

АРХІТЕКТУРА КОМП'ЮТЕРІВ-2

1 лекція

Процеси

Одночиповые микроконтроллеры

Одночип. микроЭВМ - обобщ. класс, который включает микроконтроллеры (МК), траньютеры, сигнальные вычислители, специализиров. вычисл.

МК предназн

Д/з: федерат "Обзор по МК различных фирм" (2-3 недели) (8,16, 32 разр.)

(Intel, Motorola, A... (5) min
параметры не растасывать;
нужно: объем памяти, возможн.
перепрограм; подкл. внешн. памяти
не интерес. история, особенности
выпуск по фирмам

5 багов
(не больше 10 ; 7-10 стр. ; min 5 стр)

Базовый МК 1816 BE48 (Intel 8748A)

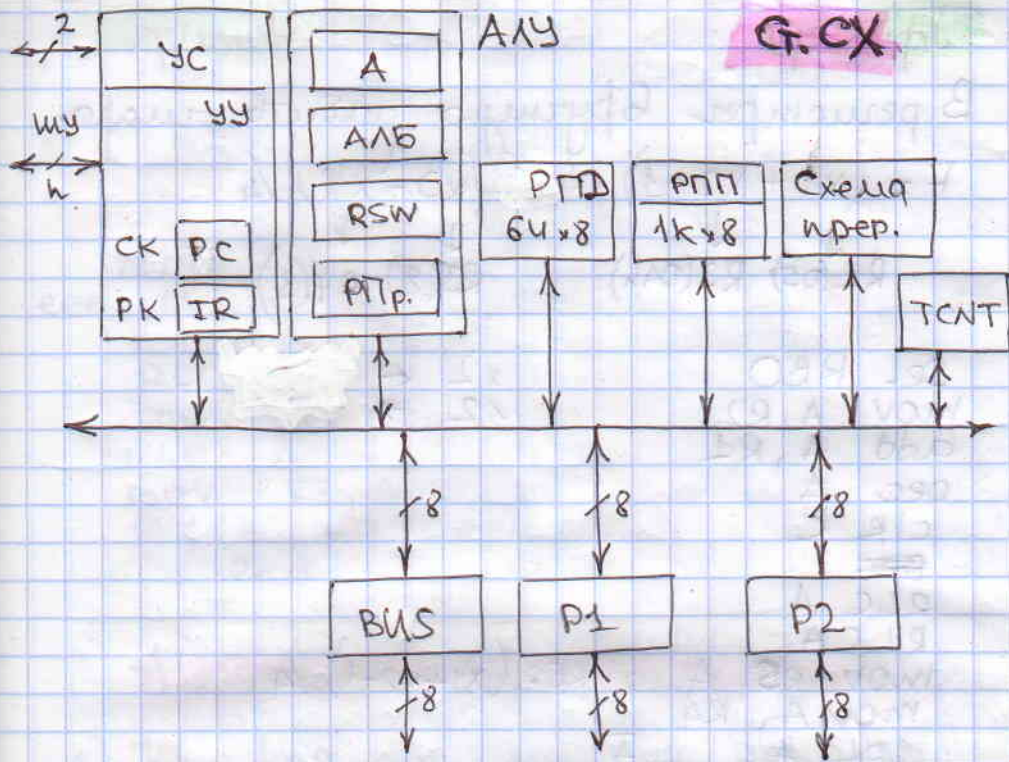
TTL технология; 40вх/вых;

есть окошко для ультрафиол. прописки.

УГО

1	TO				
2	BQ1			RD	8
3	BQ2			PME	9
4	SR	SMC		WR	10
5	SS	CPV		ALE	11
6	INT				
7	EMA			↔	
39	T1			P20	21
25	PROG			...	
	↔			P23	24
12	DB 0			P24	35
13	DB 1			...	
	...			P27	38
19	DB 7			↔	
20	GND			P10	27
26	Vddc			...	
40	Vcc			P17	34

- Д/З:
1. Назначение Вх / Вых
 2. Функцион. назначение выводов
 3. Модель программиста (распечатайте)
 4. РПД (рисунок) 1
 5. Лаб. фоб. №1 +.
 6. Распечатать систему команд



- **РПП** - резидентна пам'ять програм;
- **РПД** - резидентна пам'ять даних;
- **УС** - пристрій управління і синхронізації;
- **АЛУ** - арифметико-логічний пристрій;
- **RSW** - регістр слова стану програми;
- **TCNT** - таймер / лічильник.
- **P1, P2** - порти вводу/виводу.
- **BUS** - порт вводу/виводу.

• **A** - акумулятор - вказівник адреси,

- **A** - акумулятор - вказівник адреси, приймає та дисер. операнди;
- **ALB** - арифметико-логічний блок - виконання арифм. і лог. опер.
- **PSW** - ознаки переносу і допоміжн. переносу бітс.; ознаки, ознаки банків реєстрів; показувач стеку
- **РГр** - реєстри;
- **РС** - лінійник команд;

Тренировка

В перемножении группировать числа

$$F = (x_1 + x_2 - 1) \cdot 4 + (x_3 - x_4) / 4$$

\downarrow \downarrow
R1(05) R2(04h)

\downarrow \downarrow
R3(8) R4(3)

```
sel RBO
mov A, R2
add A, R1
dec A
clr C
```

$x_2 \leftarrow$
 $/2 \rightarrow$

```
RLC A
RLC A
RLCA
```

mov R5, A ; R5: $(x_1 + x_2 - 1) / 4$

mov A, R4

cpl A ; \overline{A}

add A, #1

add A, R3

clr C

rrc A

rrc A

mov R6, A

anl A, R5

mov R7, A

end

RO. R1

```

      + 10101010
      10010110
  C 1 01000000
  RO
  ↓
  [ ]
  
```

Процесс



```
MOV R7, #32
eee: CLR C
MOV A, R5
RLC A
MOV A, R3
RLC A
MOV R3, A
DJNZ R7, eee
end
```

Назначение вх./вых. УГО

- 1) **TO** - вх/г, по мест. ком. $\overline{INT0}$; вх/г такою счн. CLK.
- 2) **BQ1** - вх/г для подкюог. зови. днсер. синхронизації соо кварц. резонатора.
- 3) **BQ2** - " -
- 4) **SR** - скидання (CLR)
- 5) **SS** - покрокове виконання програи
- 6) **INT** - переривання
- 7) **EMA** - встанови. ренс. роботи з зови. пам'ятю програи
- 8) **T1** вх/г, по мест. командами $\overline{JT1}$, $\overline{INT1}$; вх/г зови. події.

- 25) **PROG** - линия строки для программирования входу/выходу;
- 12-19) **DB0-DB7** -шина д.; порт BUS (**BUS0-BUS7**)
- 20) **GND** - заземление;
- 26) **V_{dc}** - дополнительная нагрузка напряжение +5В;
- 40) **V_{cc}** - напряжение питания +5В;
- 8) **R** - шт. зовн. пам. данных;
- 9) **PME** - дозвол шт. зовн. пам. д;
- 10) **WR** - запись у зовн. пам. данных;
- 11) **ALE** - сигнал адреса зовн. пам;
- 21-24, 35-38) **P20-P27** - восьмизначный порт вводу/выводу
- 27-34) **P10-P17** - восьмизначный порт вводу-выводу

лекция 2

25.09.13

(кроме памяти данных и таймера
все описать)

Стр. схема:

АЛУ 2.2

Память программ 2.3

лекция 3

2.4 Резидентная память данных

Пам. дан. предназна. для хранения, записи и хранения д., полуз. в рез. обраб. инф.

Пам. д. (АЗУ) состоит из **64 ячеек**
памяти (регистров)

Условно П.д. раздв. на 2 банка регистров, стек и просто область АЗУ.

П.д. обращении к банкам регистр. использ. прямая адресация.

По всей ячейке П.д. обращ. с помощью косвенной адресации.

Указателями адр. могут быть только **регистр нулевой** или **регистр первый**, **банка регистра нулевой** или **банка регистра первого**.

Переключ. банка регистра выполн. командой

```
sel RB0  
sel RB1
```


Прямая адресация:

```
mov A, R5
add A, R3
inc R2
```

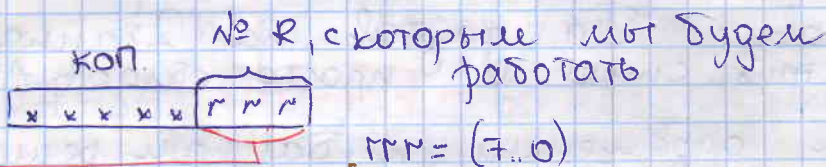
Косвенная адресация:

```
sel RB1
mov R1, #3Ah
mov A, #R5
mov @R1, A
```

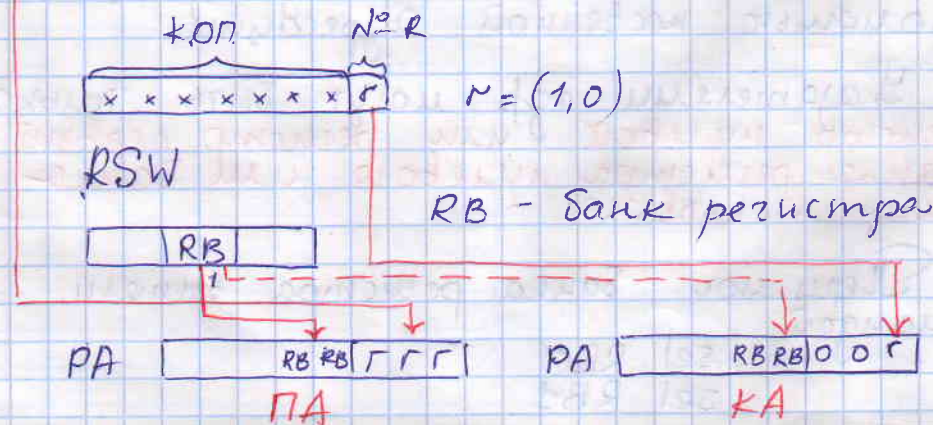
из A → шир. вез. <3Ah>

R1 - указатель адреса

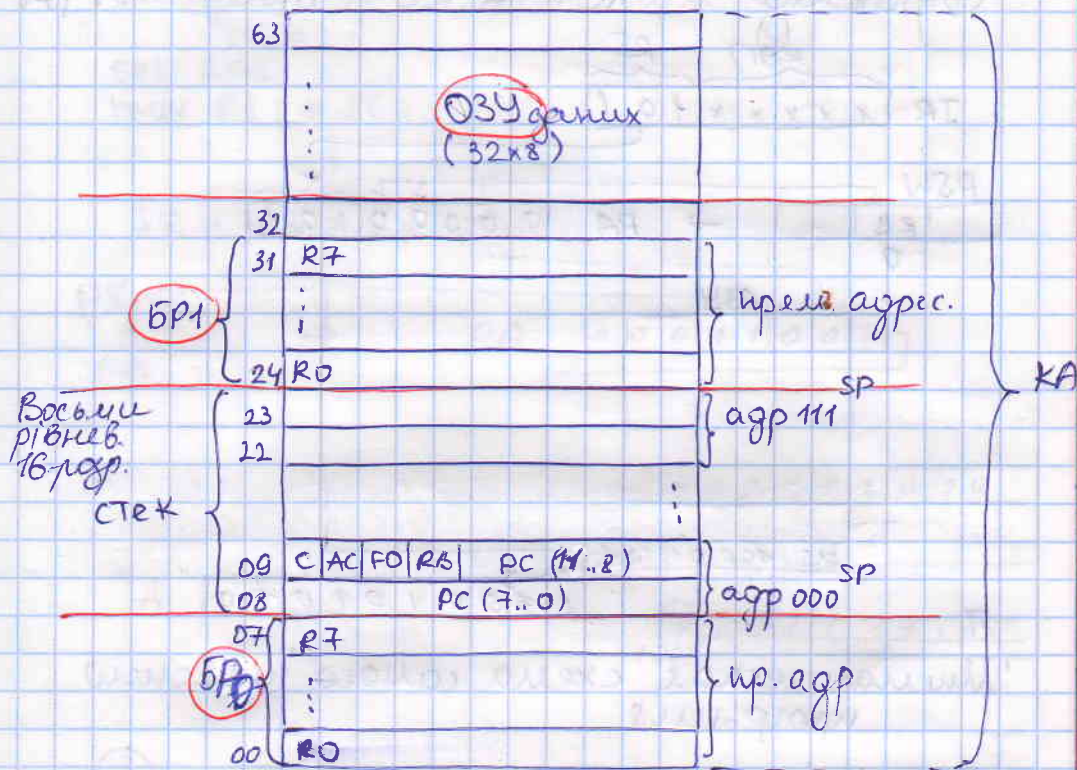
Как формируется адрес при косвенной адресации?



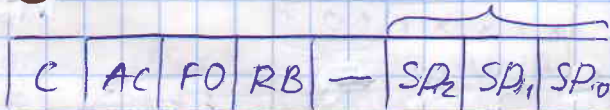
Формиров. адр. при косв. адре-ции:



Распределение памяти данных:

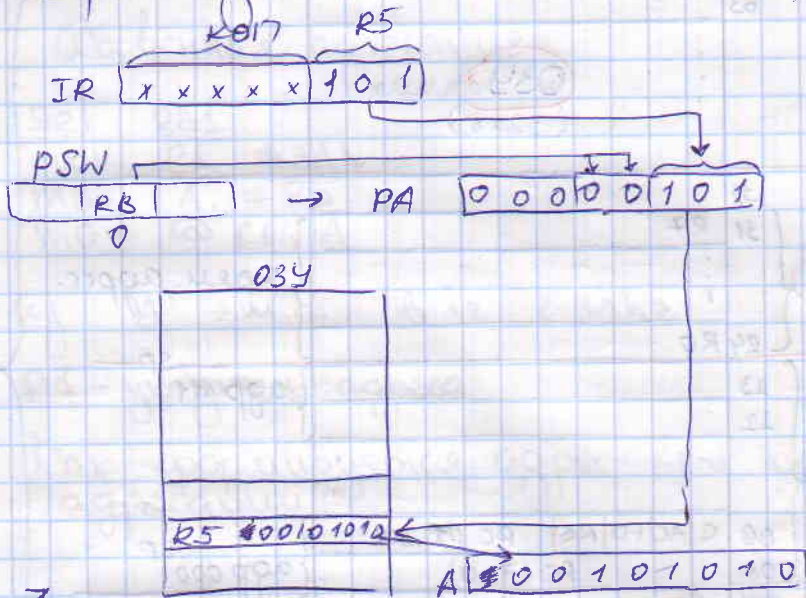


PSW



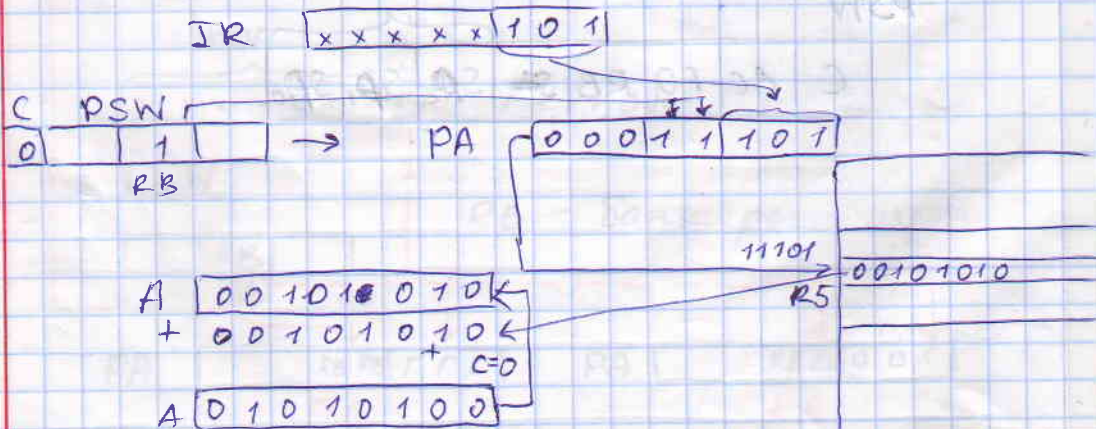
Мнемоническая схема формирования информации в регистре для обращения в АЗУ.

Пример: `mov A, R5 (BR0) : ПА`



Мнемоническая схема самого выполнения

Пример: `addc A, R5 (BR1) : ПА`



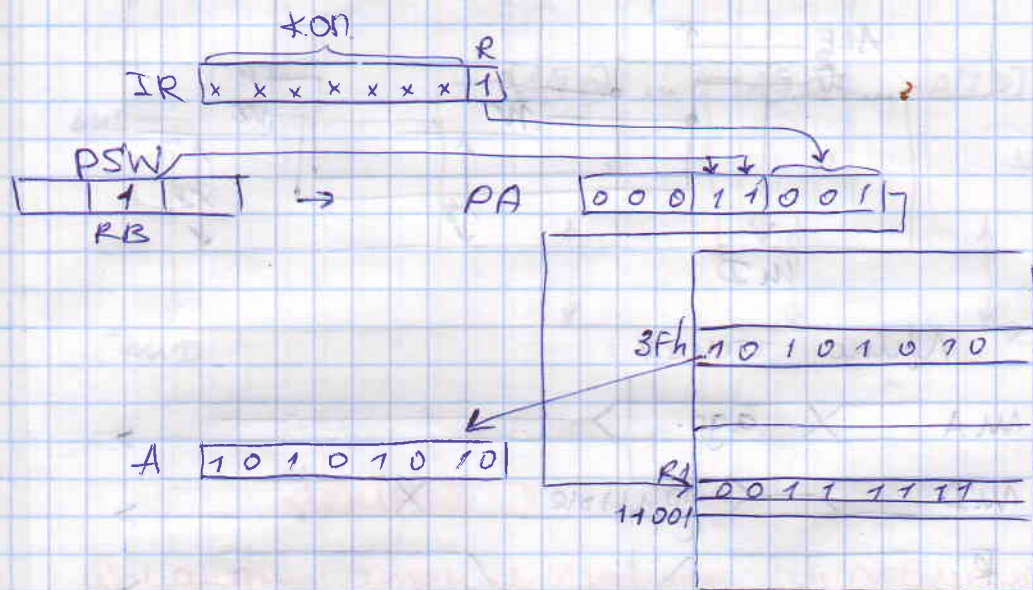
Орг. адр. образ. к пам. - ~~КА~~

Пример: `mov A, @R1; BR1`

На пом. вынос этой пом. должно
быть:

`sel RB1`

`mov R1, #3Fh`

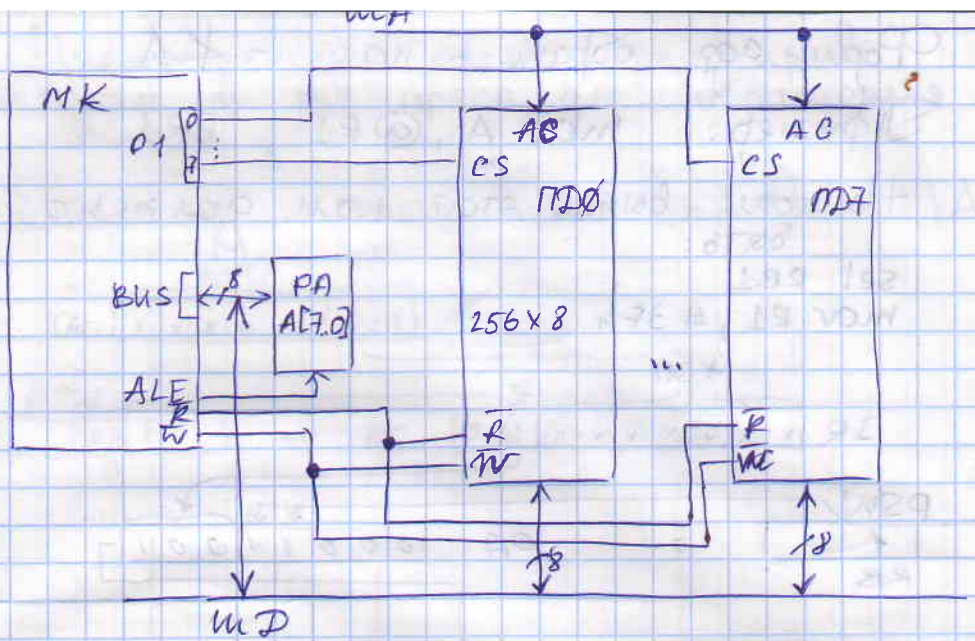


Для увеличения объема вн. пам. г. использ. дополн. внешние микросхемы п.г.

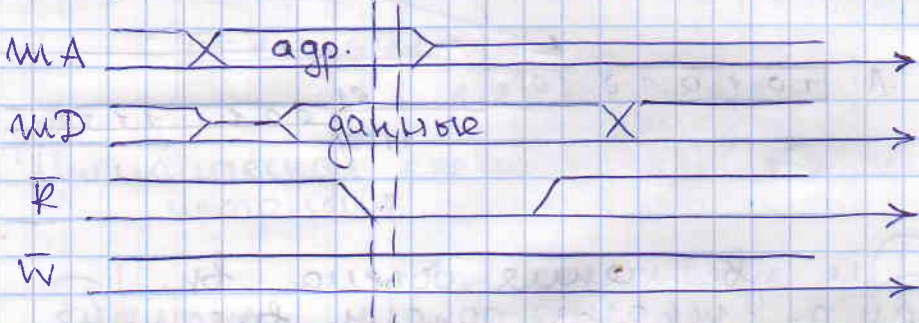
Каждая миксх. (страница) имеет объем 256x8 Адрес считыв. 7-3 порт bus 8 разр. в рег адреса.

Структурная схема подключения внешней. Шаги данных

Страница выбор. 7-3 порт P₁, P₂

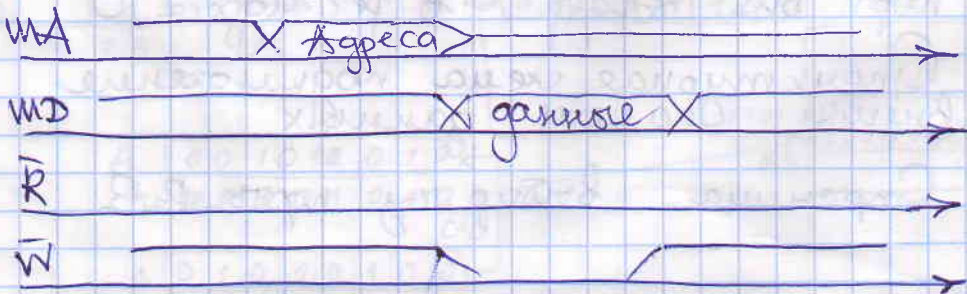


Линии чтения:



момент записи.

С! Временная диаграмма для записи

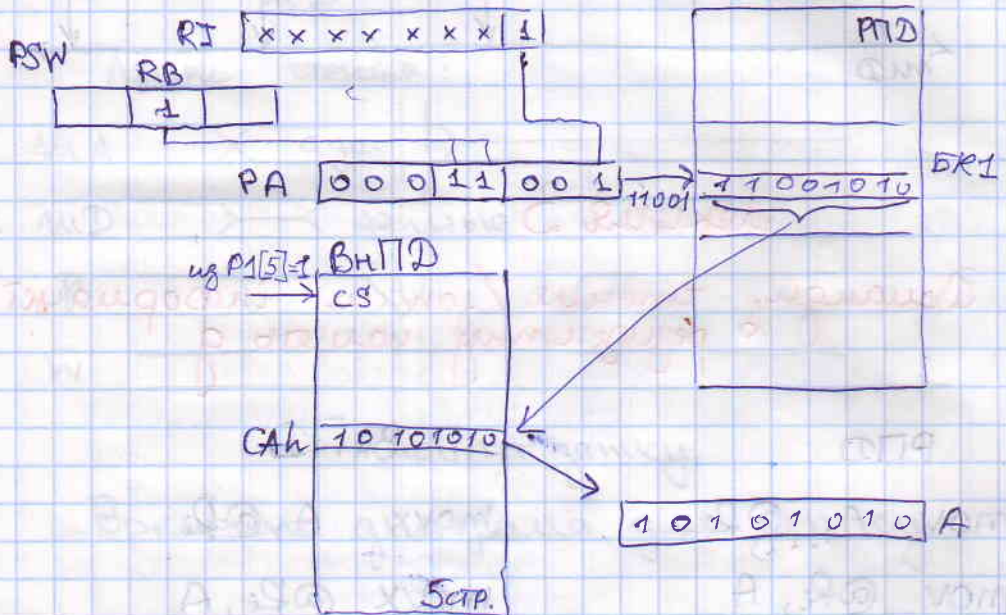


Мнемоническая схема выполн. команды
при работе с ВнПД

1. Выбор страницы ПД
2. Подключить БР
3. В регистрах должна быть информация (RA)

Пример

movx A, @R1



Задача 2

Переслать мас. д. [20 байт] из
ВнПД (CAh) в РПД (наз. сег. 30h).
Информация записана в стр. 5
ВнПД.

; 1. Выбор стр. нач. данных

```
mov A, # 00100000b
```

```
out P1, A
```

; выбор BR0

```
sel RBO
```

```
mov R0, # CAh ; адр. ВнПД
```

```
mov R1, # 30h ; адр РПД
```

```
mov R7, # 20 ; кол. ч.
```

; пересылка мас. дан.

```
lll: movx A, @R0
```

```
mov @R1, A
```

```
inc R1
```

```
inc R0
```

```
djnz R7, lll
```

```
end
```

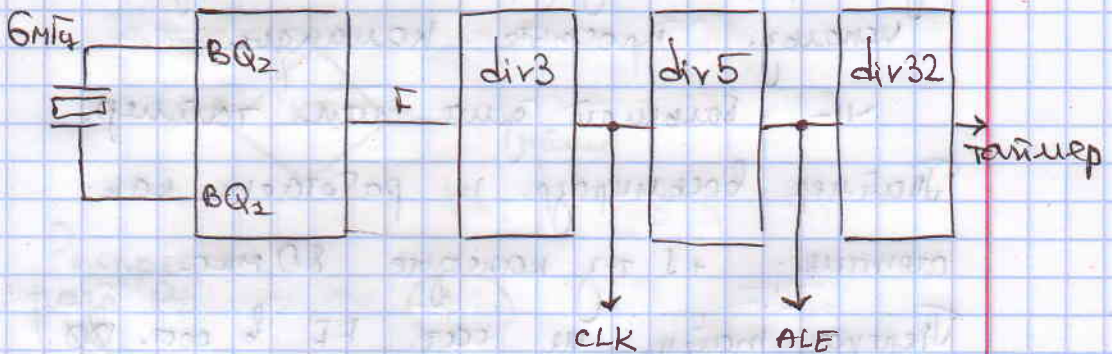
C!

2.5 Память програи (РПП, ВПП)

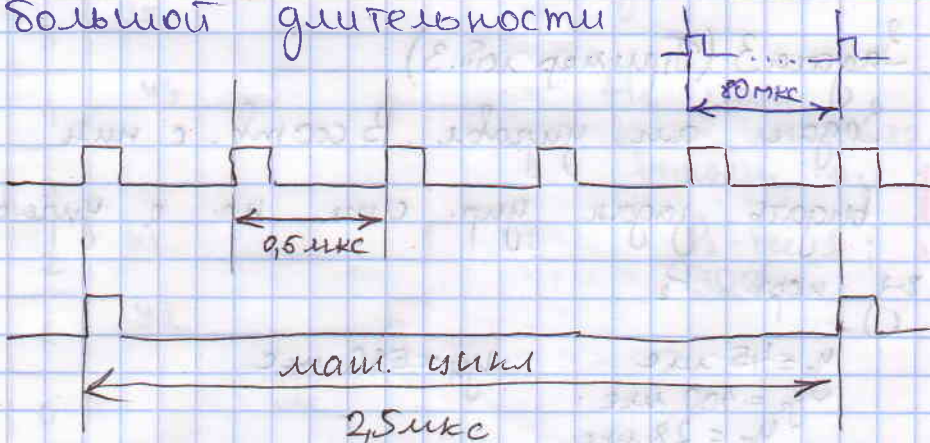
2 банка регистров БРР - РПП
БР1 - ВПП

Узел синхронизации

Для того, чтобы сформир. необх.
для работы миконт. делительн. мкшип.
к микроконтр. подключ. квартовый
резонатор (в пример 6 МГц)



ALE идентичн. вых. 1 кан
Таймер исп. для формир. задержк.
большой длительности



Иногда необходимо выдавать на
о. управления управ. сигнала
различной видности:

$$y_5 = 720 \text{ мксек}$$

$$y_6 = 580 \text{ мксек}$$

где формиру. задержка небольш. длит.
использ. простые команды.

-11- большой длит. исп. таймер.

Таймер восьмиразр. и работает как
счетчик: +1 раз каждые 80 мкс.

Переход тайм. из сост. FF в сост. ~~00~~
фиксируется в триггере TF - перенос.
таймера

Задача 3 (Пример 10б.3)

Задан ал. управл. В соотв. с ним

выдать задан. управ. снн. на о. управл.

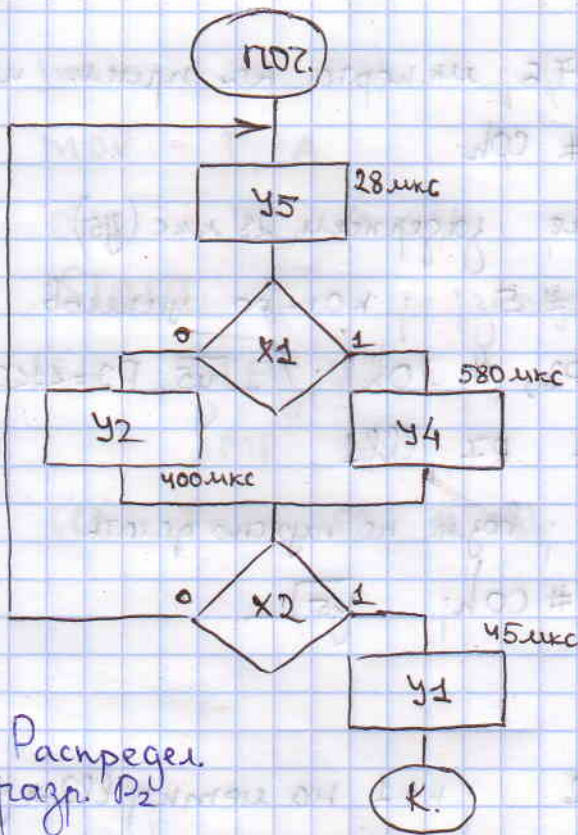
из порт P_2

$$y_1 = 45 \text{ мкс}$$

$$y_2 = 400 \text{ мкс}$$

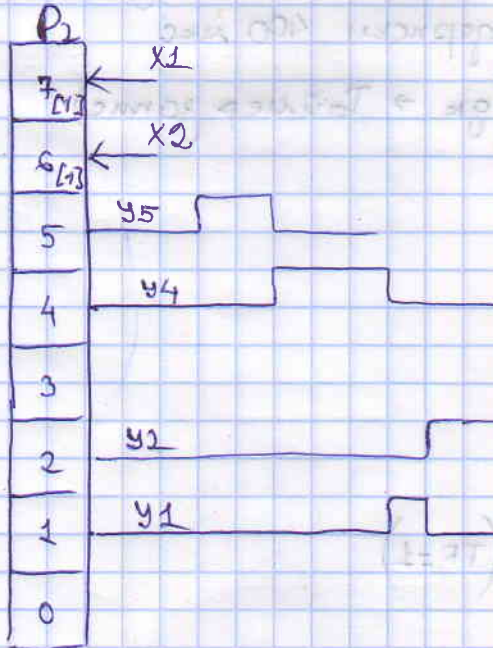
$$y_5 = 28 \text{ мкс}$$

$$y_6 = 580 \text{ мкс}$$



(Л.4 1сп.мн. →
2 или 4сп.мн.
на след. изд.
сделать 23.р.)

Распредел.
разр. P₂



Разр. порта
программир. как
угодно мне.

Но пом.
выкл. эк-контр.
выкл. пр.

P₂-все "1";
BR0

установка 7,6 мк.порта на прием инпр.
ANL P2, # 00h

формирование задержки 28 мкс (y5)

MOV R7, #5 ; кол-во циклов

~~ORL~~ ORL P2, # 20h ; y5, P2 = 11100000

ll1: DJNZ R7, ll1

NOP ; возм. не нужно делать

ANL P2, # 00h ; y5

IN A, P2

JB7, ll2 ; по 1 на четку ll2 → y4

формиров. задержки 400 мкс

400:80=5 (-5 дк → Таймер затмс)

1 1 1 1 1 0 1 1

+ 1

1 1 1 1 1 1 0 0

+ 1

1 1 1 1 1 1 0 1

+ 1

1 1 1 1 1 1 1 0

+ 1

1 1 1 1 1 1 1 1

+ 1

0 0 0 0 0 0 0 0 (TF=1)

mov A, # F3h ; (-5gx) → A

mov T, A

ORL P2, # 04h ; $\sqrt{y_2}$

START T ; запустить таймер

ll4: JTF ll3

JMP ll4

ll3: ANL P2, # 00h ; h2

в. кр!
1.03

с
0!