

1. Qual a diferença entre microprocessador e microcontrolador?

Resp: **Microcontrolador é o componente que incorpora em um só CI todos os elementos necessários a um microcomputador:**

- **Microprocessador (ULA + Registradores + Unidade de Controle);**
- **Memórias (programa e Dados);**
- **Interfaces.**

**Já o microprocessador apresenta num único CI apenas ULA + Registradores + Unidade de Controle e com apenas ele não é possível executar nenhuma tarefa.**

**Resp: tempo gasto para executar uma instrução por completo. Pode necessitar de mais de um ciclo de máquina.**

2. Sabendo que o microcontrolador 8051 apresenta arquitetura Von Neuman Modificada, como é classificado quanto ao tipo de conjunto de Instruções?

Resp: **apresenta conjunto de instruções do tipo CISC.**

3. Qual a função do ponteiro PC? E que ocorre com ele quando o microcontrolador é “resetado”

Resp: **O PC contém o endereço da próxima instrução a ser executada e quando o microcontrolador é resetado ele é carregado com o valor do endereço a 1ª posição do mapeamento de endereço de memória de programa (EEPROM), ou seja valor 0000h**

4. Dado o programa apresentado na Figura 1, responda os itens de 4.1 a 4.7:

```
ORG 0000h
LJMP INICIO

                ORG 8008h
INICIO:        MOV DPTR,#9000h
                MOV A,#23h ;
                MOVC A, @A+DPTR
                SUBB A,#06h
                MOVX @DPTR,A
LOOP:          NOP
                LJMP LOOP
                ORG 9023
Dado:         DB 20h, 30h
                END
```

Figura 1

**RESPOSTA:** explicação o que cada instrução executa:

ORG 0000h  
LJMP INICIO

Quando liga-se o 8051  
O PC é inicializado com  
0000h, e com essa  
instrução o PC é  
carregado com 8008h

ORG 8008h ; início do programa  
INICIO: MOV DPTR,#9000h; DPTR = 9000h  
MOV A,#23h ; (A) = 23h  
MOVC A, @A+DPTR; acessa EEPROM no endereço A+DPTR = 23h + 9000h = 9023h que é 20h  
SUBB A,#06h ; subtrai do valor do acumulador( 20h) o valor 06h , (A) = 1Ah  
MOVX @DPTR,A ; armazena o valor do acumulador(1Ah) na posição da RAM externa indicada  
;pelo DPTR  
LOOP: NOP  
LJMP LOOP ; cessa a execução aqui criando um loop infinito  
ORG 9023h ; define o endereço da área de dados como 9023h  
DADO: DB 20h, 30h ; usando a diretiva DB define valores a bytes consecutivos, no ex: 20h e 30h  
END ; diretiva END define o fim físico do programa, necessário para informar ao software, que  
;vai armazenar esse programa na memória EEPROM, que é até a instrução anterior a  
;diretiva END, que o software deve armazenar a memória

Figura 1

4.1 Qual o endereço do rótulo DADOS acessada na EEPROM?

Resp: **9023h da EEPROM**

4.2 Qual instrução acessa a RAM externa? Qual endereço foi acessado? E qual ação foi executada (leitura ou escrita)?

Resp: **A RAM externa é acessada pela instrução MOVX @DPTR,A no endereço 9000h da RAM externa, e executa gravação do valor armazenado em A**

**Conteúdo Memória RAM externa:**

Endereço(h)	Conteúdo(h)
.	.
.	.
.	.
<b>9000</b>	<b>1A</b>

4.3 Quais as diretivas utilizadas no programa da figura 1 e qual sua função?

Resp: As diretivas são ORG e END e suas funções, respectivamente, são de indicar o endereço da instrução seguinte e mostrar o final físico do programa para finalizar a compilação.

4.4 Quais instruções da Figura 1 apresentam endereçamento imediato?

Resp:

MOV DPTR,#9000h

MOV A,#23h

SUBB A,#06h

4.5 Consultando os OPCODES no conjunto de instruções do 8051, qual o endereço e o conteúdo da memória EEPROM, em hexadecimal, para o programa da Figura 1

Resposta:

Conteúdo de Memória de Programa EEPROM da Figura 1:

Endereço(h)	Conteúdo(h)
0000	02
0001	80
0002	08
.	.
.	.
.	.
.	.
8008	90
8009	90
800A	00
800B	74
800C	23
800D	94
800E	06
800F	F0
8010	00
8011	02
8812	80
8813	10
.	.
.	.
.	.
9023	20
9024	30

Explicação:

LABLE	Endereço(h)	Conteúdo(h)	Instrução
INICIO:	8008	90	MOV DPTR,#9000H; Instrução de 3 bytes: o 1º contém o OPCODE (90h), o 2º contém os 16bits MSB do endereço que está na instrução, o 2º contém os 16bits LSB do endereço que está na instrução.
	8009	90	
	800A	00	
	800B	74	MOV A,#23H; Instrução de 2 bytes: 1º contém o OPCODE (74h), o 2º contém o dado de 8 bits ( 23h)
	800C	23	
	800D	94	SUBB A,#06H; Instrução de 2 bytes: 1º contém o OPCODE (94h), o 2º contém o dado de 8 bits ( 06h)
	800E	06	
	800F	F0	MOVX @DPTR,A; Instrução de 1 byte: contém apenas o OPCODE (F0h),
LOOP:	8010	00	NOP; Instrução de 1 byte: contém apenas o OPCODE (F0h),
	8011	02	LJMP LOOP; Instrução de 3 bytes: o 1º contém o OPCODE (02h), o 2º contém os 16bits MSB do endereço que está na instrução, o 2º contém os 16bits LSB do endereço que está na instrução. O endereço é representado pelo lable LOOP, cujo endereço associado é 8010h
	8012	80	
	8013	10	
.			
	9023	DA	

Observação :

O programa da Figura 1 poderia não ter a instrução NOP, como mostrado abaixo. Teria o mesmo resultado:

```

ORG 0000h
LJMP INICIO

ORG 8008H ;
INICIO: MOV DPTR,#9000H
MOV A,#23H
MOVC A, @A+DPTR
SUBB A,#06H
MOVX @DPTR,A
LOOP:  LJMP LOOP
END

```