

Nome: Mateus Alves da Rocha  
Matrícula: 11/0132661

Q1. Quais as diferenças entre os barramentos de dados e de endereços?

Os barramentos de dados são utilizados pelo microprocessador para transmitir informações entre as diferentes unidades que o compõe. Este barramento pode é bidirecional.

Os barramentos de endereços são utilizados pelo microprocessador para indicar na memória a localização do dado a ser manipulado durante a execução de alguma tarefa.

Q2. Quais são as diferenças entre as memórias RAM e ROM?

As memórias RAM permitem a gravação e regravação de dados. Entretanto, são memórias que possuem a característica de serem voláteis. Isto é, uma vez que sejam desligadas, o conteúdo armazenado nesta memória é perdido.

As memórias ROM, uma vez gravada a informação, não permite sua alteração ou remoção de dados. Porém, após desligadas não perdem os dados armazenados. Outra característica importante é que a escrita é bem mais lenta do que na memória RAM.

Q3. Considere o código abaixo, de 12 linhas:

```
1. #include <stdio.h>
2. int main(void)
3. {
4.     int i;
5.     printf("Insira um número inteiro: ");
6.     scanf("%d", &i);
7.     if(i%2)
8.         printf("%d eh impar.\n");
9.     else
10.        printf("%d eh par.\n");
11.     return 0;
12. }
```

Para este código, responda:

(a) A variável 'i' é armazenada na memória RAM ou ROM? Por quê?

Na memória RAM. O uso dessa variável faz se necessário apenas durante a execução deste programa, fazendo se desnecessário mantê-la armazenada mesmo após o desligamento da máquina. Uma outra razão, é que para a gravação na memória ROM é necessário um procedimento muito mais complexo do que apenas declarar a variável durante o código.

(b) O programa compilado a partir deste código é armazenado na memória RAM ou ROM? Por quê?

Na memória RAM. A memória ROM é geralmente programada apenas uma vez com programas que são essenciais para o funcionamento da máquina. O código listado acima não é indispensável para o funcionamento do computador, sendo necessário utilizá-los para objetivos momentâneos e descartado posteriormente.

Q4. Quais são as diferenças, vantagens e desvantagens das arquiteturas Harvard e Von Neumann?

Na arquitetura Harvard, é utilizado espaços diferentes na memória para armazenar dados da memória RAM e da memória ROM. Sendo assim, essa arquitetura possui alguns benefícios, com: leitura simultânea de dados e do programa; Barramentos de dados e programas separados e otimizados, armazena dados constantes, como por exemplo as look-up tables, na memória de programa requerendo para isso instruções específicas.

A arquitetura Neumaan, é mais simples que a arquitetura Harvard. Nesta, é utilizado o mesmo espaço de memória para a memória RAM e para a memória ROM, o que a torna menos eficiente que a arquitetura anterior, pois não permite leituras simultâneas tornando-as lentas. Entretanto, podem possivelmente ser menores e consequentemente ter um menor custo.

Q5. Considere a variável inteira 'i', armazenando o valor 0x8051ABCD. Se 'i' é armazenada na memória a partir do endereço 0x0200, como ficam este byte e os seguintes, considerando que a memória é:

(a) Little-endian?

0x0200 – receberia 0xCD;  
0x0201 – receberia 0xAB;  
0x0202 – receberia 0x51;  
0x0203 – receberia 0x80;

(b) Big-endian?

0x0200 – receberia 0x80;

0x0201 – receberia 0x51;

0x0202 – receberia 0xAB;

0x0201 – receberia 0xCD;

Q6. Sabendo que o processador do MSP430 tem registradores de 16 bits, como ele soma duas variáveis de 32 bits?

A primeira variável necessita de dois registradores para armazená-la, por exemplo R4 e R5. A segunda, de outros dois registradores: R6 e R7.

Então, o MSP430 pode executar a seguinte lógica.

**R4 + R6** – Soma-se os registradores que contém os bits menos significativos dos dois números.

Caso durante a operação aconteça um carry devido ao valor exceder o número de bits, o registrador R2 – Status register – fará o seu bit correspondente ao carry como 1.

**R5 + R7 + R2[C]** – Soma-se os registradores com o restante do número acrescido do bit de carry do registrador R2.

Ao final, caso seja um número que tenha o bit mais significativo indicando o sinal e aconteça a situação de Overflow durante a operação de soma, o bit V do registrador R2 será elevado ao nível lógico alto.