

MEMORY ORGANIZATION

Internal RAM organization

R7	1F
R6	1E
R5	1D
R4	1C
R3	1B
R2	1A
R1	19
R0	18

R7	17
R6	16
R5	15
R4	14
R3	13
R2	12
R1	11
R0	10

R7	0F
R6	0E
R5	0D
R4	0C
R3	0B
R2	0A
R1	09
R0	08

R7	07
R6	06
R5	05
R4	04
R3	03
R2	02
R1	01
R0	00

2F	7F							78
2E	77							70
2D	6F							68
2C	67							60
2B	5F							58
2A	57							50
29	4F							48
28	47							40
27	3F							38
26	37							30
25	2F							28
24	27							20
23	1F							18
22	17							10
21	0F							08
20	07							00

7F
7E
.
.
.
.
.
.
.
.
32
31
30

General purpose memory

Bit addressable memory

Working Registers

Register Banks: 00h to 1Fh. The 8051 uses 8 general-purpose registers R0 through R7 (R0, R1, R2, R3, R4, R5, R6, and R7). There are four such register banks. Selection of register bank can be done through RS1,RS0 bits of PSW. On reset, the default Register Bank 0 will be selected.

Bit Addressable RAM: 20h to 2Fh . The 8051 supports a special feature which allows access to bit variables. This is where individual memory bits in Internal RAM can be set or cleared. In all there are 128 bits numbered 00h to 7Fh. Being bit variables any one variable can have a value 0 or 1. A bit variable can be set with a command such as SETB and cleared with a command such as CLR. Example instructions are:

SETB 25h ; sets the bit 25h (becomes 1)

CLR 25h ; clears bit 25h (becomes 0)

Note, bit 25h is actually bit 5 of Internal RAM location 24h.

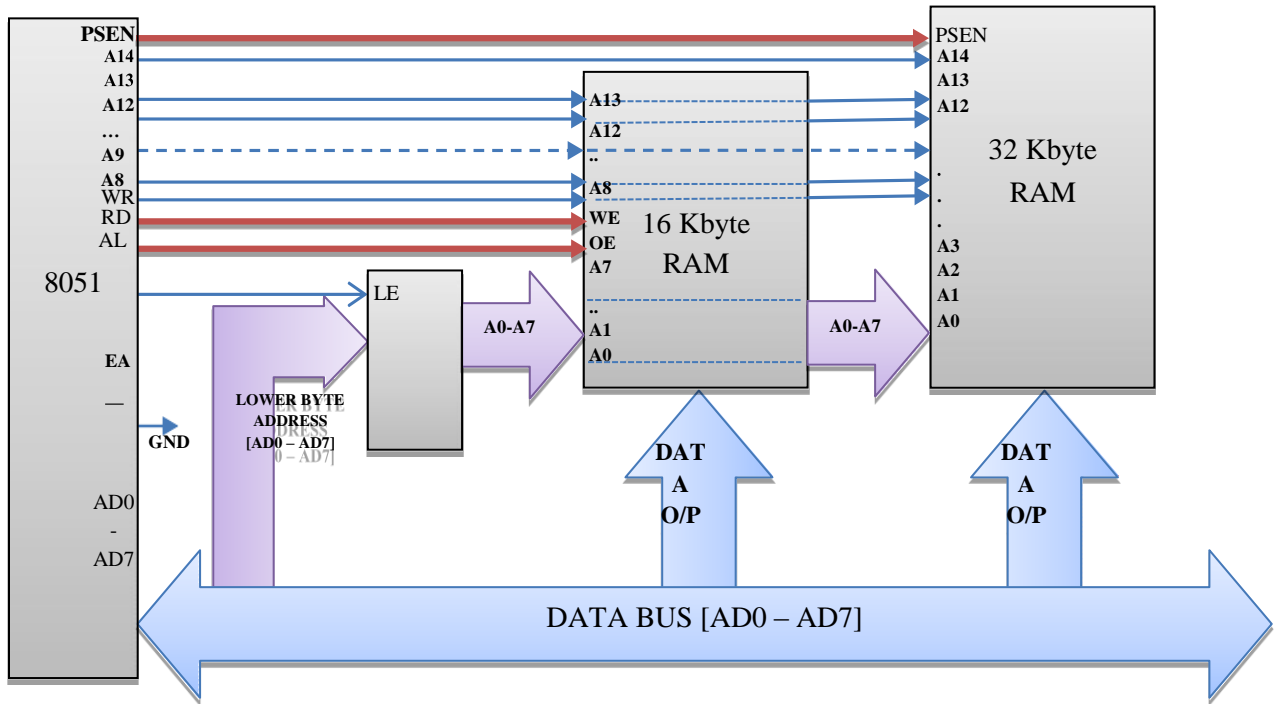
The Bit Addressable area of the RAM is just 16 bytes of Internal RAM located between 20h and 2Fh.

General Purpose RAM: 30h to 7Fh. Even if 80 bytes of Internal RAM memory are available for general-purpose data storage, user should take care while using the memory location from 00 -2Fh since these locations are also the default register space, stack space, and bit addressable space. It is a good practice to use general purpose memory from 30 – 7Fh. The general purpose RAM can be accessed using direct or indirect addressing modes.

EXTERNAL MEMORY INTERFACING

Eg. Interfacing of 16 K Byte of RAM and 32 K Byte of EPROM to 8051

Number of address lines required for **16 Kbyte memory is 14 lines** and that **of 32Kbytes of memory is 15 lines**. The connections of external memory is shown below.



The lower order address and data bus are multiplexed. De-multiplexing is done by the latch. Initially the address will appear in the bus and this latched at the output of latch using ALE signal. The output of the latch is directly connected to the lower byte address lines of the memory. Later data will be available in this bus. Still the latch output is address it self. The higher byte of address bus is directly connected to the memory. The number of lines connected depends on the memory size.

The RD and WR (both active low) signals are connected to RAM for reading and writing the data.

PSEN of microcontroller is connected to the output enable of the ROM to read the data from the memory.

EA (active low) pin is always grounded if we use only external memory. Otherwise, once the program size exceeds internal memory the microcontroller will automatically switch to external memory.

STACK

A stack is a last in first out memory. In 8051 internal RAM space can be used as stack. The address of the stack is contained in a register called stack pointer. Instructions PUSH and POP are used for stack operations. When a data is to be placed on the stack, the stack pointer increments before storing the data on the stack so that the stack grows up as data is stored (pre-increment). As the data is retrieved from the stack the byte is read from the stack, and then SP decrements to point the next available byte of stored data (post decrement). The stack pointer is set to 07 when the 8051 resets. So that default stack memory starts from address location 08 onwards (to avoid overwriting the default register bank i.e., bank 0).

Eg; Show the stack and SP for the following.

	[SP]=07	//CONTENT OF SP IS 07 (DEFAULT VALUE)	
MOV R6, #25H	[R6]=25H	//CONTENT OF R6 IS 25H	
MOV R1, #12H	[R1]=12H	//CONTENT OF R1 IS 12H	
MOV R4, #0F3H	[R4]=F3H	//CONTENT OF R4 IS F3H	
PUSH 6	[SP]=08	[08]=[06]=25H	//CONTENT OF 08 IS 25H
PUSH 1	[SP]=09	[09]=[01]=12H	//CONTENT OF 09 IS 12H
PUSH 4	[SP]=0A	[0A]=[04]=F3H	//CONTENT OF 0A IS F3H
POP 6	[06]=[0A]=F3H	[SP]=09	//CONTENT OF 06 IS F3H
POP 1	[01]=[09]=12H	[SP]=08	//CONTENT OF 01 IS 12H
POP 4	[04]=[08]=25H	[SP]=07	//CONTENT OF 04 IS 25H