Национальный технический университет
«Харьковский политехнический институт»

Кафедра «Промышленная и биомедицинская электроника»

Практическое занятие по дисциплине «Микропроцессорная техника»

Знакомство со средой разработки Keil. Группа команд передачи данных

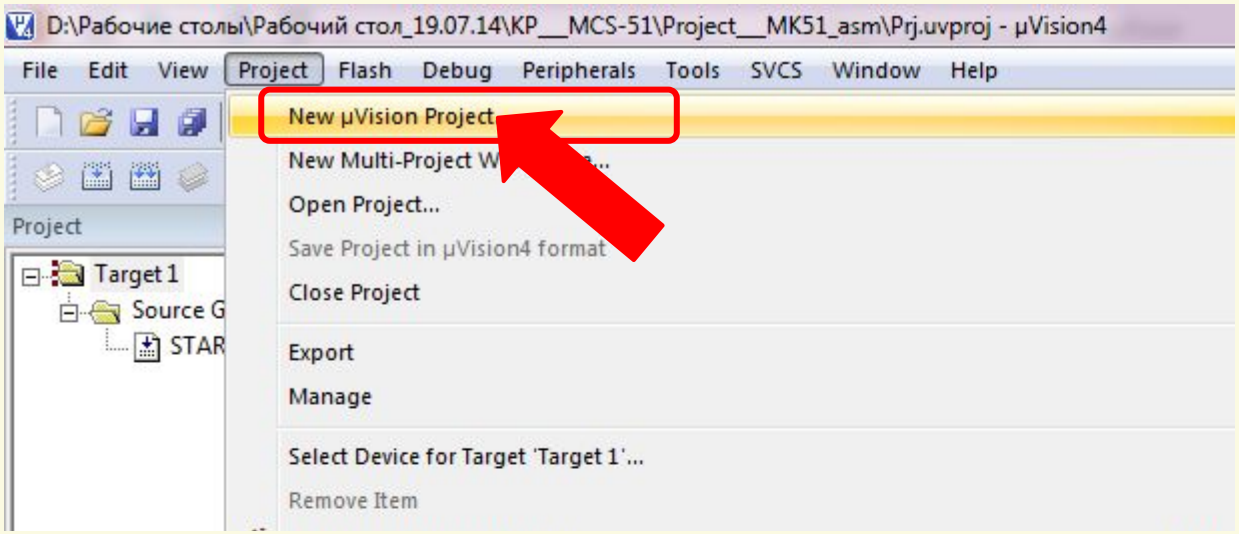
Б.А. Стысло

г. Харьков, 2014 г.

ЗНАКОМСТВО СО СРЕДОЙ РАЗРАБОТКИ KEIL-51.

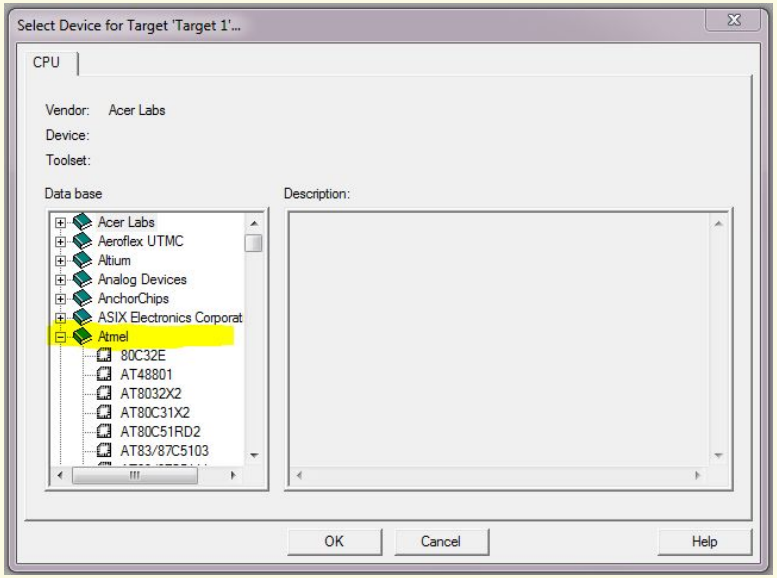
ЗНАКОМСТВО СО СРЕДОЙ РАЗРАБОТКИ KEIL-51.
Создание *проекта*

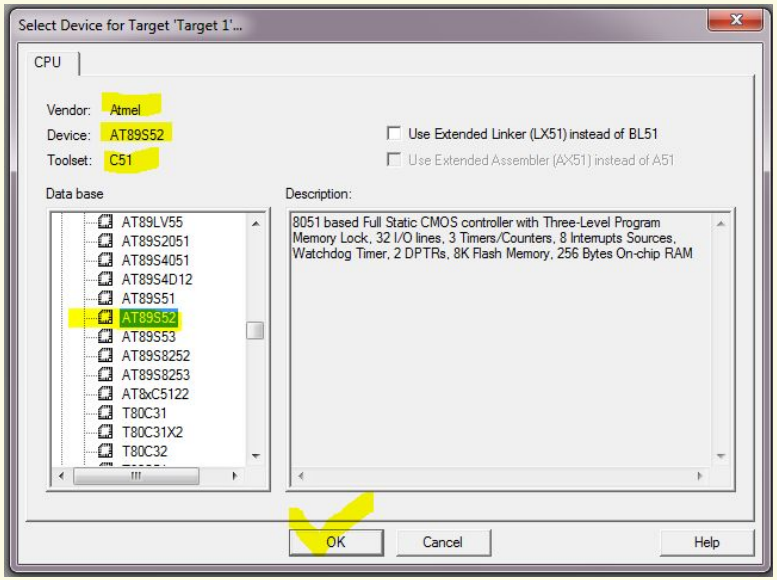
В отличии от используемых ранее средств разработки (Borland Pascal, Borland C), где весь текст программы хранился в одном файле (*.pas), Keil предполагает создание *проекта,* который может содержать множество файлов, в т.ч. библиотеки.

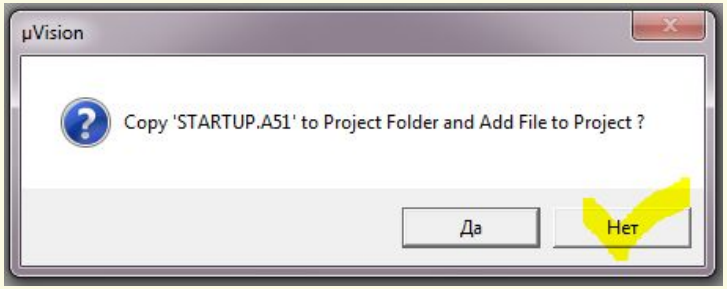


ЗНАКОМСТВО СО СРЕДОЙ РАЗРАБОТКИ KEIL-51.

Создание *проекта*

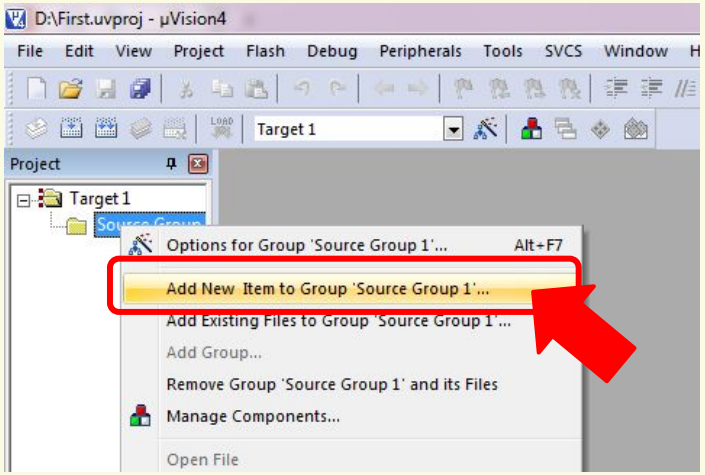


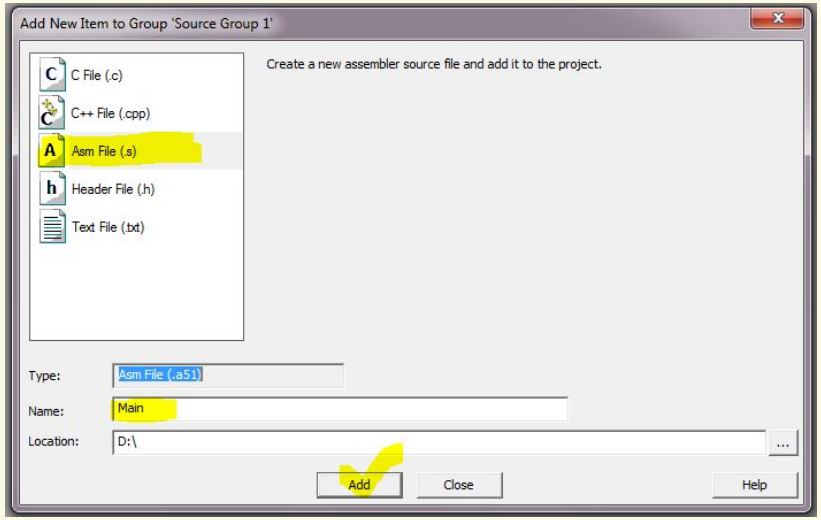


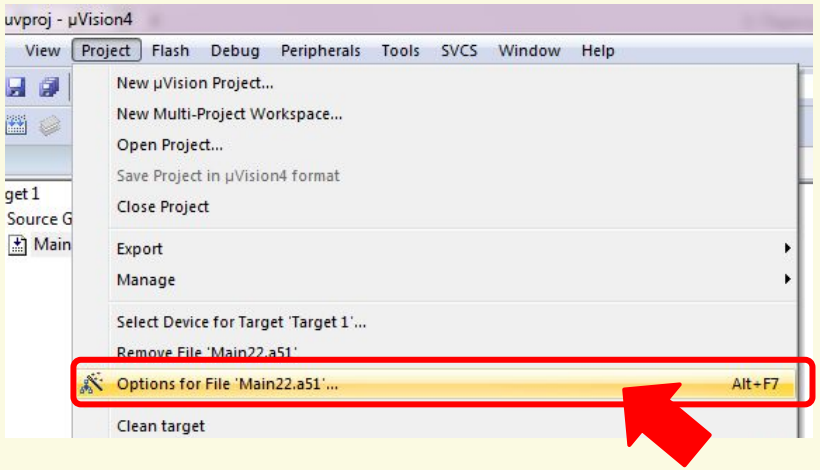


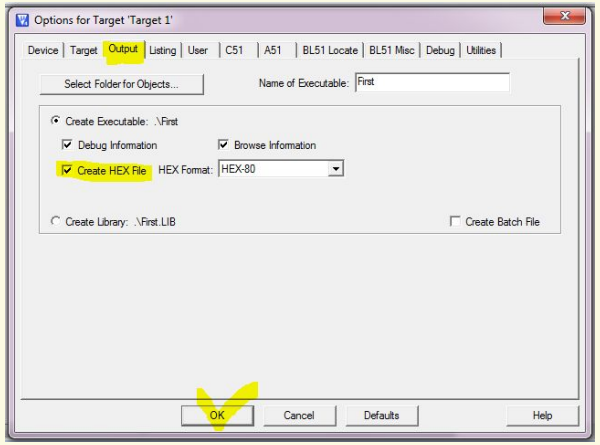
ЗНАКОМСТВО СО СРЕДОЙ РАЗРАБОТКИ KEIL-51.

Создание *проекта*









ГРУППА КОМАНД ПЕРЕДАЧИ ДАННЫХ

Данные в МК могут храниться:

- *Регистры (A, B, R0..R7);*
- *Внутренняя память данных (ОЗУ);*
- *Внутренняя память программы (ПЗУ);*
- *Внешняя память данных (ОЗУ);*
- *Внешняя память программы (ПЗУ);*

Регистры *– ячейки памяти внутри МК, обмен информации между которыми осуществляется простыми и короткими командами.*

Аналогия с языком высокого уровня Pascal:
Регистр = переменная типа byte

R0..R7 – регистры общего назначения (РОН)

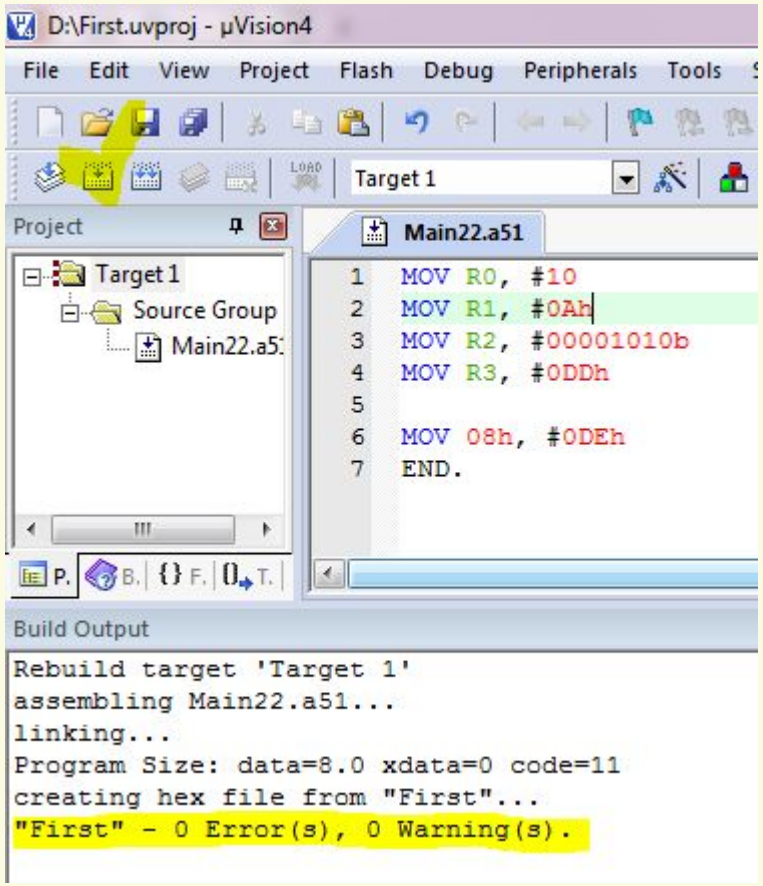
Часть команд возможна лишь с использованием специального регистра – *аккумулятора* (A)

“HELLO, WORLD!”

Первая программа

```
MOV R0, #10
MOV R1, #0Ah
MOV R2, #00001010b
MOV R3, #0DDh

MOV 08h, #0DEh
END.
```

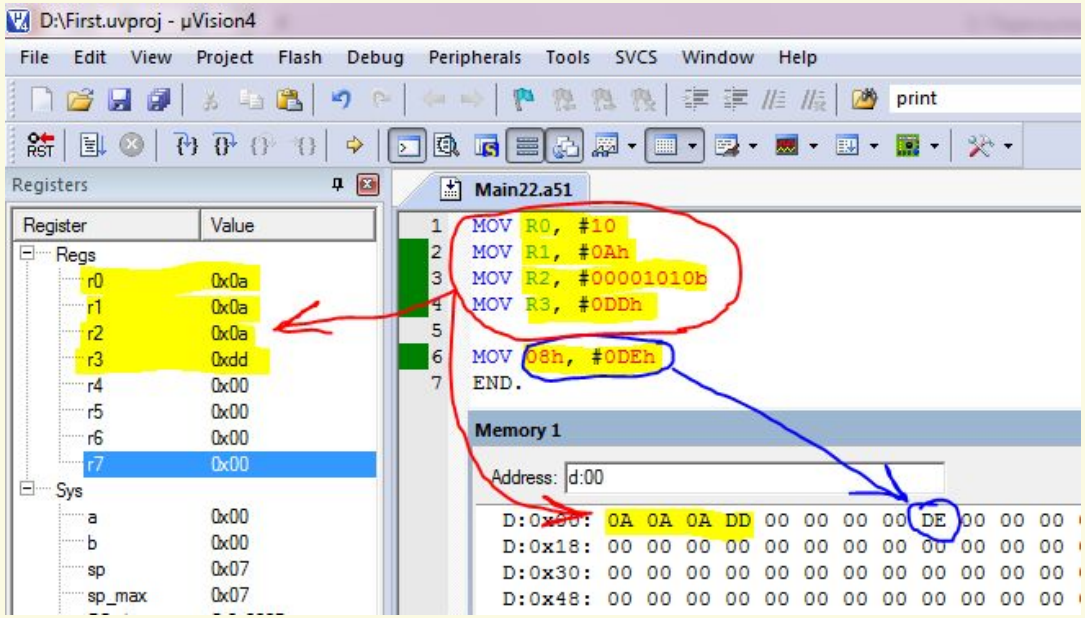


ГРУППА КОМАНД ПЕРЕДАЧИ ДАННЫХ

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Пересылка в аккумулятор из регистра (n=0÷7)	MOV A, Rn	11101rrr	1	1	1	(A) ← (Rn)
Пересылка в аккумулятор прямоадресуемого байта	MOV A, ad	11100101	3	2	1	(A) ← (ad)
Пересылка в аккумулятор байта из РПД (i=0,1)	MOV A, @Ri	1110011i	1	1	1	(A) ← ((Ri))
Загрузка в аккумулятор константы	MOV A, #d	01110100	2	2	1	(A) ← #d
Пересылка в регистр из аккумулятора	MOV Rn, A	11111rrr	1	1	1	(Rn) ← (A)
Пересылка в регистр прямоадресуемого байта	MOV Rn, ad	10101rrr	3	2	2	(Rn) ← (ad)
Загрузка в регистр константы	MOV Rn, #d	01111rrr	2	2	1	(Rn) ← #d
Пересылка по прямому адресу аккумулятора	MOV ad, A	11110101	3	2	1	(ad) ← (A)
Пересылка по прямому адресу регистра	MOV ad, Rn	10001rrr	3	2	2	(ad) ← (Rn)
Пересылка прямоадресуемого байта по прямому адресу	MOV add, ads	10000101	9	3	2	(add) ← (ads)
Пересылка байта из РПД по прямому адресу	MOV ad, @Ri	1000011i	3	2	2	(ad) ← ((Ri))
Пересылка по прямому адресу константы	MOV ad, #d	01110101	7	3	2	(ad) ← #d
Пересылка в РПД из аккумулятора	MOV @Ri, A	1111011i	1	1	1	((Ri)) ← (A)
Пересылка в РПД прямоадресуемого байта	MOV @Ri, ad	0110011i	3	2	2	((Ri)) ← (ad)
Пересылка в РПД константы	MOV @Ri, #d	0111011i	2	2	1	((Ri)) ← #d
Загрузка указателя данных	MOV DPTR, #d16	10010000	13	3	2	(DPTR) ← #d16
Пересылка в аккумулятор байта из ПП	MOVC A, @A+DPTR	10010011	1	1	2	(A) ← ((A) + (DPTR))
Пересылка в аккумулятор байта из ПП	MOVC A, @A+PC	10000011	1	1	2	(PC) ← (PC)+1, (A) ← ((A)+(PC))
Пересылка в аккумулятор байта из ВПД	MOVX A, @Ri	1110001i	1	1	2	(A) ← ((Ri))
Пересылка в аккумулятор байта из расширенной ВПД	MOVX A, @DPTR	11100000	1	1	2	(A) ← ((DPTR))
Пересылка в ВПД из аккумулятора	MOVX @Ri, A	1111001i	1	1	2	((Ri)) ← (A)
Пересылка в расширенную ВПД из аккумулятора	MOVX @DPTR, A	11110000	1	1	2	((DPTR)) ← (A)
Загрузка в стек	PUSH ad	11000000	3	2	2	(SP) ← (SP) + 1, ((SP)) ← (ad)
Извлечение из стека	POP ad	11010000	3	2	2	(ad) ← (SP), (SP) ← (SP) - 1
Обмен аккумулятора с регистром	XCH A, Rn	11001rrr	1	1	1	(A) ↔ (Rn)
Обмен аккумулятора с прямоадресуемым байтом	XCH A, ad	11000101	3	2	1	(A) ↔ (ad)
Обмен аккумулятора с байтом из РПД	XCH A, @Ri	1100011i	1	1	1	(A) ↔ ((Ri))
Обмен младших тетрад аккумулятора и байта РПД	XCHD A, @Ri	1101011i	1	1	1	$(A_{0...3}) \leftrightarrow ((Ri)_{0...3})$

“HELLO, WORLD!”

Первая программа




```
MOV R0, #10
MOV R1, #0Ah
MOV R2, #00001010b
MOV R3, #0DDh

MOV 08h, #0DEh
END.
```

Команда **MOV** выполняет пересылку данных
из второго операнда в первый.

ГРУППА КОМАНД ПЕРЕДАЧИ ДАННЫХ

```
MOV R0, #10
MOV R1, #20
MOV A,  R0 ; поместить в аккумулятор содержимое R0

MOV 08h, #0DEh
MOV R5, 08h ; поместить в R5 содержимое ячейки ОЗУ с адресом 08h

END.
```

ГРУППА КОМАНД ПЕРЕДАЧИ ДАННЫХ

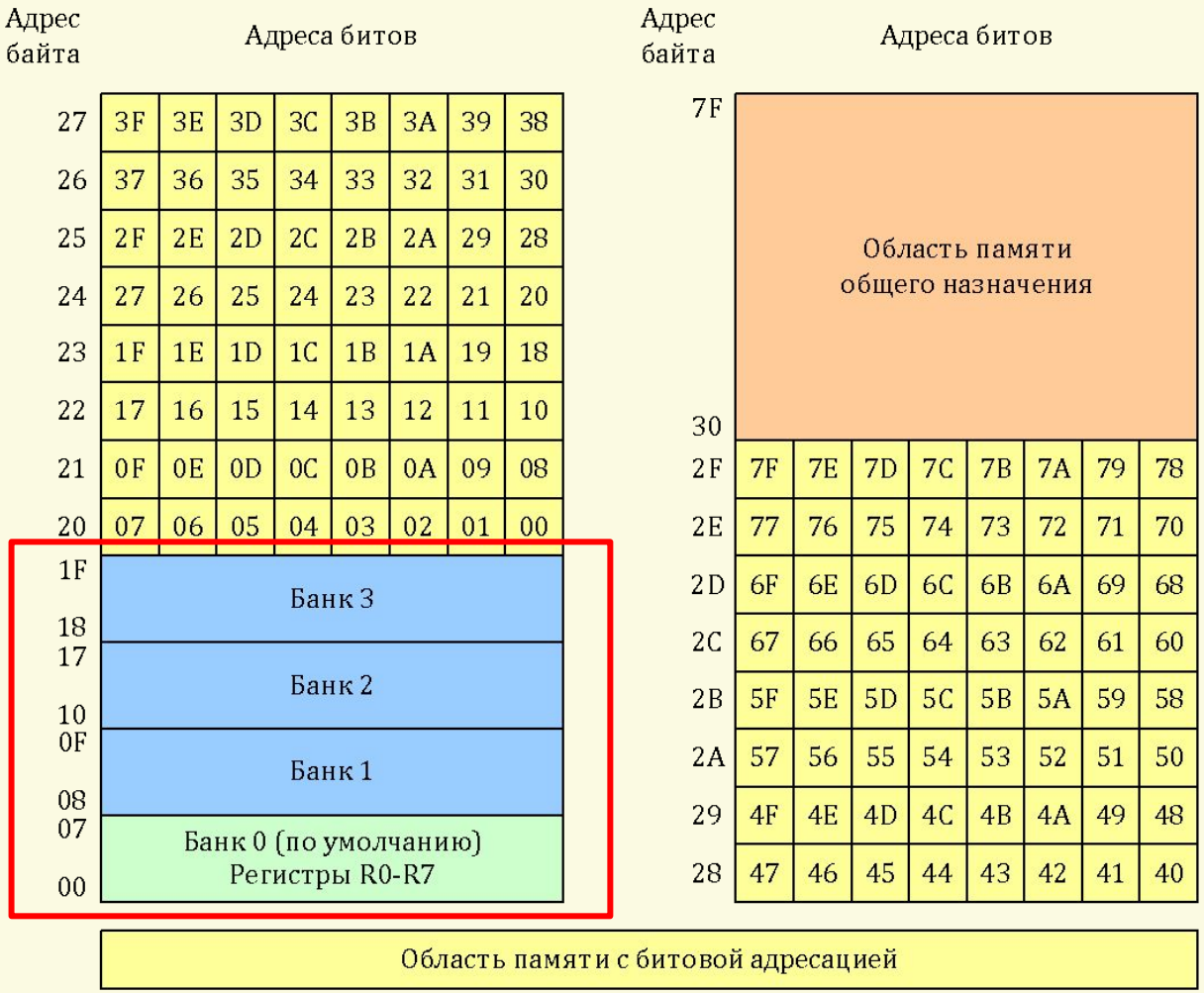
Косвенная адресация

Косвенный способ адресации предполагает указание операндов посредством адреса, содержащегося в регистре либо в регистровой паре. В команде указывается регистр, который в свою очередь указывает адрес операнда.

```
MOV 08h, #0DEh
MOV R0, #08h
MOV A,  @R0 ; поместить в аккумулятор содержимое ячейки ОЗУ , адрес которой хранится в регистре R0
END.
```

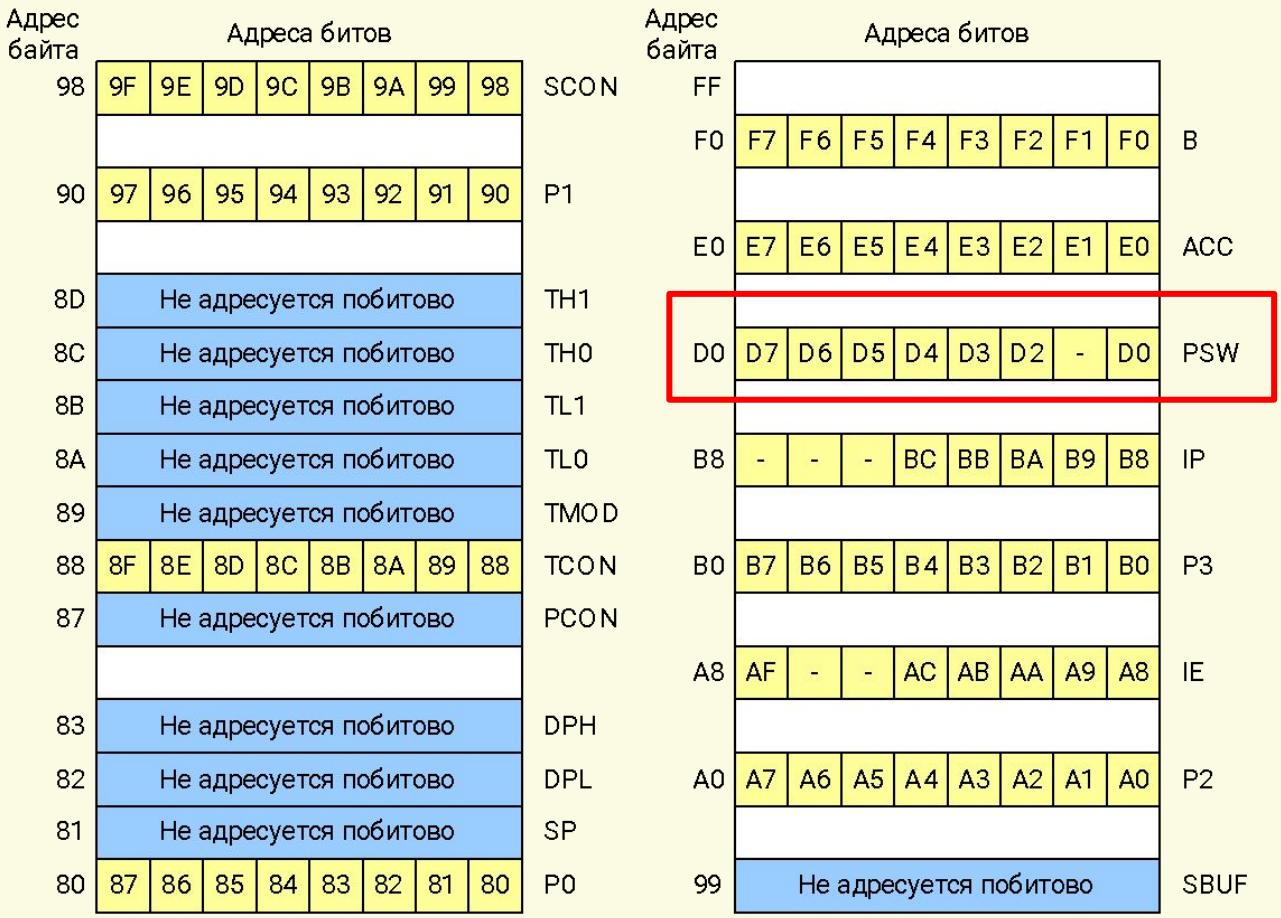
ГРУППА КОМАНД ПЕРЕДАЧИ ДАННЫХ

Карта памяти AT89S52



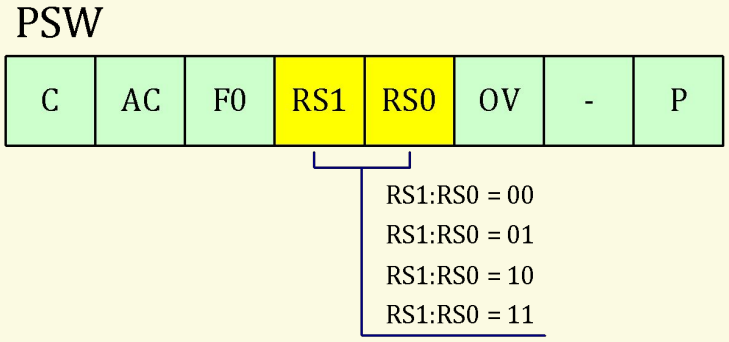
ГРУППА КОМАНД ПЕРЕДАЧИ ДАННЫХ

Карта памяти AT89S52 (продолжение)



ГРУППА КОМАНД ПЕРЕДАЧИ ДАННЫХ

Регистр слова состояния PSW



Выбор банка регистров

RS1	RS0	Границы адресов R0-R7
0	0	00h-07h
0	1	08h-0Fh
1	0	10h-17h
1	1	18h-1Fh

ГРУППА КОМАНД ПЕРЕДАЧИ ДАННЫХ

Чтение и вывод информации в порт

```
MOV P1, #0FFh ; настроили порт P1 на ввод
MOV P2, #00h ; настроили порт P2 на вывод
MOV P2, P1   ; копируем содержимое P1 в P2
END.
```

Обращение к регистрам по адресу:

```
MOV 90h, #0FFh ; настроили порт P1 на ввод
MOV 0A0h, #00h ; настроили порт P2 на вывод
MOV 0A0h, 90h   ; копируем содержимое P1 в P2
END.
```

ГРУППА КОМАНД ПЕРЕДАЧИ ДАННЫХ

Задание для самостоятельного выполнения

Реализовать программу, в результате которой содержимое регистров **R0-R7*** копируется в область ОЗУ по адресу 18h..1Fh, а затем обнуляется.

**Изначальное содержимое регистров – произвольное (отличное от 00h)*

The End.

Thank you for your attention!