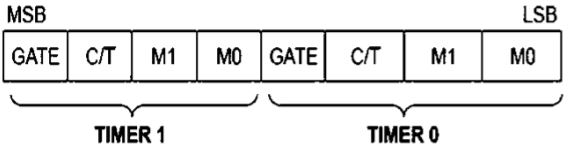
**3 & 4 UNIT- 2 MARKS**

**1. Illustrate the format of TMOD register in 8051 Timer.**



**2. List the features of 8051 serial communication.**

(i) Full duplex operation

(ii) Receiver buffered

(iii) Access using single double buffered register SBUF.

Options to use fixed or programmable baud rate.

**3. Distinguish between microprocessor and microcontrollers.**

Microprocessor: contains ALU, general purpose registers, stack pointer, program

counter, and clock timing circuit and interrupt circuit.

Microcontroller: contains the circuitry of microprocessor and in addition it has built-in

RAM, ROM, I/O devices, timers and counters.

**4. Illustrate the functions of DPTR.**

DPTR is meant for pointing to data. It is used by the 8051 to access external memory using

the address indicated by DPTR. DPTR is the only 16-bit register available and is often used

to store 2-byte values.

**5. Summarize the features of 8051 microcontrollers.**

4KB bytes on-chip program memory (ROM), 128 bytes on-chip data memory (RAM), Four

register banks, 128 user defined software flags, 8-bit bidirectional data bus, 16-bit

unidirectional address bus, 32 general purpose registers each of 8-bit.

**6. How is external memory accessed by 8051 microcontrollers?**

Up to 64 k-bytes of additional data memory can be addressed by the 8051. The external

data memory is accessed using the “MOVX” instruction. The 8051&#39;s internal data memory

is split into three sections: Lower 128 bytes, Upper 128 bytes, and SFRs.

**7. Define cross compiler.**

A compiler which can convert instructions into machine code or low-level code for a

computer other than that on which it is run.

**8. Discuss the necessity of embedded C programming instead of assembly language**

**programming.**

1. Writing the code in Assembly Language is too difficult.

2. C is very easier than the

Assembly Language and internally does nothing but it converts the code.

**9. Outline the simple C program for 8051 interrupts.**

#include &lt;REG52.H&gt;

#include &lt;stdio.h&gt;

#define BIT(x) (1 &lt;&lt; (x))

void interrupt0() interrupt 0

{

IE ^= BIT(1);

}

void serial0() interrupt 4

{

unsigned char x;

unsigned int i, z;

unsigned char yes[]=&quot; YES &quot;;

unsigned char no[]=&quot; NO &quot;;

unsigned char nvalid[]=&quot; NOT VALID TRY AGAIN ;;

while(RI==1)

**10. Summarize the structure of embedded C programming.**

A number of variables of the same type that are logically related to one another can be

grouped as an array. Working on a group rather than a collection of independent variables

allows us to arrange the data and use it more conveniently.

**11. Compare the features of microprocessor and microcontrollers.**

Microprocessor: contains ALU, general purpose registers, stack pointer, program

counter, and clock timing circuit and interrupt circuit.

Microcontroller: contains the circuitry of microprocessor and in addition it has built-in

RAM, ROM, I/O devices, timers and counters

**12. Illustrate the functions of stack.**

The stack is a section of a RAM used by the CPU to store information such as data or

memory address on temporary basis. The CPU needs this storage area considering

limited number of registers

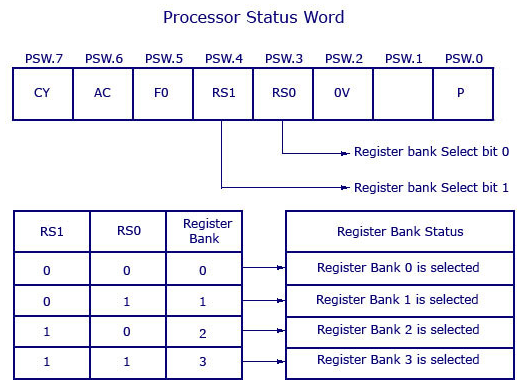
**13. List the addressing mode supported by 8051 microcontrollers.**

Immediate Addressing Mode, Register Addressing Mode, Direct Addressing

Mode, Register Indirect Addressing Mode, Indexed Addressing Mode, Implied

Addressing Mode

**14. Illustrate the format of program status word register in 8051.**



**15. Summarize the features of 8051 microcontrollers.**

4KB bytes on-chip program memory (ROM), 128 bytes on-chip data memory (RAM),

Four register banks, 128 user defined software flags, 8-bit bidirectional data bus, 16-bit

unidirectional address bus, 32 general purpose registers each of 8-bit.

**16. Build a simple C program for 8051 interrupts.**

#include &lt;REG52.H&gt;

#include &lt;stdio.h&gt;

#define BIT(x) (1 &lt;&lt; (x))

void interrupt0() interrupt 0

{

IE ^= BIT(1);

}

void serial0() interrupt 4

{

unsigned char x;

unsigned int i, z;

unsigned char yes[]=&quot; YES &quot;;

unsigned char no[]=&quot; NO &quot;;

unsigned char nvalid[]=&quot; NOT VALID TRY AGAIN &quot;;

while(RI==1)

**17. Why embedded C programming is chosen instead of choosing assembly level**

**programming?**

1. Writing the code in Assembly Language is too difficult. 2. C is very easier than the

Assembly Language and internally does nothing but it converts the code

**18. Enumerate the structure of embedded C programming.**

A number of variables of the same type that are logically related to one another can be

grouped as an array. Working on a group rather than a collection of independent variables

allows us to arrange the data and use it more conveniently.

**19. Identify the memory allocation in C.**

Each variable in C has an associated data type. Each data type requires different

amounts of memory and has some specific operations

**20. Explain about cross compiler.**

A compiler which can convert instructions into machine code or low-level code for a

computer other than that on which it is run.