

Микропроцессорные системы

Конспект лекций Хартова В. Я.

Автор: **Екатерина DanCat Флегонтова**

Подготовка выпуска: **Алексей Avenger911 Маянц**



Распространение осуществляется на условиях [Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 Unported License](https://creativecommons.org/licenses/by-nc-nd/3.0/). Никакая часть этого материала не может быть использована в коммерческих целях или для создания производных произведений. Распространение разрешается только при надлежащем упоминании автора и указании лицензии. Все права защищены. Для получения права на использование данного материала на желаемых вами условиях [свяжитесь](#) с авторами.

Микропроцессорные системы.

4 наб. заучита ; Хартов Вячеслав Яковлевич
в мед. сем-ре курсовая ; экзамен

1) В.А. Хартов, микроконтроллеры

AVR. Практикум

изд-во МДТУ, 2007г

2) Н.Н. Щенкунов, А.П. Дуанов
средства

„Микропроцессоры и системы“

М., Радио и связь, 1989г.

3) В.В. Сташин, А.В. Зусов, О.Ф.

Многомигунца, Проектирование

устройств - в на функционировании микроконтроллера

Энергетический издатель, 1990г.

Тренировочные продукты в наб.-ом пакете
0. www.ubinst.ru / file / hartov / archive.zip - набр

1. www.sinc.com, Preview 32. (MCS-51)

www.phyton.ru

2. пакет www.oitme.ru AVR Studio 4.xx
(xx: 6...13)

скачать

3. пакет VM Lab с сайта vmlab.ru
описание 123.pm.ru

4. пакет www.labcenter.co.uk
Proteus 6.73. (или 6.90)
(профессиональный пакет 8/разных видов МК)

5. CV AVR
слайд-работы: (STR 500 набор)

1. Порты ввода/вывода
по два человека

2. Арифметическая обработка данных в МК AVR

3. Таймеры

4. Канал посылки бв./бав VART
(Universal Asynchronous)

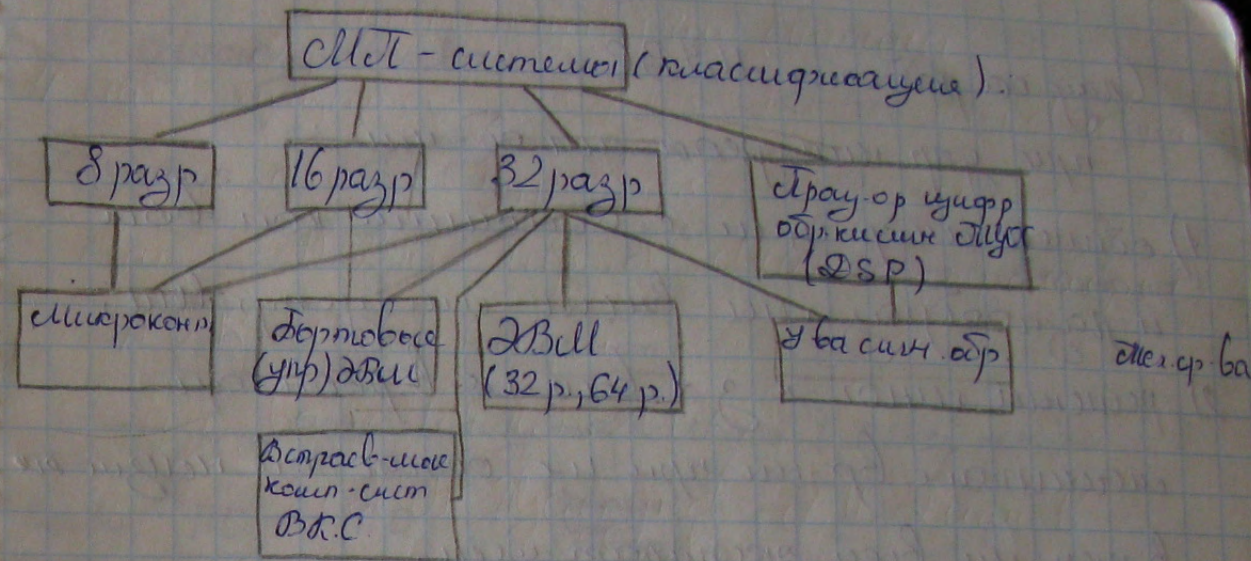
5. Канал посылки бв. бав

6. Отладка программ на дж. МК
(используя CV AVR)

с сайта кафедры /участники работы/

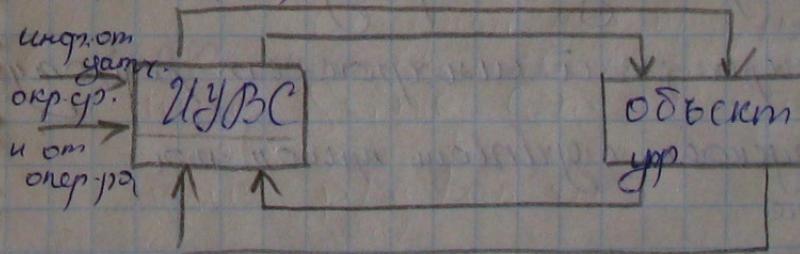
пакет подобие Хартмана

программа на Avr Studio



Основные понятия:

1) Информационно-управляющая вычислительная система представляет комплекс аппаратных и программных средств, способных под управлением оператора выполнять автоматическую обработку информации (ИУВС)



2) Управление - целенаправленное воздействие на объект с целью перевода его в требуемое состояние.

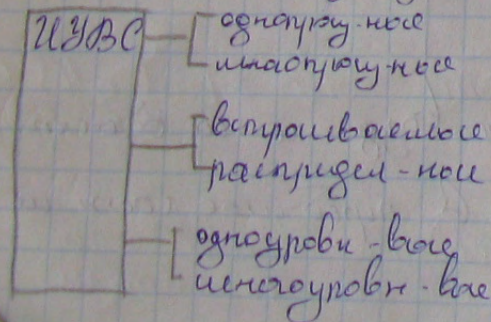
Сложность работы ИУВС

при управ-нии реал-ми об-м:

- 1) обмен инфор-цией с большим коли-вом и разнообраз-ми внеш-ми устр-ми;
- 2) решение много-х з-т вопр-ов в реал-ом масштабе вр-ни при их относ-ой неизм-ти в теч-ии всей эксплуат-ции;
- 3) обеспечение продолж-ть непер-ой работы при высокой уст-ти к отказам.

Эти фак-ры ок-ют влиянии и на СТО:

- 1) Разнообр-зи прогн ср-ств вв/выв. инфор-ции.
- 2) Необход-ть учета реал-го вр-ни, т.е. нап-тие ср-ств контроле вр-ни ф-но присутств-ть.
- 3) Наличие диспетчера, к-й управл-ет работой всех прогн, запускает с учетом приор-тов.



Многоуровневая система управления:

OS Windows

структурная схема

корпоративная сеть (АСУ предпр.)
(авт. сист. для управ-ия)

OS Win
Scada
гидро-мех
отобр-ие об-та

информационная сеть (Ethernet)

промышленная сеть (Modbus, Profibus, CAN)

OS DOS

OS-9, VxWorks

полевые шины

PC - рабочие станции

S - серверы

ПО - пункт оператора

ПЛК - программируемый логический контроллер (PLC)

вынос-ые устр-ва вв/выв.

одноразовые ЭВМ (микроконтроллеры).

≈ 12 бит, 1/4-16 бит, 1/4-114 бит

Технические средства, используемые
на различных уровнях управляющей
системы.

1. Верх. ур-но - ПК, РС (раб. станции)
2. Ср. ур-но - Прогр-ве мош-е контрол-р
(SMART, Schneider Electric,
Группа Advantech.)

одноплатные коши-ры,

модули v_b/v_a (до 300 нами ≈ 200 фюз.)

3. Личн. ур-но - микроконтрол-р: 1) CISC-
арх-р; прил-р, MCS-51 (Intel) ~ 1980г.

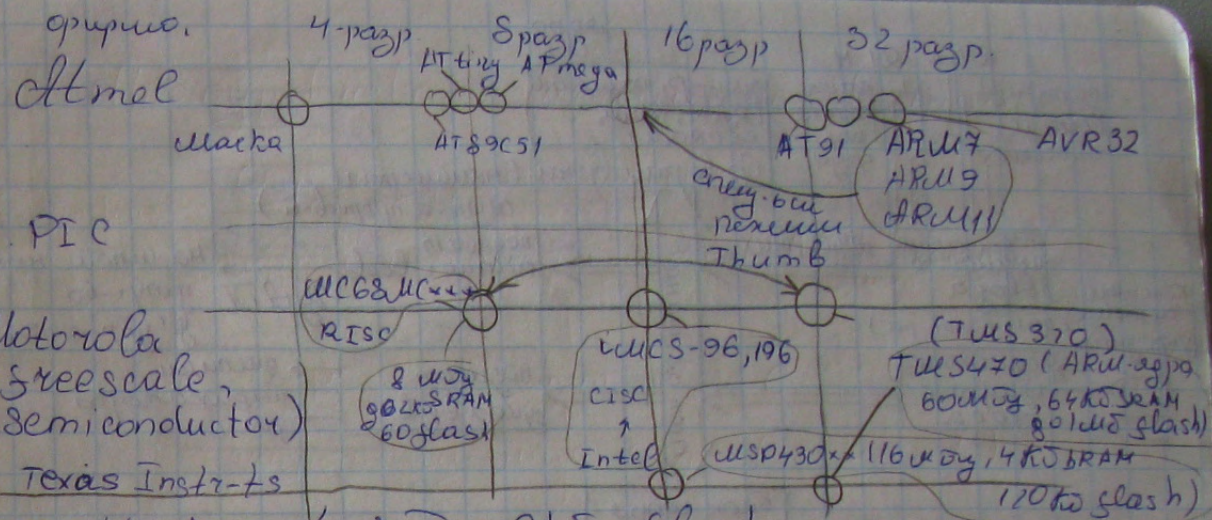
- 2) RISC-арх-ры; Microchip, PIC-комм-
(корр. малое количество)
роллер (до 35 команд), Atmel

(до 130 команд)

- #### 4. Системы на кристали (Suppress)

Любой МЭ имеет проц-ор, уст. вв^д вв^д вв^д, понимать,
Схемы проц-а по применению.

Микро- и мезо-ресурсы; изъём в виде ^{суммы}



Marka - 4Mg, 8k5 flash

A489C51 - (capex-on CISC, uzr na c nom-ro
flash mexre-uñ), 64 kb flash
64 kb SRAM (onep nom.)

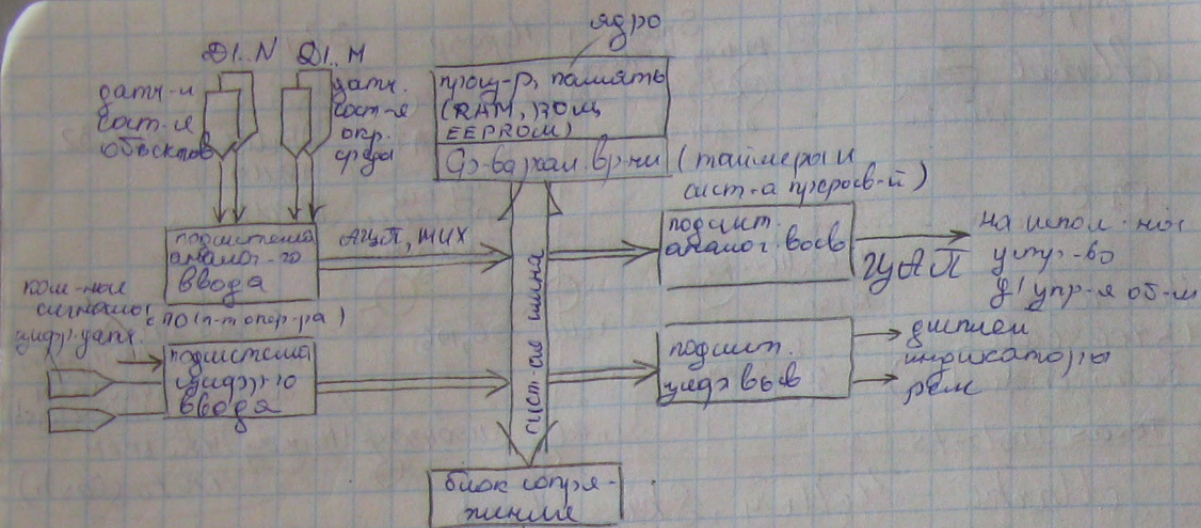
Affinity } approx RISC
 ATmega } - 16 MHz flash
 - 4 KB SRAM

ARМ7 - ; орак ки орокурит-ае ЭВМ,
; Вах-ае сает-а с баъоииии возм-ии

ARM7 - +5 mDu
ARM9 - 120 mDu, 256 K5 SRAM
2048 K5 flash

✕ ✕ ✕

Структура управляющей микро ЭВМ.



→ передача информации

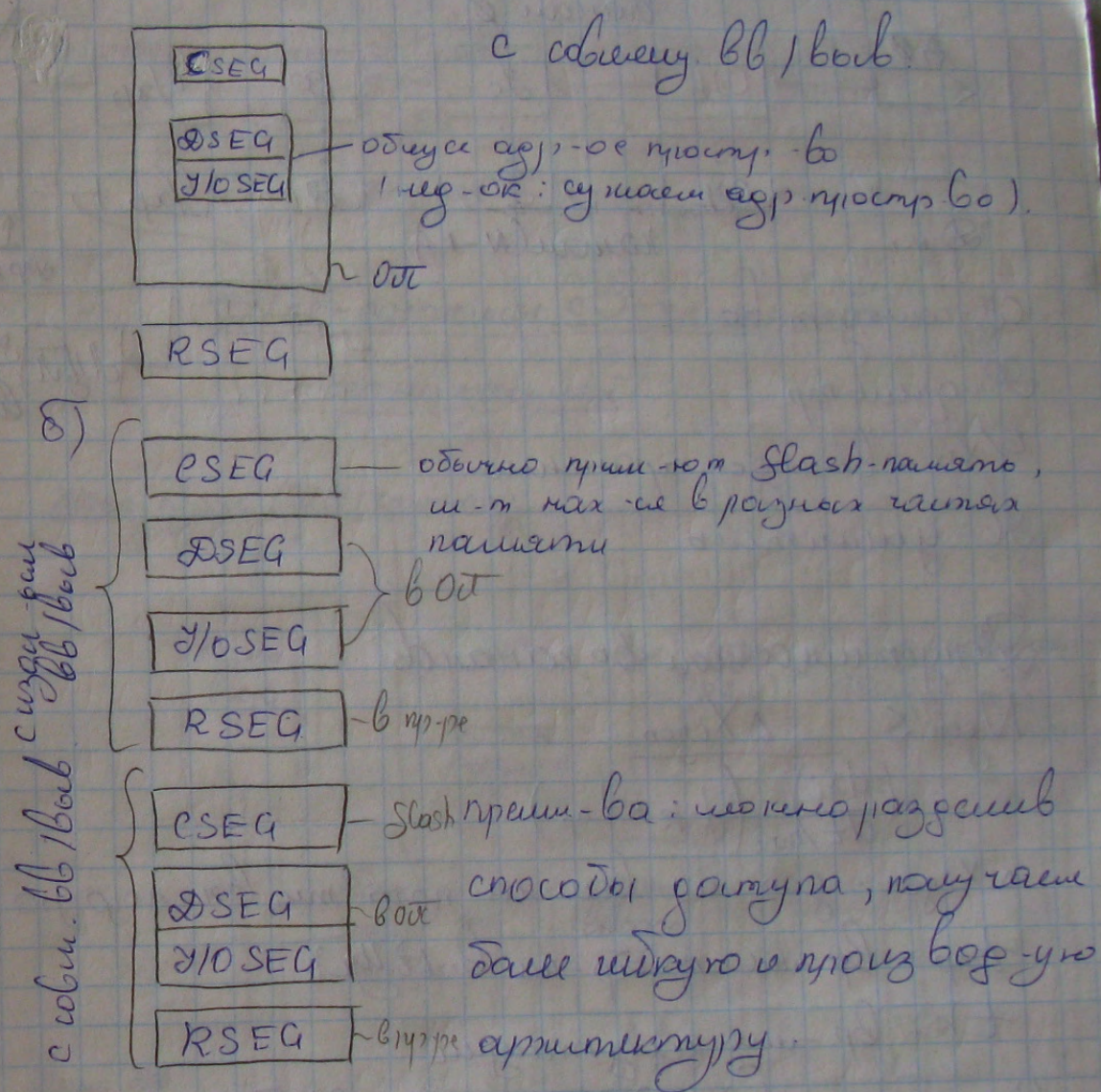
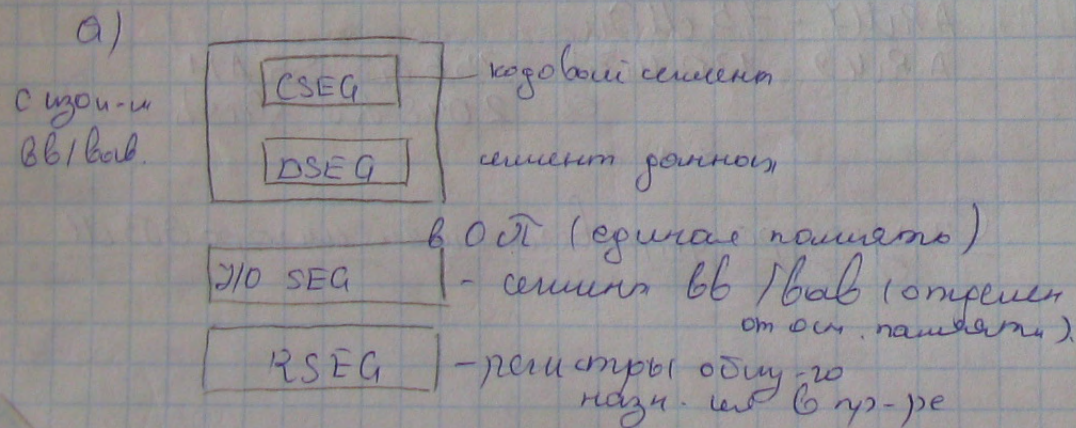
→ сигналы

АЦУ - арифметико-логическое устройство

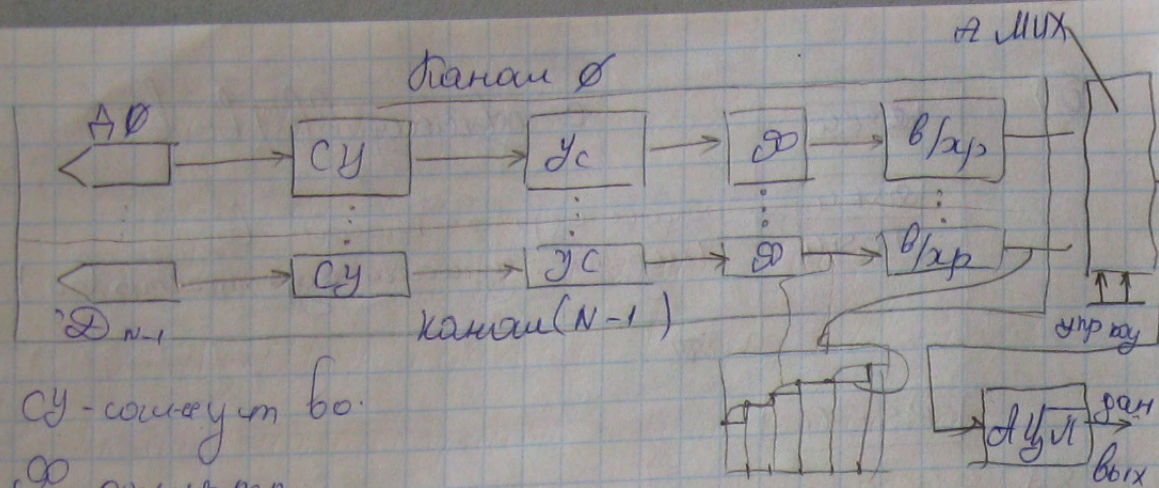
А. Организация "ядра"

Арх-ры: а) мейнфреймового типа

б) гарвардского типа



Б. Подсистема аналогового ввода многоканальная с несколькими АЦУ



СУ - суммирующее
 Д - делитель
 В/Хр - выходное значение
 УС - умножитель

Допустимое кол-во каналов

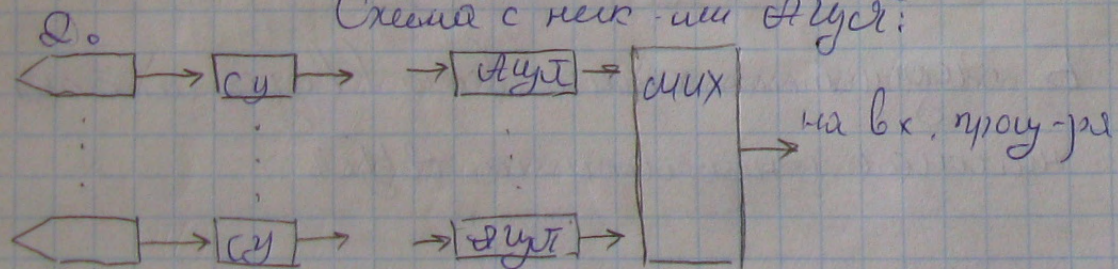
$$N_{\text{дон}} \leq \frac{\Delta X_{\text{дон}}}{\left(\frac{dx_i}{dt}\right)_{\text{max}} (t_k + t_{\text{пр}})}$$

$\Delta X_{\text{дон}}$ - дон-ая величина погр-ты вы-ход-ных

$t_{\text{пр}}$ - вр-мя прот-ра в АЦУ

t_k - вр-мя коммутации

Схема с нек-ми АЦУ:

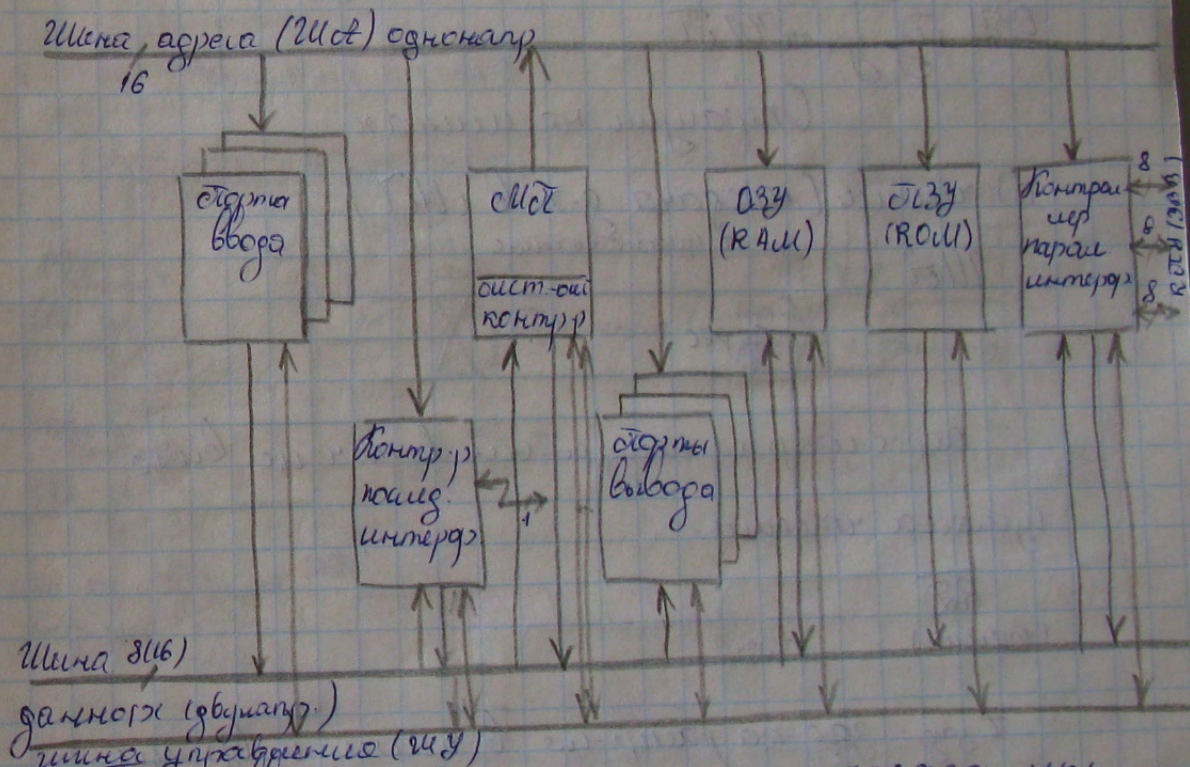


$N_{\text{дон}}$

Ну-ка: много АЦУ

$$N_{\text{дон}} \leq \frac{\Delta X_{\text{дон}}}{\left(\frac{dx_i}{dt}\right)_{\text{max}} t_k}$$

ММХ-системы с 3-х шинной организацией.



Шина 8(16)

данных (двухнапр.)

шина управления (МУ)

Порты вх-вых - электр-ые узлы, к-ми предет-ны

ПЗУ ввода/вывода

Стр-е ввода данных с порта:

1) Д-е порт-ют на ММХ и далее в ММХ.

Странный вид $g(x)$:

- ## Операции на числах

[illegible]

$\overline{RZ}$
(inverse) $+ \text{gay}$

24) гашение
выходное
состояние

3-е сраст-ие

Ула

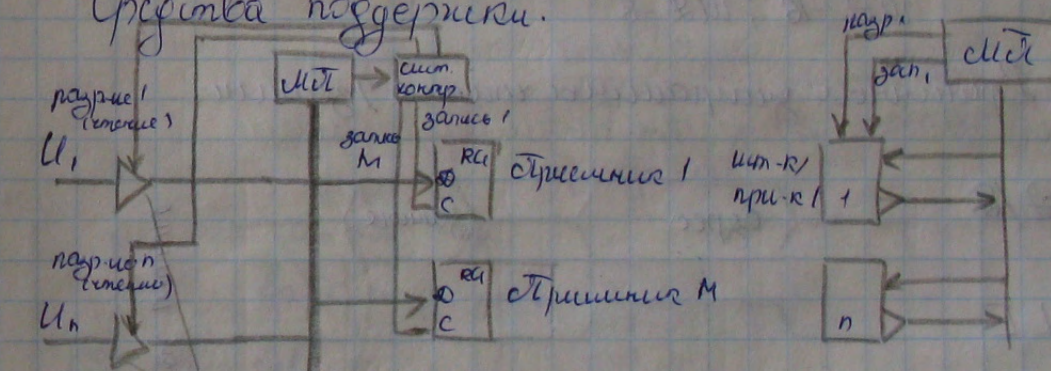
WR
(запись)

MLD

данные
из листа

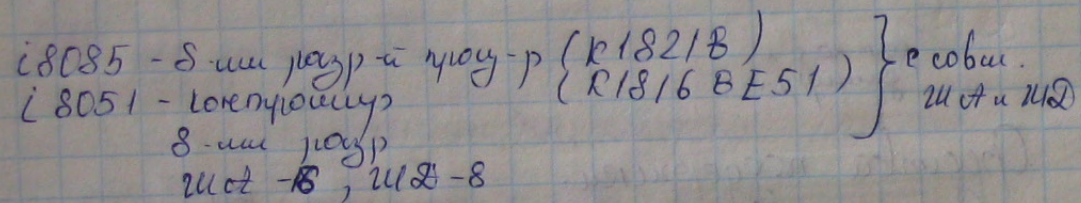
дест. кар. х.
дест. кар. х.

Средства поддержки.

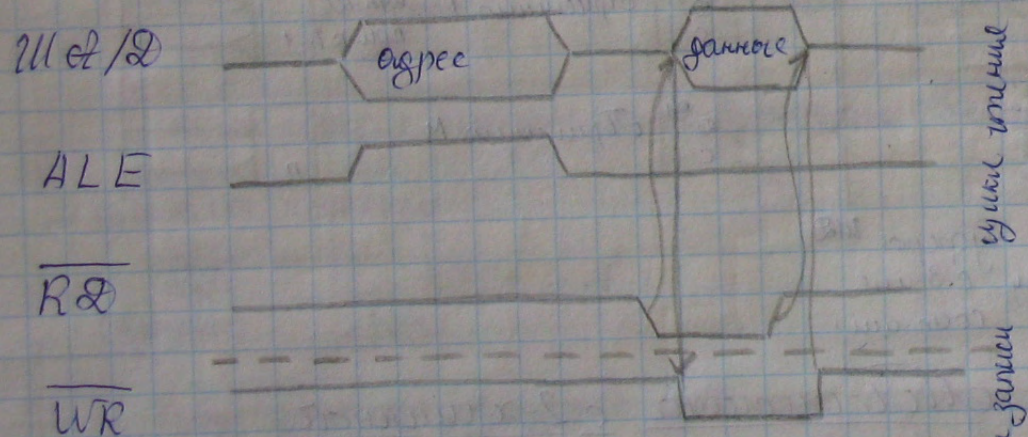


ит-ке бюджетте
инд-ми сз-ме
сост-ми,

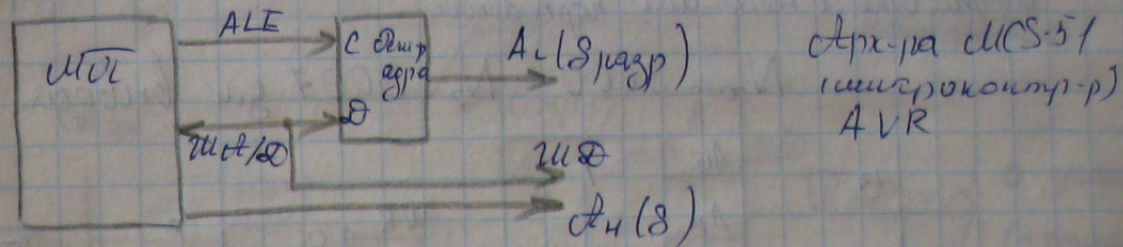
Мст-система с 2-х минной
организацией



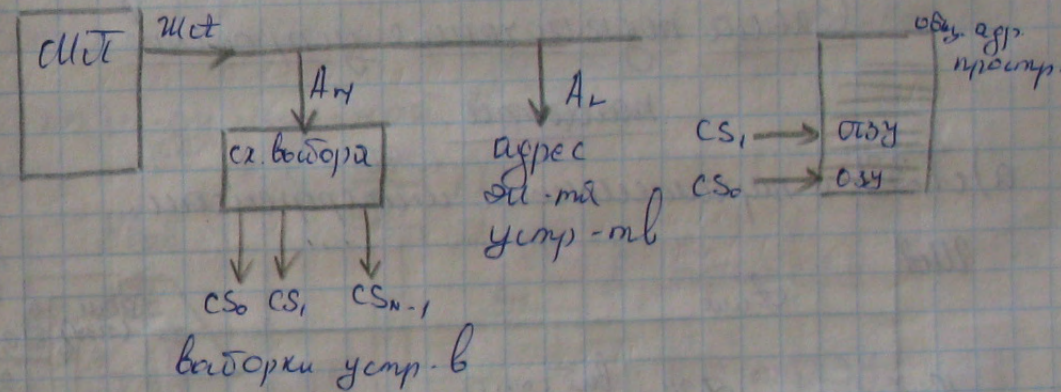
Временное диаграммное течение жизни.



Преобразование 2-х битной
адресации в 3-х битную.



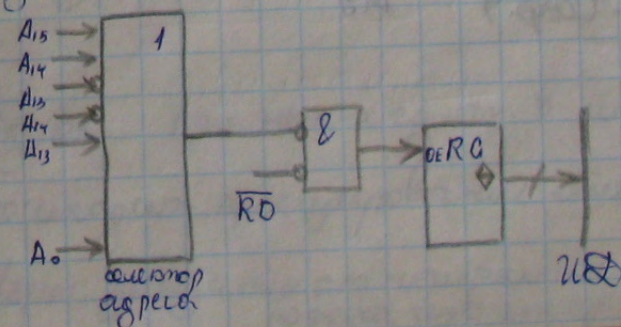
Несомненно, и с помощью ИИТ-систем можно
создать воборуд устройств подконтрольных к оедр-
шине



CS - chip select

Примеры подглагольных позы и уступов:

а) одиночный порт ввода с номером $N = 3000$ h



а) работа с несколькими портами

$$N_0 = 3000 \div N_7 = 300 \text{ \# где вывoda}$$

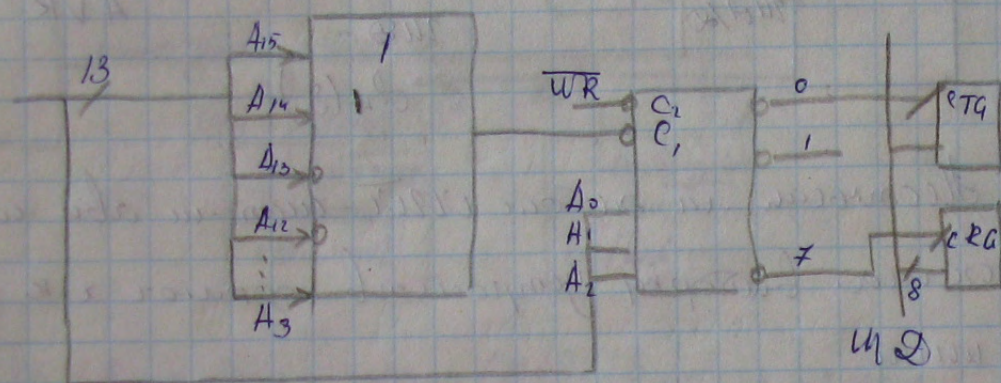
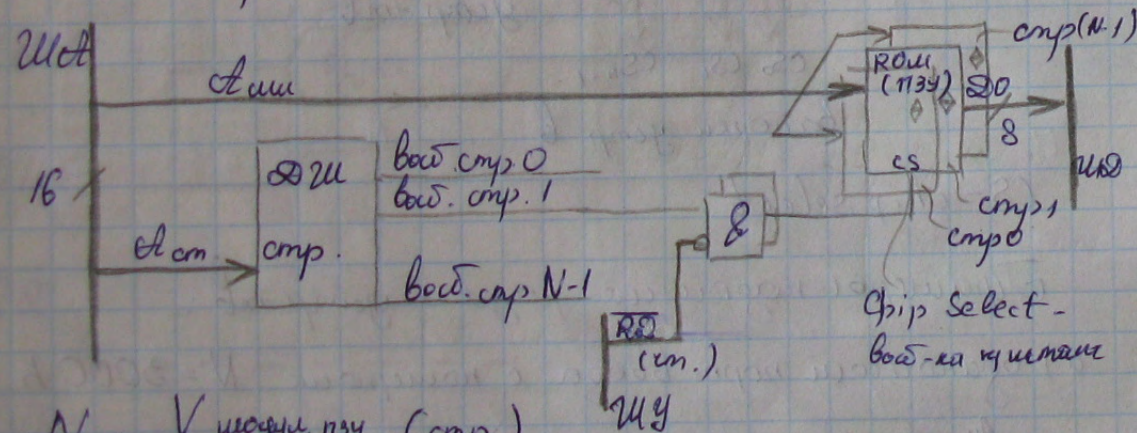


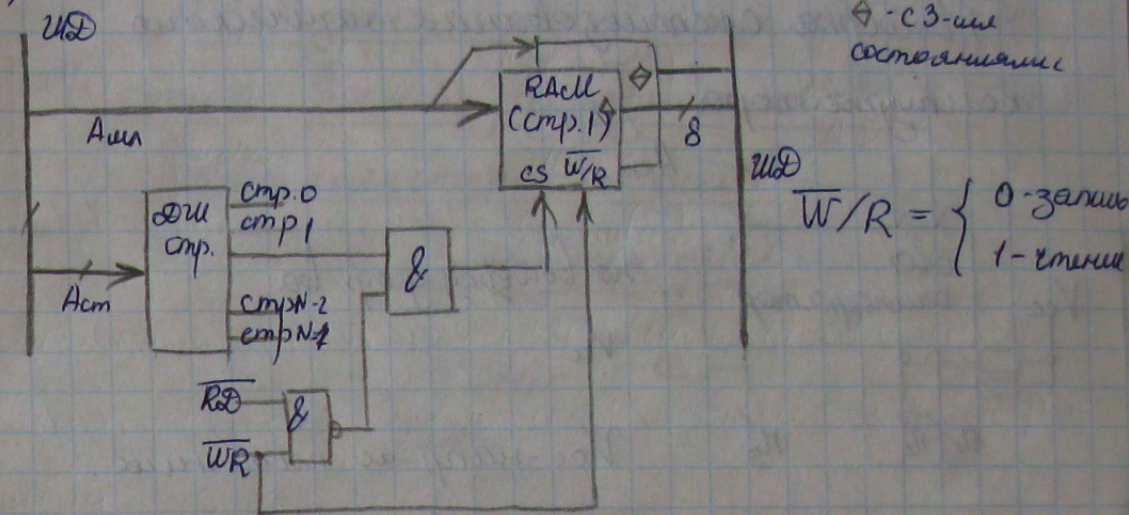
Схема подключения устройств памяти.

а) ПЗУ с параллельным интерфейсом.



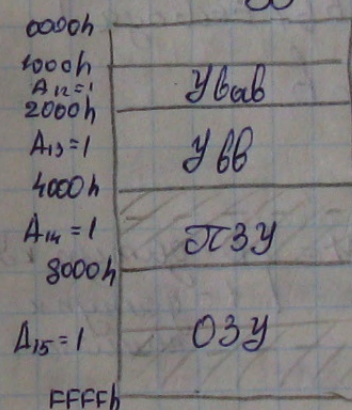
$$N = \frac{V_{\text{памяти ПЗУ}}}{V_{\text{чипа}}} (\text{чип.})$$

б) ОЗУ

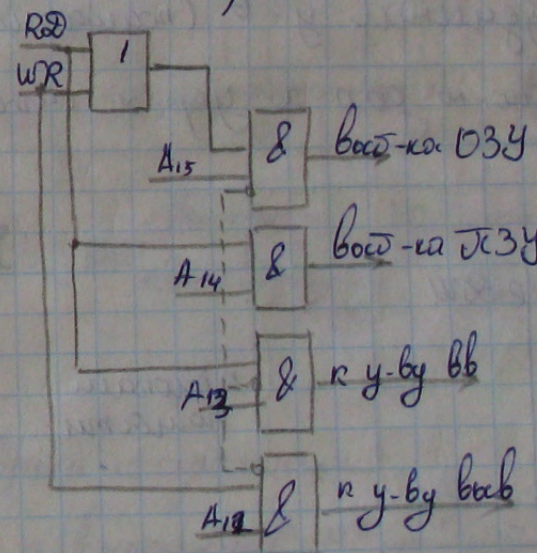


Дополнительные приемы доступа к устройству:

а) по ш-ду шин-ной выборки

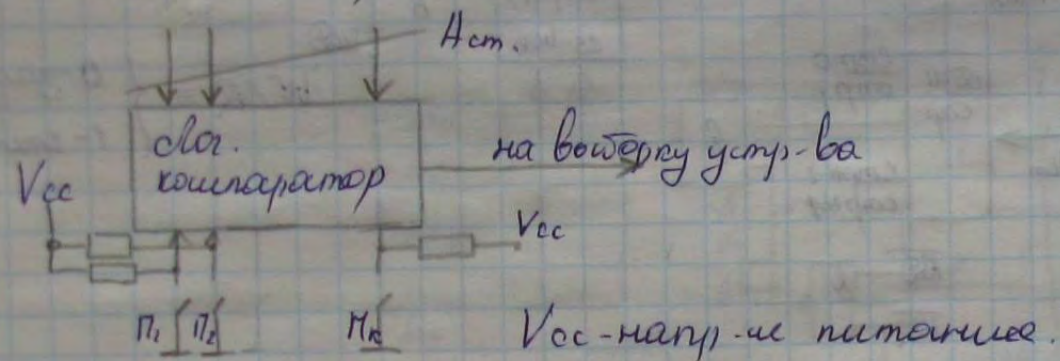


Спроектировать



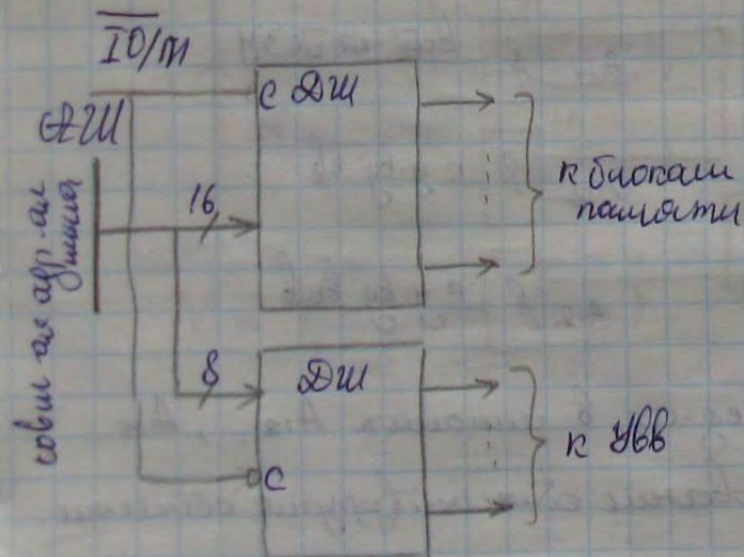
Рез-ок: при нек-х ед-ах в сигналах A12, ..., A15 происходит процессуирование данных в других областях. Сперва защита от некорректных действий: по факту ш-ва ед-а (непосредственно по)

б) выборка с использования почтового
торца



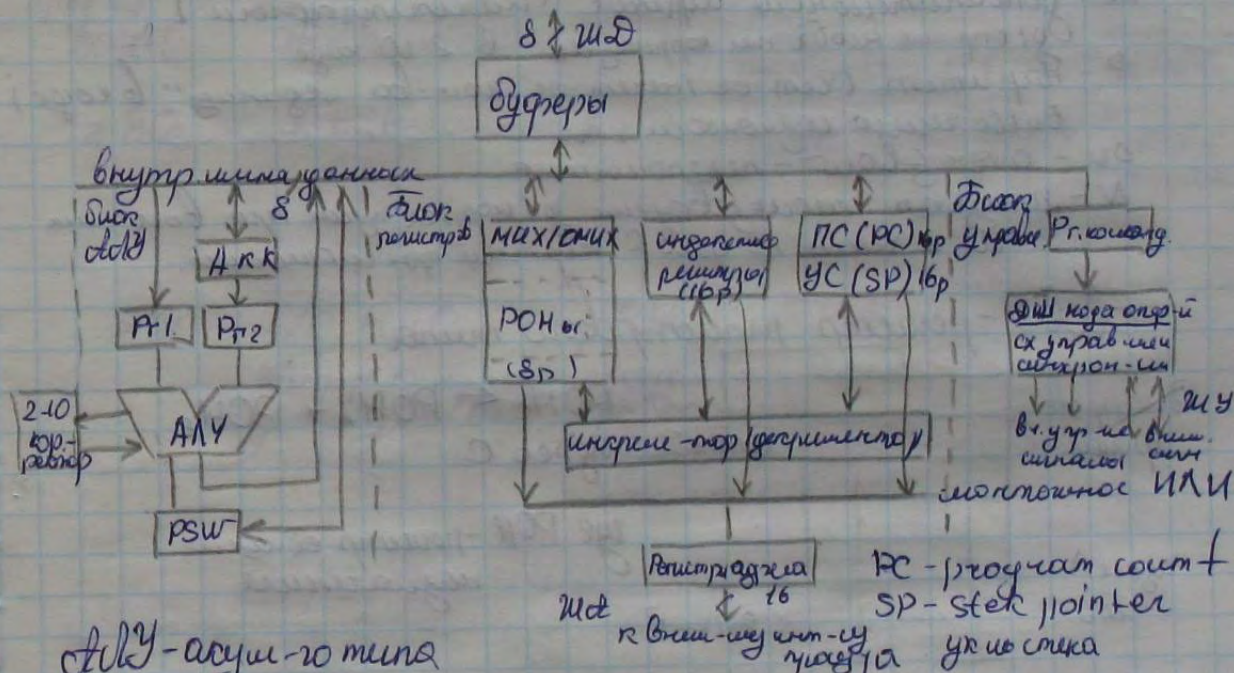
задаем с помощью $P_1, P_2, \dots, P_k$ номер
устройства,

в) разделим у-в (памяти, вв/вывода)
с пом-ю рег-а упр-я чтения в



$\overline{IO/M} = \begin{cases} 0 - \text{доступ к УВВ} \\ 1 - \text{доступ к} \\ \text{памяти} \end{cases}$

Обобщенная структурная
схема восьми разрядного МЭ.
Алгоритмические функции
восьми разрядная МЭ.



АЛУ - арифметико-логический

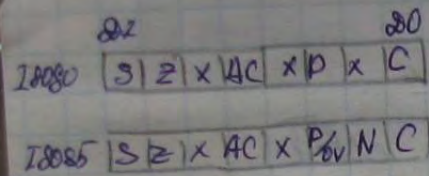
$(\text{АЛУ}) * (\text{СЧ}) \rightarrow \text{АЛУ}$

P_1, P_2 - регистры временного хранения

АЛУ кодов-го типа у/выполнения "+", "-", НЕ, И, ИЛИ, ⊕.

2-10 корр-р иел-се при обр-ке 2-10-х операндов, д/того чтобы получить 2-10 резу-тат.

PSW - регистр состояния программы



8-ми битный для микр. Intel 8080 (8085)
 * - произв.-ые

S - знак рез-та операций (ст. разр. режима (ст. к))
 Z - признак 0-результата
 C - перенос (заем)
 AC - дополнительный адрес (инкрементный)
 P - признак переноса в 2-10 коде
 N - признак переноса в 2-10 коде
 OV - overflow - переполнение
 N - признак переноса в 2-10 коде

Accu - регистр регистрового типа



$POH_C \leftarrow POH_A * POH_B$,
 адрес C
 еще POH - регистр общего назначения

Состав микр.
 микр. 8080

Data Bus In
 DB IN, WR, READY (от внешнего у-ва),
 WAIT (от внешнего у-ва)

микр. 8085

RD, WR, WAIT (от внешнего у-ва),
 HALT (от внешнего у-ва)
 сигнал: входы и выходы микр.

8080: HOLD HLDA INT INTE
 8085: BUSRQ BUSAC INT NMI
 запрос в канал г/у прямого доступа к памяти (HOLD)
 подтверждение приема сигнала
 запрос прирав-ия
 not mask interrupt
 подтверждение прирав-ия

прог-ие: 8080: RESET E_nim Общ. шин
 8085: RESET E_nim Общ. шин
 сброс, вын-е при с ошибкой адр-а
 + MI MREQ
 сигнал, отпр-е в выход командной памяти

8085: I/O REQ, RFSH
 запрос уст-бу вб/выб
 сигнал работы с динам. памятью

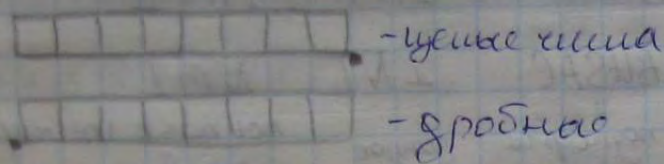
Программистские модели

Это форматы данных, способы агрегации г-х, ресурсы.

1) Форматы данных

Беззнаковые целые числа от 0 до 255
 знаковые целые в диапазоне -128 ÷ +127
 отриц-ые числа приняты в 2-х коде

$[-64]_2 = 11000000$



Соборные и др. не предполагают работы с обратными кодами.

2) Способы адр.-ии:

1. Беспрямое адр.-ие: операнды нах. в регистре.

2. Непосредственное адр.-ие: оп.-код нах.-ся в самой команде, указ. на опер.-да.

3. Прямое адр.-ие: указ. адр.-с запис. в к-й нах. операнд.

4. Косвенное регистровое адр.-ие: в регистре указ. адр.-с опер.-да, обычно пара 8-ми разряд. рег.-ов. Частный случай, стек. адр.-ие: адр.-с опер.-да в стек. регистре.

3) Ресурсы

i8085 (K1821 BM85)

A - аккум.-ор

B, C, D, E, H, L - 8-р. рег.-ы адр.-а нах.-ие

(B-C, D-E, H-L) - 16-р. (пары рег.-ов)

PC - Program Counter } 16-р.

SP - Stack Pointer

PSW - Program State Word 8-р.

IM - ^{содерж. см. мануал} ~~регистр~~ маска прерыв.-ий, Interrupt mask
= 80:

+ A'

B', C', D', E', H', L'

8-ком. ие команд. прерыв.-ий
пр.-ий и доступа к рег.-ам

+ $\frac{IX}{IY}$ } указ.-ые рег.-ы (16-р.)

+ IV - рег.-тр прерыв.-ий

i8085 и z80:

64 Кбайт эл. пам.

256 портов $\overline{b\overline{b}}$

256 портов $\overline{b\overline{a\overline{b}}}$

Набор команд

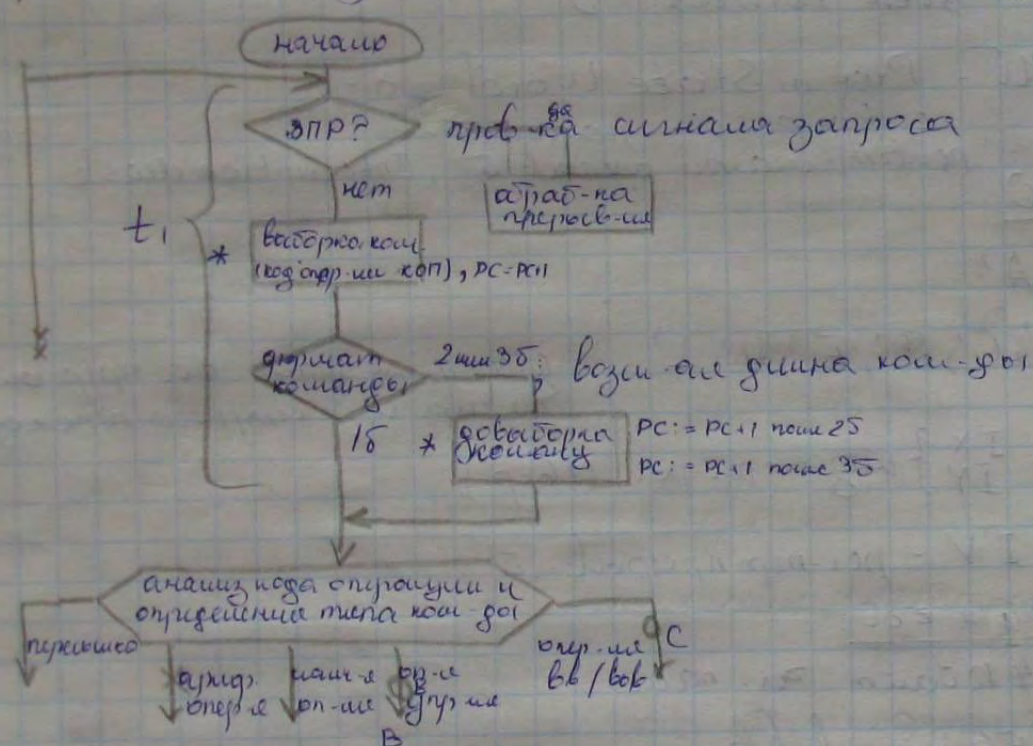
i8085 - 80 команд

z80 - более 80 команд

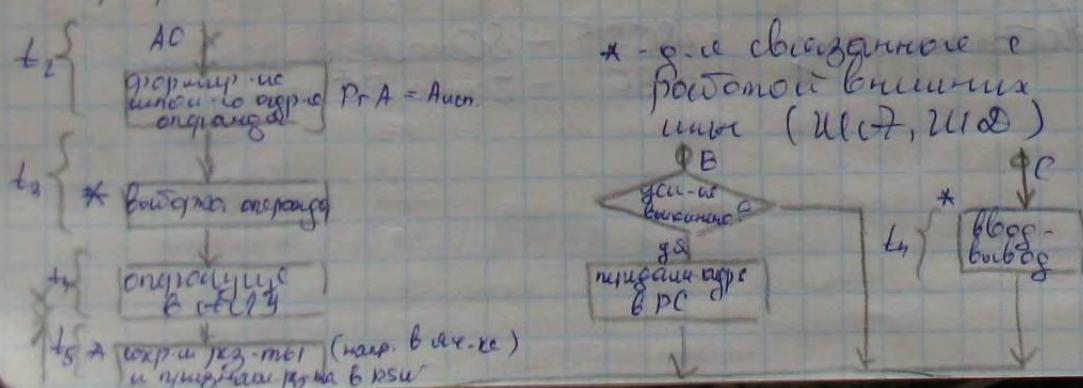
Функционирование первичного управляющего автомата.

Циклы выполнения команд.

УЗ главной функ-ии имеет перем-ие команд.



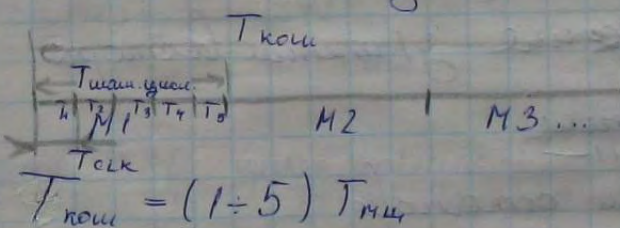
Ариф-исе операции АО



Основные фазы выполнения команд.

- t₁ выб-ка команды
- t₂ дешифр-ия
- t₃ выборка опер-га
- t₄ выполн-ие опер-ии
- t₅ сохран-ие рез-та

Временные хар-ки выполнения
команд.



где ИИ7: 8085, 780

Tком = (1 + 2) Tмц

CISC

где МК: MCS-51

Tclk - период одного машинного такта

$$T_{clk} = \frac{1}{f_{clk}}$$

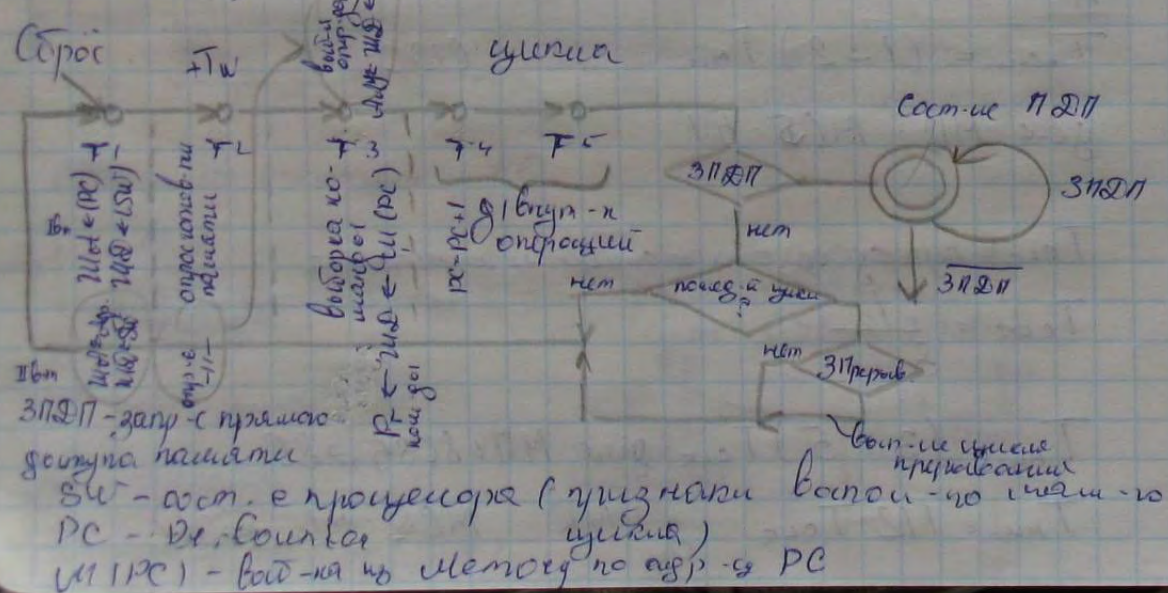
Tмц = (3 + 5) Tclk где MMi8085, 280

Tмц = 12 Tclk где MK MCS-51, i8051

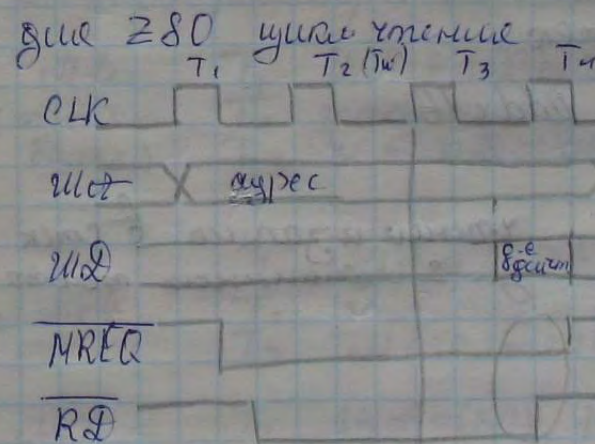
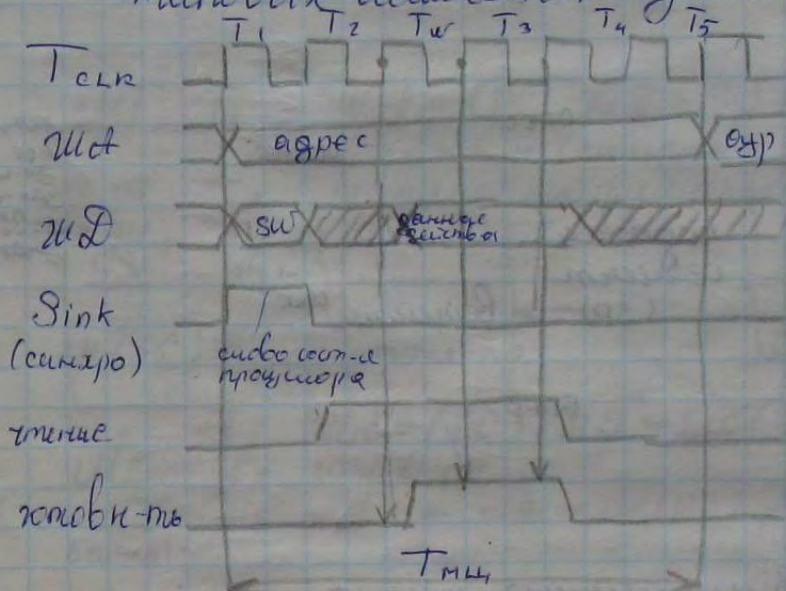
Жизнь машинных циклов.

1. Выборка команд
2. Чтение ячейки памяти
3. Запись в ячейку памяти
4. Чтение из стека
5. Запись в стек
6. Ввод с порта ввода
7. Вывод в порт вывода
8. Цикл прерывание / чтение вектора
9. Цикл прерывание
10. ^{II}Служба при останове

Диагностика восприятия и мышления

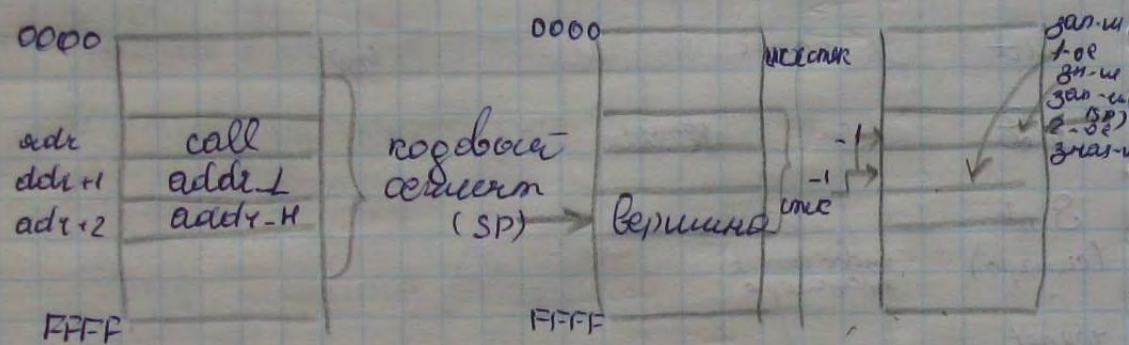


Временное диагональное
типовых машинных удилищ



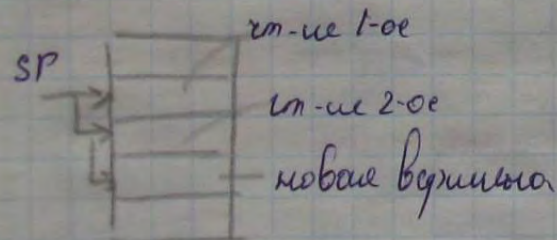
подорож
до моря - ми

Пример выполнения команды CALL addr16.



35.

1. Выбрать за 3 машинных цикла
2. Сохранить (PC) в стек
3. Заменить (PC) на addr16



чтение и запись в стек
за 2 машинных цикла

Микропрограммы

5мц, 16сц (маш. такт.)

М1	T ₁	Ист ← (PC), ИД ← SW (выборка команд)
	T ₂	готовность (памяти к передаче)
выб-ка	T ₃	принять по регистру ИД ← Ист (PC)
ком-да	T ₄	PC ← PC + 1

И(PC) - из памяти по PC

М2	T ₁	Ист ← (PC), ИД ← SW (чт. памяти)
	T ₂	готовность
чт-ие 2-го байта	T ₃	Z ← ИД ← И(PC), PC ← PC + 1, SP ← SP + 1

Операции, инструкции W_{CH} Z_{CH} SP - SP = 16p.

М3	T ₁	Ист ← (PC), ИД ← SW (чт. памяти)
	T ₂	готовность
чт-ие 3-го байта	T ₃	W ← ИД ← И(PC), PC ← PC + 1, SP ← SP + 1

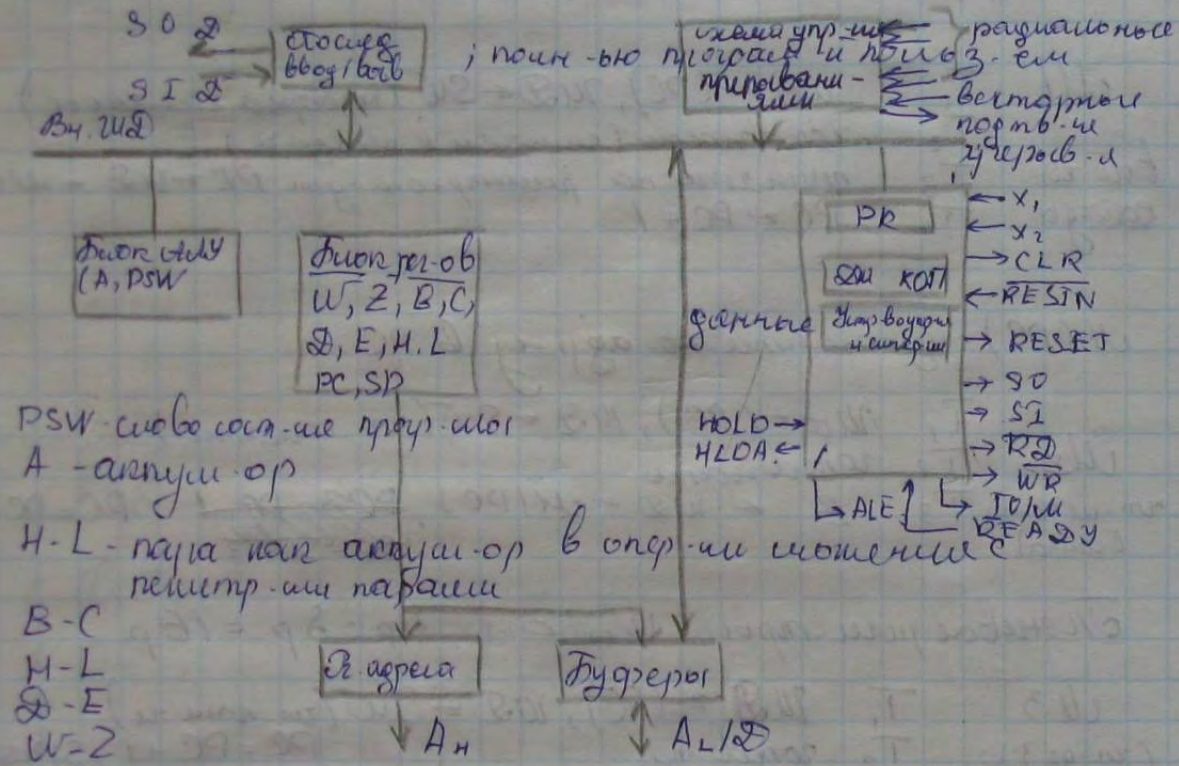
М4	T ₁	Ист ← (SP), ИД ← SW (запись в стек)
	T ₂	готовность к обмену
запись в стек	T ₃	И(SP) ← ИД ← (PC _н), SP ← SP - 1

М5	T ₁	Ист ← (SP), ИД ← SW (запись в стек)
	T ₂	готовность
	T ₃	И(SP) ← ИД ← (PC _н), PC ← (W, Z), 3 прдпр

3 пр - запрос, пр - разрыв прерывания (проброс)

Архитектура ИМТ 18085

и системное устройство.



RESET - вых. сброс

RESET - вых. сброс для ИМТ-сист.

RD, SI - сигналы сброса процессора

SI	RD	сост.
0	0	HALT (останов)
0	1	WRITE (запись)
1	0	READ (чтение)
1	1	FETCH (выб. код.)

RD, WR - сигналы синхронизации чт/записи на шину

IO/M = { 0 - адрес памяти
1 - адрес шина

READY - вход готовности

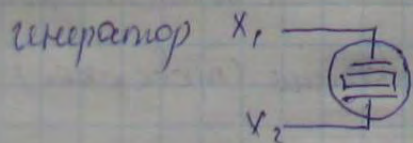
ALE - строб сопровождения адреса

HOLD - запрос внешнего ресурса
HLDA - подтверждение ресурса.

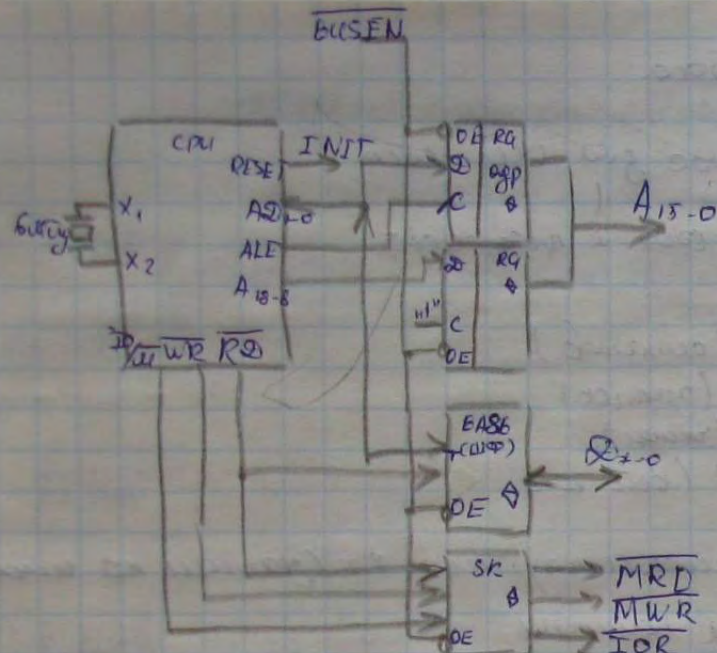
Каждый вход управляющих сигналов имеет одну точку подключения

PC - регистр команд; DM COP - для опроса

x₁, x₂ - внешние входы синхронизации, пороговые



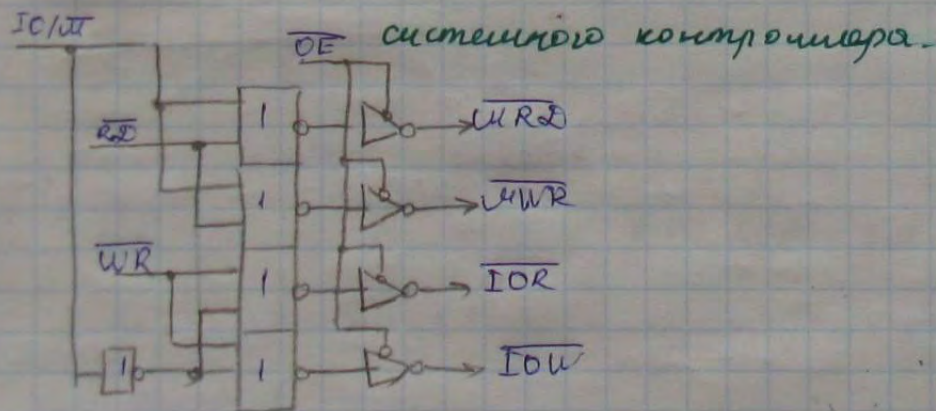
CLR - вых. синхронизации



ШФ-шиновый драйвер - ив
СК - системный контроллер-р.
Т - тактовый генератор

OE - разряд выходов

Функционально-логическая схема



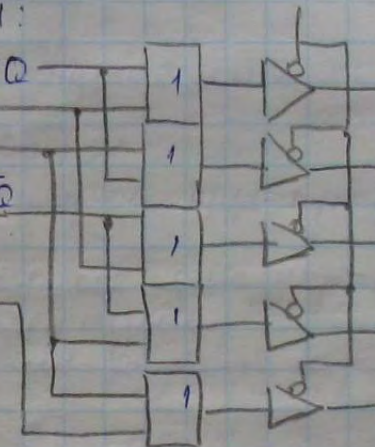
Системный контроллер 280

Вх сигналы:
Запросы:

наименование
м-а
зап-б
ум-ба
сб/бав
мгновенный
выбор
уникальный
код

UIREQ
RD
WR
IOREQ

MI



UIELUR инициализация

MEUW запись в память

IOR

IOW

INTA (интерпретация и прерывание)
пункт системного вектора

"Ядро" МК i8051 (80C51), i8052

Структурная схема "ядра"

Функции:

RAM

RAM - 128 байт, random access memory

RPM - random program memory

RAR - регистр адреса

SP - указатель стека

EEPROM - электрически стираемая и программируемая память

SFR - регистры специального назначения

IR - регистры инструкций

CU - управление

1) P0, ..., P3 - порты ввода/вывода

2) при программировании

3) через P3 вывести управляющие сигналы

Упр-ие шин-ой на шит-се.

RST - сброс

V_{PD} - цеп-ие при прог-мме

$\overline{EA}$ - исп-ие для упр-ия внеш-й памяти
программы,
если внут-й внутр-й памяти не хватает.

$\overline{ALE}$ - сброс адреса

$\overline{PSEN}$ - разрешение работы внеш-й памяти
чтением.

регистр P3:

P3.0 - вход пошл-го канала; RxD

P3.1 - TxD - выход пош. канала;

P3.2 - вход внутр. прерыв-ия $\overline{INT0}$

P3.3 - вход - " - $\overline{INT1}$

P3.4 - вход таймера T0

P3.5 - вход таймера T1

P3.6 - запись $\overline{WR}$

P3.7 - чтение $\overline{RD}$

}; важно адекват порт

Самост-но: 1) AT89C51 Datasheet

(8051, 80C51, 816BE51)

програм-ие flash-памяти программ.

2) Изучить книгу 2, библио-е соотв-ие со схемой 1.

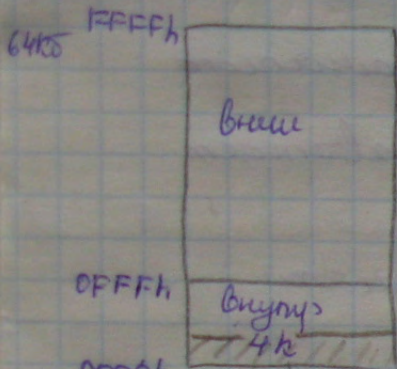
Словесное описание памяти.

Память программы (CSEG)

Вар. мож. исп. - ие:

1) $\overline{EA} = 1$ - внут.-ая память

код прог.-и, табл. конст.-т в памяти программы.



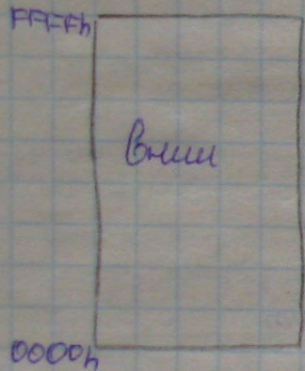
x x 51 - сегмент (архитектура)

табл. прерыв.-й (см. выше).

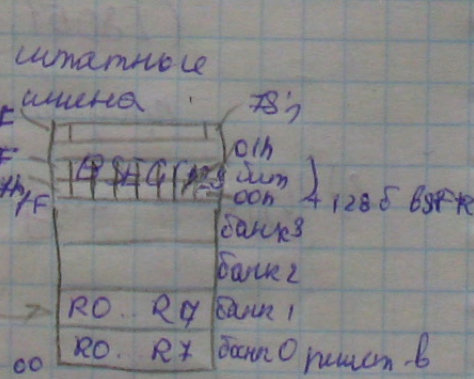
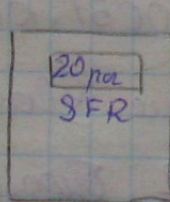
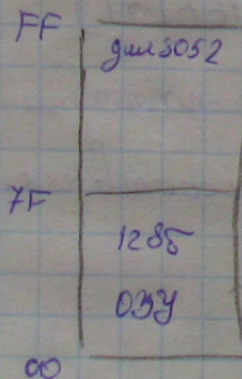
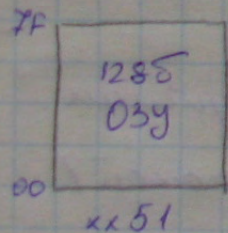
2) $\overline{EA} = 0$

не исп.-е внут.-ая память.

x x 31 - память прог.-и за пределами МК

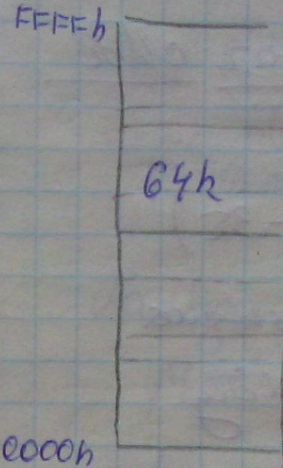


Память данных (DSEG)



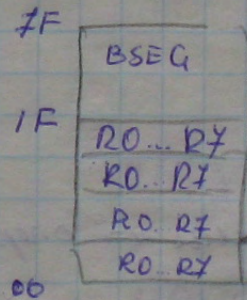
внутр

к р.мем.-и можно отп.-а
код к адрессам, если нет
ит в кон.-се памяти



можно разбить (использ.-то)
на части.

Объем сегмента

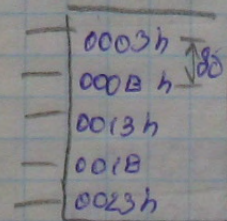


↑ расчет

$(SP)_{исх} = 07h$

(SP) при операциях PUSH

табл. прерыв.-й (см. выше)



Обработка данных в 8-разряд МК.

Система rang MK51:

111 код - 9 (49-15, 45-25, 17-38)

Кол-во вост-ие за 1 или 2 мкм. диметр
по 1 мкс (12 мактов). 12 мкг

Уин, ден. за 4 мом. уикли (4 джк)

- 1) Пересыпши
- 2) мол. операцион
- 3) битовое опер.
- 4) арифм
- 5) передача упр-ия

1) Терминал е уред - ик и нумен ка опер-ра

MOV dst, src

exam: $dst = A$, no SRC = $\{R_n, ad, n = 0, \neq\}$

@ Ri # data 3
 ndab-no. 8-um
 app-ae pazp-i
 aruma onpang

enum dist = R_n, no src = {A, ad, #data}

own dst = ad, no src = {A, Rn, aol, @ Ki,
#data}

cam dst = @ Ri, music - { A, ad, # data }

если $olst = DPTR$, то $src = \#data16$
16-ти разг-и уяз-во
гармаз

Ваша просьба ^{из} во вним. памяти:

MOVX A, { @DPTR, @Ri }

коб. 16р. 8р. (допущ-е
уп-е 2р. 2р. на выпущ)

намер, стр.
указ-ие
депутатом РД

MOVX {@DPTR, @Ri} ; A - адресная в
базе памяти

MOVС
из код. сегм. по
предела на

A, @A + DPTR
данный сегм.
L опер. чин
по Page

000

main
комм.

FFF

MOV A, @A+PC
относительная адресация

PUSH ^{src} ad записано в стек с адресом $+(sp) \leftarrow (ad)$

POP ad $\text{sumamo iz smera c opreleniu}$
 $(ad) \leftarrow (sp) + 1$

Тыпиер ^{101st} PUSH Acc - an - p, ^{body ← (sp)} sp ⁺ →

PUSH DPL } связь по меньшей мере 2-м
PUSH DPH } инкремент

Ошибки:

XCH A, {Rn, ad, @Ri}

XCHD A, @Ri; A₃₋₀ ↔ (@Ri)₃₋₀ (меморизация)

SWAP A; A₄₋₇ ↔ A₃₋₀

Пример. Счит. табл. - и константы с исп. инкремента.

sin 0 = 0
sin 1 = 0.017
sin 2 = 0.035

sin 89 = 0.99
sin 90 = 1

$\frac{1}{256} = 0.0004$ - цена дискрета.
0.00000100B, 0.0001001B.
в байте макс 255

MOV A, #1
call sinx

(mov DPTR, #TABL)
не нужно

mov A, #2
call sinx

sinx: inc A
movc A, @A+PC
ret

TABL sin: DB 0, 0.00000100B, 0.0001001B, 0.0010010B

2) Операции с данными.

ANL
лог. и-е

ORL
лог. и-е

XRL
лог. и-е

сдвиги и повороты

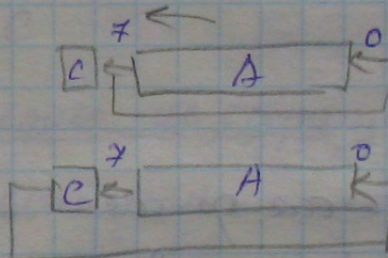
A, {Rn, ad, @Ri, #data}

— " — ad, {A, #data}

CPL A; A ← A̅
инверсия

RL A
вправо по кругу

RLC A
вправо по кругу с
ар по C.



CLR C - сброс флага

3) Операции с данными:

C-битовый арифмет.

3.1. Однобитовые операции:

обнуление бит-а ар-ра CLR C

— " — бит-а сема CLR bit; бит-а сема

SETB C - установка

SETB bit

CPL C - инверсия

CPL bit

3.2. ANL C, bit - лог-кое сложение

ANL C, /bit

3.3. ORL C, bit - лог-кое умножение

ORL C, /bit

3.4. Термины, битов:

MOV C, bit
MOV bit, C

Пример. Дана лог-ая ф-ия

$$y = x \bar{v} \vee w (x \vee v)$$

Пусть x, v, w - поступают через вх. порт.

PI по линиям 0, 1, 2 (линии портов)

y - Выводим в порт PI. 3

x bit PI. 0

v bit PI. 1

w bit PI. 2

y bit PI. 3

MOV C, X

ANL C, /v

MOV FO, C

FO - бит-ое сопр-ение по флан
FO в PSW

MOV C, X

ORL C, v

ANL C, w

ORL C, FO

MOV y, C

4) Арифметич-ие операции

4.1. ADD - сложение

ADDC - " - с переносом A, {Rn, ad, mi-mp}

SUBB - вычитание с заимом

@ Ri, # data }

коэф. по
адресу
данных

SUBB A, Rn ; $A \leftarrow (A) - (Rn) - C$
присоединение C

если без заимов, то CLR C перед SUBB

4.2. Умножение: MUL AB;

$BA \leftarrow A \times B$; В по знаку C

43. Деление:
 $DIV AB$; $AB \leftarrow A/B$
 от. или произв.
 частн. остаток

$OV = 1$, если $B = 0$

44. Инкремент (декремент):

$inc \{A, R_n, ad, @R_i, DPTR\}$
 увелич. пер-по.

$dec \{A, R_n, ad, @R_i\}$

45. Дв-гес. - инкремент:

$DA A$; где op - дв-гес-х чисел.

Обработка многобайтовых операций.

1. Сложение

$A+B$

, AL - млад. байт
 ($C=0$)

а) $AL + BL$

(ADD) - м.р. в AL

б) $AH + BH + C$

($ADDC$)

2. Вычитание

$A = 130_{16}$

$0D FE$

$B = 0503_{16}$

1) $AL - BL$

$0000 0001$

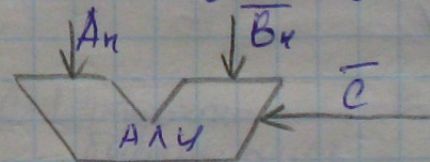
$\frac{1111 1101}{1111 1110}$ - $90H$ код $[-03]_{16}$ г.н.

$C=1$ - заим., отсутствие переноса

2) $AH - BH$

$0001 0011$

$\frac{1111 1010}{0000 1101} = 0D_{16}$ - $0D_{16}$ код $C=1$



Программная процедура сложения и вычитания

$addr_A \quad EQU \quad 10h$

$addr_B \quad EQU \quad 20h$

$MOV R2, \#2$; op - код 2-х байтов

рег. $R1$

рег. $R0$

AL

AH

$20h$

BL

BH

$MOV R0, \#addr_A$

$MOV R1, \#addr_B$

$CLR C$

loop: MOV A, @r0
 ADDC A, @r1 (SUBB)
 MOV @r0, A ← (DAA) 8, 2-10.
 Inc r0
 Inc r1
 DJNZ R2, loop (если не = 0 R2, то loop).

3. Сложение 2-10 чисел.

$$A = 29'85$$

$$B = 49'36$$

$$79'21$$

1) $A_L + B_L$

$$\begin{array}{r} 1000 \ 0101 \\ 0011 \ 0110 \\ \hline 1011 \ 1011 \\ \hline 1100 \ 0110 \\ 0110 \\ \hline \end{array}$$

- сложение 6

$$C \leftarrow 0010$$

$$\begin{array}{r} 2) \ 0010 \ 1001 \\ 0100 \ 1001 \\ \hline 0111 \ 0011 \\ 0110 \\ \hline 0111 \ 1001 \end{array}$$

$$A_H + B_H$$

; пример, в 9х-ке 8/2-10.
 без DAA числа приноса
 сдвиги - 10, DAA

4. Вычитание 2-10 чисел.

а) $A = 04'93$ $A - B ?$
 $B = 02'98$

$$\begin{array}{r} 0493 \\ - 0298 \\ \hline \end{array}$$

1) $A_L - B_L$

$$\begin{array}{r} 99 \\ + 9A \\ + 08 \\ \hline 02 \\ + 93 \\ \hline 95 \end{array}$$

- заплата

2) $A_H - B_H$

$$\begin{array}{r} 99 \\ + 99 \\ + FE \\ \hline 97 \\ + 04 \\ \hline 9B \\ + 06 \\ \hline A \\ \hline 6 \end{array}$$

зачисли до 10⁴

C = 1 - разность
 больше нуля

б) $B - A = -0195$

1) $B_L - A_L$

$$\begin{array}{r} 99 \\ + 9A \\ + 60 \\ \hline 04 \\ + 98 \\ \hline 9F \\ \hline 6 \\ + A \leftarrow 5 \\ \hline 6 \end{array}$$

разн-то
 отриц-ая

$$C \leftarrow 05$$

2) $B_H - A_H$

$$\begin{array}{r} 1 - C \\ 99 \\ 9A \\ FE \\ + 96 \\ \hline 02 \\ + 98 \\ \hline \end{array}$$

addr_A -11-

addr_B -11-

MOV R2, #2

MOV r0, #addr_A

MOV r1, #addr_B

SETB C

loop MOV A, #99h

ADDC A, #0

SUBB A, @r1 ; вычитание B

ADD A, @r0 ; + умножение на A

DA A ; кор-ли

MOV @r0, A ; копи-ли вычитание 1-го опер-го

INC r0

INC r1

DJNZ r2, loop

5. Умножение

A_H A_L

B_H B_L

A_L × B_L

A_H × B_L

A_L × B_H

A_H × B_H

6. Деление

Преобразование 2-го числа в акк. в 2-10.

MOV B, #100

DIV AB;

MOV R0, A; кол-во 100-ен

MOV A, B; остаток

MOV B, #10

DIV AB

SWAP A

; поменяем десятичные и шестнадцатеричные

OR A, B; в акк. десятичные и шестнадцатеричные

имеет 25: ход оп. и сечения; $PC \leftarrow PC + 2$
 $PC \leftarrow PC + \text{rel} \quad (-128 \div 127)$

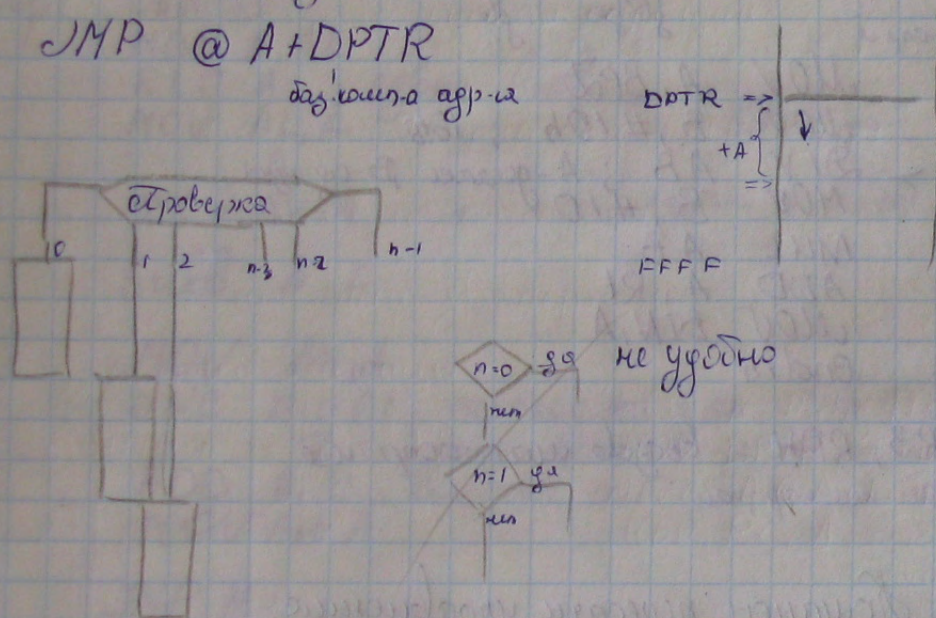
JMP <label> - комп-ор сам вычисляет
 CALL <label> - комп-ор вызывает
 вспомог-ие

Jmp - ; относит-ый переход от-но PC на
 более 2648.

Классический переход относительный:

JMP @ A+DPTR

диз-компа адр-а



Пример: ветвление на n направлений.

Пусть селектор направлений имеет имя MEMSEL

MOV R3,

JMP - n ; процедура перехода

① MOV A, MEMSEL

② MOV DPTR, #TBL

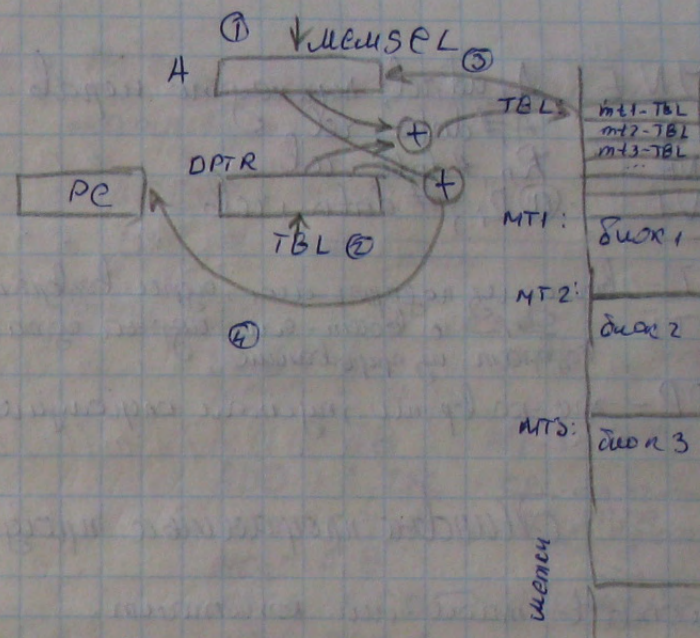
③ MOVC A, @A+DPTR

④ JMP @A+DPTR

TBL: DB m+1-TBL, m+2-TBL,

DB m+3-TBL, ...

m+1: ; блок 1
 RET
 m+2: ; блок 2
 RET
 m+3: ; блок 3
 RET
 RET



JZ rel

; переход по нулю

JNZ rel

; переход если не 0; сравнение с нулем в
 регистре-операнде

JC rel

; переход по переносу

JNC

; переход если нет переноса

JB bit, rel

; 3-х байт. операнд

1-ый байт - адрес
 2-й байт - бит
 3-й байт - селектор

код инициализации, адрес бита, инициализация
 по 1-му значению бита из бит-10 или-та
 на инициализацию

JNB

; переход по 0-му значению бита в BSEG

JBC bit, rel

; по 1-му биту, инициализация бита вы-
 нудит. Инициализация как селектор 10 или
 по 1-му значению бита в BSEG

DJNZ Rn, rel

; уменьшит и проверит на нуль-е
 значение (с регистром), по регистру

DJNZ aol, rel

; уменьшит и проверит на нуль-е
 значение (с адресом), по регистру

CJNE A, rel, rel - переход по нерав-ву $\geq$ ср-ве.
 CJNE A, #data, rel
 CJNE Rn, #data, rel
 CJNE @Ri, #data, rel

$C \leftarrow 1$, если $A < \#data$;
 переход по $C=1$

RET - выход из подпрограммы, далее возврат из стека;
 RETI - с возм-ем прерывания; далее прерывание;
 NOP - загрузка в регистр, пустая операция;

Типовые программные процедуры

1) Работа с таблицей констант.

одномерные или двумерные

а) Одномерная таблица:

0:	1111100
1:	
2:	
...	
9:	

"0" $\frac{1}{2} = \frac{1}{2}$
 "1" $\frac{1}{4} = \frac{1}{4}$
 "2" $\frac{1}{8} = \frac{1}{8}$
 "9"

MOV A, #digital
 MOV DPTR, #TBL
 MOVC A, @A+DPTR
 MOV Px, A; отобразить цифру

б) Двумерная таблица (матрица)

a_{11}	a_{12}	a_{13}	...	a_{1n}
a_{21}	a_{22}	a_{23}	...	a_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_{m1}	a_{m2}	a_{m3}	...	a_{mn}

Использовать эл-ты a_{ij}

или загрузка по i и j .

$(R_0) = i$; $(R_1) = j$

адрес R_0

$(DPTR) = START$ (начало адрес массива элементов матрицы)

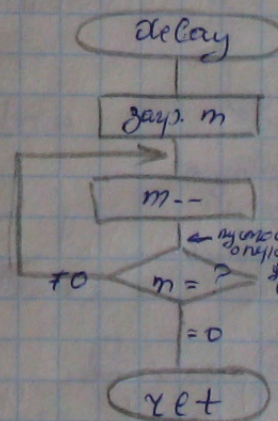
$$ADDR = \underbrace{START + n(i-1)}_{DPTR} + \underbrace{(j-1)}_A$$

start	a_{11}
start+1	a_{12}
	a_{13}
start+n-1	a_{n1}
start+n	a_{21}
start+2n-1	a_{2n}
start+mn-1	a_{mn}

MATRIX: MOV A, R0; $i \rightarrow A$
 DEC A
 MOV B, #n
 MUL AB; $m(i-1) \rightarrow BA$
 ADD A, DPL; DPL - адрес
 MOV DPL, A; $m \cdot \text{начало матрицы}$
 MOV A, B
 ADDC A, DPH; $(A) + (DPH) + C$
 MOV DPH, A
 MOV A, R1; j
 DEC A
 MOVC A, @A+DPTR; (1)
 ; θA как эл-т a_{ij}

2) Программная реализация временных задержек

а)



$$t_{\text{заг.1}} = \frac{12}{f_{\text{CLK}}} \cdot (t_{\text{заг.пр}} + m \cdot t_y + t_{\text{ret}})$$

$t_{\text{заг.пр}}$ - время задержки при загрузке
 t_y - время цикла

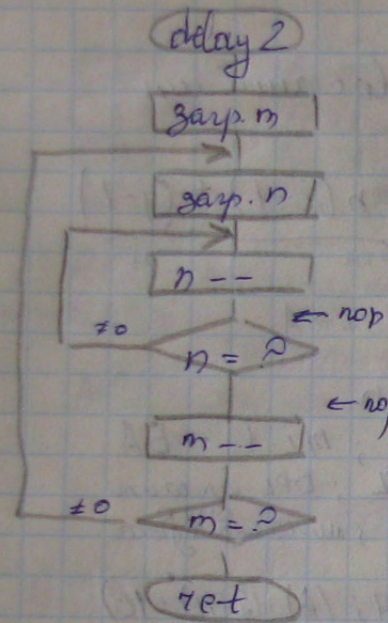
delay: MOV R2, #m
 loop: DJNZ R2, loop
 RET

1 м.с.
 2 м.с.
 2 м.с.

$$t_{\text{заг.1}} = \frac{12}{f_{\text{CLK}}} (3 + 2 \cdot m)$$

м.с. = мкс

при $m = 235$ $t_{\text{заг.1}} = 513 \mu\text{s}$



mov R2, #m
 mov R3, #n
 loop: DJNZ R3, loop
 DJNZ R2, m0
 ret

$$t_{\text{заг. 2}} = 1 + m(1 + 2n + 2) + 2 = 3 + m(3 + 2n)$$

t^* - разность бр. тик переклада и پردازи-и-е
 тубограми-и-е.

$$t_{\text{заг. 2}} \approx 0,1e$$

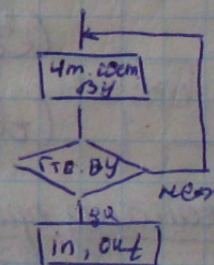
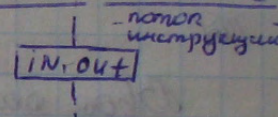
Система ввода-вывода МЦ. Классификация:

1. Программно-управляемая система обмена

- безумный обмен

- умный обмен

(проверка готовности)



- обмен по прерываниям

- обмен с квитированием

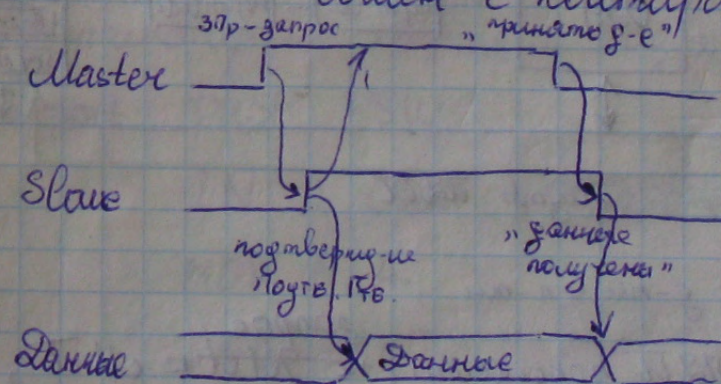
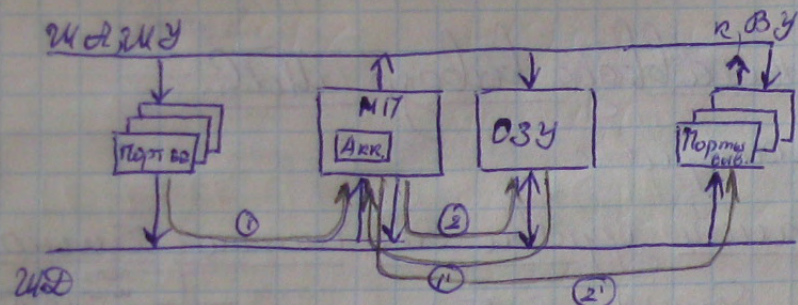


Схема взаимод. вие портов
 ввода-вывода в МЦ

с архитектура-ой архитектуры:

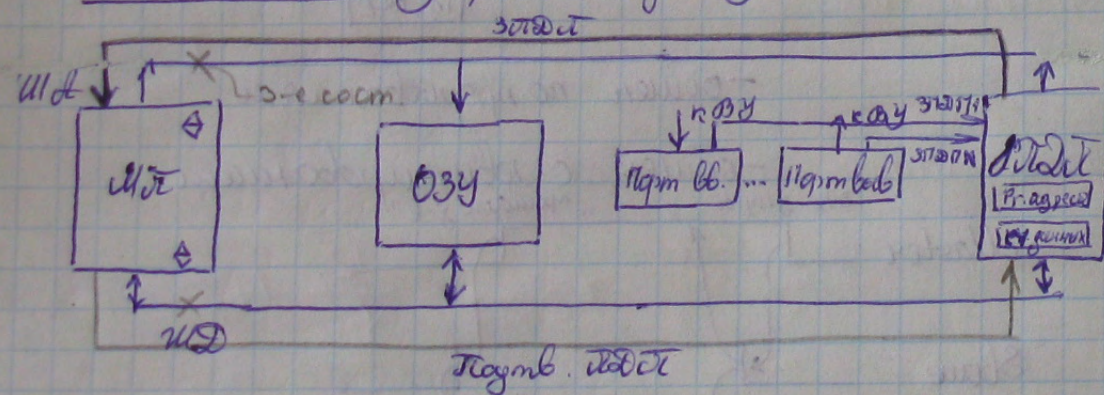


Ввод данных: Порт вв. → МЦ → АКК МЦ (1)
(А) → ОЗУ (адрес) (2)

информ. из арх-ра

Вывод г-2: ОЗУ (адрес) → АКК. (1')
(ст) → Порт выв. (2')

2. Обмен по каналу прямой доступ к памяти.



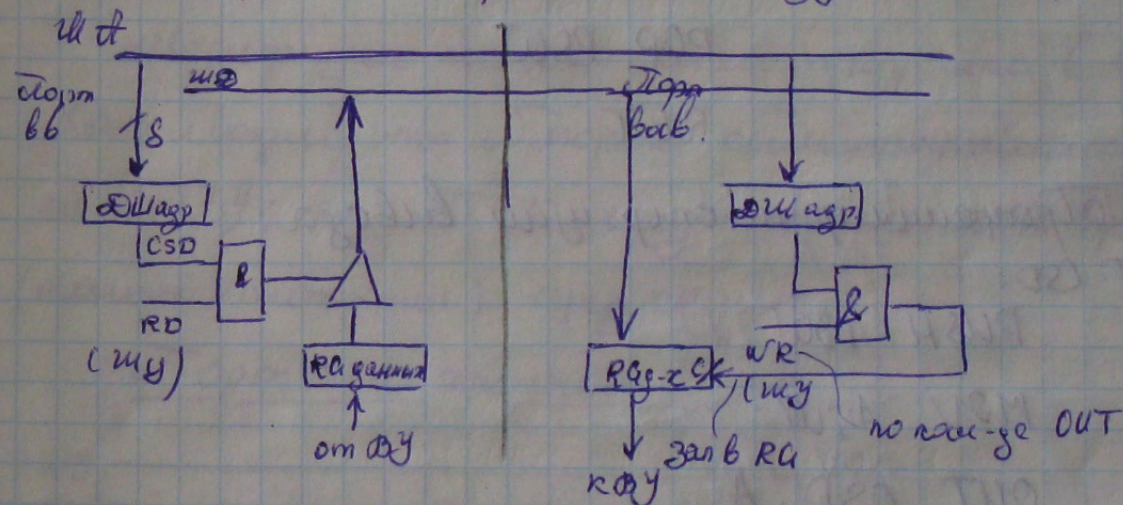
МДПТ - канал пр. до г-на с п-ти

на МДПТ запрос МДПТ получает, но МДПТ к МЦ,
подготов-ие, МЦ-выходной / входной в 3-е состояние,
МЦ приходит в состояние под проверку МДПТ.
Передача данных до тех пор, пока не-г-2

не будет равен 0.

Схемотехника портов ввода-вывода.

Схема портов синхр. обмена (безусловного).



Программные операции ввода:

IN-port: PUSH PSW

IN A, CSD ; CSD - символ-симв. порта

MOV M, A.

POP PSW

RET

Если номер порта указан в регистре од. адрес-а с
и исп-е обдуплю-ру работает с портом ввода


```

IN_CSDx:  PUSH PSW
          +1 MOV A, C; номер порта в арег.
          +2 STA IN_CSDx + 6
          +5 IN A, 0
          +6 MOV M, A
          POP PSW
          RET

```

Программа-операция вывода:

```

OUT_CSD:  PUSH PSW
          MOV A, 0
          OUT CSD, A
          POP PSW
          PUSH PSW
          RET

```

с переключением порта "CSD"

Программа модификации номера порта из регистра C:

```

OUT_CSDx:  PUSH PSW
          +1 MOV A, C

```

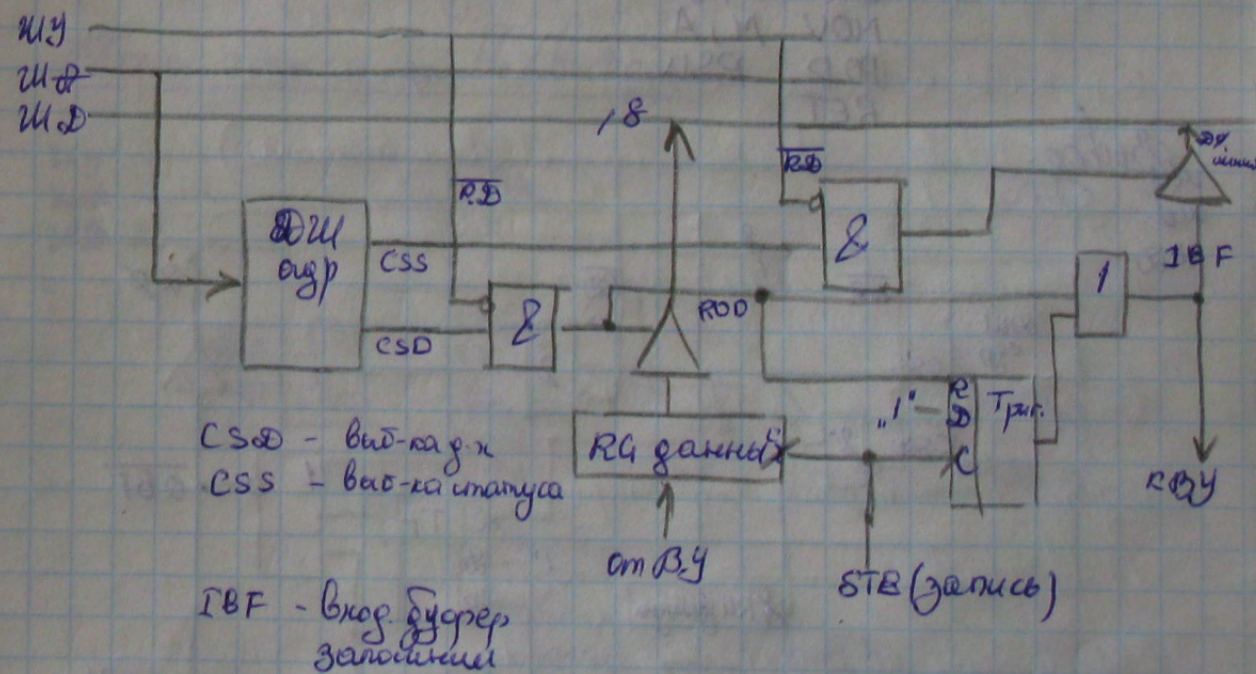
```

+2 STA OUT_CSDx + 7
+5 MOV A, 0
+6 OUT 0, A
  POP PSW
  RET

```

Модифицирующая программа, с помощью которой, в ОЗУ. Как модифицировать порт, с помощью программы, с помощью ОЗУ: через СТЕК, через порт памяти программы (только константы), через регистры.

Порт асинхронного ввода.



Адаптер программируемого параметричного интерфейса (КР580ВВ55)

Структурный параметричный интерфейс (ППИ).

Можно иметь 3 порта 8-и разряд (2-а 8 разряд и 2-а 4-разряд, которые можно объединить в один 8 разряд.)

РА
8р

РВ
8р

РС
2x4р

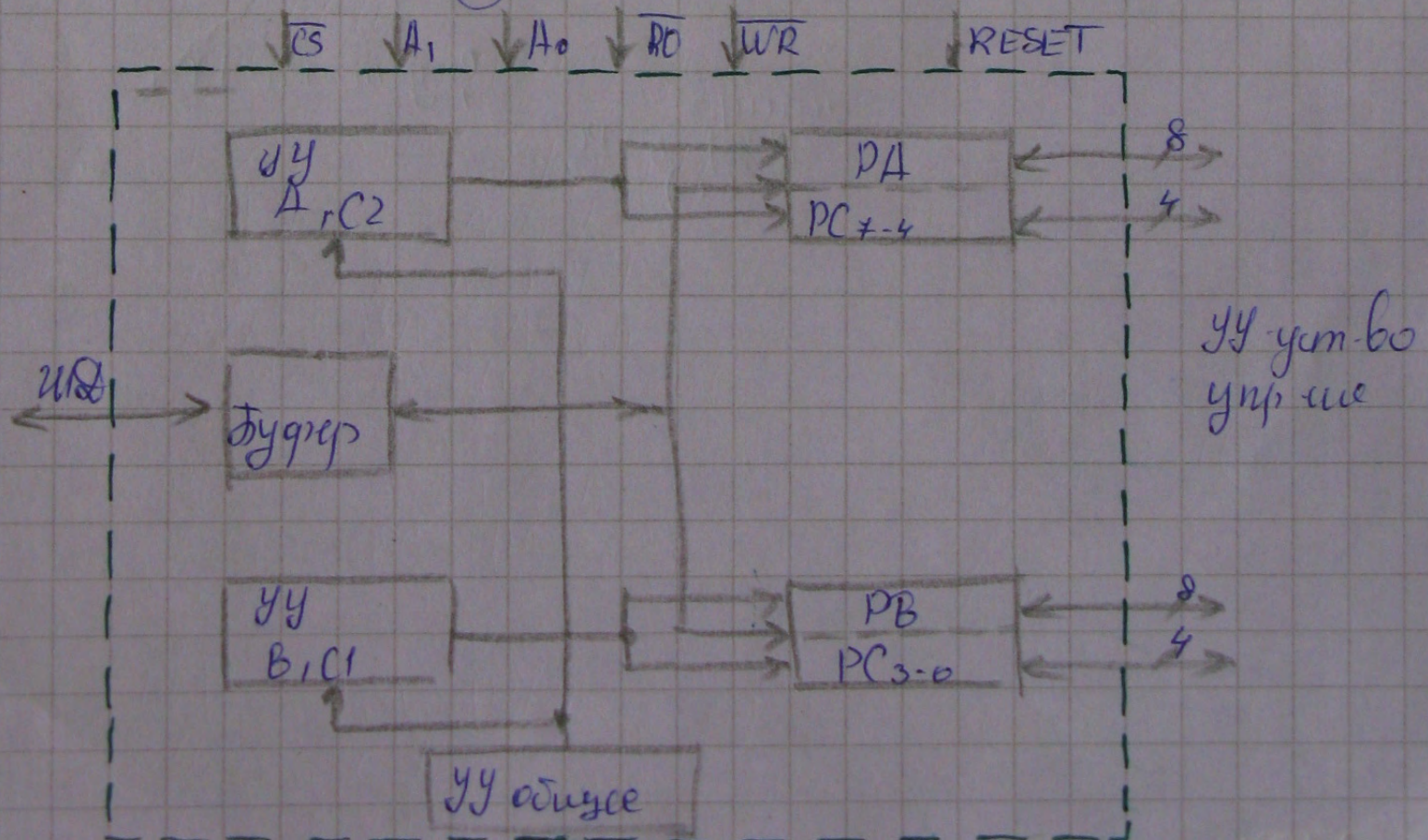
РС с локальным управлением в каждом бите.

Режимы: - синхронный ввод/вывод по командам IN/OUT

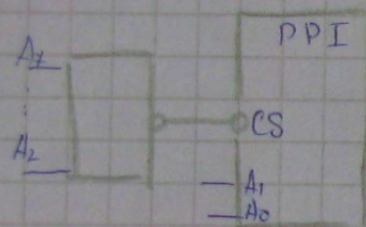
- асинхронный ввод/вывод РА и РВ по
формированию запроса прерывания

- асинхронный двусторонний РА совместно с
работой РВ

Структурная схема ППИ.

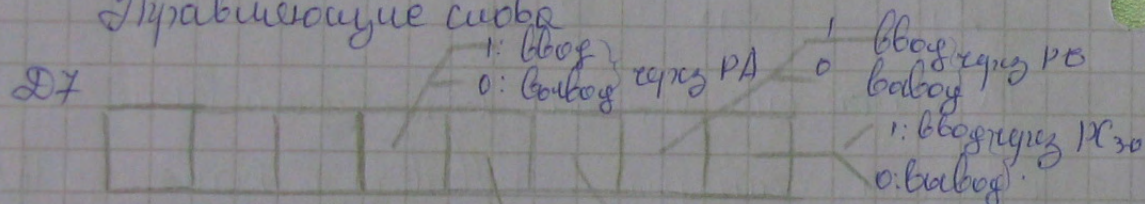


Адрес	Порт
0 0	PA
0 1	PB
1 0	PC
1 1	RQW - п-р упр-ю слова



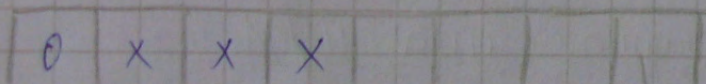
PPI задает адреса, напр-р, FC, FD, FE, FFh

Управление слова



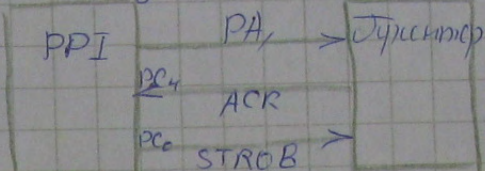
00 интр.
01 асинхр.
1x асинхр.
дву интр.

1: асинхр.
0: интр.
1: ввод, 0: вывод; 1: ввод, 0: вывод; 1: ввод, 0: вывод; 1: ввод, 0: вывод; 1: ввод, 0: вывод; 1: ввод, 0: вывод; 1: ввод, 0: вывод; 1: ввод, 0: вывод.

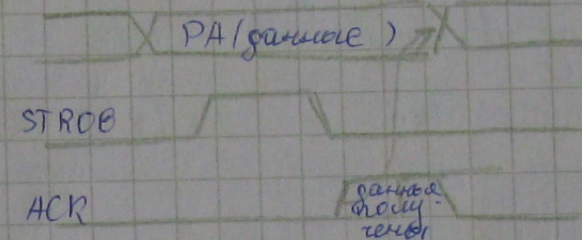


напр-р, упр-ю PC; 0, упр-ю PC; 1, упр-ю PC; 0, упр-ю PC; 1, упр-ю PC; 0, упр-ю PC; 1, упр-ю PC; 0, упр-ю PC; 1, упр-ю PC.

Пример передачи данных по протоколу Centronix где принтер с использованием 0.



АКК - поств-ие передачи



```
count EQU 8
PA EQU FCh
PB EQU FDh
PC EQU FEh
RQW EQU FFh
STROB EQU 01h
NSTROB EQU 00h
CW EQU 10001000B
DIASK-ACK EQU 10h
```

; упр-ю слова
д/настр-ки прог-ю парю
инт-са (ПЗУ).

; команда инициализации

```
LXI H, addr - MES
MVI B, count
MVI A, NSTROB
OUT PC, A
MOV A, H
OUT PA, A
MVI A, STROB
OUT PC, A
```

; адр. соод-ие
; код-коперис-х инт-в.
; инт-в из памяти
; вывод инт-ва

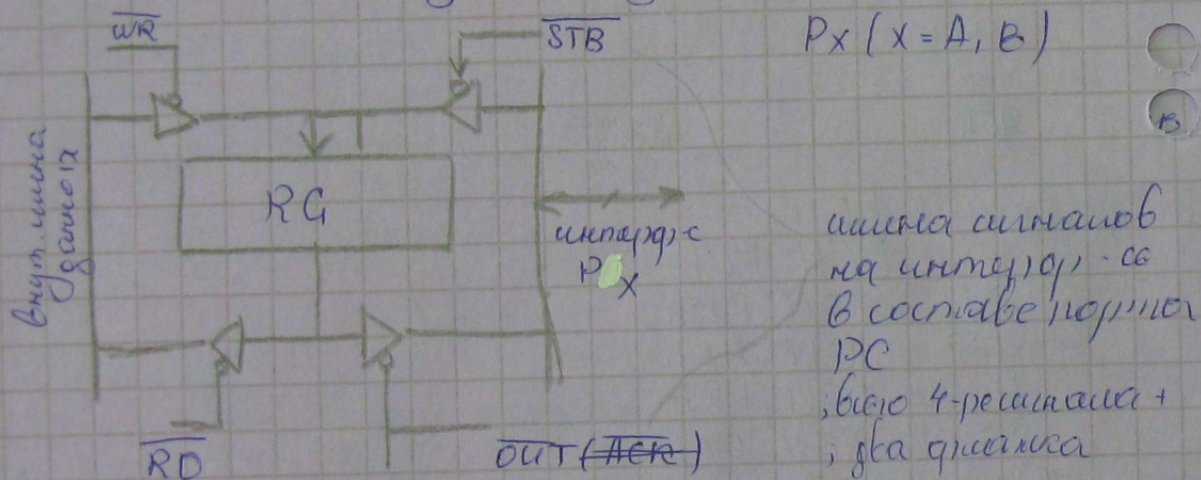
```
MVI A, NSTROB
OUT PC, A
wait: IN A, PC
```

< ; много задержки
; нет строба

ANI A, MASK ACK
JNZ wait

INX H; +1 указ. рег-тр
DCR B
JNZ loop

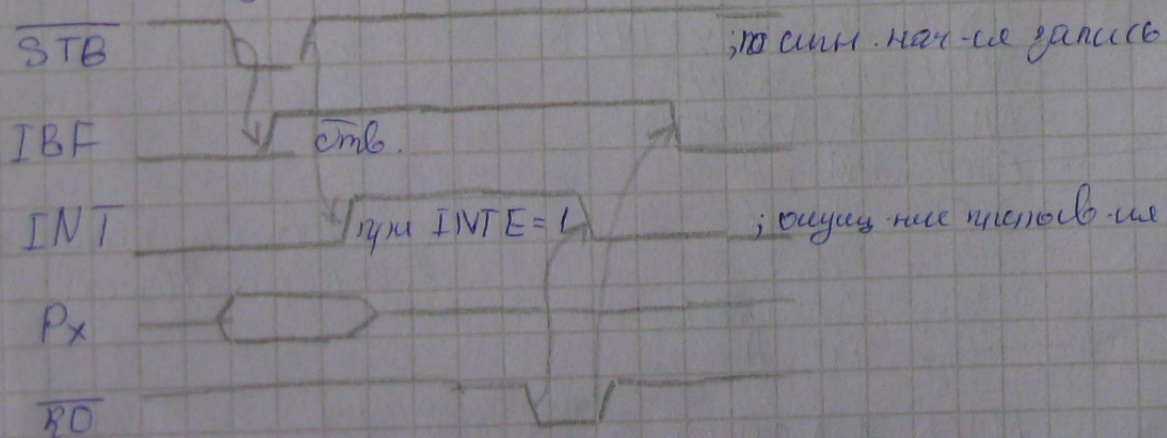
Асинхронный обмен с кэшированием
(т.е. с подтверждением)



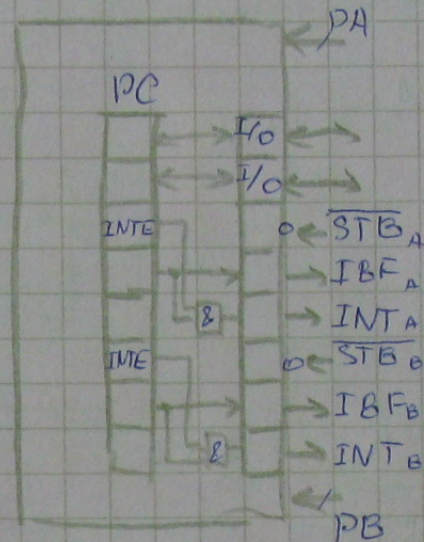
В PC используется 6-битный.

Операции ввода.

Врем.-ые диаграммы



Интерф. -ые сигналы при вводе:

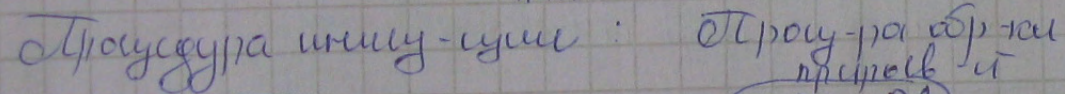


Врем.-ые диаграммы ввода:



Операции вывода.





Троусдур иинг-уинг : Троус-ра бор-та
прирав и

INT - PA

Зару-ш обр-ку
присованай

Corp-muo PSW

Возв. сив. из
памяти

Barbog 6 12 A

Уст-во стробв "1" {; 61 б

Сбориште студио в „Ø“; 00h

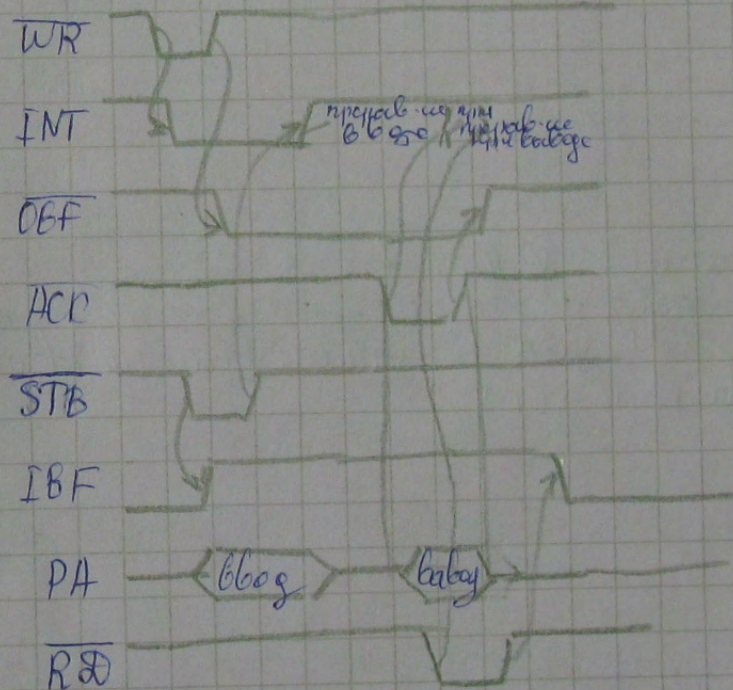
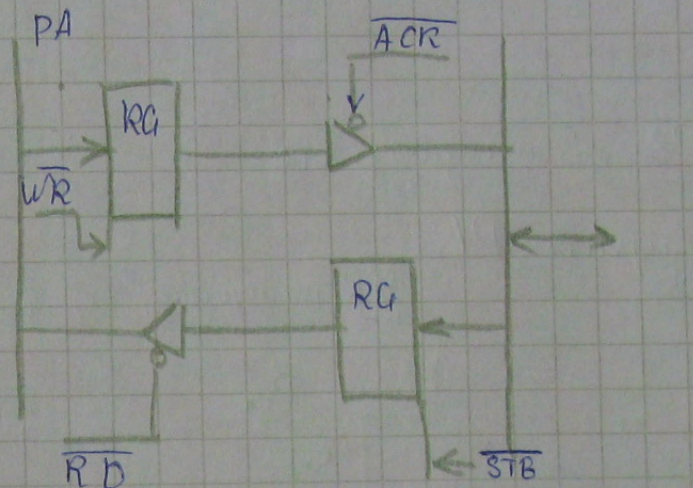
Известно, что в окружении,
успешности или неудачи

Кому
сообщить?

затраченные на переводы
от 1000

Разр-мо
прераба-на

Второй

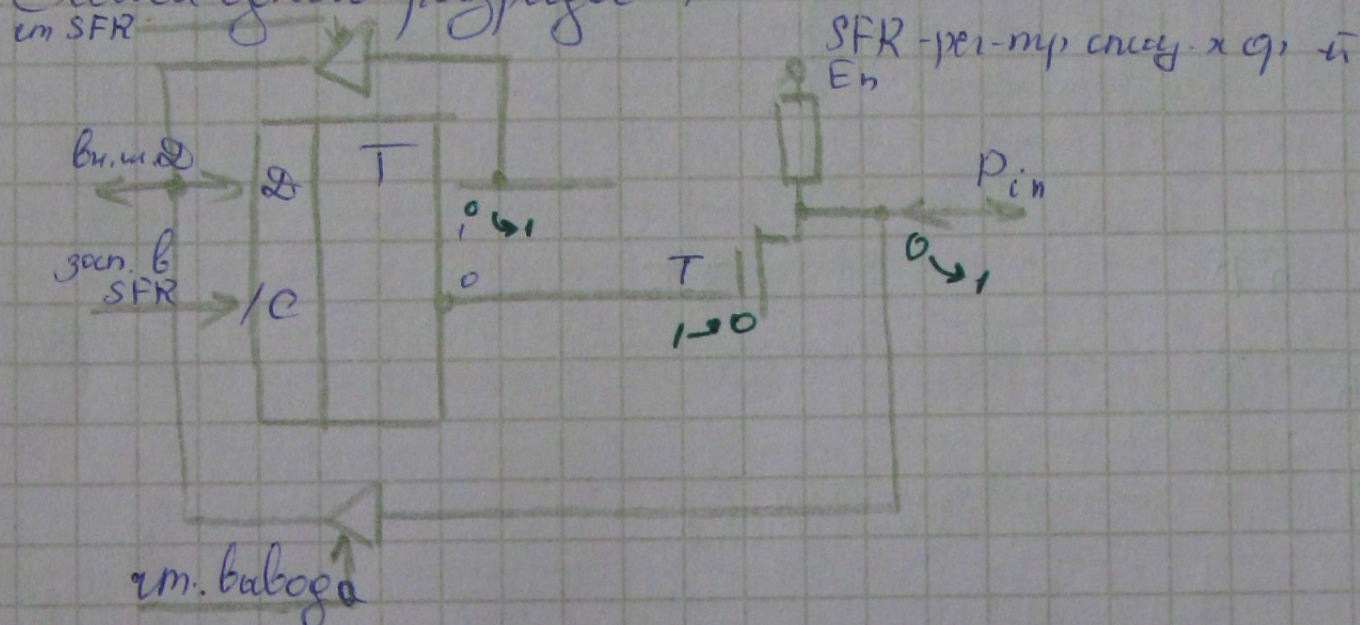


Порты микроконтроллера MCS-51.

MCS-51

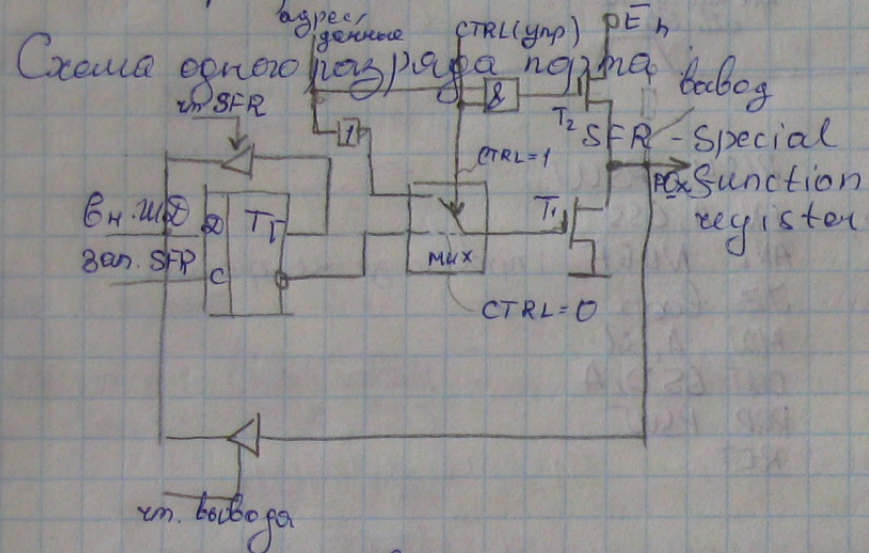
P0, P1, P2, P3 ; 8 бит разрядности

Схема одного разряда P1.



Вывод: записываем 0 или 1. inc P1
итого - инициализация записи (принимать через T-триггер)
Ввод: предв. T-триггер "1", затем вынужденный ввод.
(две команды).

Порт P0 - двунаправ-ий и мультиплексный
адрес + данные
; исп-ие д/передачи интр-х д-х: адр-ов и д-х,
при работе с внешней памятью.



При выводе во внешнюю память д-х или при
обр-нии с внеш-й памяти проц-шы:

$CTRL=1$: Енма выходе (вых. пот-и)
при адр/д-х = 1

при адр/д-х = 0 → низк-й пот-и

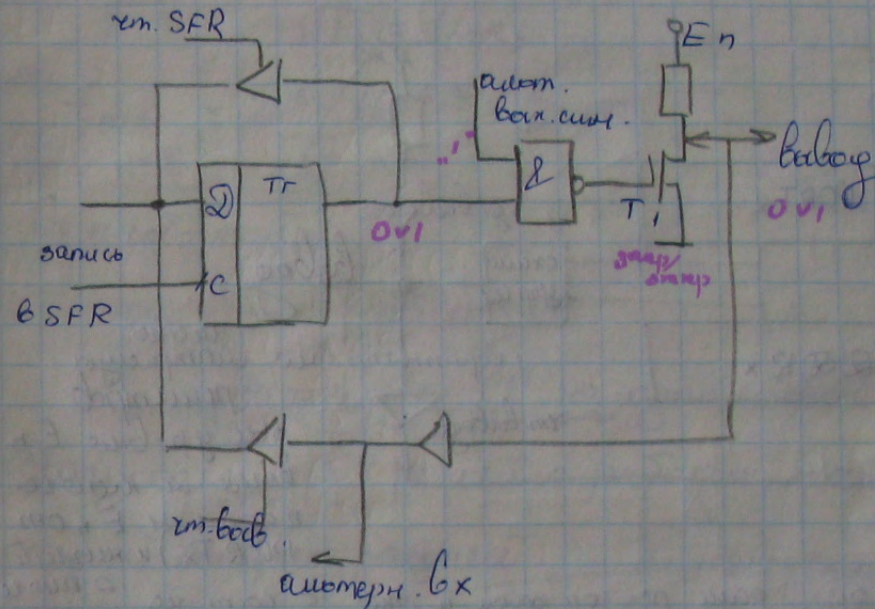
Порт ввода $CTRL=0$.

Порт P3 - порт с альтернат-м упр-м сигналами:

1) вв/выв.

2) альтерн-й сигнал вх и вых.

Схема одного разряда



Ввод/вывод

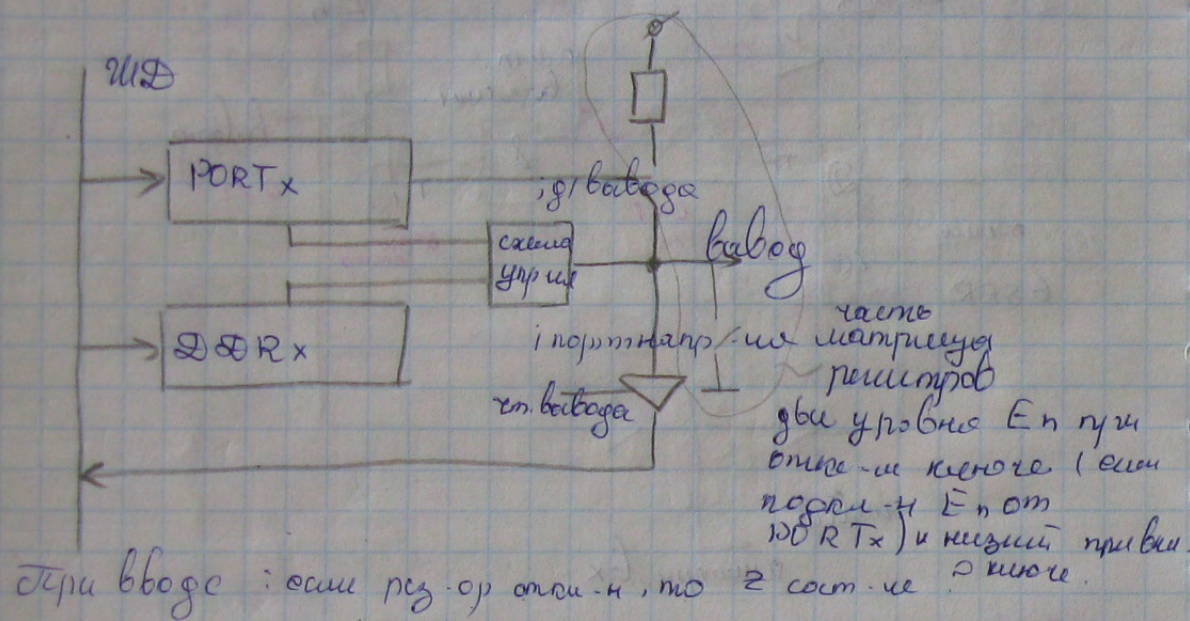
При выводе данных $адрес = 1$

При вводе д-х $Тг \leftarrow 1$, $адрес = 1$.

При сбросе $Тг \leftarrow 1$.

Порт P2 поодино P0, без передачи данных; д/передачи
адресов в режиме вв/выв.

Порты МК AVR



Таймер МК

1) Таймер MCS-51 (МК51)

2 таймера на 16 разрядов (макс.); T0, T1

- таймерный режим (относит интервал в р.)

- счетчик событий

↑ событие
внешний вход по таймеру

Режим таймеров: синхронизированный по переключениям.

Событийный режим:
 $\frac{S_p}{12}$ при $S_p \approx 12 \text{ МГц}$
 $T_{bx} = 1 \text{ мкс.}$

$$t_{\text{max}} = 2^{16} \cdot 10^{-6} \approx 65 \text{ мс.}$$

Событий:

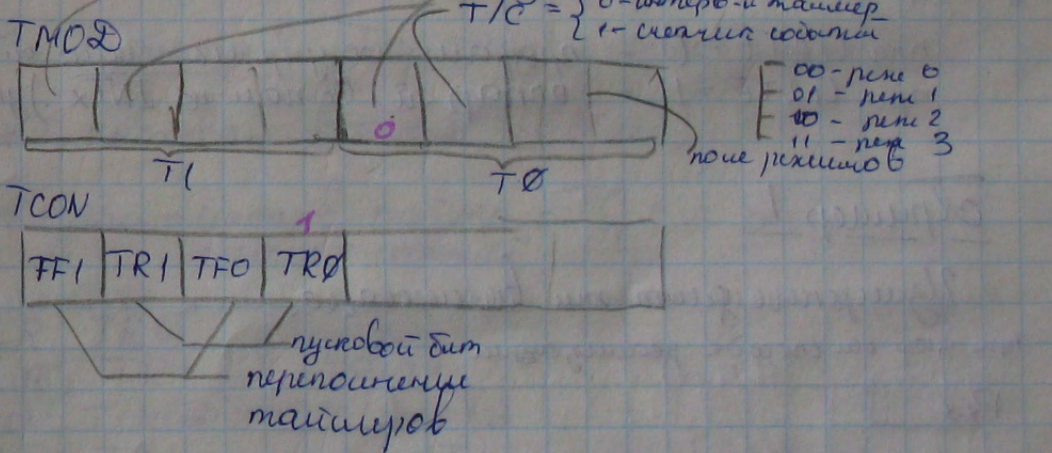
при $S_{\text{max}} = 500 \text{ кГц}$

два периода СС на одно событие

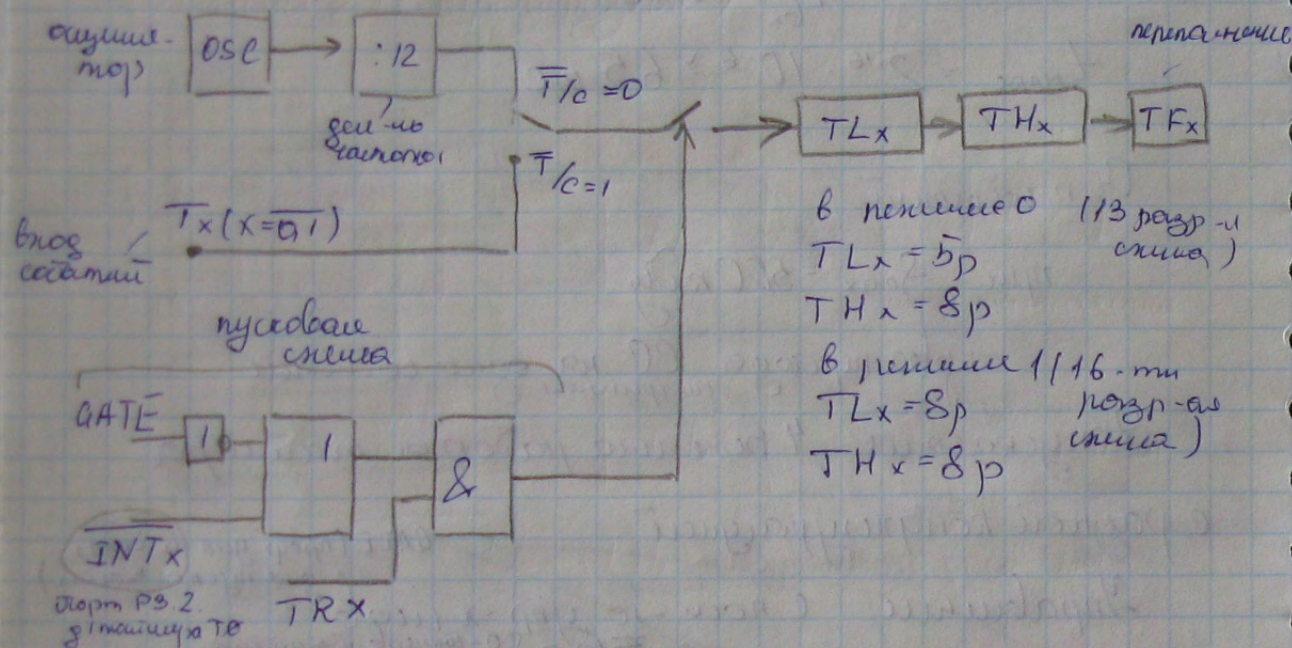
Допускается 4 режима работы таймера

с разной конфигурацией.

Управление с помощью упр-х и рег-ов



Режим 0 и 1. (конфигурация таймера)



при GATE=0 - программ-ое управ-е таймером
при GATE=1 - аппарат-ое управ-е таймером (с помощью INTx)

Пример 1.

Измерение длительности внешнего сигнала
при помощи способа реализации



Пусть

г/Т0 P_{x.y} = P3.2

г/Т1 P_{x.y} = P3.3

Проц. процедура г/Т0

CLR TR0 ; броси Т0

mov TMO0, #00001001B

CLR TL0

CLR TH0

SETB TR0 ; броси Т0

wait1: JNB P3.2, wait1

wait0: JB P3.2, wait0

CLR TR0 ; броси.

mov 00h, TL0 ; по адресу 00h, 01h
mov 01h, TH0 ; по адресу 00h, 01h

Пример 2. Подсчет событий за заданный интервал времени (50 мс).

Выбираем - интерв. таймер - Т0
счетчик событий - Т1

CLR TR0 ; запр-ем

CLR TR1 ; Т0, Т1

mov TMO0, #01010001B

CLR TL1 ; сброси Т1

CLR TH1

MOV TL0, #low(-50000)

MOV TH0, #high(-50000)

SETB TR0

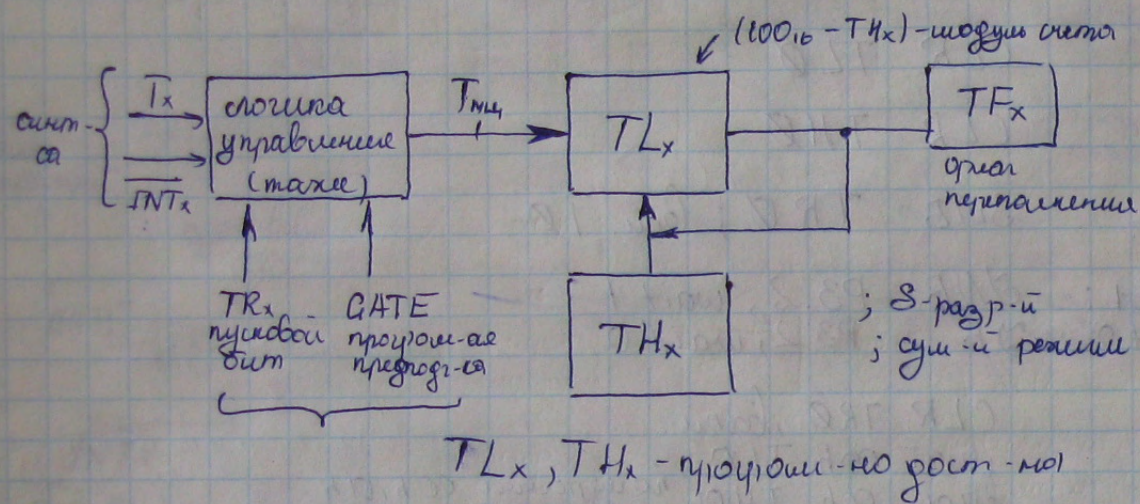
wait: JNB TF0, wait < CLR TR0, CLR TR1

mov 00h, TL1

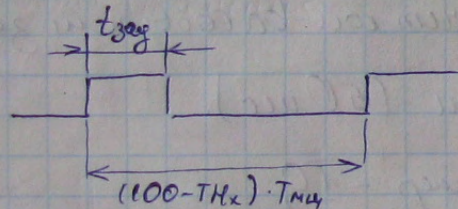
mov 01h, TH1

продолжение на биол. шитал 1,2,3,4

Режим 2 (с перезагрузкой)



Схематическое:



1) сформирование широтно-импульсной модулированной (ШИМ) сигнала

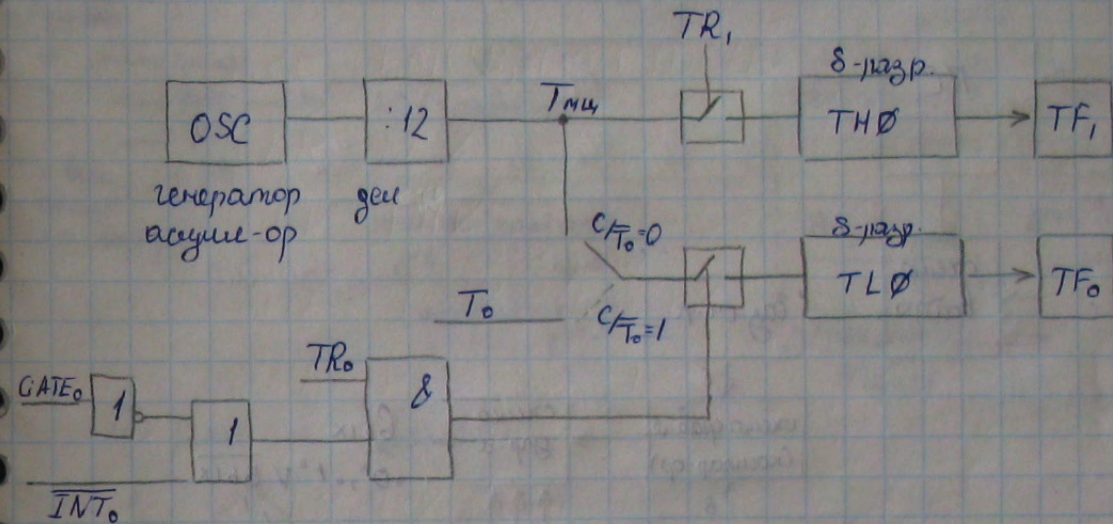
2) использование UART (работа последовательного канала).

; UART - универсальный асинхронный последовательный канал приема/передачи

таймер T_1 в режиме 2.

Режим 3. Можно сделать из 2-х таймеров 3.

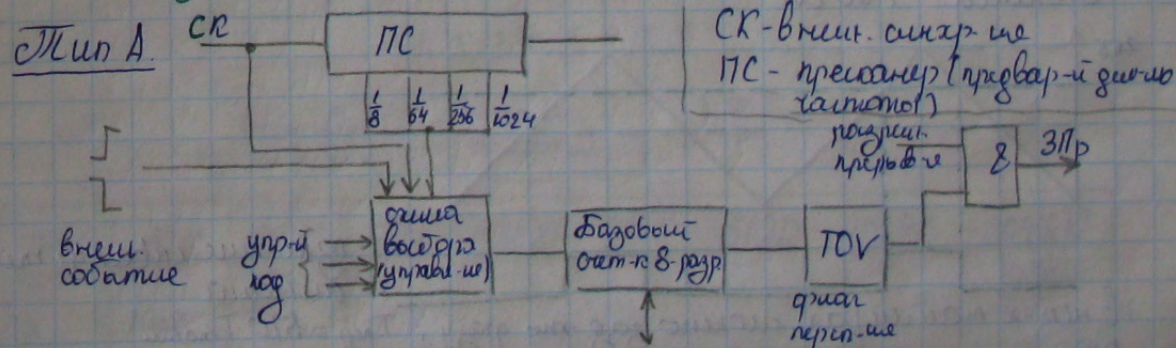
Конфигурация таймеров:



T_1 работает авт-ки: запуск при программировании TMOD; останов при задании режима 3 для T_1 .

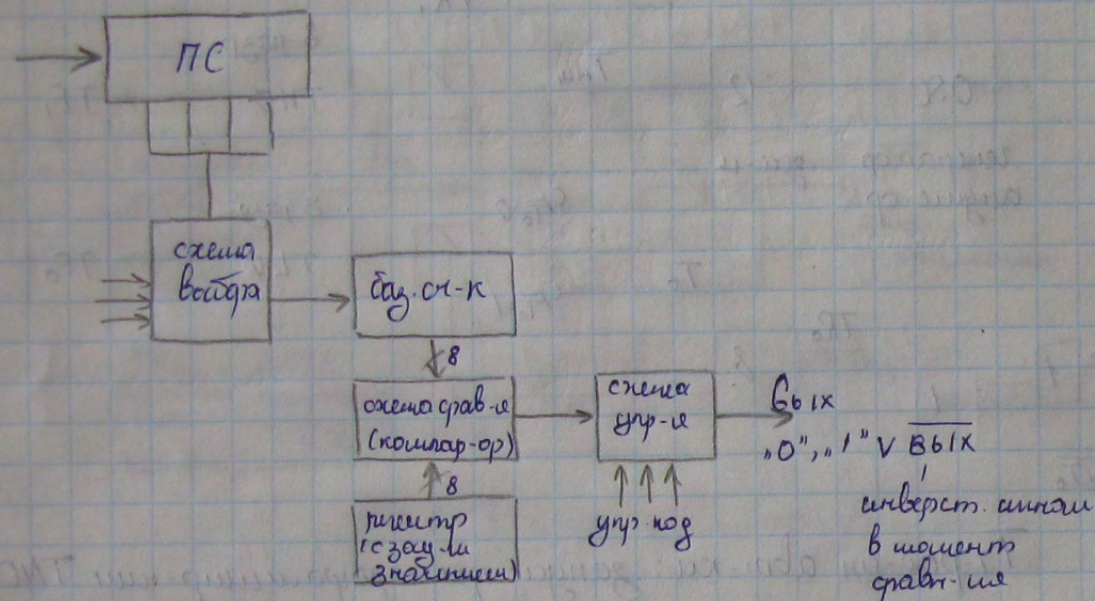
Сл. T_1 используется как генератор (генератор частоты) син-ов сигнала 8/полюс-го канала.

Архитектура и функциональные возможности таймеров в составе AVR.

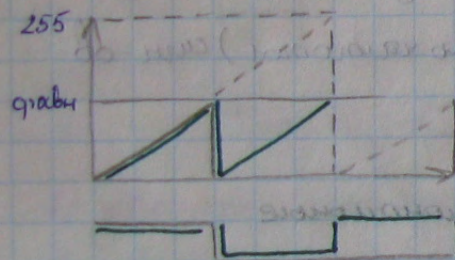


Тема В. (таймер + др-ие сравн-ние, РШМ (или))

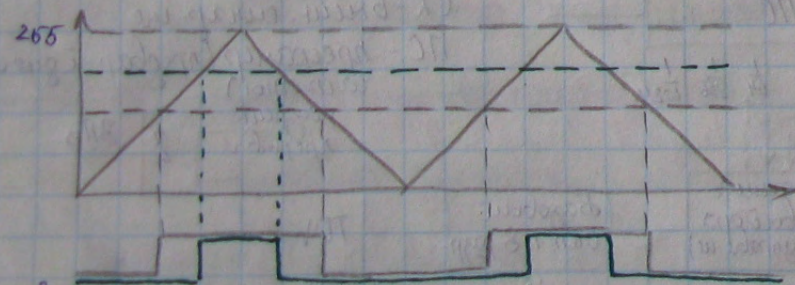
22121 - широколиственные шор. лес;



Решение сравнение:



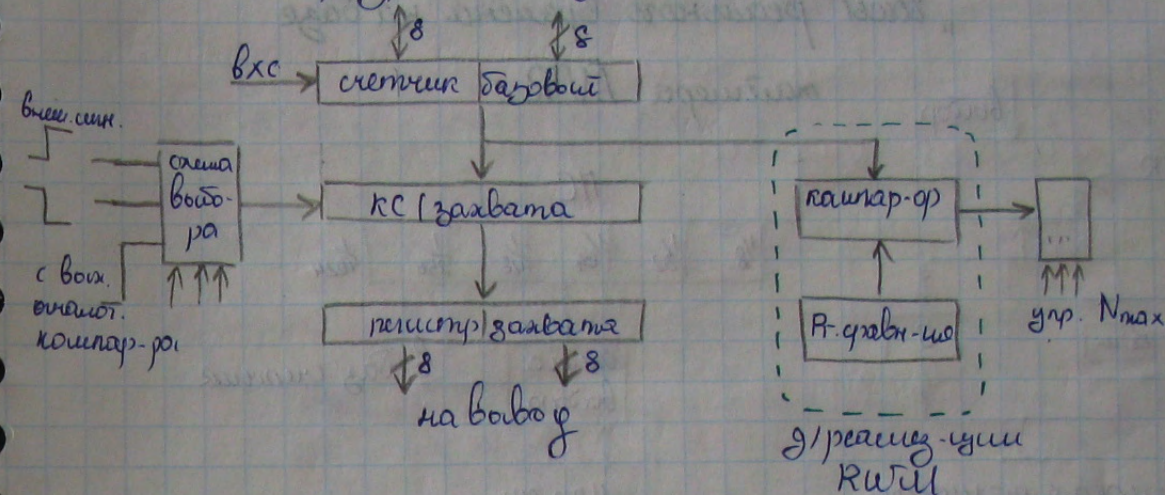
Решение РШМ



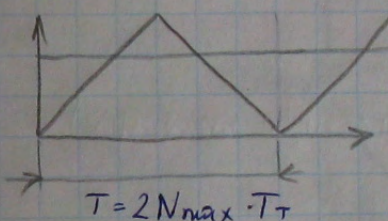
изм-ние свачен-ти
сигнала

Внеш-х таймирах можно зад-то сразу; $T_{\text{пер}} = 5/10 T_{\text{нашии}}$
 вых. сигнала (прямою или инверс-я аттн).

Мин 2 (16 разр. с-ин, захват, сдвигание (РШМ)).



Björn Rasmussen



$$N_{max} = \begin{cases} 255 \\ 511 \\ 1023 \end{cases}$$

Сторожевой таймер (WDT).

Watch Dog Timer c/croc WDT uz -
 программа - WDR



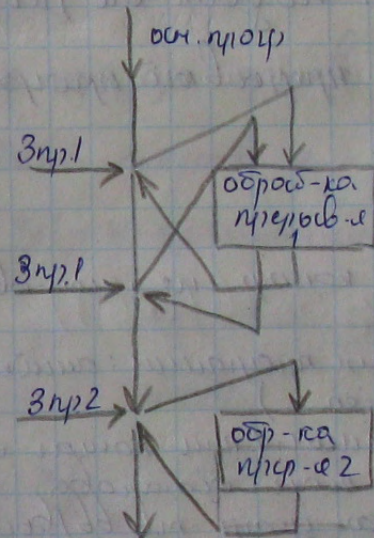
W	16K	32K	64K	128K	256K	512K	1024K	2048K
5B	15mc	30mc						1.9c
3B	47mc	94mc						6.0c

"Часы" реального времени на базе таймера AVR.



Системы прерываний.

Сис-а пр-й - комплекс алг-р-х и проц-х, обесп-х
обр-ку зап-ов прерыв-ий



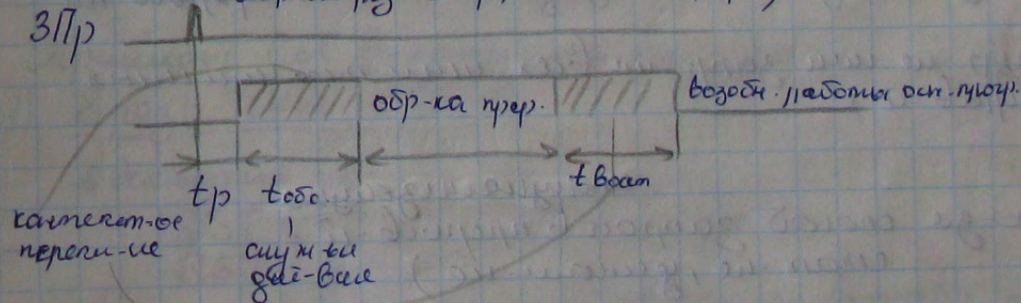
2048K

1,9с

6,0с

Время реакции на прерыв-ие, время обслужи-ия
прерыв-ия, время выполн-ия, задержка прерыв-ия
(сост-ие из-ов прерыв-ий и осн. проц.)

ЗПР



Э-е, кот-ые вып-ие при прерыв-ии:

1) сост-ие адр-а возвр-та / осн. проц-ии в
памяти (в стеке обраб-ки), вып-ие авт-ки

2) сохр-ие слова сот-ие прер-ий PSW
(все приу-ки прерыв-ой оп-ции и актуал-ор, по необход-ти)

3) сохран-ие все приор-ногост-ые рег-истр, к-е м-т быть исп-ны прерыв-ой прер-ой

3) ^{пост-ие в сме-у прер-ий} пр-е-ие на категорию:

- 1) аппаратные (от схемы контр-ля, исп-ов питания, и.т.д.)
- 2) программные (от польз-им программ: ошибки дел-я на 0, access violation...)
- 3) от операционной системы (при помех-ой работе, нечел. ошибках...)
- 4) внешние приу-вание (от устр-ств вв/выв-а (внеш-х устройств явл-ие внутр-ним))

Функции, выполня-ие сист-ой прер-ия (ССТ)

1) инициал-изация ССТ: задание маски прерыв-ий

1.1) (кажд-ие или загр-ие всех или тех или иных прерыв-ий)

1.2) ука-зе способ запросов прерыв-ий: стат-ие, динам-ие

1.3) уст-ка приоритетов (есть приор-ты по умолчанию); в арх-ре 51-го два ур-ня прер-ий

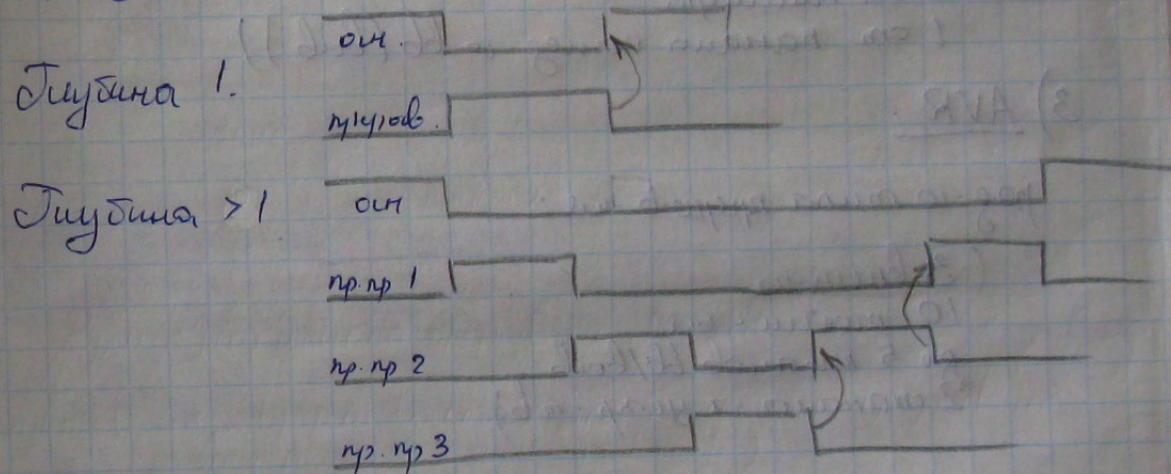
1.4) уст-ка вектора прерыв-ий в тех случ-ях, когда приу-ие специал-й контр-р прерыв-ий.

2) при патунти-нии загр-ие ват-ие конт-а прерыв-ия

3) при окончании обр-ки ^{прерыв-ие ват-ие} конт- прерыв-ия и вост-ании ССТ, если она была заблокир-на

4) инициал-изация обр-ки запросов.

Глубина прерыв-ия - количество принятых на обр-ку ЗПР, и возможность внешнего прерыв-ия.



Примеры ССТ в разных архитектурах:

1) 8-разр. МПЛ i8085

8 векторных прер-ий

RST 0
RST 1
RST 2
RST 3
RST 4
RST 5
RST 6
RST 7

Точка входа:

$n \times 8$ - адрес в табл. прерыв-ий

$n = 0, 7$

4 доп. прерыв-ий: 1) прерыв-ие по уровню
 радиал-ые прер-е - запросы кот-н связаны
 с адр-ами тайм-уов прерыв-ие заранее

2) MCS-51

5 бит-ов запроса прерыв-ий от разн. устр-в:

все запросы радиальные:

(2 внешних
 2 от таймера
 1 от канала помех-го вв (во в))

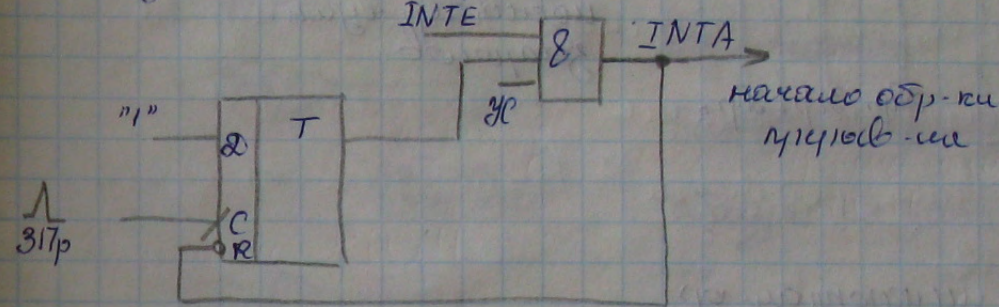
3) AVR

раз-го типа прерыв-ие:

(3 внеш-их
 10 тайм-ов
 до 5 каналов вв/во в
 2 канала-х устр-ва)

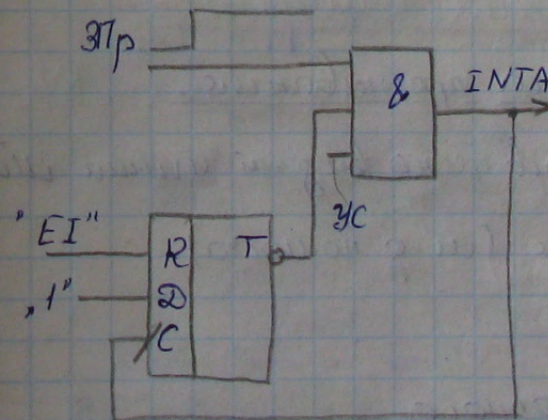
Схемы обработки статических и динамических запросов прерываний.

а) динам-ий ЗПР



INTE - interrupt enable - разрешение пр-а
 ЗС - управл-ий сигнал
 INTA - int-pt acknowledge - подтверждение прерыв-ия

б) статический запрос прер-ие ЗПР

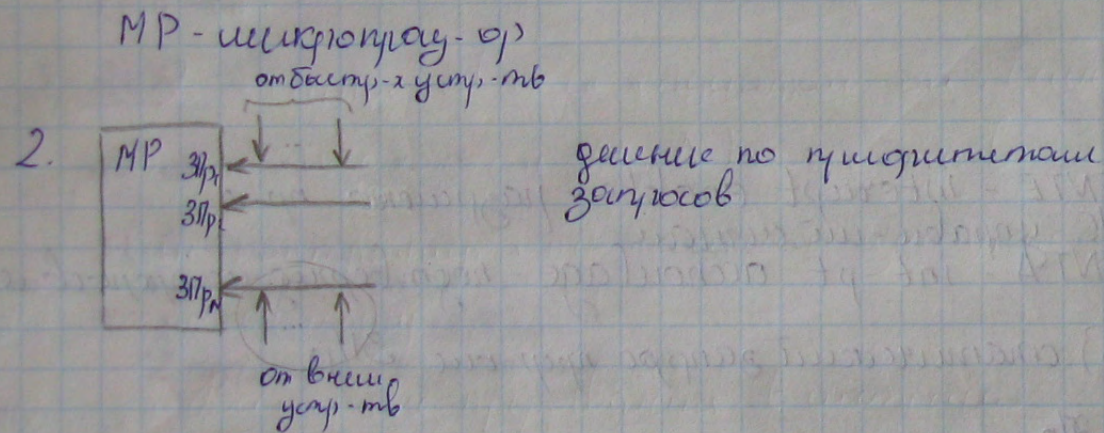
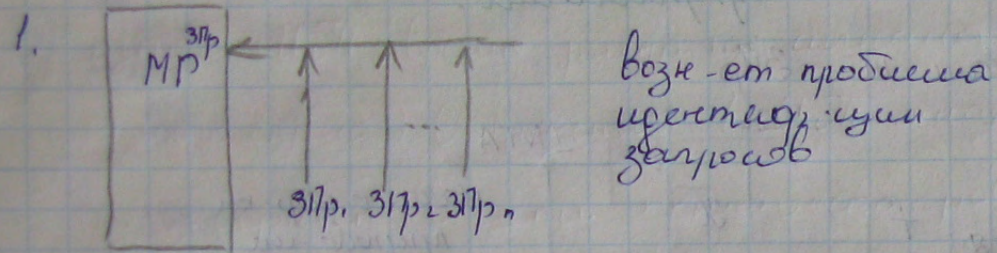


В конце обработки или ранее
 сбросить запрос пр-а от
 источника (процессин-ая
 процедура дл-внеш-устр-ва).

EI - enable interrupt

Обработка ЗПР:

1. по одноуровневой схеме прер.-ий
2. по многоур.-ой схеме прер.-ий



Идентификация запроса прерывания.

П.е. устан-ся чет-к прерыв-е на вход-й линии мп

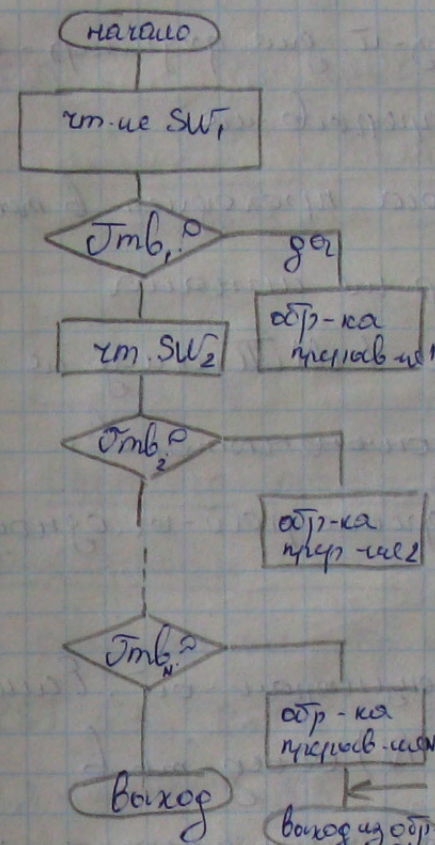
а) программ-ое иден-ие (м-д поминга)

б) аппарат-ое ир-ие

а) м-д поминга - ам-ти отчуса

$$t_{\text{обсл}} = t_{\text{созр}} + t_{\text{идент.}}; \text{ PОН - регистры общего назначения}$$

PSW
PОНов



SW - статусное слово (ист-ия)

Преим-во и-ра:

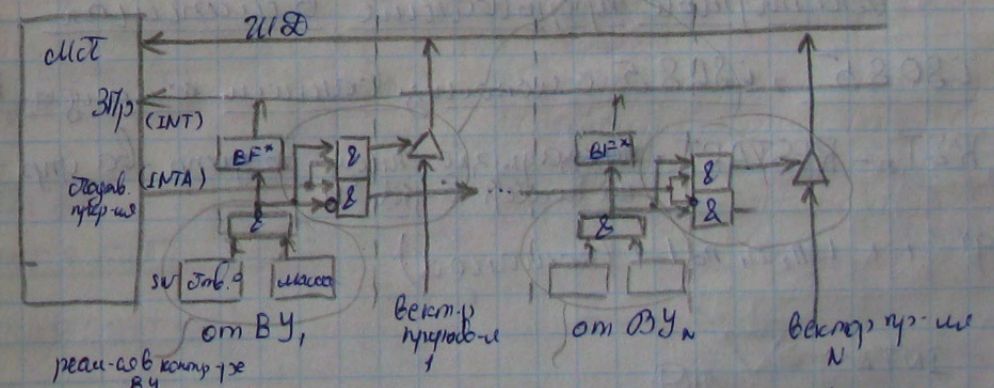
простота реализации,
гибкость программ-ия

Недостаток:

большое сред-нее время
обслужив-ия запроса.
(тогда)

б) аппаратный способ (цено стирания "реализи")

доп-ая аппар-ая реализ-ия



BF* - буфер с отсрочкой-и коммутации / реализации
монта-го или-и).
К-е ВУ представ-но микроконтр-ам.

Принимая: сигнал опроса проходит в течение
одного МТТ, время распространения сигнала
(одного опроса) в пределах 1 МТТ, иная
из задержек в цепи конечных устройств,
т.е. время реакции $\approx$ время работы одной
ком. ро.

Восторные прерывание в системах
i8085, i8085 с использованием команды RST_n.

Пример

1 1 1 n_2 n_1 n_0 1 1 (команда)

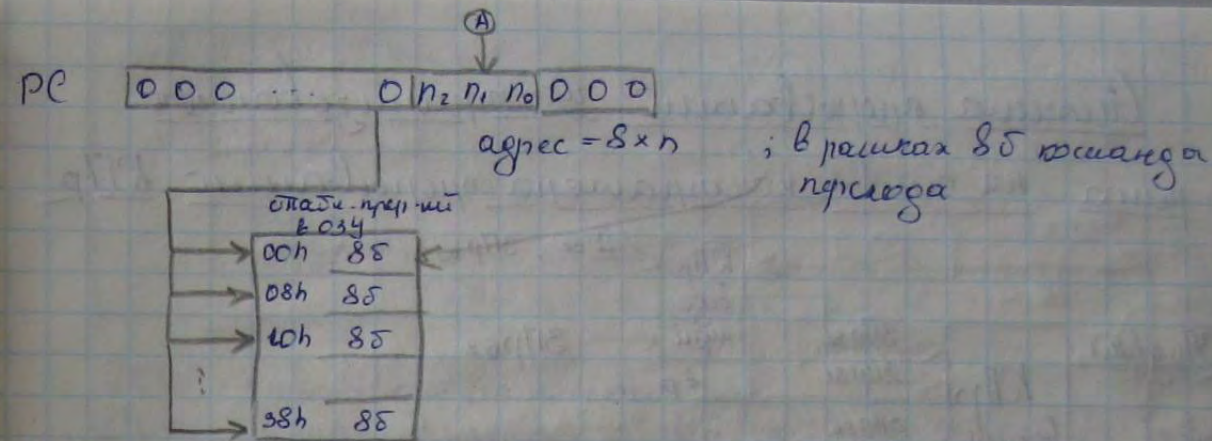
INTA → Δ ИД

ПК ИД

Блок код

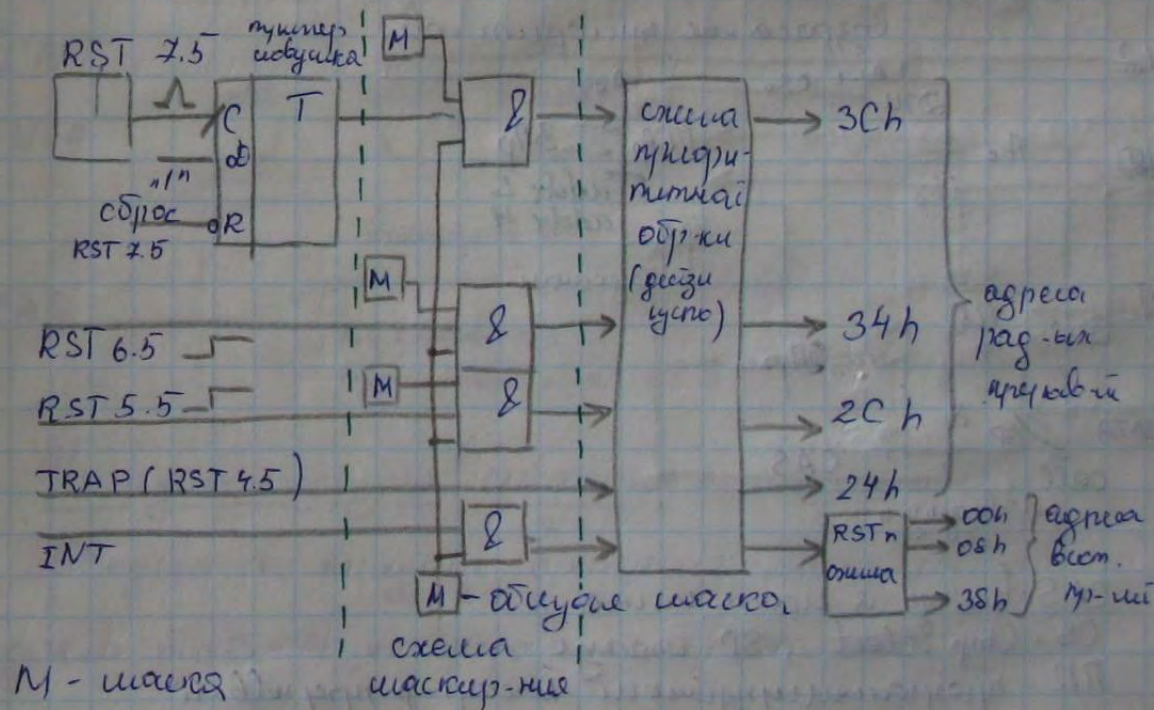
А

исполняет команду
код операции

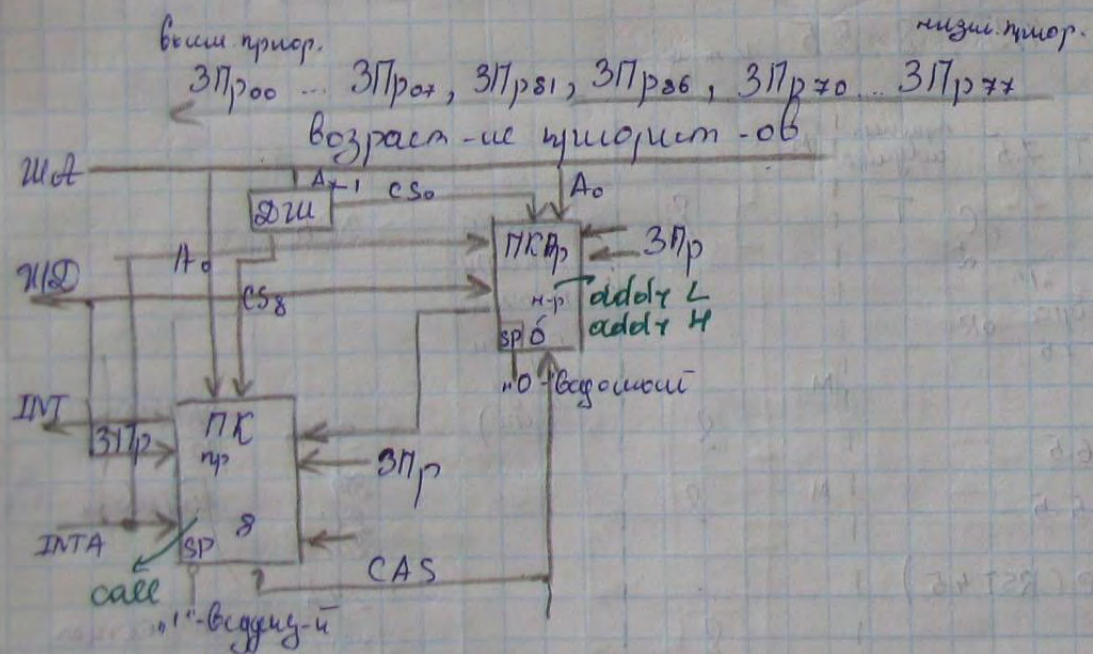
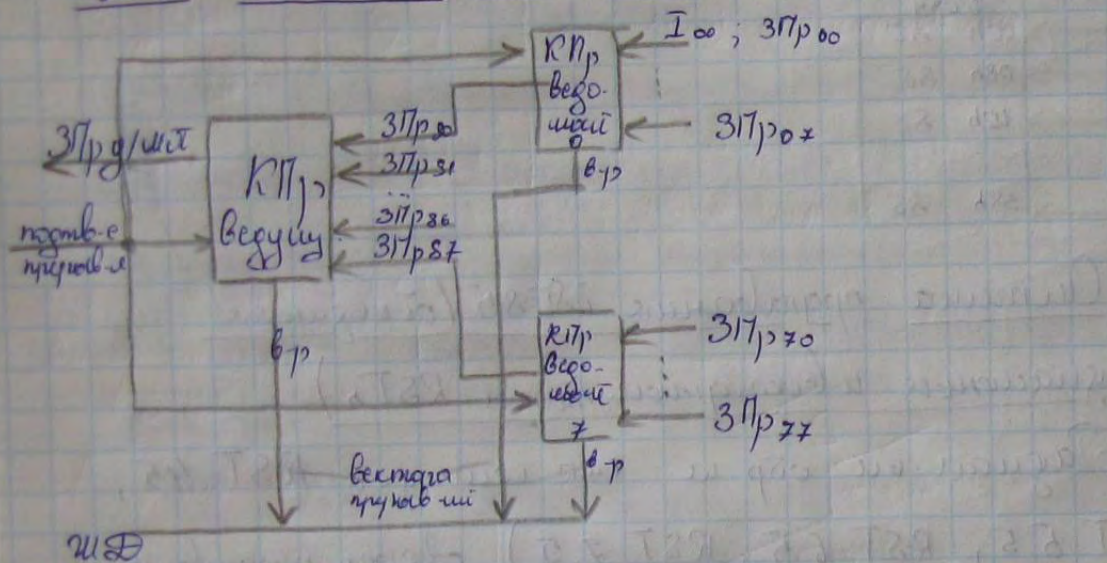


Система управления i8085 (объединяя радиальные и векторные типы RSTn)

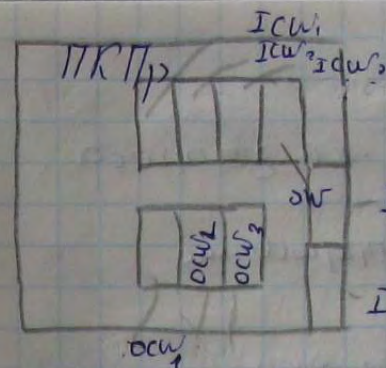
Датум-и одр-и на кои год RST 4.5,
RST 5.5, RST 6.5, RST 7.5). Агрегирани
процеси 8×4.5 , 8×5.5 и.т.д.



Система прерываний централизованного типа на основе контроллера прерываний КИПр.



CAS - вынужденная логическая единица
CS - Chip Select, SP - статус
ПК - программный контроллер прерываний



IRR - регистр запросов прерываний

ISR - регистр адресов прерываний

ICW - управляющее слово инициализации
OCW - управляющее слово обслуживания

Доступны для записи

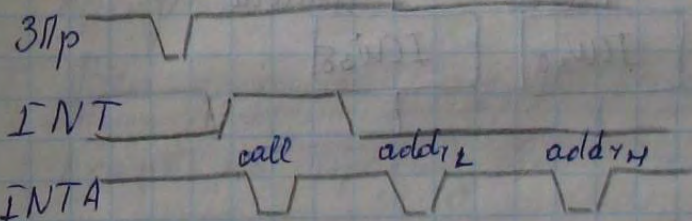
Р2 инициализации ICW₁ - 3

OCW₁ - 3

Доступны для чтения

IRR, ISR, SW

Временные диспетчеры контроллера



Вектор + баз. адрес

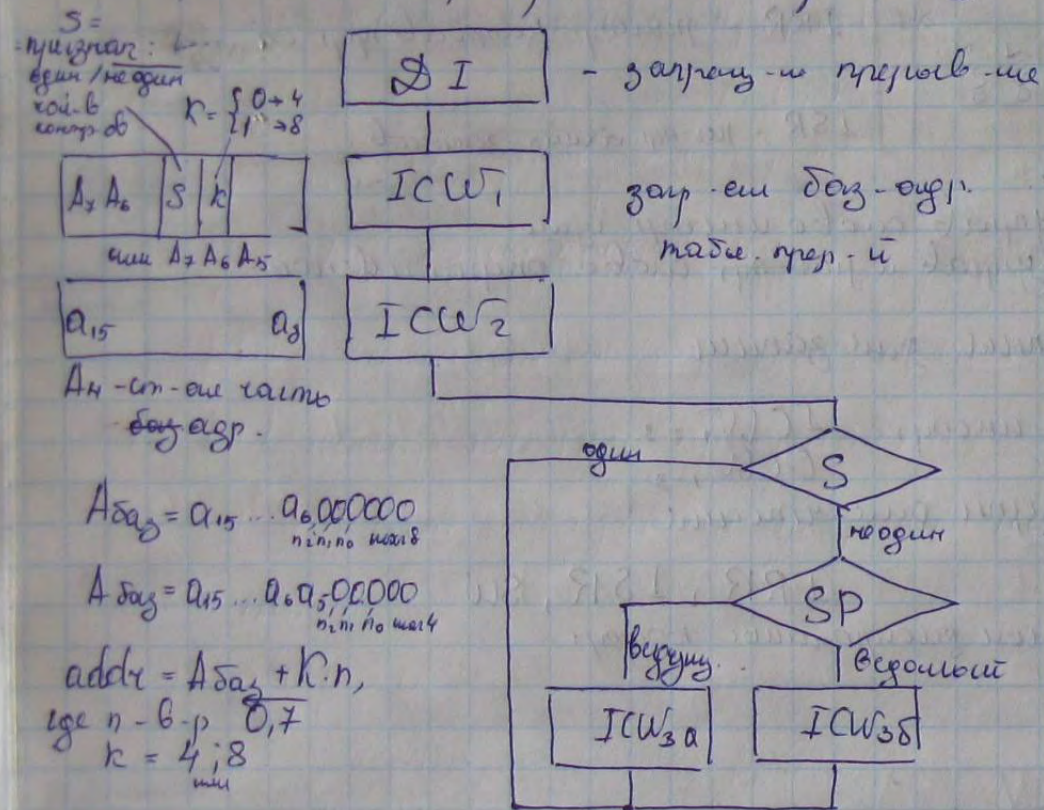
CAS - код с номером выходящего контроллера

Инициализация системы прерываний.

Сводка к передаче в конструкцию: 1) баз. состав - адреса таблиц прерываний, 2) инициализация

векторах в табл. прер-ий,

3) способа приорит-ой обр-ки запроса.



ICW3a и/ведущего:

$S_7 \dots S_0$

$S_i = 1$ - если вект от ведомого

$S_i = 0$ - ведомого на вх. нет.

ICW3b и/ведомого:

$00000 S_1 S_0$

S_{10} - н-р вх. ведущего

Способы обслуживания
прерываний в контроллере пр

1. Вект-ое прер-ие с фикс. прер-ом
 2. Век. прер-ие с учетом-ием приоритет-ов / все системы прер-ов после обслуживания тех-го з/тр изм-ие по кругу при этом обслужив-ые пр. занимают следующий приор-т, все ост-ные все-таки по кругу, следующий номер - более высо-й прер-т!)
 3. Век. прер-ие с агре-вом распр-ки прер-ов (похоже на прерыв-ий, уст-ка запр-са низшего прер-та уст-ке программ-ым путем)
 4. Прер-ие по рез-ту опроса (и-д Ломана). (програм-ные путем уст-ки все-таки прер-т)
- OCW₁, OCW₂, OCW₃ - группа прер-ов

OCW₁ ^{Q₇} M₇ M₆ M₅ M₄ M₃ M₂ M₁ ^{Q₀} M₀ маска 317j

8 битов по шине вх/перыв-ий

OCW₂ R SL EOI ... L₂ L₁ L₀

R - признак уцен-ю прер-тов,

SL - статус прер-тов

EOI - окончание прерыв-ия, сброс бита ISR

L₂, L₁, L₀ - номер низшего приоритета

ISR - бит обслуж-ия прерывания

Пример

а) 0 0 1 0 0 x x x

1 - бит прерыв-ия с данного приор-та
со сбросом одного ISR (при условии копирования Пр.)
или двух (при условии ведения и ведения КПр)

A₀ = 0 при A₇₋₁ = $\tilde{a}_7 \dots \tilde{a}_1$ - произв-ые значения

б) 1 0 1 0 0 x x x

уценка смена приор-ов

до IR₅ IR₆ IR₇ IR₀ IR₁ IR₂ IR₃ IR₄

IR - 317j

поиск сб-ки прерыв-ия

IR₆ IR₇ ... IR₃ IR₄ IR₅

в) с приор-ой уст-ой прер-ов

0 1 1 0 0 0 1 0

до IR₅ IR₆ ... IR₄

IR₃ IR₄ IR₅ IR₆ IR₇ IR₀ IR₁ IR₂
и.т.д.

OCW₃ - ESMY SMM 0 1 P

отменяет все прерыв-ия и приор-ты

ESMY - биты разрешения спущ-маски-ия

SMM - задание спущ-маски-ия

ESMY - 11 - прерыв-ия прер-ны, обраб-ка 317j
по мере поступл-ия

10 - биты записи

0x - операции не проводятся

P - разрешение способа "Поиска" (опрос),
считывание IR_R - вх-ий запросов

ISR - рег-ры обслуж-ия запросов

; програм-м путем реализ-ции системы прер-ов

В ш-ге Поиска и-но четкость SW:

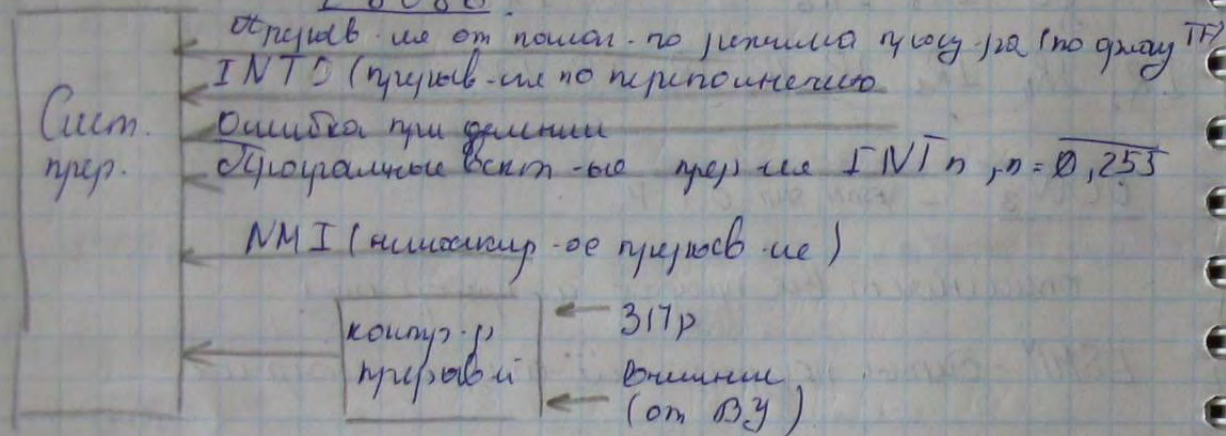
SW I < I₂ I₁ I₀

I - приоритет 317j

I₂ I₁ I₀ - номер самого прер-го запроса

Особенности системы прерываний 16-ти разрядных процессоров на примере

8086



Приоритет:

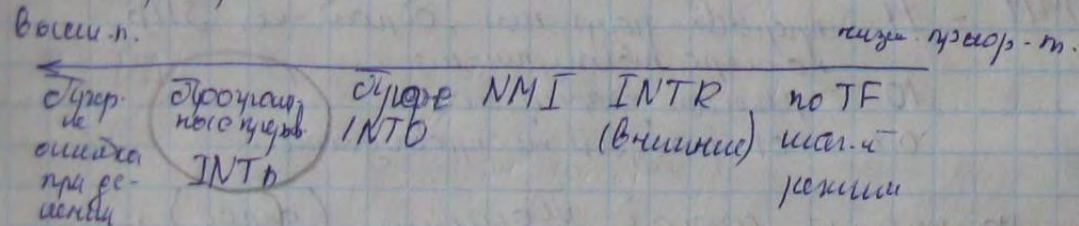
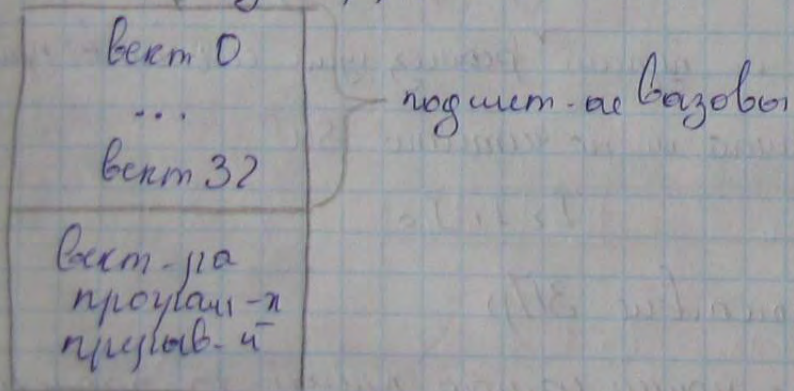


Таблица прерываний

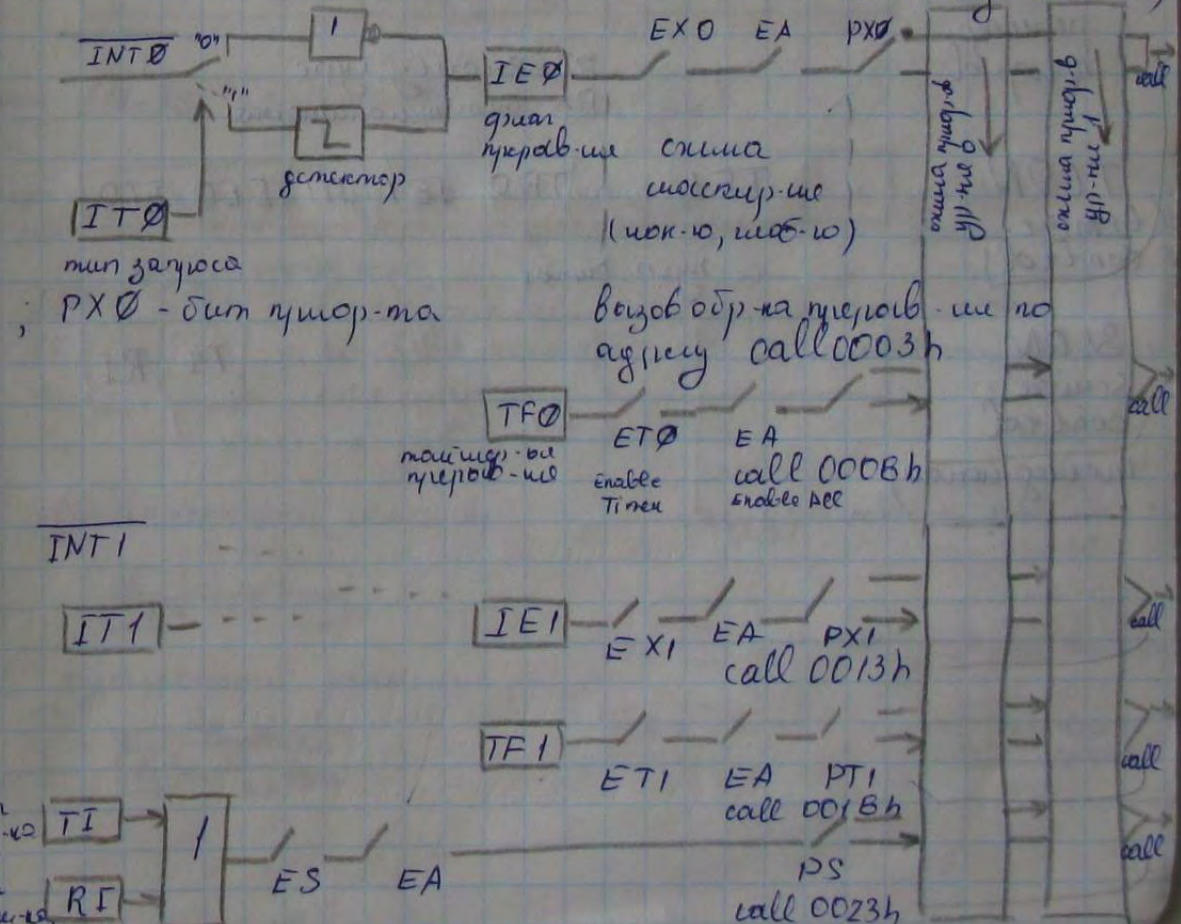


МК 80188, 80186, 80187 - 16-ти разрядные процессоры и микроконтроллеры, а также микро- и мини-компьютеры.

Система прерываний i8086-51.

(аппаратные ресурсы и программные средства).

а) аппаратная часть (взаимодействие 2-х каналов, 2-х каналов, 2-х каналов)



8) Строительные средства упр-ия

$$E_i = \begin{cases} 1 - \text{загр.} \\ 0 - \text{загр.} \end{cases} \text{ н.ч.}$$

PT - Priority Type
PX - Cumulative

пущ. Битов

SCON
serial
control
(no interrupt)

Камачи посигроватиньного
ввода / вывода.

Стерсача индустријата е бистама подготвена,
начинае индустријата, смирено биста, сојузнична,
обично, бистама контрола чистоти.

Два способа передачи:

- 1) синхр.-ая передача /прими
- 2) асинхронная передача /прими

1)

порт периферии $g-x$

синхрон-ий сигнал

serial peripheral interf

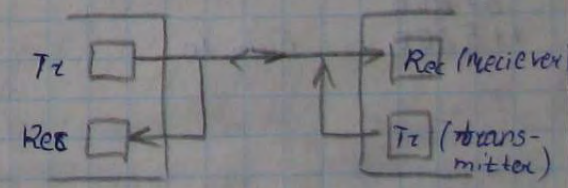
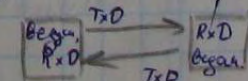
serial peripheral inters.
I2C, SPI - интерфейсы с синхр-м режимом
(2) (3) - ком-во шинный, SPI (3), т.е. ~~одно~~ двунаправ-
ная передача д-х

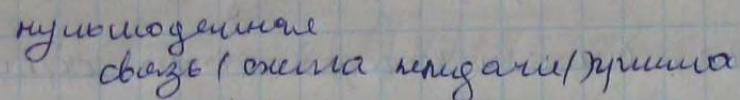
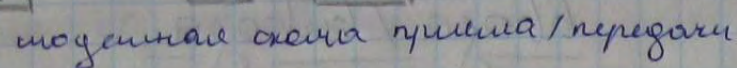
2) синтетический метод:

послудисковой жизни:

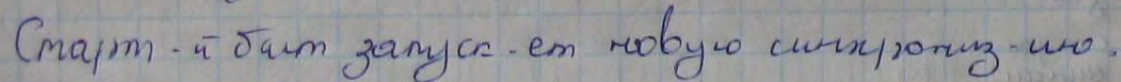


душистый, резкий:





7, 8, 9 место вступили в союз
(16, 17 место в союз) с 10 по 11
до 12

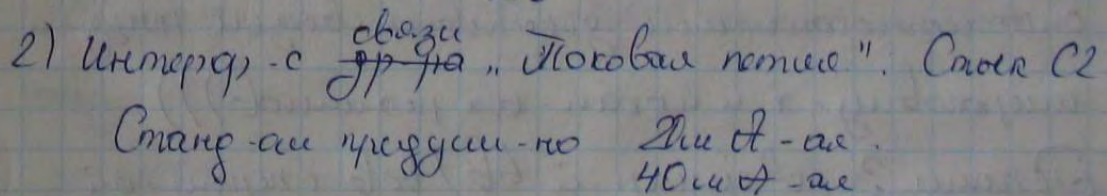


1200	бит / с	} МКБ1	Или можно сказать передачи между банком надежность!
2400			
4800			
9600			
19200	бит / с		
38400			
57800			
115200	бит / с		

Hand-drawn schematic diagram of a serial communication system. It shows a 'uik' block connected to a 'MAXIM' block via 'TxD' and 'RxD' lines. The 'MAXIM' block is connected to a 'UART' block via a 'C=15u' line. The 'UART' block is connected to a 'uor' block via 'TxD' and 'RxD' lines. A note 'change to max & uor' is written below the diagram.

$T \times D$ - типично прегради
 $R \times D$ - типично прикрити

cu. DataSheet p/
comb. 20 p/1-12



Решим: $\emptyset$ - снхр-и ввог/снхр. ввог, $S_{\text{прим.}} = \frac{S_p}{12}$ (прим. р. 1115)
 $\frac{+}{2}$ где S_p - площадь рез. пр.
 Две пластины $\emptyset R \times D$ и т.д. г/ пер. снхр. прим.



Режимы 1. Асинхронный вб/выв с переменной частотой, определенной работой таймера T1.
(примечание: 10 бит в кадре: 8 бит данных + старт и стоп).

Режимы 2. Асинхронный вб/выв с фиксированной частотой $S_P/32$ или $S_P/64$ (формат 11 бит: старт, стоп, 8 бит данных + 1 бит данных (кадр, бит четности или десятичная дробь (разряды шестнадцатеричных и двоичных режимов)))

Режимы 3. Асинхронный вб/выв с переменной частотой, определенной работой T1 (11 бит формат)

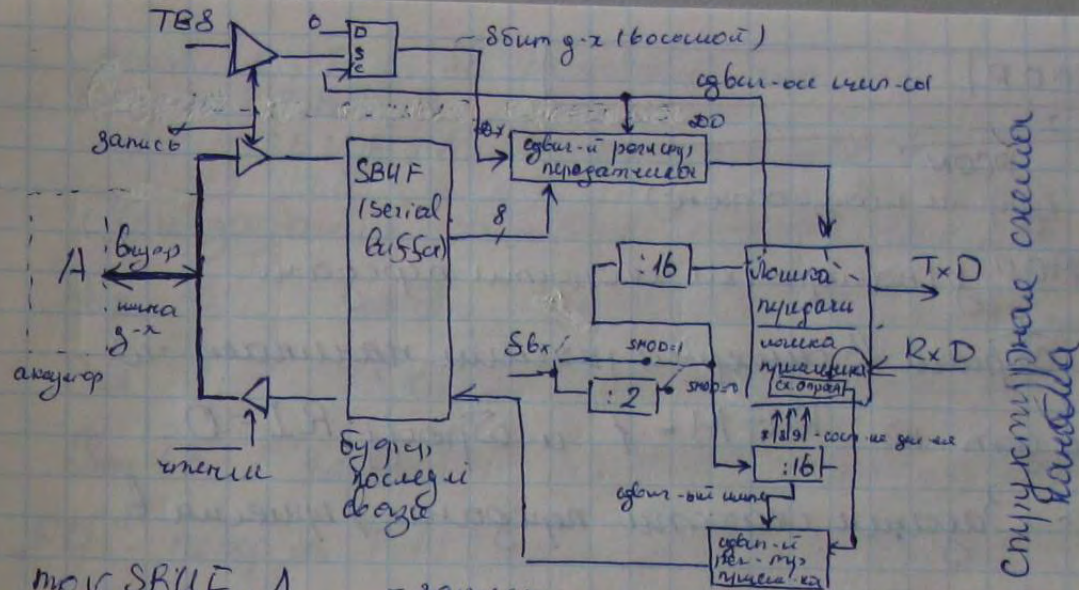
Для всех режимов:



где $S_{ov_{T1}}$ - частота переключения T1

$$S_{переб} = \frac{S_P \cdot 2^{SMOD}}{12 K_T \cdot 32}$$

K_T - K-м делитель таймера T1



mov SBUF, A - запись

mov A, SBUF - чтение

всего 11 бит передатчика 8 бит данных + старт.

при передаче передатчик $\rightarrow R_T \rightarrow$ значение, записанное в таймер,

состояние отсчета с частотой 7,8,9 отсчетов - 11 бит получены.

Работа:

Упр слова SMOD (serial mod)

SM0	SM1	SM2	REM	TB8	RBS	TJ	RI
-----	-----	-----	-----	-----	-----	----	----

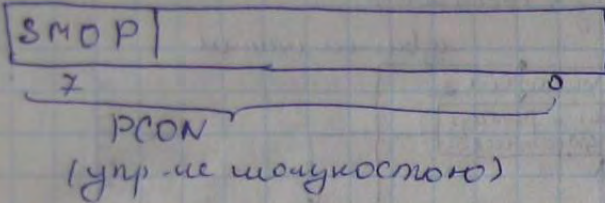
режимы работы:
00 - режим 0
01 - режим 1
10 - режим 2
11 - режим 3

SM2=1 - запрещен прием данных с RS=0

REM - разрешен прием данных

TB8 - 8-й бит 11-битного формата (передатчик)
RBS - 8-й бит 11-битного формата (приемник)

Сигналы приема сигнала



SMOD 8/настройки скорости передачи

Прием в синхрон. режиме начинается по уст-ке REM=1 и сбросу RI=0.

Задаем скорость передачи/приема в асинхронном режиме.

$$S_{заг} = 2400 \frac{\text{бит}}{\text{с}} \cdot \frac{1}{S_p} \quad \text{1 бит SMOD=0}$$

$$2400 = \frac{11059 \cdot 10^6 \text{ Гц}}{12 \cdot K_1 \cdot 32} \cdot 2^{\text{SMOD}}$$

$$K_1 = 12$$

Время T1 = 2 с перезагрузки:

$$\frac{100 \text{ нс}}{F_4} = \text{в TH1}$$

ANL PCON, #7Fh ; стираем SMOD=0
CLR TRI ;
mov TMOD, #00100000h ; режим 2
mov TH1, #0F4h
setb TRI ; включим таймер

Для программ. а бита читности (в 11-битовом) исх. байт в A кк:

ADD A, #0 ; "P" в PSW

mov C, P ; в битовом рег.
mov TB8, C
mov SBUF, A ; начало передачи

При задании частоты передачи/приема:

Рех. 0 1 МГц Sp=12 МГц, Smod = x - произв.-ое RI

Рех. 2 375 кГц Sp=12 МГц, Smod=1

Рех. 1,3 62,5 кГц Sp=12 МГц, Smod=1, "FF"

"FD" - коэф-т TH1 (таймера 1)

P 19,2 кГц Sp=11059 Гц, Smod=1, "FD"

9,6 Smod=0, "FD" - 3

4,8 "FA" - 3

2,4 "F4"

1,2 "F8" RI, 24

Программирование передатчика и приемника
(для передачи) на линии P3.1 (TxD) (для приема) на линии P3.0 (Rx) RI

орд 30h ; из внутр. памяти XSEG орд 30h
MOV DPTR, #1000h ; нач. адр. код-а MOV DPTR, #2000h
MOV R0, #32

mov psw, #0 ; рег-и байт=0
clr EA ; запрет всех прерыв-ий

ORL PCON, #80h ; SMOD=1

ORL P3, #02h ; стоп линия

MOV SCON, #010xxx1xB ; режим ; 10 бит-и, TI=1

CLR TRI ;

MOV TMOD, #00100000B

MOV TH1, #0FDh

SETB TRI

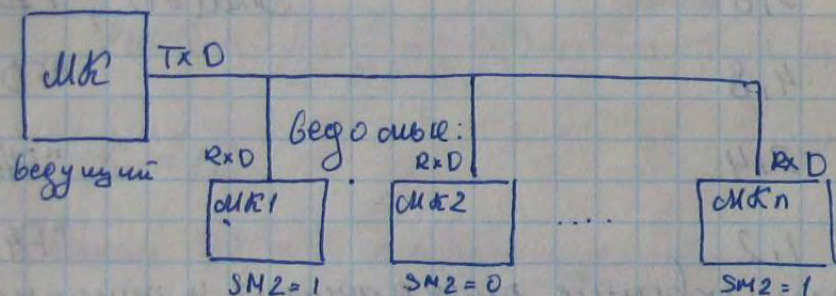
loop: movx A, @DPTR ; символ з/передачи ; JNB RI, loop
JNB TI, \$; ожидание пожел-ия CLR RI

MOV R0, #32 ; 32 символа

CLR TI	MOV A, SBUF; значение з-е
MOV SBUF, A	MOVX @DPTR, A; копирование
inc DPTR	inc DPTR
DJNE R0, loop	DJNZ R0, loop
JNB TI, \$	end
end	

Точно, но упр-е [↑] вывод и ввод, но лучше
знать с пом-ю призываний, цикличес-
кую вып-ю с пом-ю процедуры.

Сетевое использование канала UART.



1) "Микроковену - и", 11-битовый формат с

$$TBS = 1$$

Усл. соот-ие $SM2=1$ $g/UK(1 \div N)$

CTBS MKO передает адрес абон-та

$SMZ=0$ при совпадении адресов вер-20

МК, данные при $TBS=0$ пред-е только

и $\mathcal{H} \subset \mathcal{H}_2 = 0$; заменим $\mathcal{H}_2 = 1$ по-новому.

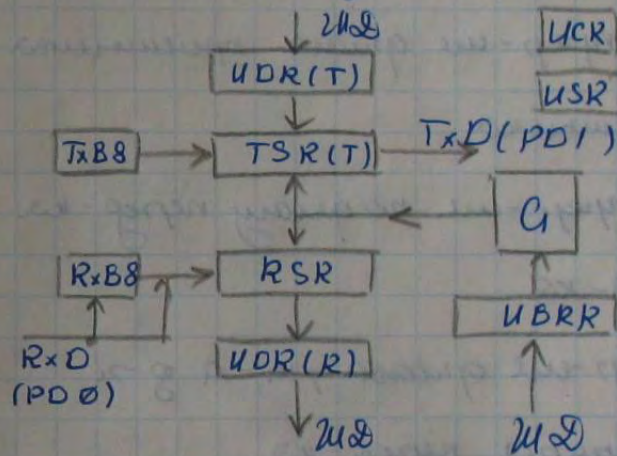
Особенности каналов UART (USART) AVR

USART раб-т в сем-бе Мега.

UART "Begebe"

$UDR(T)$ - при- ту, гачных пере- ка

TSR (T) - $\gamma_{\text{виз-и}}$ перископ перес-ка



USR - per - mp ynp - me

УСР - пер-ту, ост-ли
пачинае

RSR - cобуз-и jес-мы
нр-ка

UDR(R) - пер-тр данных
пути и чикой

а - генератор

UBRR - per - p Zagadmo
tactom

Тх В8-гон-и бум

RxB8 - ^{переходящий} год - и ^{принимая} тут

Эти присаженные данные пом-т в рег-тр ф-х
и если он пуст, то перед-ие сразу по-соби-т
рег-тр. У рег-тр ф-х есть флаж, кот-й говорит
пуст он или нет.

Если нет, то 9-е соуд, - и в нем. Тогда
поддерж - и 2-ие признами. ТхС (орна
нехватки (сбиз-роп-ра)), и DRE - том-ть

per-ia g-x UDR.

TxC ит-е g/ канализации популяционной
связи.

UCR	$RxCIE^*$	$TxCIE$	$UDRIE$	$RxEN$	$TxEN$	$RxB8$	$TxB8$
USR	RxC^*	TxC	$UDRE$	FE	OR		

$RxCIE$ - разр-ие прер-ие сигнала приемника

RxC - сигнал приемника

$TxCIE$ - разр-ие прер-ие по каналу перер-ка

TxC - сигнал перер-ка

$UDRIE$ - разр-е прер-ие сигнала пер-а g-2
перер-ка

$UDRE$ - пер-тр сигнала перер-ка

$RxEN$ - разр-е работы приемника

$TxEN$ - разр-е работы перер-ка

$RxB8$ - зон-й бит (сигнал-и при перер-и)
(напр-я, бит сигнала)

$TxB8$ - зон-й бит (запис-е при перер-и)

FE - ошибка при приеме

OR - ошибка,
- обнаружение при перер-ке.

$FE = 1$, если на шесте стопового бита 0.

Второй бит OR, связан с тем, что если мы во время не прочитали информацию возникает переполнение, сбрасывается при следующем прочтении.

Скорость (UBRR) приема / передачи:

$$f_{\text{приема/передачи}} = \frac{f_{\text{CLK}}}{16(UBRR + 1)}$$

Диапазоны: 4800, 9600, 14400, 19200,
28800, 38400, 57600, 76800, 115200 б/с
(два ряда, определяют стандартные частоты
приема / передачи).

В микроконтроллере МEGA в составе регистра управления есть бит к-й и бит сброса. В работе микроконтроллерной сети, работает по принципу похожему на работу соответствующей AVR.

Интерфейсы последовательной связи SPI в AVR.

Основные характеристики.

Полудуплекс (или дуплекс) режим:
прием и передача одновременно
синхронный 3-х проводной (3-е линии -
синхронизация)

Может работать как Master и как Slave.

Скорость передачи: от 100 кбит/с до 10 Мбит/с.
Также внутри при 16 разномощности.

4 скорости передачи данных:

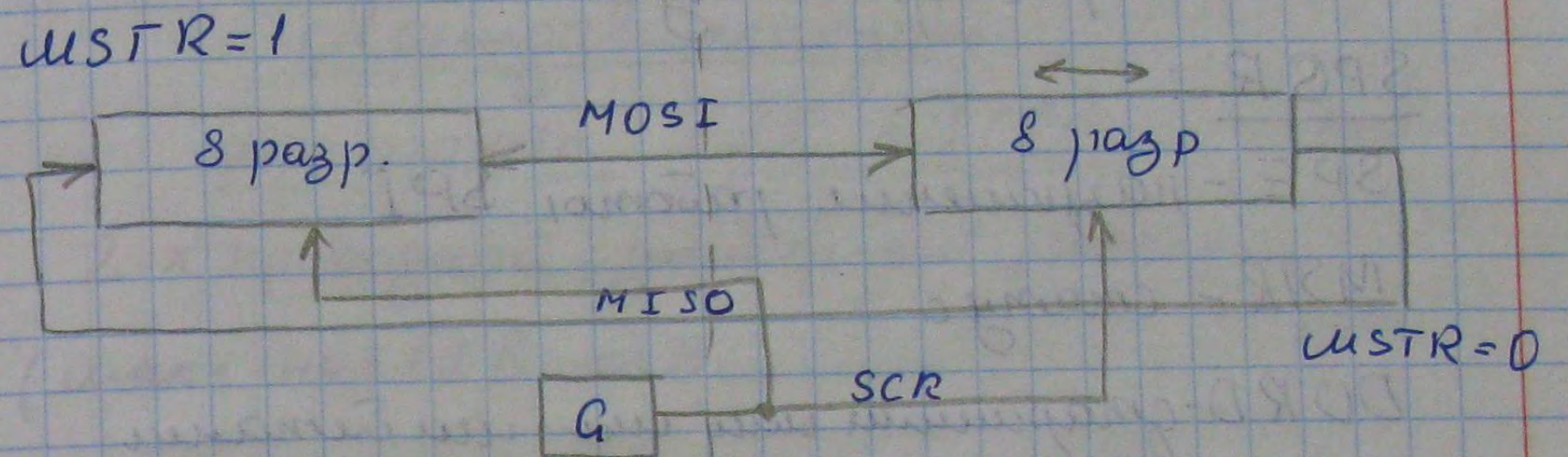
$\frac{f_{clk}}{4}$; $\frac{f_{clk}}{16}$; $\frac{f_{clk}}{64}$; $\frac{f_{clk}}{128}$

Пример: ATx8515, 8 МГц

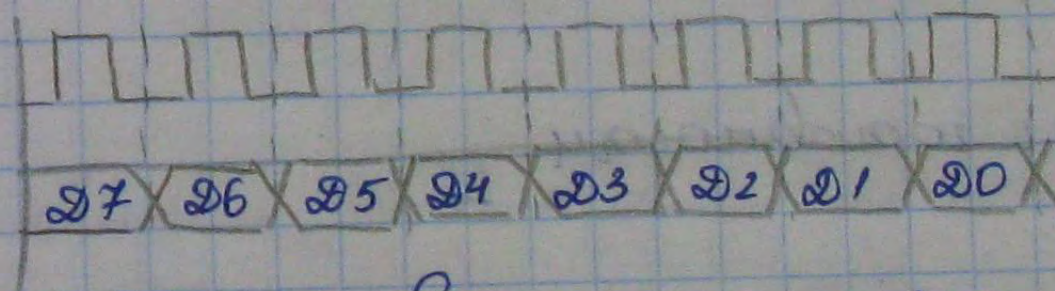
$f_{max} = 2 \text{ МГц}$

Можно активировать работу по умолчанию
из режима пониженного энергопотребления
IDLE - в режиме МК.

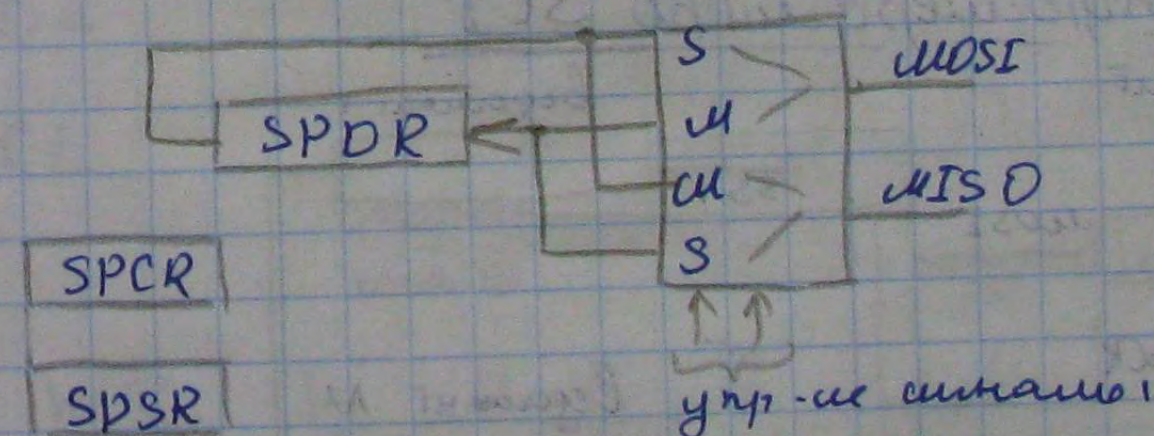
Структура интерфейса.



MOSI Master Out Slave In
MISO Master In Slave Out
SCR - шинный сброс
 $\overline{G}$ - генератор импульсов
SCR - шинный сброс в Master и в Slave
MSTR - бит мастера



Структура (ресурсы) канала.



Способ управления инициализации и работы.

SPDR - регистр данных
посылка и прием данных

SPCR - jci-my ynp-ur
SPSR - jru-my manyca

SPE - планирование работы, SPI

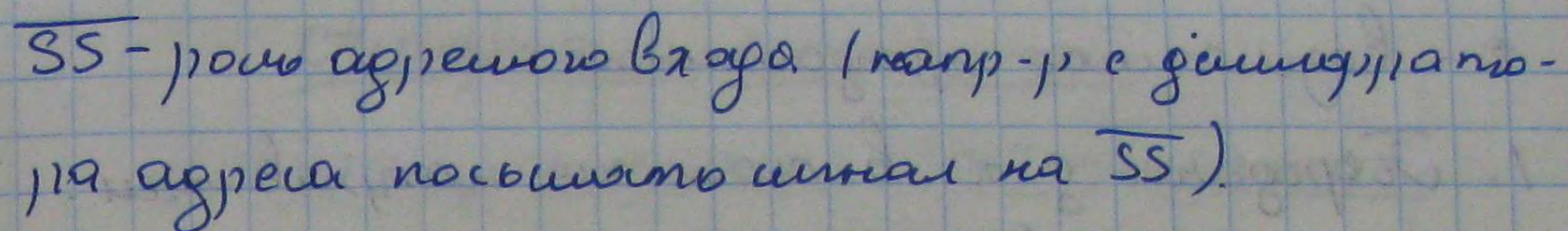
DVRD-старшими или или-или Ботани

SPRO: SPR1 - стоимость неуправли

SPSR :

Работа с числовыми

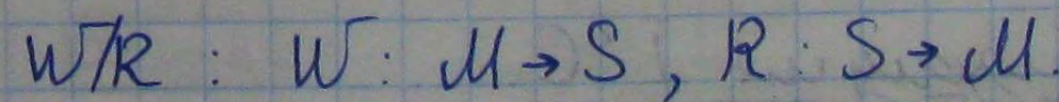
1 веруиши



2-х проводной, универ-ый, работа с
(макс-но) 128 ЛУ.

Hand-drawn schematic diagram of an I2C bus. It shows multiple slave devices labeled $Y_1, \dots, Y_N$ connected to a common SDA (data) line and a common SCL (clock) line. The SCL line is connected to a pull-up resistor network consisting of two resistors R in series, connected to VCC (labeled "питание").

Цитоплазматические ВД:

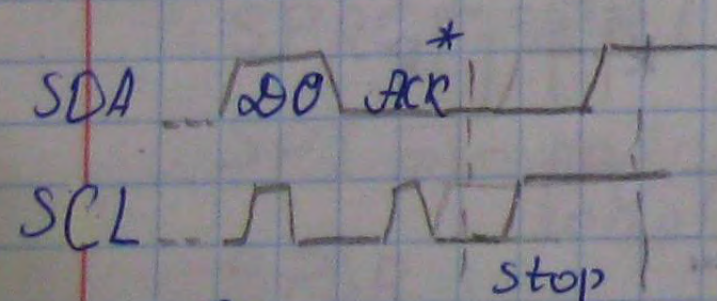


АСК - подтверждение того, что устаревшие адреса
есть.

Если $ACK=0$, то передаем след-ий

* - возможное продолжение:

1. Передача 8-х в том же направлении.
2. Выбор др-го ведомого или смена направления по отношению к тому же Masterу (или и тому и другому) и выдача др-го нового адресного байта
3. Окончание отклика с освобождением шины и переход в состояние Stop



Средняя скорость:

$$BR = \frac{f_{clk}}{16 + 2(TWBR) + (1.6 \div 4.8)}$$

где TWBR - к-т в рег-ре скорости;

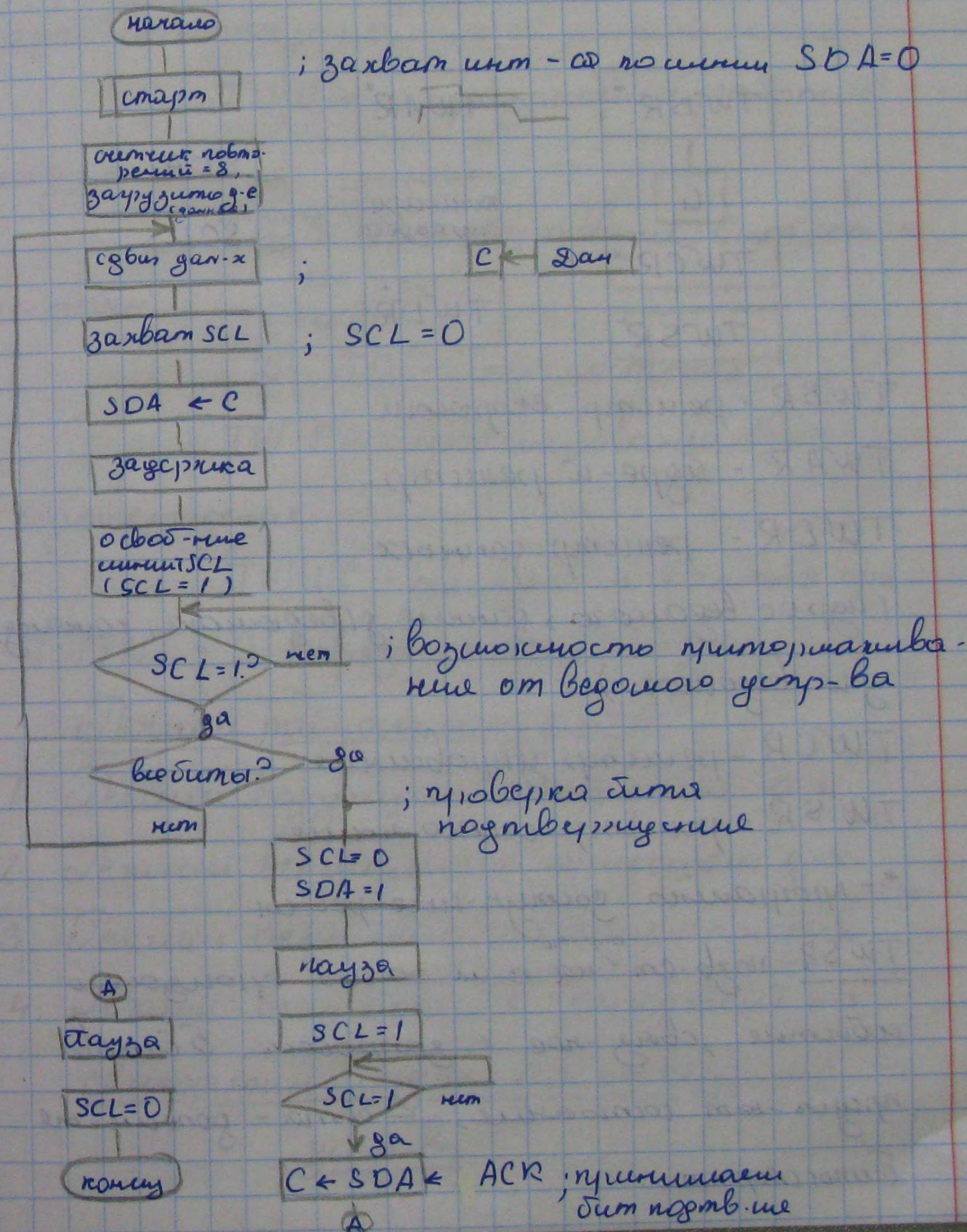
$$BR = \frac{8 \text{ (МГц)}}{16 + 2 \cdot 8 + 1.6} = 0.24 \text{ МГц}$$

рек-ое TWBR > 7

Реализация I²C.

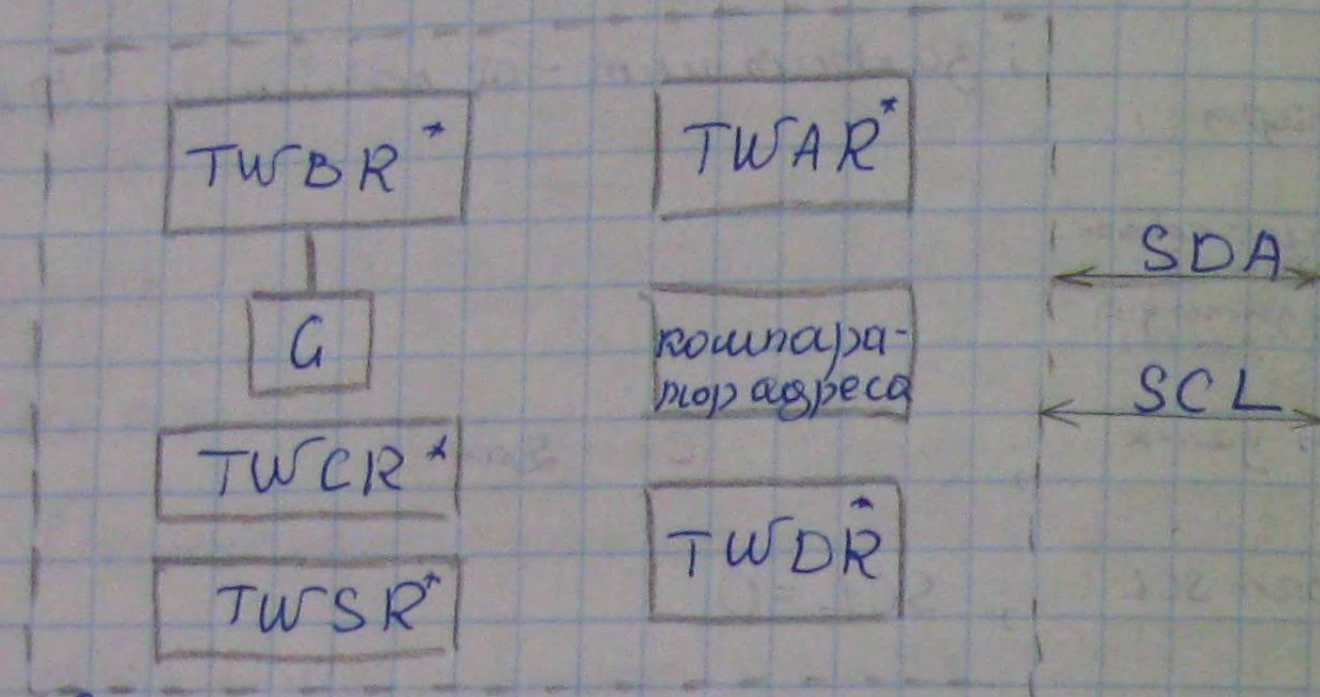
1. программная (при отсут-вии канала)
2. используя аппарат-ые сред-ва канала TWI.

Алгоритм программной транзакции записи байта данных.



Модуль TWI (I2C).

Структура модуля.



TWBR - регистр скорости.

TWAR - адрес-и регистр.

TWDR - регистр данных.

(адрес ведомого, данные от ведомого, команды для ведомого).

TWCR - регистр управления.

TWSR - регистр состояния.

* - программно доступные ресурсы.

TWSR позволяет на каждом этапе производства события, связанное с запросом. 5 битов присутствуют в состоянии, 2 бита - готовые биты.

TWCR: TWINT - флаг (запрос) прерывания. Для сброса ист.-е TWINT = 1, после TWINT вычит-е аппаратно.

TWEN - разрешение работы модуля.

TWSTA - стартовый бит, ст.-е состояние.

TWSTO - стоп состояние.

TWEA - разрешение бита подтверждения.

TWWS - при записи данных при TWINT = 0 (связан с ошибкой).

Режимы работы модуля.

1. Режим ведущего с передачей д-х
2. Режим ведущего с приемом д-х
3. Режим ведомого с приемом д-х
4. Режим ведомого с передачей д-х

Алгоритм передачи д-х от ведущего устр-ва.

Шаг 1. $TWSTA, TWEN, TWINT$ (выполняет роль старта). Автоматически формируется захват ит-са и форми-се $SDA=0$

Шаг 2. Проверка состояния линии проверки кода состояния

Вывести адрес в $TWDR$ по адресу, в $TWINT$ пометить "1". Проверка сов-ия.

Шаг 3. В $TWDR$ загрузить $g-e$, $TWINT$ в "1". Проверка.

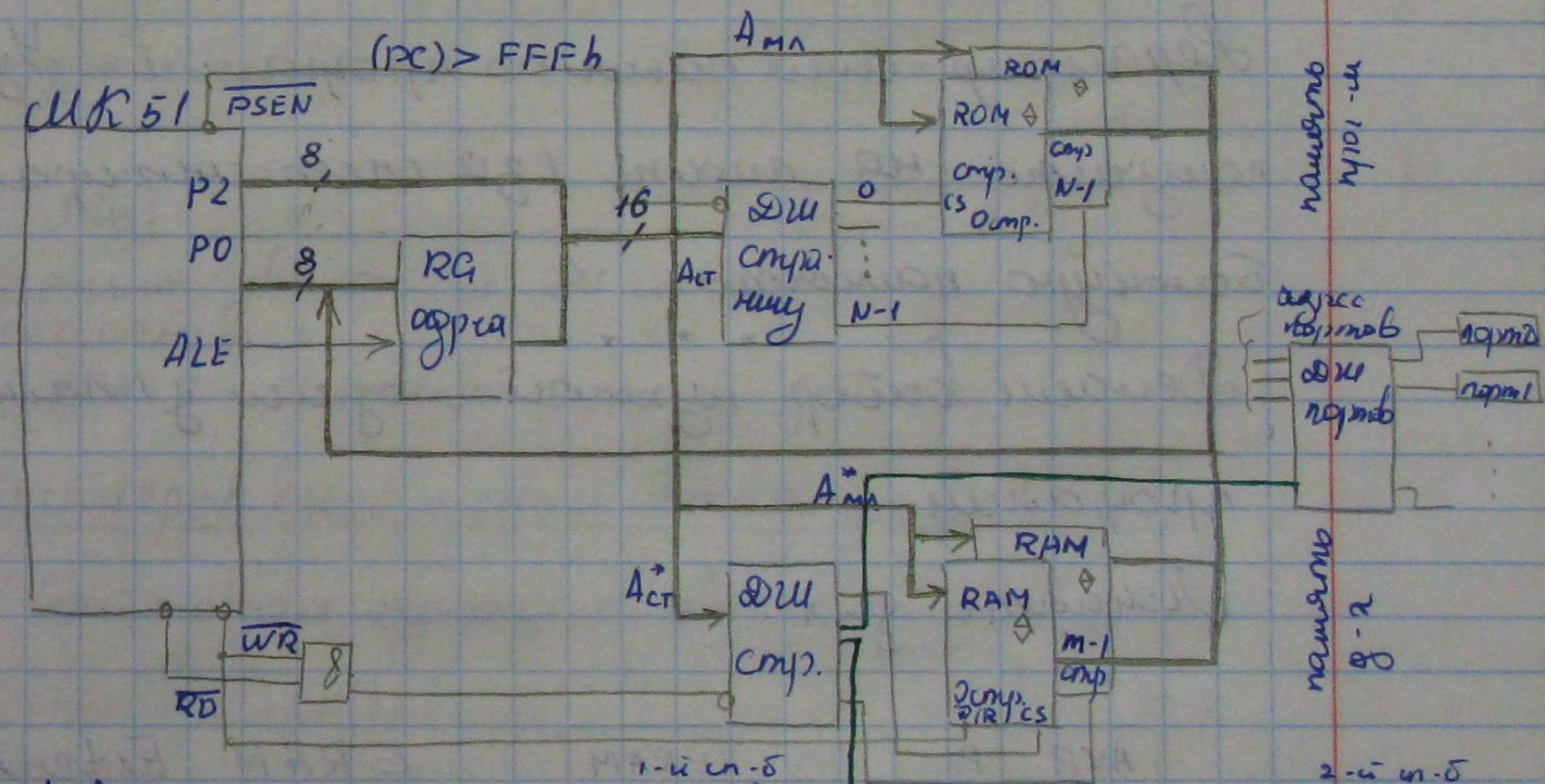
Шаг 4. Пометить бит стоп, $TWSTO$, $TWINT$ в "1".

* Основные прог-ры в учеб-м пособии.

Все 4-ре прог-ры написаны на работу с 2-м МК. Другой подход; реализ-то аппаратное констр-е ведомых контр-ов $\Rightarrow$ необх-но прог-то только ведущий контр-ер, также исл-ие ПЛД ведомых контр-ов (пар-но прог-шам ит-са).

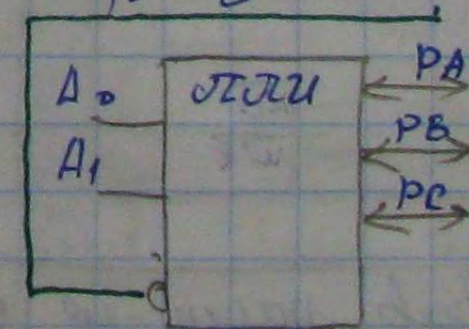
Организация микроконтроллерной системы на базе MCS-51.

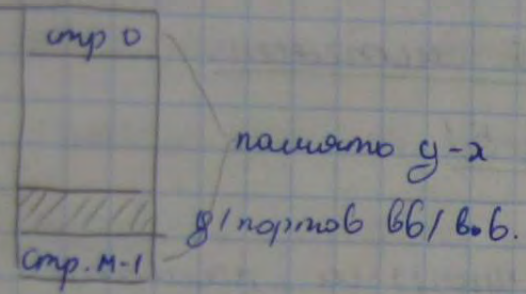
Расширение памяти $4K$ $128-256B$, $g-n$, портовой системы (4 порта)
 При этом $64K$ ПЛД, $64K$ ПД
 ПС расшир-се за счет ПД.



$A_{ст}^* A_{мл}^*$ - сов-ое при-ра (DPTR) $-16B$ или P2 - номер стр-ы, $(DPH: DRL) - 8:8B$ (KOLVR) - адр. адр. $8B$ стр-ы
 $PSEN$ - разрешение обращения к бчм. памяти

ALE - строб адреса
 CS - Chip Select





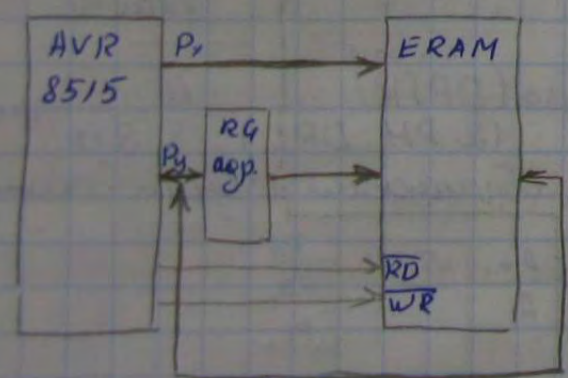
Исп-ие 8-и разн-ых портов ДШ портов и ПЛУИ (см. выше).

Пример: МК AVR.

Автоматизированная память программы. Удобен поучить на плате (за счет интеграции) базовую память.

Делаем байтер нужной модели 8-и памяти программы.

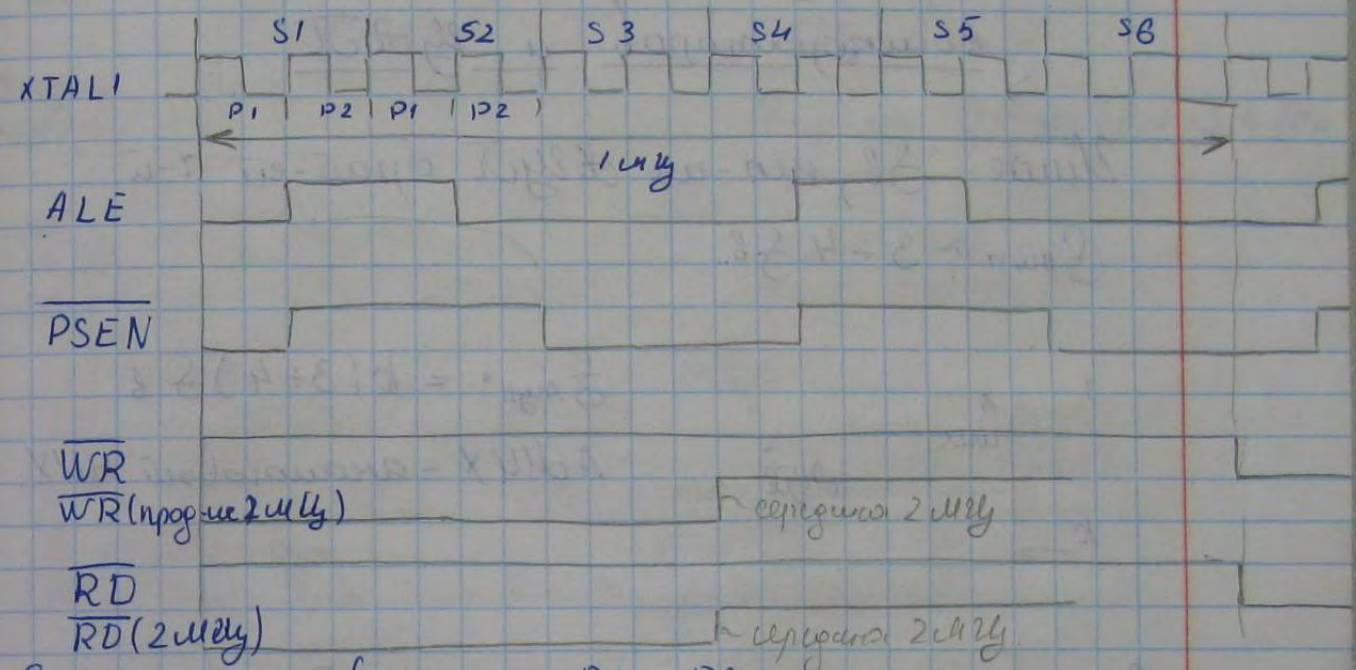
Память 8-и:



ERAM - External RAM.

Порты вб/воб. разн-ые отобраз-ен на адр. протр-во памяти данных.

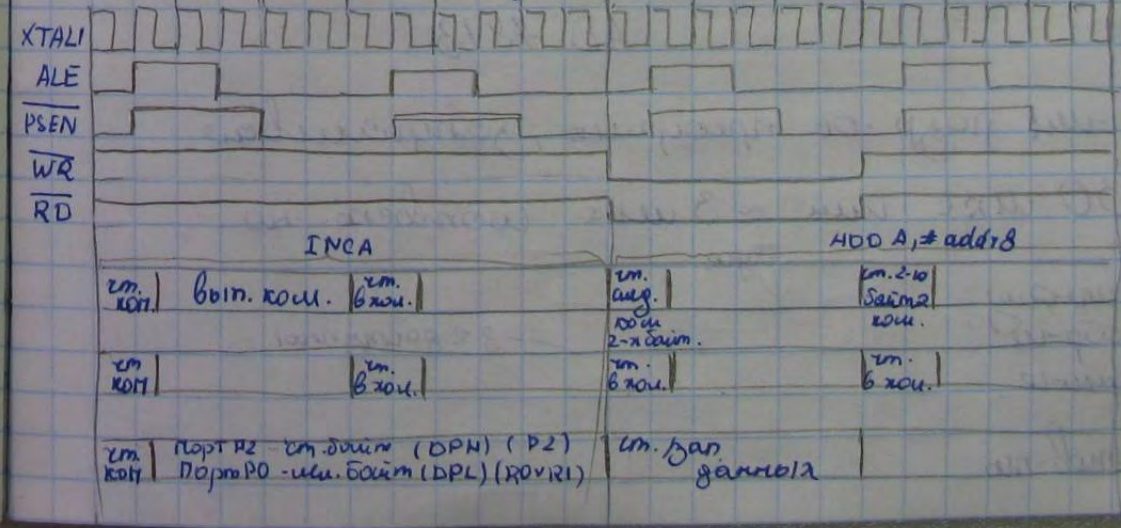
ВД сигналов управление на интерфейсе МК51.



S_i содержит две фазы P1 и P2,
 S_i - состояние i.
 XTALI, XTAL2 - сигналы синхронизм

Примеры выполнения команд МК51.

1-байтовая арифметическая INC A.

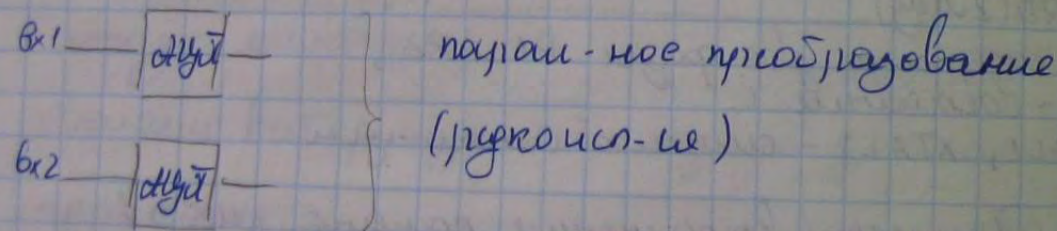
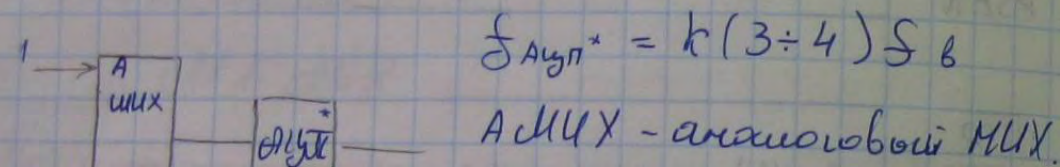


INC DTR

1-й и 2-й унк-ые кон-га MOVX

Обработка аналоговых сигналов с помощью АЦП, аналоговых компараторов и ЦАП.

Число $\Sigma \text{в}$, исп-ое АЦП с раз-ей n -й
 $\Sigma_{\text{АП}} \approx 3 \div 4 \Sigma \text{в}$.

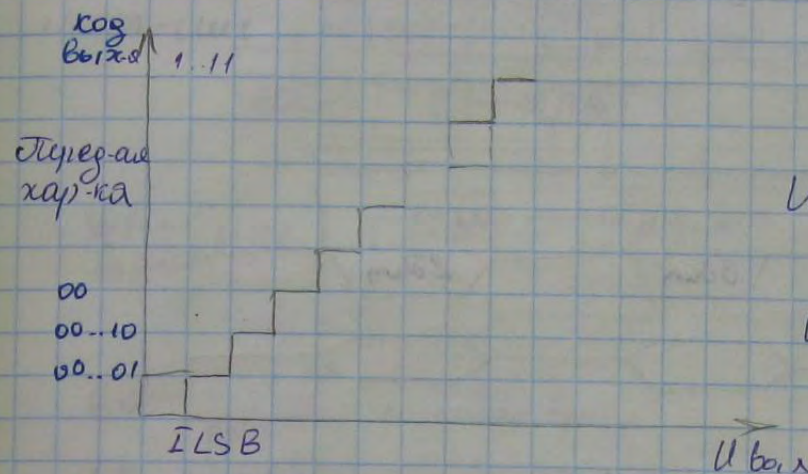


На примере АЦП К1113ПВ1 или
AD 7813

8-ми разр-ое преобр-ие, затрачивая
30 мкс или ≈ 3 мкс соответ-но.



Применение АЦП в МК-системах.



AD 7813

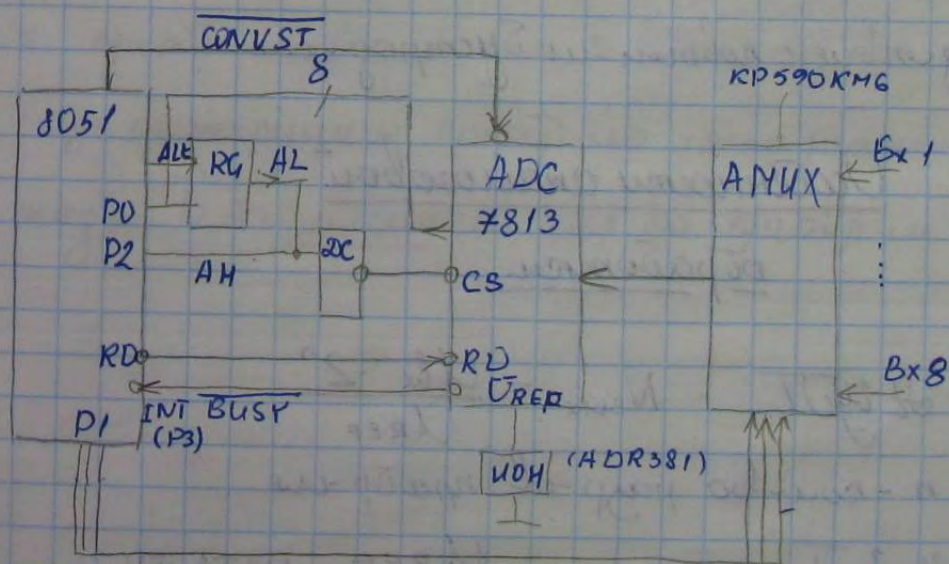
8-и разр. бита
10-и разр. бита

$$I_{\text{LSB}} = \frac{V_{\text{REF}}}{1024}$$

I_{LSB} - цена дискрета

$$V_{\text{REF}} = I_{\text{LSB}}$$

Схема включения AD 7813 в систему
МК 51

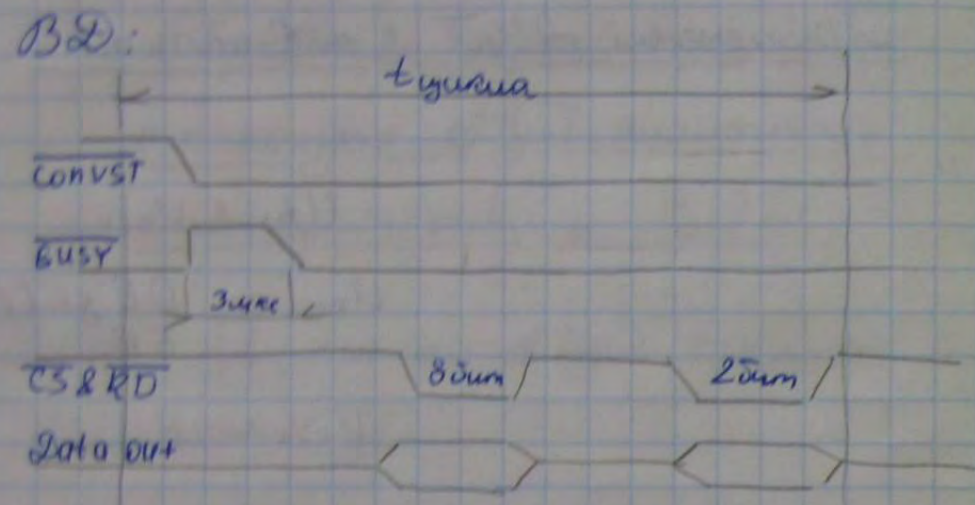


V_{REF} - опорн-й сигнал

ALE-сигнал
CONVST-сигнал
преобразование

BUSY-сигнал
готовности
ищ-м-к нагр-е
сигнала

Наимен-е
11-го бит-а
связи



Режимы работы: 1) 8-битный и 10-битный режимы (однозначный или цифровой (авт.-кал-б-е) преобразов.)

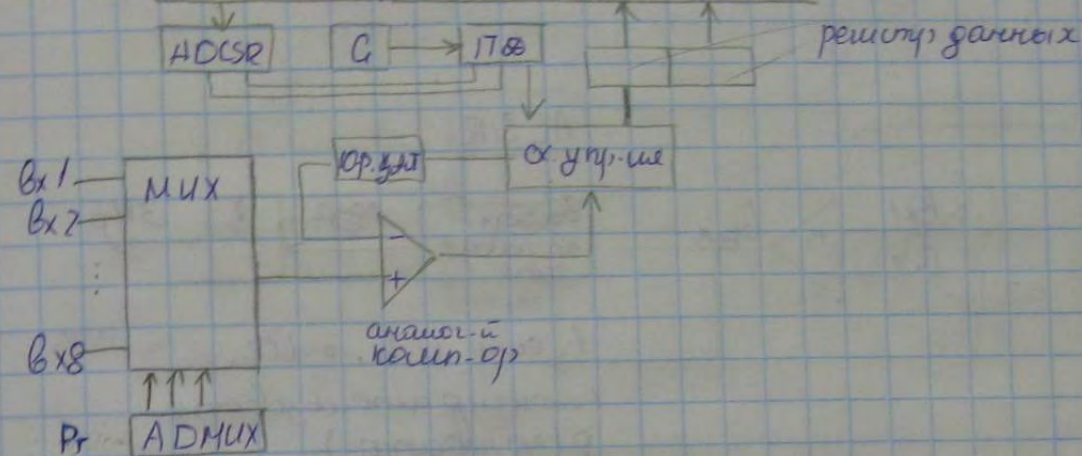
Еще один МК: ADμ8215 с АЦП 5мкс, входит в серию AD7813, авт.-кал-б-е и интегрирующ., и выдает в 10-битном формате.

Особенности аналоговой обработки

Для АЦП $N_{вых} = \frac{U_{вх} \cdot 2^n}{U_{REF}}$
 n - кол-во разрядов преобразов.

Для ЦАП (AD8303) $N_{вых} = \frac{U_{REF}}{2^n} \cdot DATA_{IN}$

Заключение. 7 схем АЦП с группировкой по числу битов, напр-р SPI и др.



ADCSR - рег-тр управ-ия част-ой част-ой АЦП
 ЦАП - преобразователь;

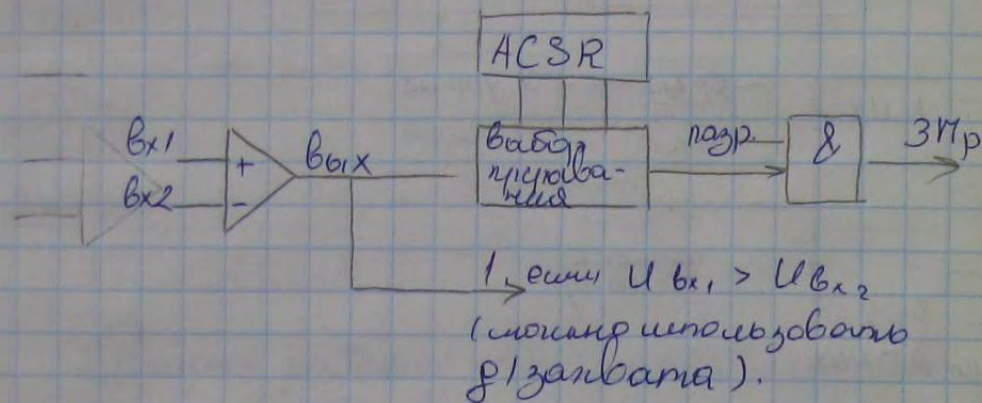
$$S_{анал.} = 50 \dots 200 \text{ кГц}$$

↓
5 мкс

При 5 мкс треб-е примерно 65 мкс на преобраз-е, т.к. преобр-е длится ≈ 13 тактов.

; Еще один режим преобр-е - это режим хвостового кода, в этом случае как-то преобр-е выше.

Аналоговый компаратор (в составе МК AVR).



ACSR - analog control start register

Выбор прерывания можно осуществлять по изменению сигнала (событию) $\square, \square, \square, \square$

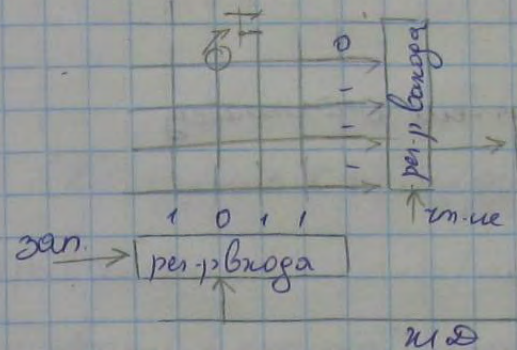
Взаимодействие МК-читингов с оператором

состав
пульта
оператора

{ кнопки, переключатели и клавиша (орудийная)
дисплеи: 7-ми сегментные, ЖК и матричные

Пример: для отсчета отсчета дуплекс-а клавиша
должна иметь клавиши 0...F, ADDR, STEP, RUN,
RT, REG (если хотим проинтерпретировать). Всего 21 клавиша +
+BREAK.

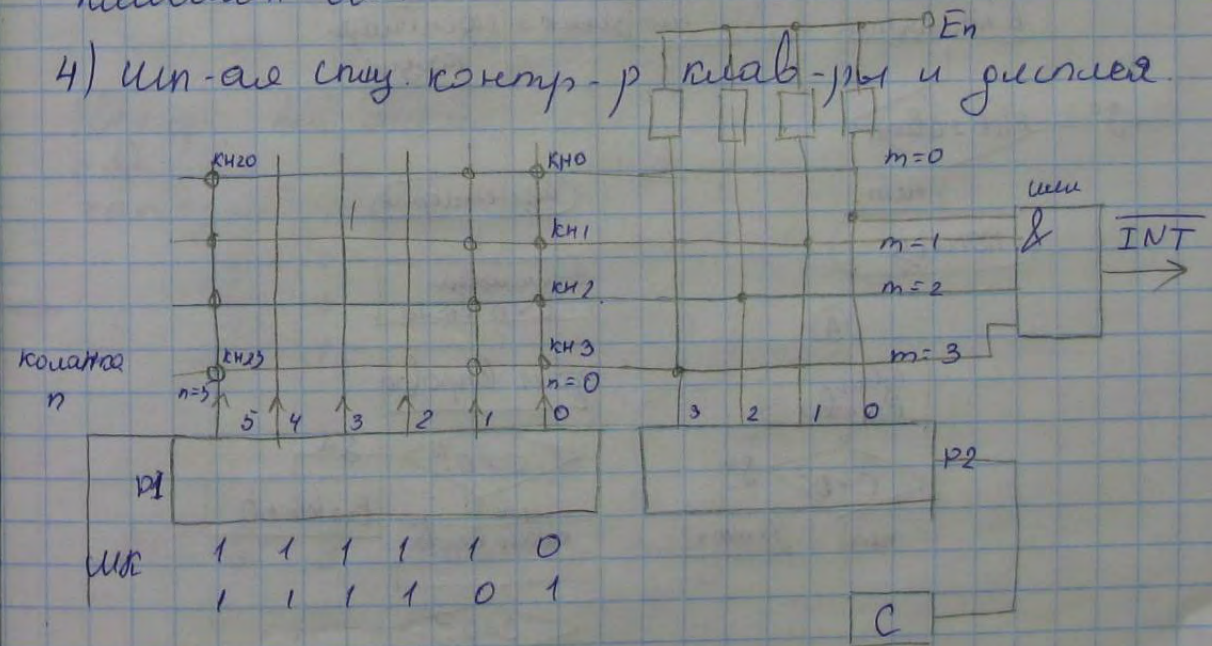
1. Взаимодействие с клавиатурой.
1) Взаимодействие с клавиатурой, ип-ае для подключения
клав-рог. разнотр. ввода и разнотр. вывода.

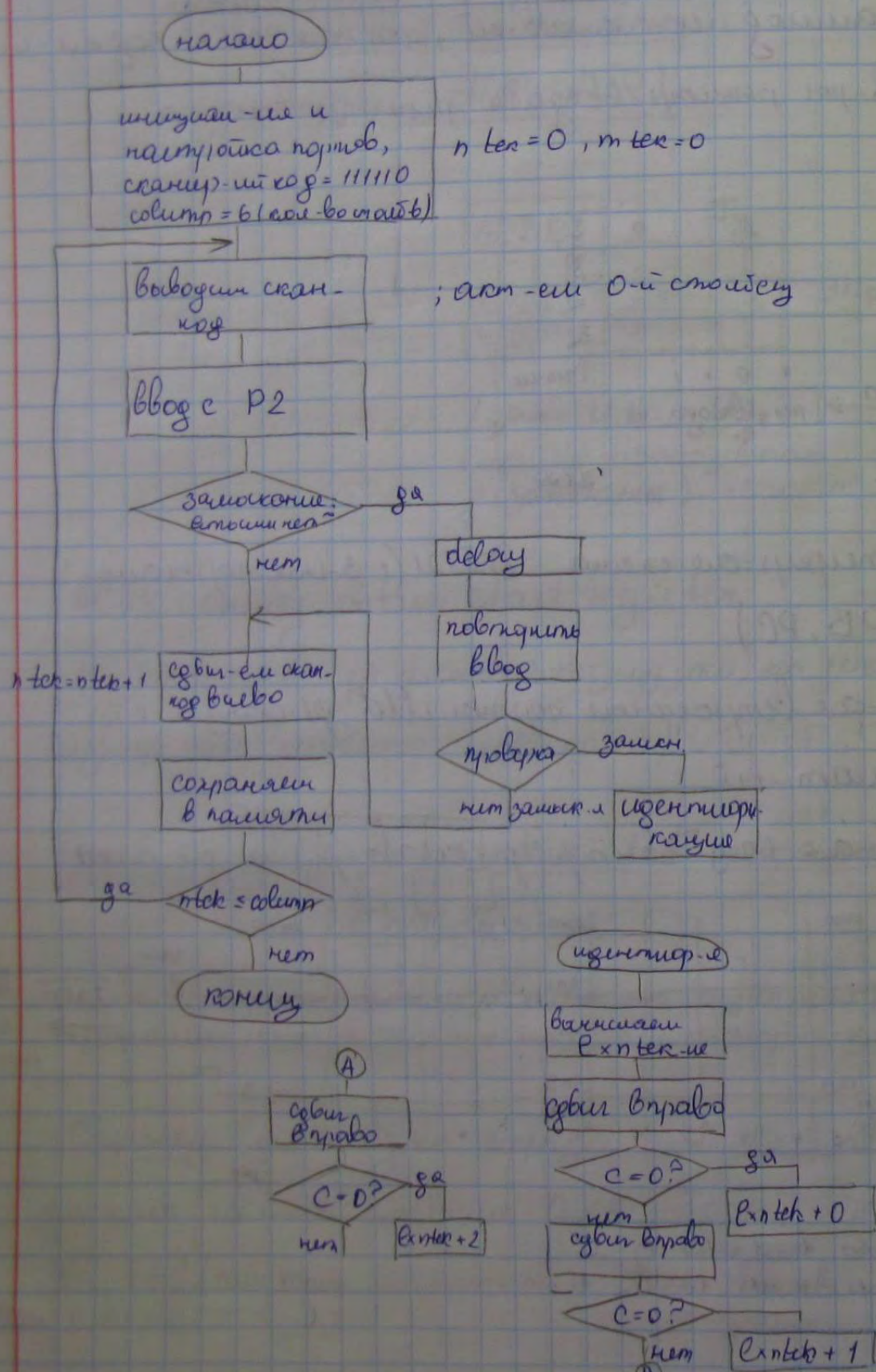


2) ип-ае смена $\square$ (с 3-ми портами
РА, РВ, РС).

3) ип-ае вступлении портов МК с связи с
клавиатурой.

4) ип-ае спец. контр-р клав-ры и дисплея.





Поме идент-ции, имея номер кнопки:

номер кнопки

Поминица

код с функ-ии стр-ем

stop

вып-ие
качанг

выполн-ие по:

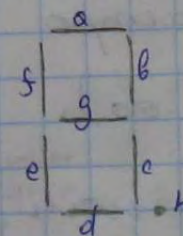
Обращение к прайс-листу

1) прерыванию.

2) прерыванию по таймеру с интервалом $\approx 70 \text{ мс}$.

2. Дисплейная часть.

1) Семисегментные индикаторы.



h g f e d c b a

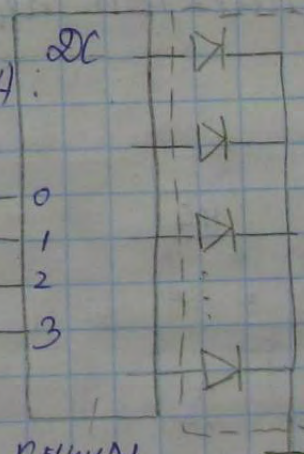
АЛС 314А

Упр-ор АЛС 314А:

с общ-им
катодом

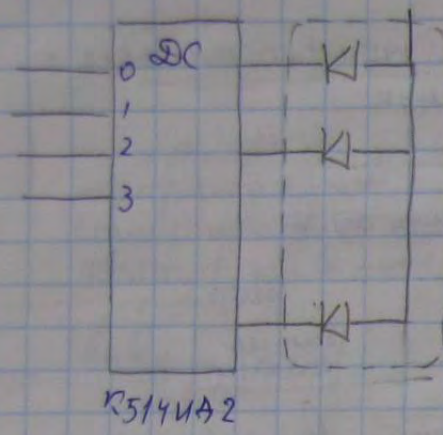
упр-й
код

0
1
2
3



АЛС 314А

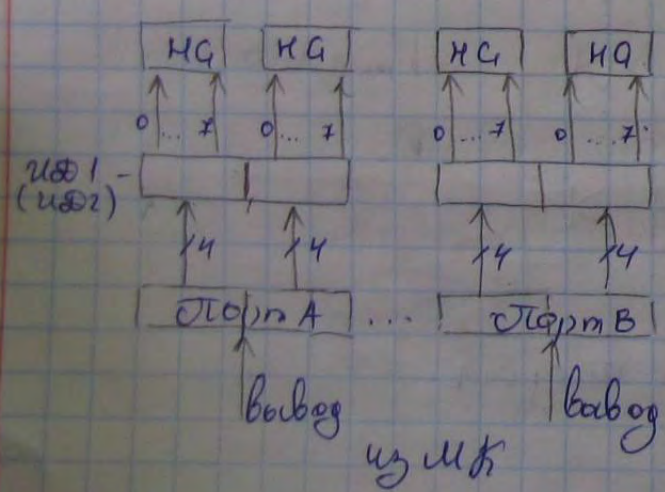
АЛС 324 В с общим анодом:



Статический и динамический способы управления дисплеями.

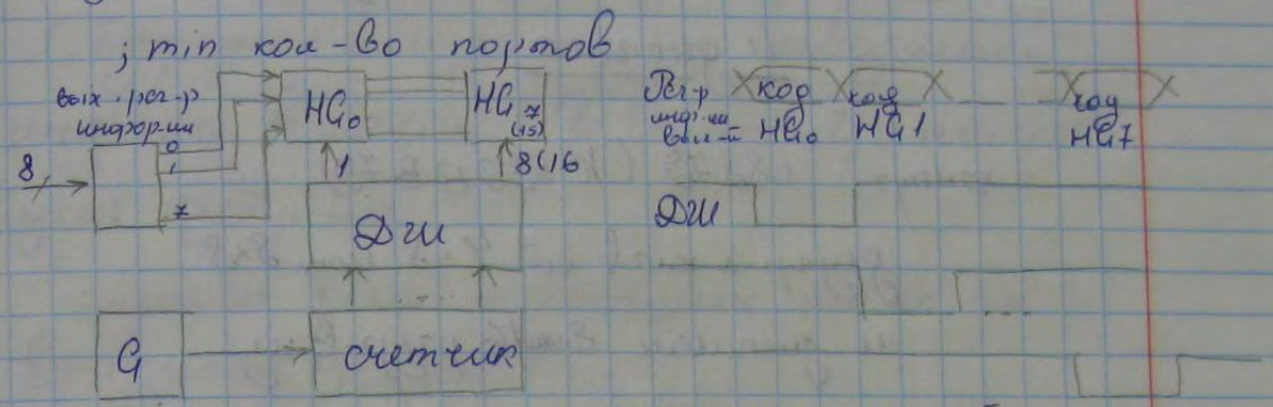
а) статический

Адресат-к: код-во оборуд-ия, кот. должно разгр-се с увелич-м код-во. НС (инд-ов), такую схему приме-ют при небольшом числе НС.



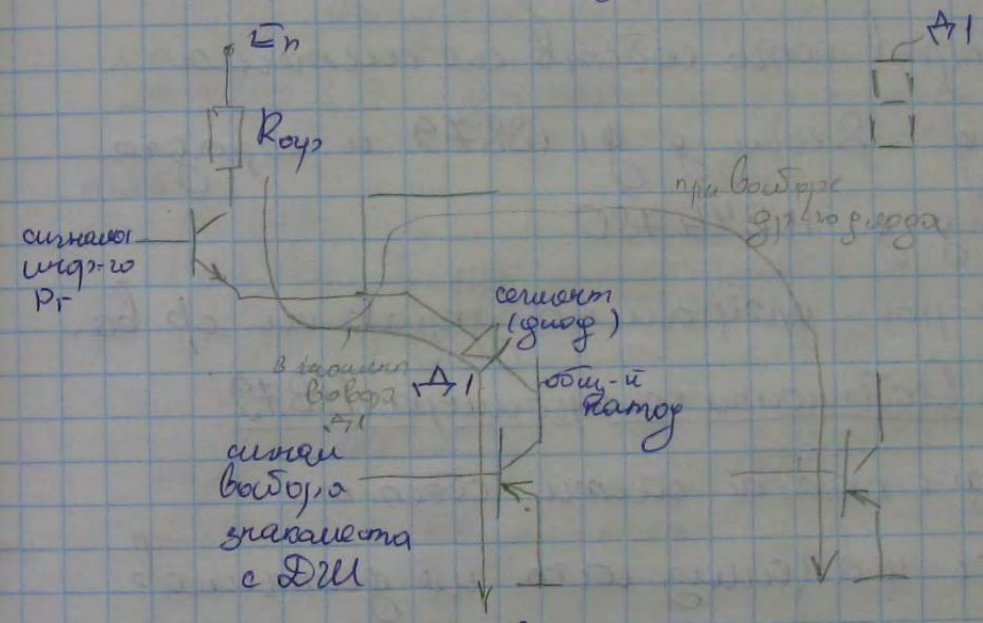
или выв-е порта МК, или обрат-ое на внешн. мат-о д-а, как в примере с АЦП.

б) динамическая регенерация

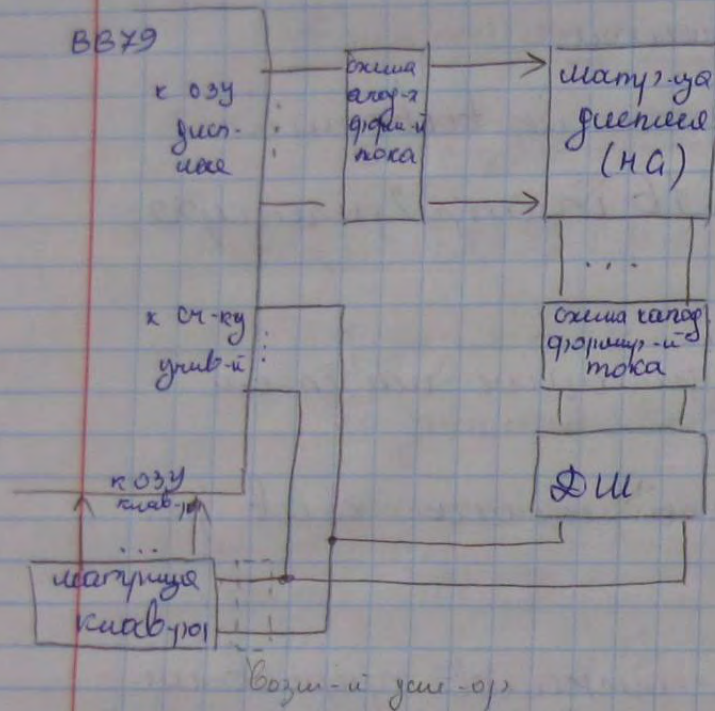


Основ-а задела: выбор, топ-а протек-то через диоды в НС-индек-ров. схема с регенерацией динам. матрицы.

Электроника цифрового сегмента индикации:

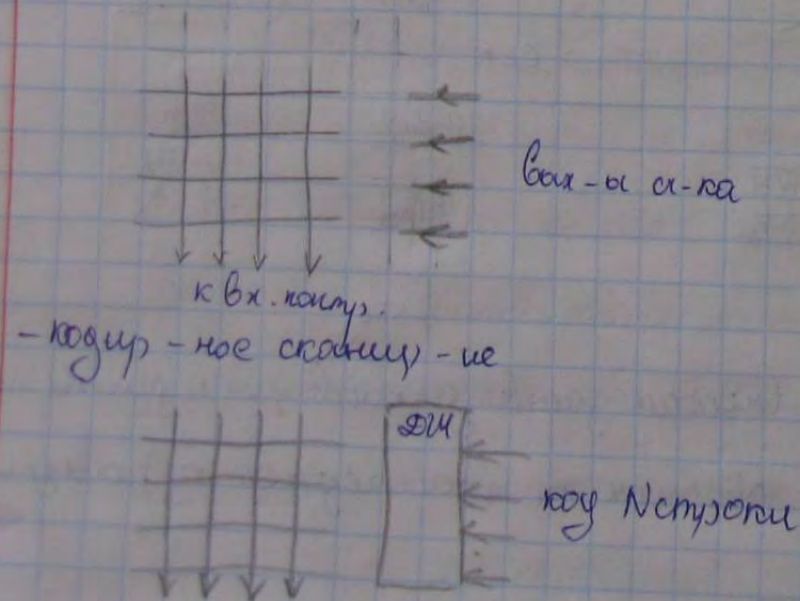


Регули-ка тока в цепи диода с помощью выв-а R_{огр}.

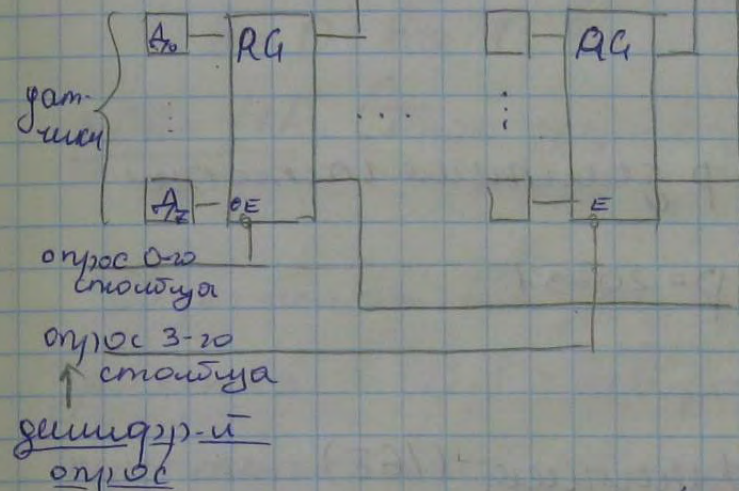


Способы подключения клавиатуры.

- дешифр-ое сканир-ие (напрямую)



- с шифр-и кошипут-и на вх. контр-ра



- с кодир-и сканир-ем (декод-ое сканир-ие) если вместо РГ будет МХХ



Команды контроллера BB79

7 6 5

КОП

коде парам-ров

КОП = 000.

Можно задать: кодир-и дешифр-ое скан-ие клав-ра, 4-режимная клав-ра

Блок-ка счит-и матрица датчиков n-клавиш-ое сканир-ие ввод по упр-му сканир-е

8, 16-разрядный и набор выв. интр-ов
(слева, справа)

KOIT = 001.

уст-ие зн-ие р дшитие частоты:

$\frac{CLK}{P}$

$P = 20 \div 31$

KOIT = 011.

чтение из ОЗУ ушнее (165)

из к-то пер-ца прох-то интр-ов и т.д.
автоинт-м

KOIT = 010.

чтение из ОЗУ клав-рой (85).

заг-е адр-са и интр-ов

Формат чтения (при чт-ии из ОЗУ).

1.

shift	ctrl	Nctrl	Nctrl
-------	------	-------	-------

 - 85.

2. чтение байта:

к-й байт - состояние одной клавиши.

KOIT = 100. Запись в ОЗУ ушнее: адрес ах-и,
в ком. произ-ве запись.

KOIT = 101. Запрещение записи / стирание
ушнее.

KOIT = 110. Таблица дшитие:

При тапении можно выв-ть:

все 0, все 1, все ... - 20 h ASCII

KOIT = 111. Вброс притывание.

* * *

В рег-ре кот-ие указ-ие: код-во амер,
занятость в ОЗУ клав-рой; признак
притывание при потытсе записи в ОЗУ
клав-рой; признак опуст-ие ОЗУ клав-рой;
признак негот-ти ОЗУ ушнее к пишью
г-х.

Синхронизирование контр-омера

г/ работы с клавиатурой

clock EQU 14h; P=20

mode EQU ...; y/ KOIT = 000

клав-кей EQU 40h; ко-и. KOIT = 010

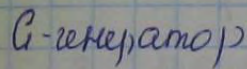
огу 30h $\nearrow$ орг3h
отр key

start: (*) mov A, #clock ; зн-ие дшитие частоты
mov R0, #80h ; адр-с вброса
mov R1, #81h ; адр-с вброса
(*)

movx @R0, A ; коб. ас. арг. и рег. R0
 mov A, mode ; режим
 mov @R0, A
 mov IE, #01 ; разгр. мо INT0
 setb EA ; разгр. мо чис. коб. ас.
 jmp \$1 ; окн. ас.

key: push PSW
 mov A, #read-key ; чтение к-та
 mov @R0, A ; в клав-ру
 mov A, @R1 ; чтение ОЗУ клав-ры
 mov 40h, A ; сохранение в память
 TBL: DB 3Fh, 06h, 5Bh, ...
 ↑ ↑ ↑
 цифры 0-й строки 831-й
 1-й строки
 mov DPTR, #TBL
 mov A, @DPTR; чтение из памяти
 mov 41h, A ; сохранение
 pop PSW
 reti

Обобщенная структурная схема
AVR - микроконтроллера.



РС-генератора
настройку
выполнен в
МК.

КР-кварцевый резонатор

PC - счетчик
процесса

dash 16-mu razp-
are

8 КБ = 4 Кслов, слово = 16 разрядов, ЕРАМ - внешняя память

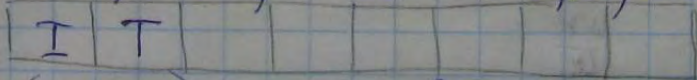
SRAM - статическая память, 5128 (g/nyozp-acc)

ЕИРКА энергонеав-ая паллято, 5/28 (у(ранных)

длина инт. ш $\approx 2,5 \div 4$ мс

Набор ПЗ зав-ит от модели (max 64 уст-ств)

SREG-регистр состоит из программы в СРЧ



орган
прехват
ниче

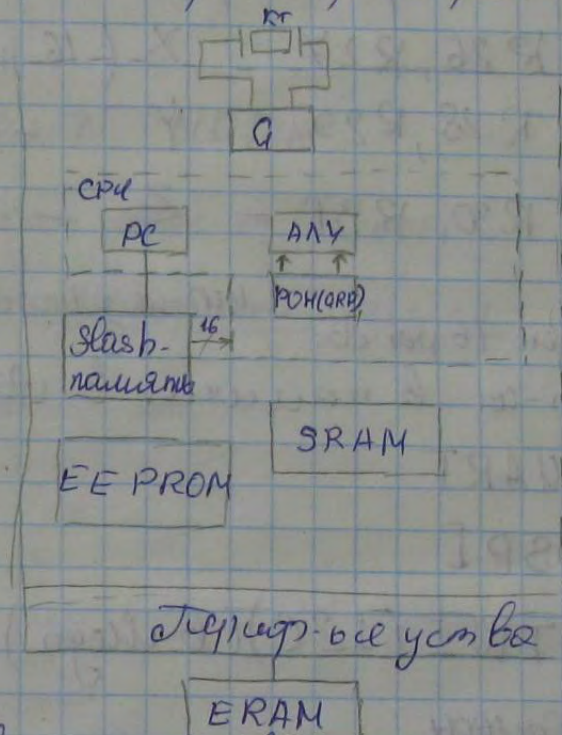
уточ-но
доступи
отнас

С-переносиме зами

V — переполнение

N - 3400

$\sum_z$ - NDV-знач, незав-и от перес-и.



Регистры делятся на две части:

1) 16-разовые верхние регистры с номерами:

R26, R27 - X (16-разовые регистры)

R28, R29 - Y - "

R30, R31 - Z - "

2) нижние 16 регистров. Используются для адресации

Интерфейсы в чипах AVR:

UART

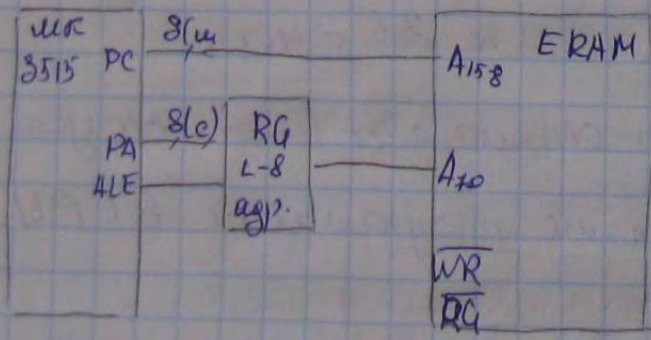
SPI

I2C (TWI) (в Меге)

Таймеры:

T0, T1, T2, WDT

Внешняя память - ERAM:



Работа с ERAM осуществляется так же как с SRAM, Значения на работе с ERAM и SRAM - 1ц.

PC, PA вводят адрес (16 раз-ов)

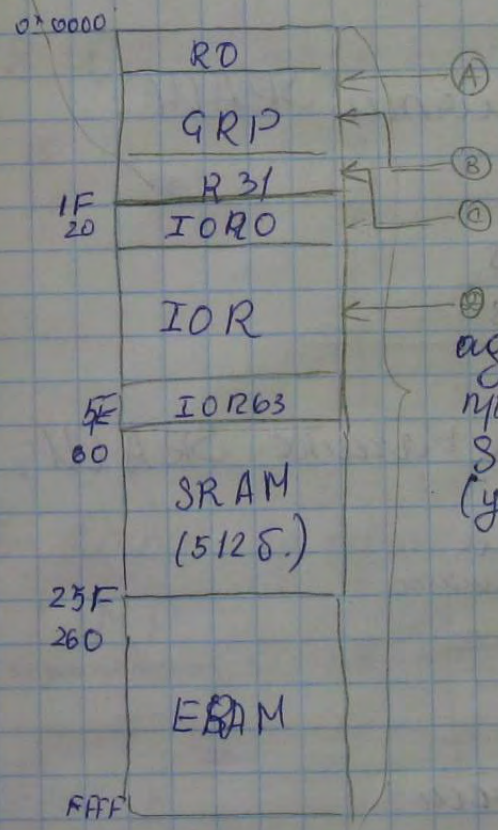
Режимы энергосбережения:

1. Idle

2. POWER DOWN MODE

Выход из режима с помощью

Логическая организация памяти

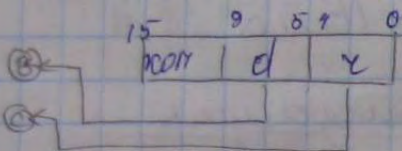
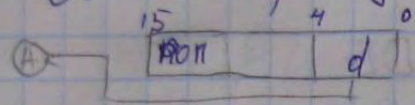


IOR - pin 61
Ввода / вывода
адресное пространство
SRAM (учитывая 16-разовые)

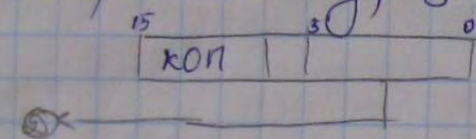
Способы адресации AVR.

1. Прямая регистровая адресация.

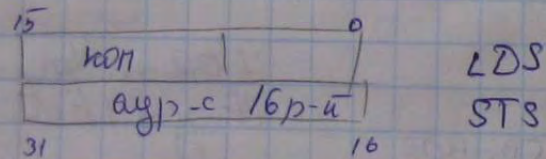
М-м обращение к рег-ам GPR.



2. Прямая адресация области 66/666.



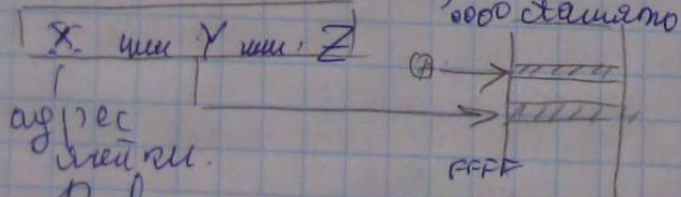
3. Прямая адресация памяти SRAM:



LDS
STS

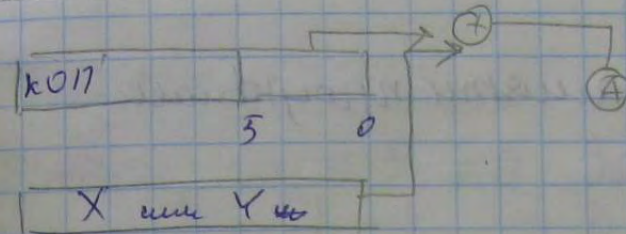
дает возм-ть адресации к ячейке SRAM.

4. Косвенная регистровая

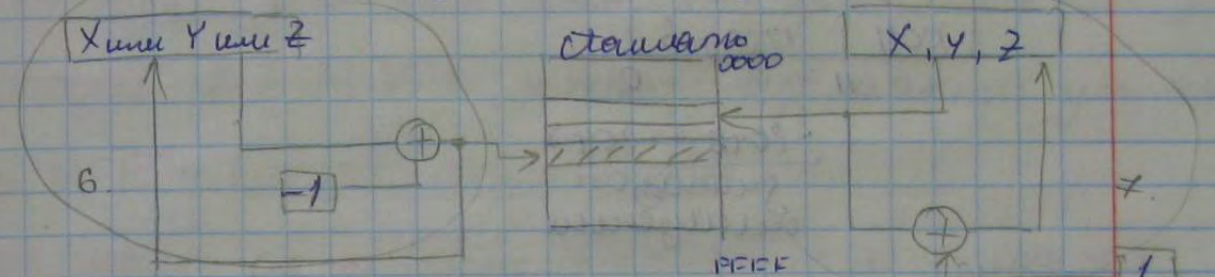


5. Косвенная по смещению:

Y или Z +



6. Косвенная адресация с предускориванием



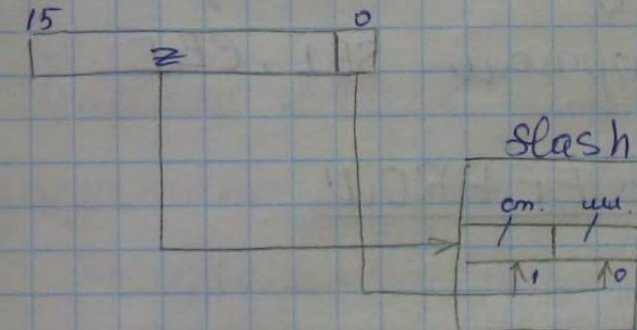
7. —" — с постдекриментом.

8. Обращение к памяти программы.

команды:

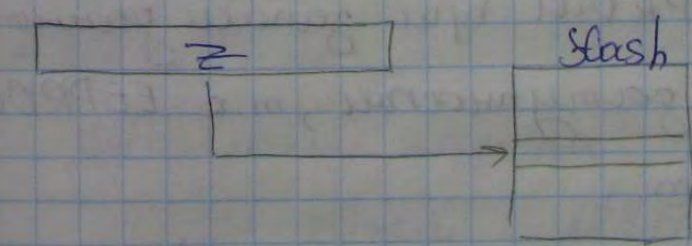
LPM [R0]

SPM [R1:R0].

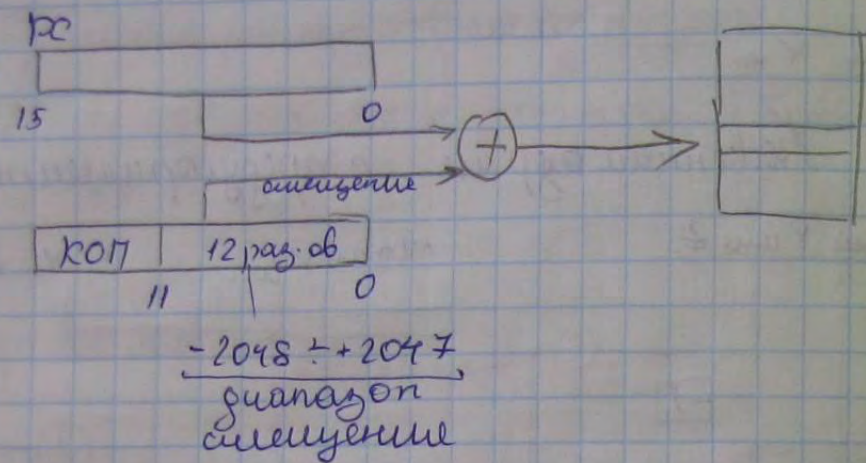


9. Косвенная адресация к памяти программы

LPM (возврат с переносом) ICALL (возврат по адресу)



10. Относит-ся к памяти проца



11. Непосредственная адресация

$Ld\ i$ - непосред-но в ком-де зад-ся адр-т

12. Битовая адресация - можно одр-но к рег-ам PCN и портам SBI, CBI

Работа с EEPROM.

Рг. адр-та

Рг. данных

Рг. управ-ия

Обращ-ие к EEPROM при записи данных:

→ 1. Проверка доступности, т.к. EEPROM может не работать.

ожидая одуливания сигнала в Рг. управ-ия

EEWE

2. При $EEWE = 0$. Запись адреса в Рг. адр-та. Записываем данные в Рг. д-х.

В Рг. управ-ия уст-ть бит произвольного значения EEMWE и через 4-ре такта уст-ть $EEWE = 1$.

Начинает-ся процесс записи, к-й длит-ся 2 ÷ 4 мс; в зав-ти от уровня сигнала того аппарат-и микропроцессора.

Прим.

Имеется 2 бита защиты: по к-м можно запретить запись, чтение.

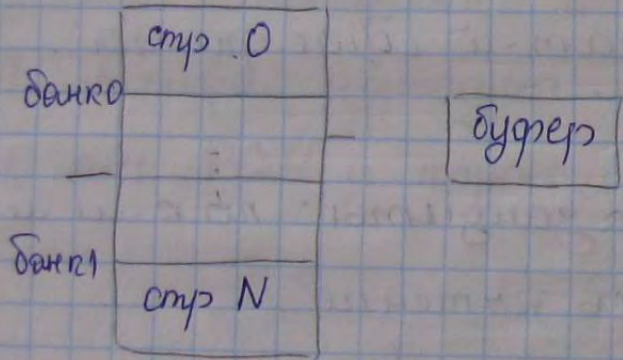
Память пользовательного типа EEPROM.

Расширение памяти: с микропро-ом I2C можно подключить внешнюю память AT24.

Особен-ые хар-ки:

объем . 128Б - 128КБ

напр-е питания 1,8V - 5,5V



Записи в стр-у первоначально
записываются в буфер и по команде
Stop на инт-се запускается процесс стирания
всей стр-ы и запись буфера в стр-у.
При записи производится контроль
целостности стр-ы, к-й приводит к записи в
записи на стр-у при выходе в-а адреса

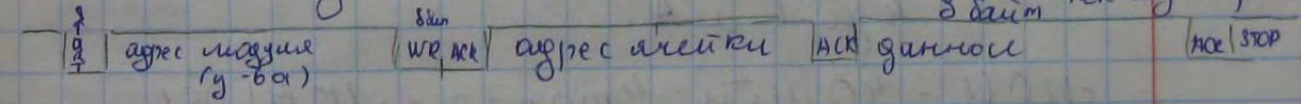
стр-уе.

При чтении контроль целостности стр-уе
не проводится. AT2402 256б при 8 мод-х = 2Кб
WP-бит настр-е стир-е записи настр-е.
Страницы м-т иметь 1, 2, 4, 8, 16, 32, 64, 128, 256 байт.

AT24	32 Кб	64	128	256	512	1024	Кб
	4 Кб	8	16	32	64	128	Кб

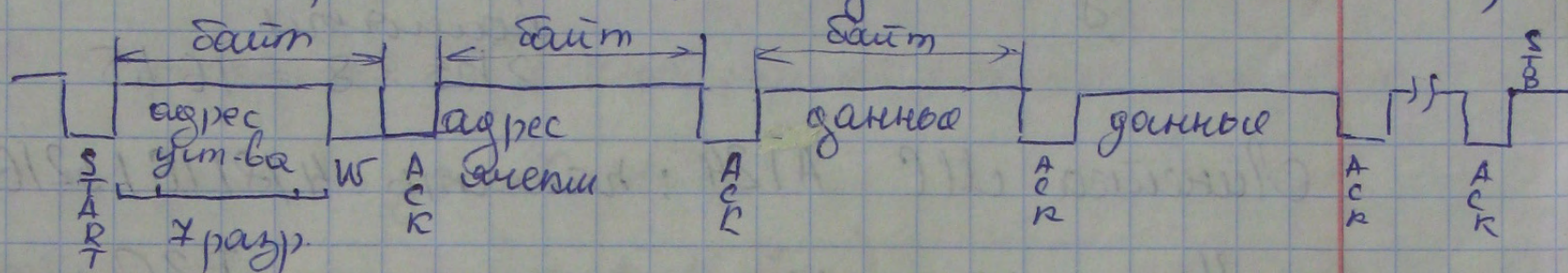
Некоторые мод-ы внутри производятся на
банки. Мод-ы с 2, 4, 8 банками,
при этом кол-во адрес-х (инт-ва)
кодов уменьшается на 1, 2 и 3 соотв-но,
т.е. драг-ки происходит интенсификация
модулей по 2, по 4 и под. в один "мод-ый"
модуль.

Запись одиночного байта:



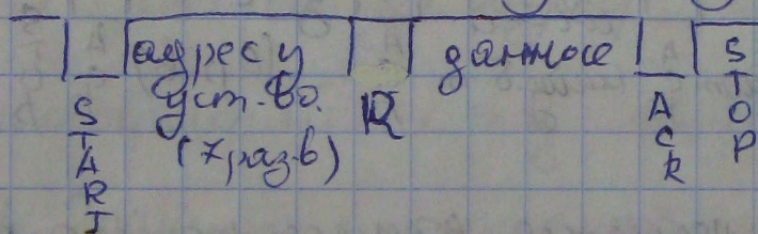
Энергонезависимая память АТ25

Запись страничная (последовательность байтов):

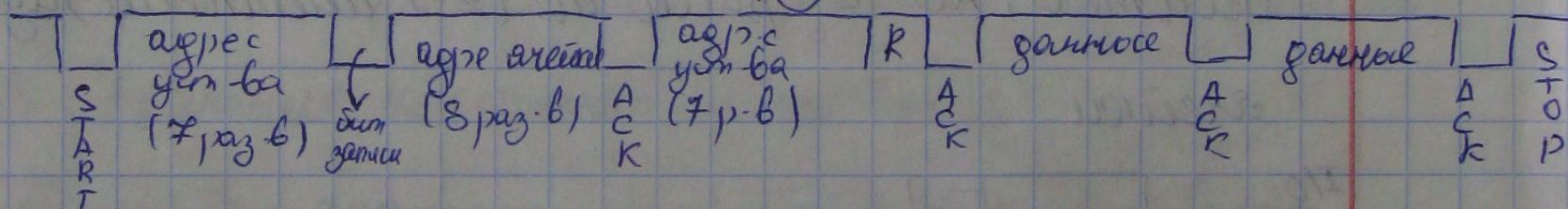


Адрес послед-ей ячейки для записи в ней байта с пош-ю вступн. счетчика.

Чтение по текущему адресу:



Чтение по заданному адресу:



Сериовый адресовый устрой-о (АТ):

Если АТ дан-ся на банки (2, 4 или 8 байтов), произв-ся изме-ие в протоколе: адрес уст-ва

1010 $A_2 A_1 A_0$ модиф-се на запись 1010 $A_2 A_1 P_0$ - для 2-х байтов.

$P_0 = 0$ - банк 0

$P_0 = 1$ - банк 1.

1010 A_2, P_1, P_0 - для 4-х банков
 1010 P_2, P_1, P_0 - для 8-ми банков.
 ; 1 МС на 2 КБ

Новый вид: $A_2, A_1, A_0, P_2, P_1, P_0$ - при обзоре объема памяти
 $2 \text{ КБ} \times 8 = 16 \text{ КБ}$.

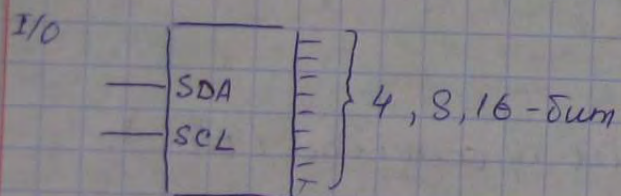
Микросхема МС AT24; память 4/8/16/32/64/128 КБ

Исп. с расширенным протоколом I2C;
 для передачи адреса требуется 2 байта.

Пример: Запись одиночного байта.

адрес	адрес	адрес	д-е
усл-во	байт	байт	(8 б-т)
(7 б-т)	(стар. байт)	(конт. байт)	
$A_7 \dots A_0$	$A_7 \dots A_0$	$A_7 \dots A_0$	S, D, P

Подать в образе как-то запись послед. т-х байтов, но с разн-ми (2-х байтными) адрес-ми ячейки.



Память DataFlash (семейство AT45).

Объем $V = 1-64 \text{ Мбит}$

AT45DB62 : 64 Мбит, 2 интерфейса связи
 (3-х проводной SPI-интерфейс (чтение, запись, синхронизация), до 20 МГц
 парал-л инт-с до 5 МГц).

Структура DataFlash.

Схемы, обеспечивающие хранение данных об объеме
 д-х, инт-с д/оцифр-ки звука, хранение
 протоколов и т.д. Страницы:

1. Хранение д-х для работы приложения
2. Хранение программы, к-е и-т счит-ся в ОЗУ микро-контроллера
3. Хранение констант и таблиц.

Страницы: хранение интр-ли в памяти в виде.

Датчик температурный.

Управление считыванием:

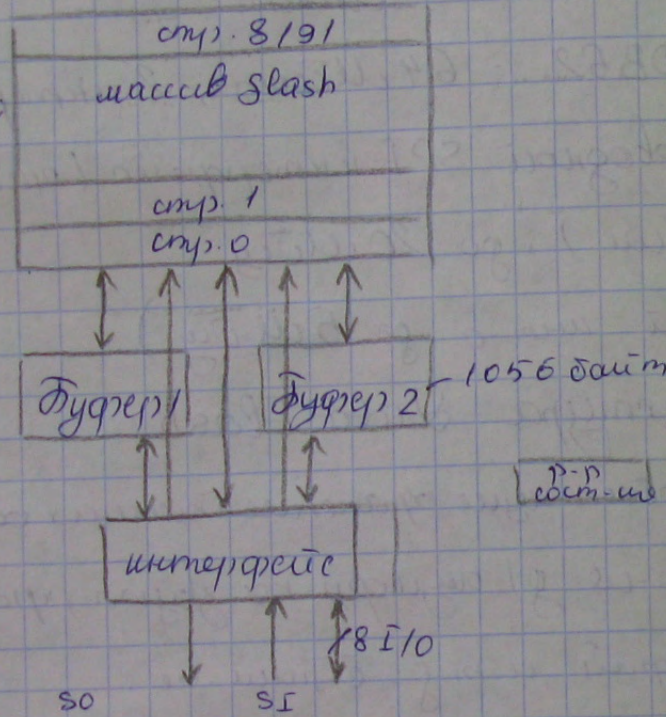
RESET
CS

DER / PAR

READY / BUSY

WP

SCR



Serial Output

Serial Input

CS - выбор и управление ИС

DER / PAR - проверка и установка 11-го бита ИС

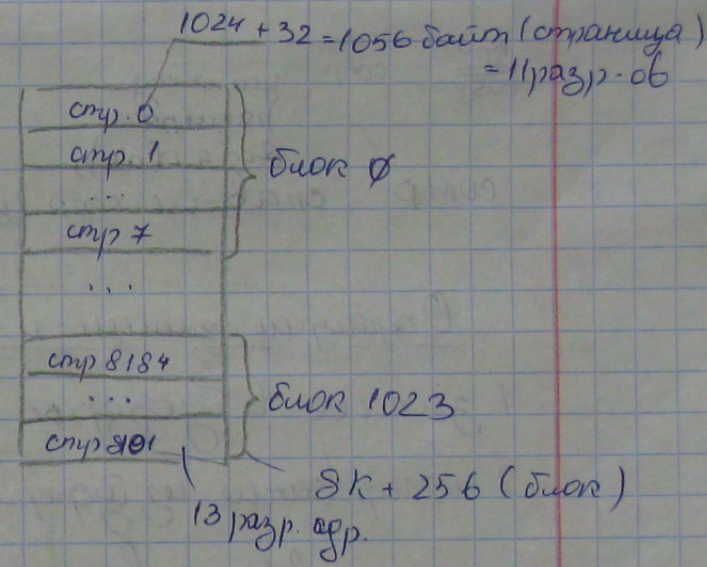
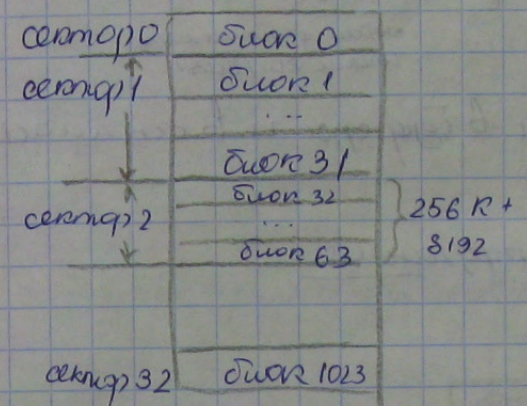
READY / BUSY - готовность и неготовность ИС

SCR - установка, сброс передаточного регистра (бита, байта)

WP - защита от записи

Логическая организация
устройства.

Секторы, блоки и страницы - уровни
памяти.



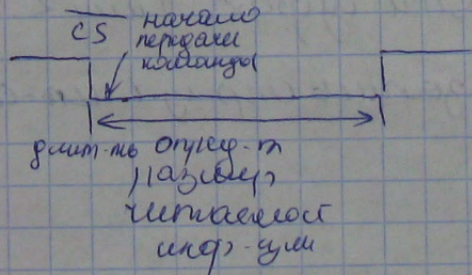
Операции:

1. чтение / запись на уровне страницы
2. стирание на уровне страницы и блоков

Операции чтения:

1. непрерывное чтение
КОП (1 байт)

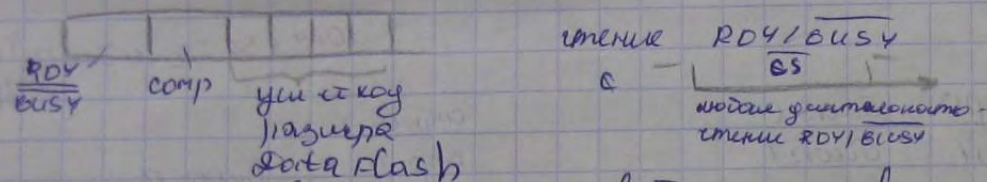
3 байта адрес (16-битовый, 5 страниц + 16-бит на страницу)



2. постраничное чтение

3. чтение буфера

4. чтение слова состояния



компр - слав-ие стр-цы в буфере и в ач-машине

Операции записи и стирания:

1. запись в буфер
2. копирование из буфера
 - 2.1. с встроенным стиранием
 - 2.2. без встп-стирания
3. запись через буфер (авт-ки две операции)
4. стирание стр-ц и блоков.

Доп. операции:

1. Копир-ние стр-ца из Flash в буферы 1 или 2.
2. Сравнение страницы Flash и буфера. 1 или 2.
3. Автоматическое перезапись стр-ц и бл-ков буфера 1 или 2.

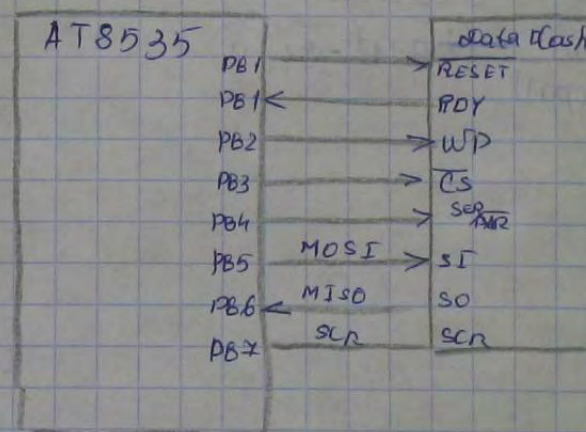
Алгоритмы записи:

1. Запись через буфер
2. Запись в буфер и затем копирование

Воп-ие возникает, тогда когда запись производится страница за страницей.

Запись двух страниц из разных чипов не возможна:

Взаимодействие МК с DataFlash



Системная кристалли

МТ, блок памяти, массив программируемой логики.

Atmel, Cypress, Trisend.

В Е5 ил-ие МК Turbo 8032.

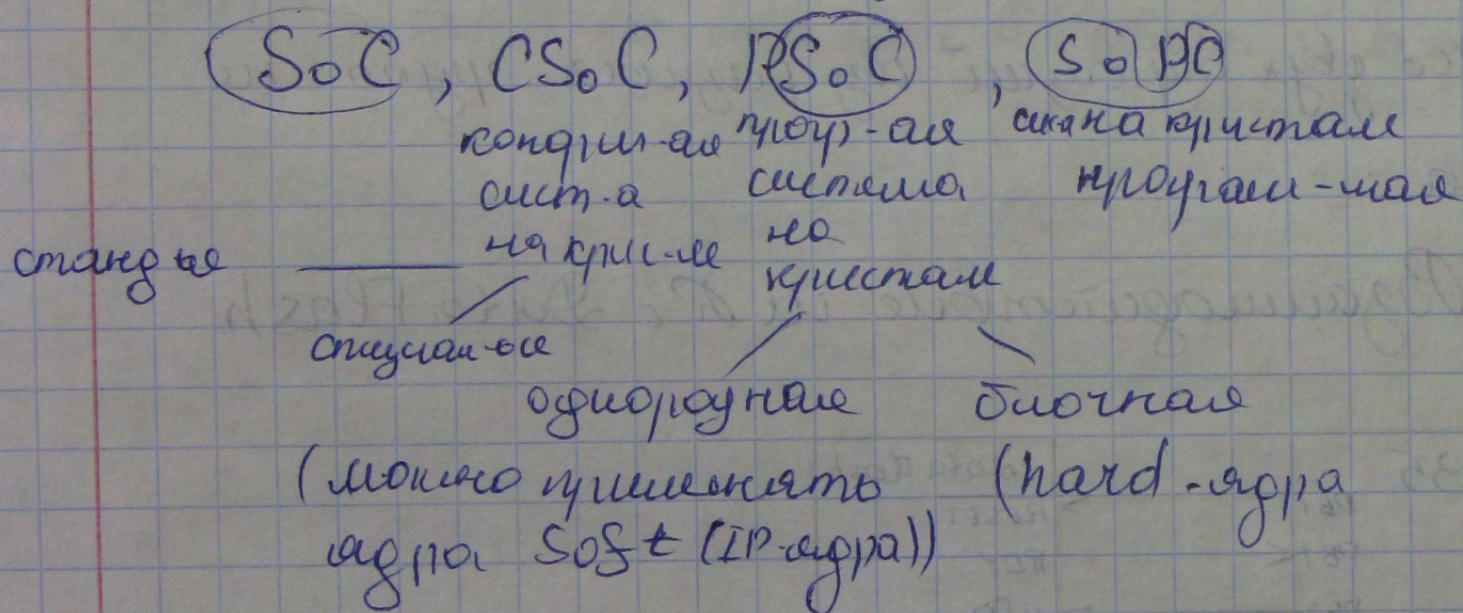
FPSLIC

AVR

Положение произв-ли, отношение энерго-

потребление, уменьшение габаритов и цены.

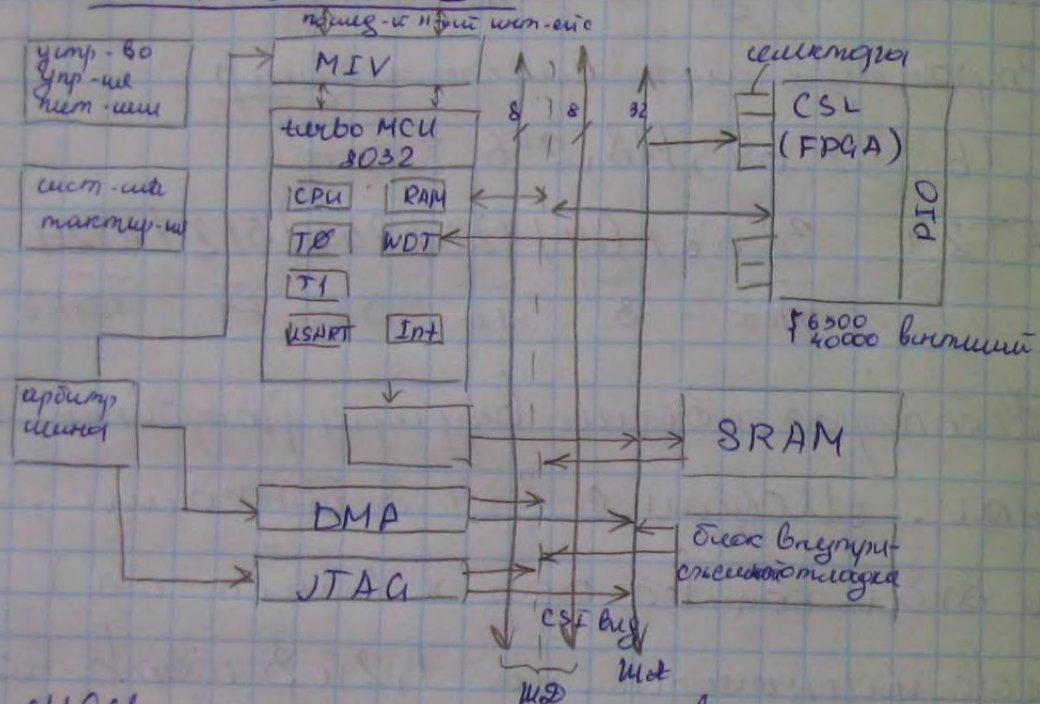
Система обозначения.



CSOC.

Структурные схемы

1. Tausend €5.



МССА - микропигу-ос ^{мд} устп. 60

CSL - конгиг - ае илтишдае ионика

РТО - порты вв/вав, с учетом ил-ив
можно обеспечить напр-но спос-то от
4 до 12 мА.

МСС - 40 МГц: 4 такта $\Rightarrow$ 10 MIPS (млн. опер. в сек.)

Средство передачи по шине 40 ШДТ.

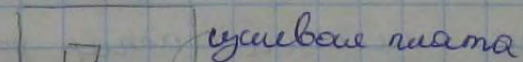
СБГ-конфигурация систем - а нугара (interconnect)

Частота работы шины 40 Мбайт/с

DM A - контролер, 17.017

МІV-контролер вилучення пам'яті об'ємом до 256
Частота 40 мГц: 10 MIPS кбайт

Частота 40 мГц: 10 MIPS



МК имеет шлейфы, вместо МК внутренняя та
система отправки со звонком с ПВМ

Получение проп-се CSL и CSI, PIO

2. FPSLIC

СМС программирование поездов - сис. сист-го

успехи и успехи-успехи.

AT94 x x



1. ойууе дьукатр-е
иникт
ССК-еинхр-уе

СКР - (сложн) - числ

Динамическое регулирование
матрицы в пр-е вып-ки
продукции.

$$\begin{array}{ll} 10K \times 16 & (10K \times 16)_{max} \\ 4K \times 8 & (16K \times 8)_{max} \end{array}$$
$$4R \times 8 \quad (16R \times 8)_{\max}$$

	$6R \times 16$	
$2R \times 16$	$2R \times 16$	$2R \times 16$

 $2R \times 16 \quad 2R \times 16 \quad 2R \times 16$

Для проектирования и использования пакетов System Designer
(включая встроенные средства проектирования AVR и FPGA).

1. AVR Assembler v1.3

2. Редактор языка описания аппаратуры
HDL Planner (VHDL - Verilog)

3. Синтезатор языка описания аппаратуры
(сверточная библиотека логических элементов FPGA, AVR)

4. Средства симуляции, разводки и транзисторной
FPGA.

Верификация и отладка:

В AVR Studio есть интрин ModelSim
и готово Seamless совместной симуляции
(посмотреть на симуляцию FPGA), и, самое
главное, работа с прототипом от FPGA при работе
симулятора.

PSOC Express.

Ядро MSC: 24 МГц, быстродействие 4 MIPS,
от 93,7 КГц до 24 МГц
136 интрин

- 173Y - 2-64R5
- 03Y - 256-20458

сложная матрица входов и выходов и
12 аналоговых блоков / цифровых: счетчики,
таймеры / (анал. блоки: пропорциональное управление,
аналоговые компараторы из R-блоков, полосовые
и низкочастотные фильтры из C-блоков,
АЦП, ЦАП с разрядностью от 6 до 14 разрядов)
Набор интерфейсов RS-232, I2C, SPI синхр.,
асинхр. UART).

3. ARM

Adventurous RISC machine

Описание на языке, поддерживаемые ядрами.

ARM-ядро.

ARM7 самый популярный тип, новое
ARM9, 10, 11.

Отработка ядра ARM7 TDMI:

на языке C в среде сложности прог-ми
на Assembler.

Flash: 100 000 циклов, 20 мс

SRAM: адресное побайтно, 32-х разряд. шифрами
и 16 разряд.

USB: сод-ит буфер, FIFO, 328 байт,
поддерживает передачу 4-х каналов г-х,
общее ядро генератора частоты

UART: высок-с-ти обмена

30 Мб/с - синхр., 6,87 Мб/с - асинхр.

SPI: 30 Мб/с, 4-ре линии, обмен г-ми
по 8 или 16 бит.

Сч/таймер: 4-х каналов сод-ит и илт-ов.

Блок ИММ: 4-х независим-х канала, и-по
прогр-то незав-мо скважностью,
помехоустойчив-сть и т.д.
общее и
интерф-с линии на нагрузку до 10 мА.

Струк: 8, 10-битовый режим,
384 кГц, 1 мА, междупроб-ми 1 мкА

Контр-р:
JTAG: обмен г-ми ПУ с памятью без
участия прог-ядра.

В составе JTAG и периф-и узлов

имеет 32-х разряд. ук-но адреса

16-х разряд. ук-но четности г-х,

повтор-ые операции г/работы одного и
подготовки группы блока.