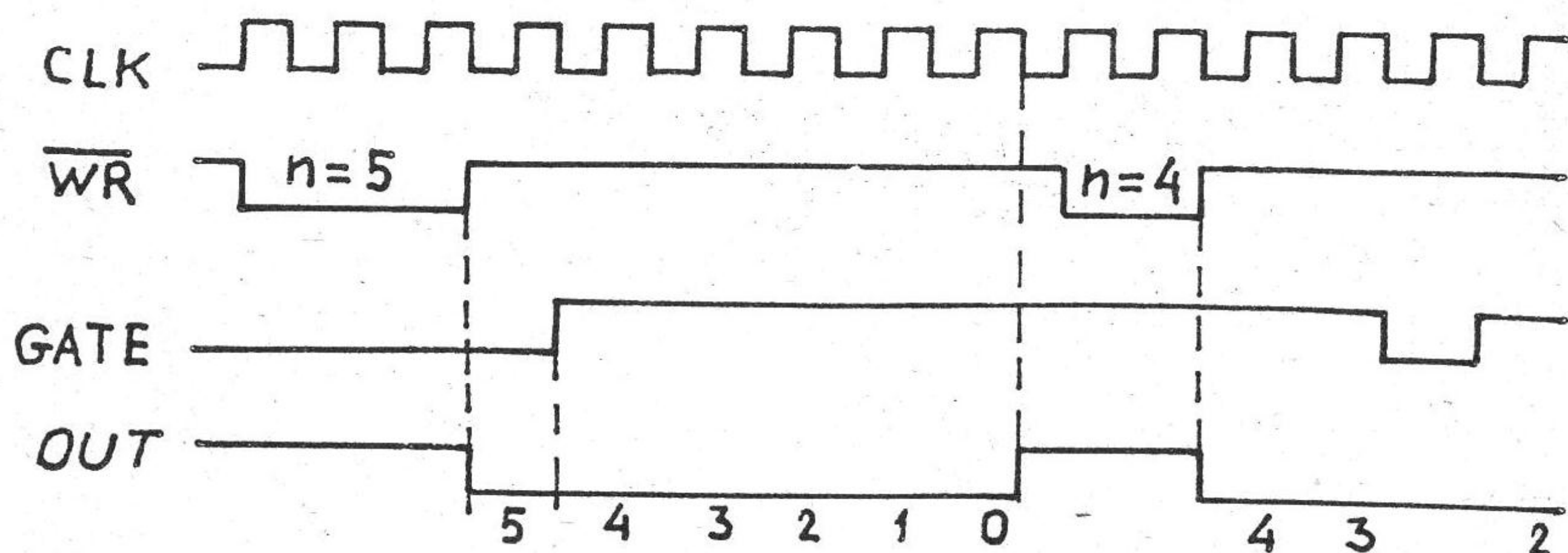
В режиме 2 счетчик работает как программируемый делитель частоты. Сразу же после загрузки управляющего слова на выходе OUT устанавливается высокий уровень напряжения. При $GATE = 1$ на выходе OUT с частотой F_{clk} / n устанавливается нулевой сигнал на время одного периода CLK (n - число, загружаемое в счетчик). Режим 2 является режимом с автозагрузкой, то есть после окончания цикла счета регистр CE автоматически перезагружается и счет повторяется. Перезагрузка канала новой константой не влияет на текущий счет. Новый цикл начинается по окончании предыдущего. При $GATE = 0$ на выходе OUT устанавливается напряжение высокого уровня и счет останавливается. При $GATE = 1$ счет продолжается, что позволяет синхронизировать работу канала с внешними событиями.

Режим 3 - генератор импульсов со скважностью два - аналогичен режиму 2, за тем исключением того, что на выходе OUT формируются импульсы с длительностью полупериодов; равной $n / 2 * T_{clk}$ при четных n ; $(n + 1) / 2 * T_{clk}$ для положительных и $(n - 1) / 2 * T_{clk}$ для отрицательных полупериодов (рис. П2.7, а). Этот режим является режимом с автозагрузкой. Перезагрузка константы во время счета не влияет на текущий счет. Новый счет начинается по окончании предыдущего. Снятие сигнала GATE приостанавливает счет, установка его - продолжает цикл счета. В этом режиме канал может работать только с константой больше трех.

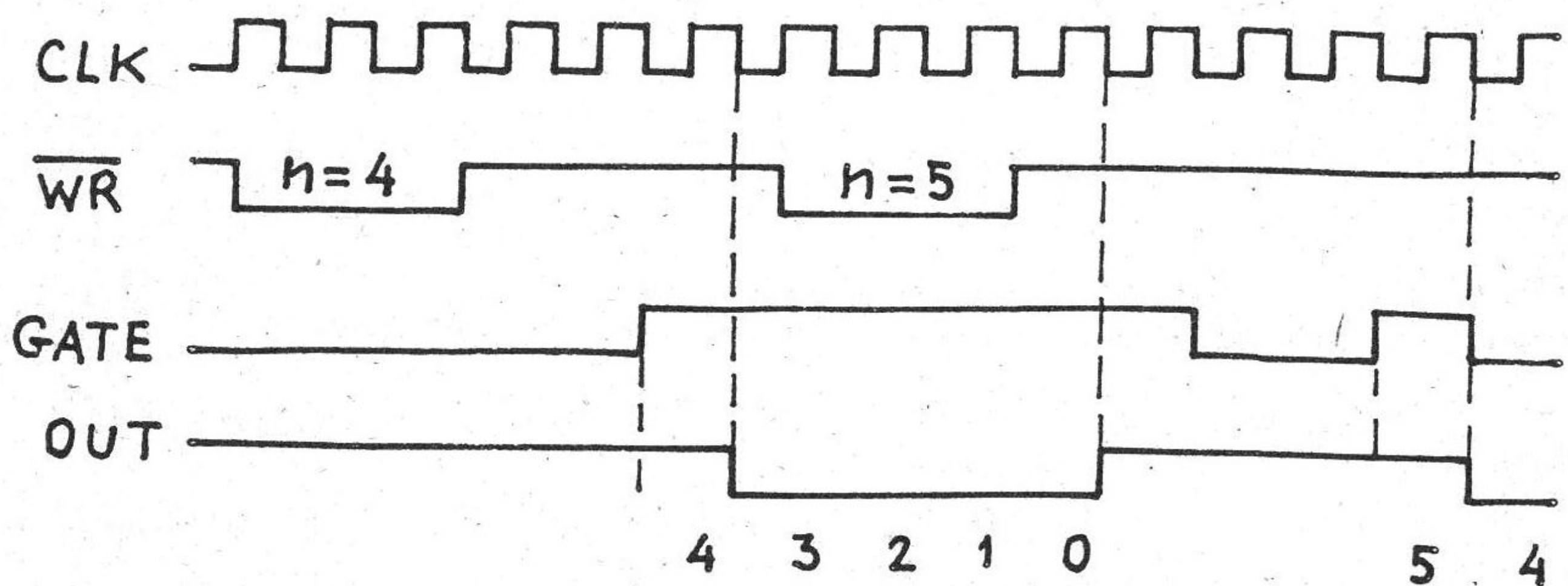
В режиме 4 - программно-запускаемого генератора - по окончании отсчета числа, загруженного в счетчик, на выходе OUT устанавливается сигнал низкого уровня на время одного периода сигнала CLK. Высокий уровень сигнала на выходе OUT устанавливается сразу после загрузки управляющего слова. Сигнал высокого уровня на выходе GATE разрешает счет, причем первый импульс CLK загружает счетчик CE константой из

CR, а второй - начинает счет. Таким образом, сигнал, равный длительности периода тактовой частоты, устанавливается на выходе OUT через $n + 1$ тактовых периода (рис. П2.7, б). Загрузка новой константы во время счета приводит: при записи младшего байта к остановке текущего счета, а при записи старшего - к запуску нового счета.

Режим 5 - аппаратно - запускаемый одновибратор - аналогичен режиму 4 по способу формирования сигнала на выходе OUT и режиму 1 по действию сигнала GATE. На выходе OUT устанавливается сигнал низкого уровня на время одного периода сигнала CLK после отсчета загруженной в CE константы. Загрузка в CE константы из CR происходит по фронту сигнала GATE (рис. П2.7, с).

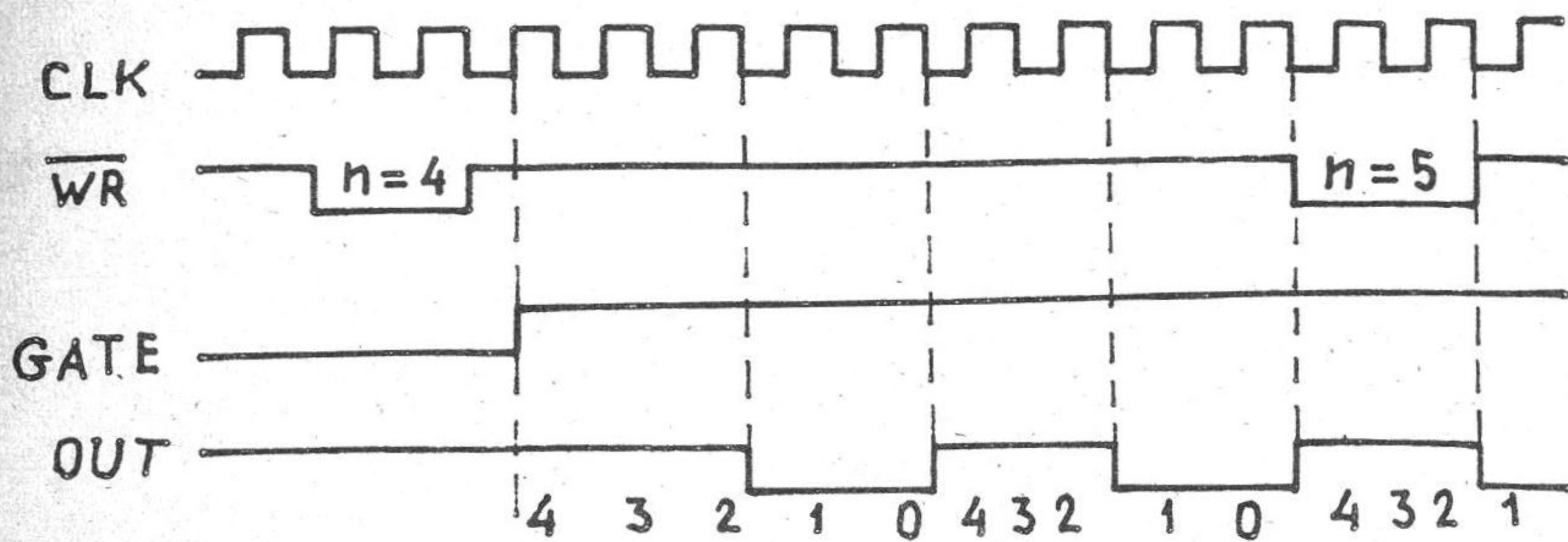


а)

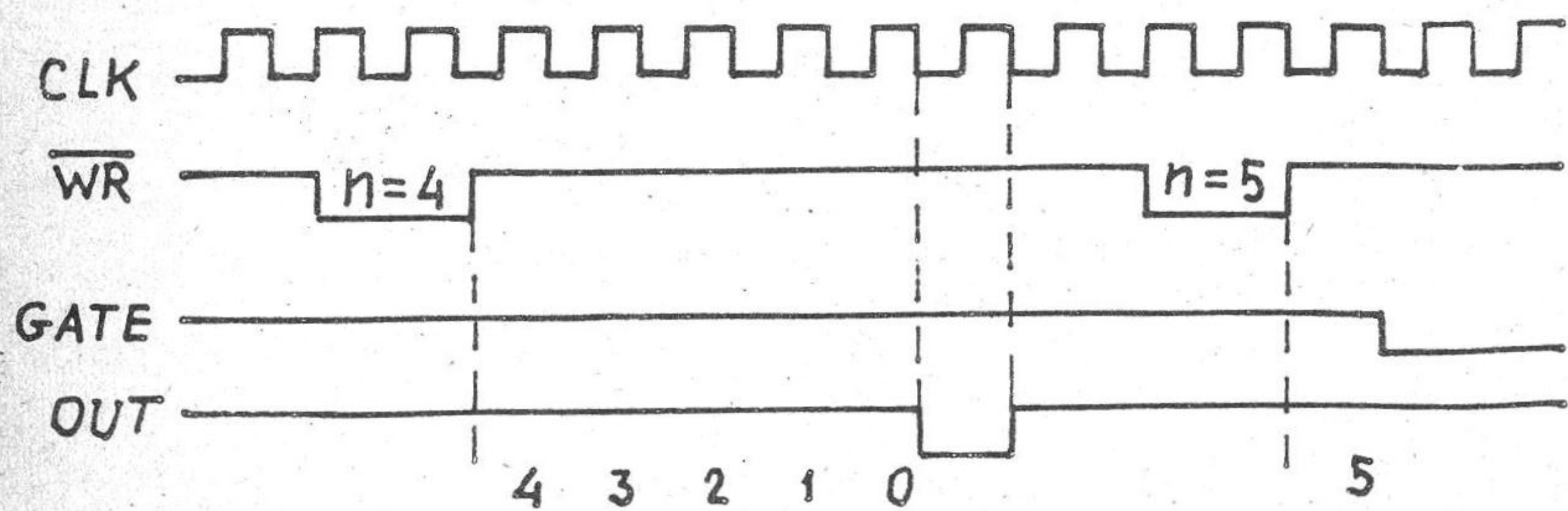


б)

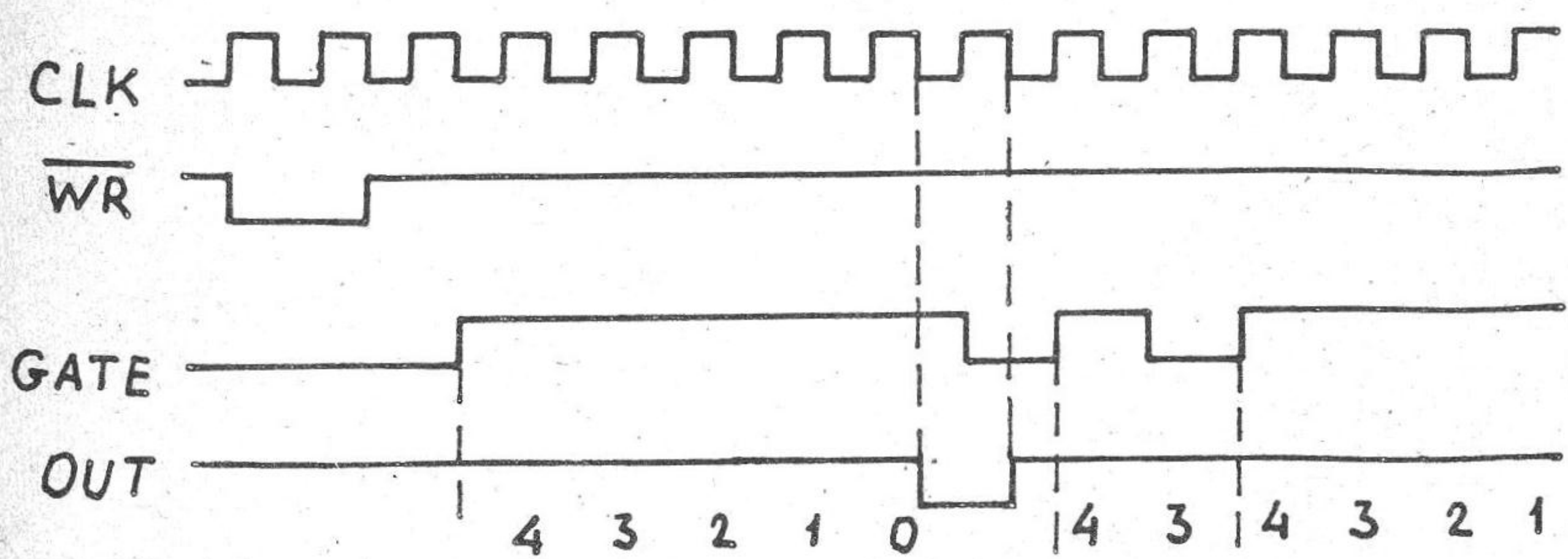
Рис. П2.6. Временные диаграммы работы таймера в 0 и 1 режимах



a)



б)



с)

Рис . П2 . 7 . Временные диаграммы работы таймера в 3 , 4 и 5 режимах

СОДЕРЖАНИЕ

Стр.

ОБЩИЕ МЕТОДИЧЕСКИЕ УКАЗАНИЯ.....	3
Лабораторная работа N1. АНАЛИЗ , ПРОГРАММИРОВАНИЕ И ИССЛЕДОВАНИЕ АДАПТЕРА ПОСЛЕ- ДОВАТЕЛЬНОЙ СВЯЗИ.....	4
Лабораторная работа N2. АНАЛИЗ , ПРОГРАММИРОВАНИЕ И ИССЛЕДОВАНИЕ ТАЙМЕРА.....	10
Лабораторная работа N3. АСИНХРОННАЯ ПЕРЕДАЧА БАЙТОВ С ИСПОЛЬЗОВАНИЕМ МЕХАНИЗМА ОПРОСА СОСТОЯНИЙ.....	13
Лабораторная работа N4. ПЕРЕДАЧА И ПРИЕМ КАДРОВ С ИС- ПОЛЬЗОВАНИЕМ МЕХАНИЗМА ПРЕРЫ- ВАНИЙ.....	20
Лабораторная работа N5. ИССЛЕДОВАНИЕ МНОГОУРОВНЕВОГО ИНТЕРФЕЙСА.....	32
Лабораторная работа N6. ПЕРЕДАЧА И ПРИЕМ КАДРОВ С АДРЕ- САЦИЕЙ.....	41
Лабораторная работа N7. МЕТОД ДОСТУПА К СРЕДЕ ТИПА МАРКЕРНОЙ ШИНЫ.....	45
Приложение 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИССЛЕДОВА- НИЮ БЛОКОВ НА ОСНОВЕ АДАПТЕРА ПОСЛЕ- ДОВАТЕЛЬНОЙ СВЯЗИ.....	51
Приложение 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИССЛЕДОВА- НИЮ БЛОКОВ НА ОСНОВЕ ИНТЕРВАЛЬНОГО ТАЙМЕРА.....	61

УЧЕБНОЕ ИЗДАНИЕ

Методические указания и задания к лабораторным работам
по дисциплине «Сети ЭВМ». Часть I.

(для студентов специальности 7.09.15.01 дневной и вечерней форм обучения)

Составитель: ГУБАРЬ ЮРИЙ ВЛАДИМИРОВИЧ

Подп. к печати 13.12.96. Формат 60x84 1/16. Бумага тип. № 2. Печать офсетная
Усл. печ. л. 3,95 Усл. кр.-отт. 4,18 Уч.-изд. л. 66 Тираж 250 экз, Заказ № 966

Типография «Новый мир»
340050, Донецк, ул. Артема, 96