**Avaliação B1**

Nome do Aluno: Jorge Antonio Terence Novaes de Santana

Questão #01 (2 pontos)

O barramento é um conjunto de linhas de comunicação que permitem a transferência de dados entre os componentes do computador.

Pede-se:

Explique a interação entre o barramento e os vários componentes demonstrados à figura abaixo. Espera-se que você explique e demonstre exemplo de interação entre todos os componentes.

|  |
| --- |
| Diagrama  Descrição gerada automaticamente |

A ULA e a Unidade de Controle são componentes chave da estrutura Von Neumann; a primeira sendo responsável por operações aritméticas e a segunda comanda as operações descritas na memória. A memória, representada na direita, é onde os dados e comandos do programa são guardados.

Os outros blocos com um espaço branco no centro são registradores, espaços que guardam dados temporários, que podem persistir entre operações. O MBR guarda uma palavra, unidade de informação guardada na memória; o MAR guarda o endereço de memória da palavra no MBR; o PC guarda o próximo endereço de memória a ser lido; o IR contém o código de operação da palavra atual; o ACC guarda resultados de operações aritméticas.

Os barramentos transmitem dados entre os registradores e componentes chave. **Barramentos internos** (preto) trocam dados entre os registradores; **barramentos de endereços** (vermelho) buscam endereços de memória e os envia aos registradores; o **barramento de dados** (verde) transmite dados entre a Unidade de Controle, a memória e dispositivos de I/O; os **barramentos de controle** (ciano) emitem sinais que coordenam operações do sistema a todos os seus componentes.

# Questão #02 (2 pontos)

Um sistema de numeração, (ou sistema numeral) é um sistema em que um conjunto de números são representados por numerais de uma forma consistente. Pode ser visto como o contexto que permite ao numeral "11" ser interpretado como o numeral romano para dois, o numeral binário para três ou o numeral decimal para onze.

Considerando o este conceito, crie um sistema de numeração hipotético (\*), demonstrando como ele é composto, seus símbolos e um exemplo de conversão para a base 10 (demonstre o passo-a-passo).

(\*) Não serão consideradas as bases já estudadas neste curso, 16, 10 e base 2.

Vamos introduzir a base 6. Seus algarismos são 0, 1, 2, 3, 4 e 5. Qualquer base numérica não possui o número de sua base como algarismo, por isso 6 não é incluído. Seis é representado como ‘10’, pois representa 1 × 61 + 0 × 60. Podemos perceber que qualquer base funciona assim; 25610 representa 2 × 102 + 5 × 101 + 6 × 100.

Para converter de base seis para base dez, usamos essa linha de raciocínio:

34346 =

= 3 × 63 + 4 × 62 + 3 × 61 + 4 × 60 =

= 3 × 216 + 4 × 36 + 3 × 6 + 4 × 1 =

= 648 + 144 + 18 + 4 =

= 81410

Questão #03 (1 ponto)

“..se a tecnologia da aviação tivesse progredido tão depressa quanto a tecnologia de computadores, um avião custaria U$500 e daria uma volta na terra em 20 minutos com 20 litros de gasolina. Entretanto seria do tamanho de uma caixa sapato”. Esta frase foi proferida por Gordon Moore (cofundador da Intel).

Pede-se: responda as seguintes questões:

* Faça uma explanação do contexto histórico desta frase
* Faça uma análise da frase, sob o ponto de vista da história da evolução da computação: Do ábaco aos dias de hoje.

Gordon Moore é conhecido pela "Lei de Moore", que prevê que a capacidade de processamento dos computadores dobra aproximadamente a cada dois anos. Na década de 1960, a computação estava experimentando um rápido avanço devido à miniaturização dos componentes eletrônicos e ao desenvolvimento de circuitos integrados. Essa tendência continuou nas décadas seguintes, levando a uma explosão de inovação e progresso na indústria de computadores.

Ao fazer essa comparação, Moore estava destacando o ritmo impressionante de inovação na computação e especulando sobre como o mundo poderia ser diferente se outras áreas tecnológicas progredissem na mesma velocidade. Ele imaginou um cenário em que os aviões seriam acessíveis para quase qualquer pessoa.

Essa frase de Moore reflete a rápida transformação que a tecnologia da informação trouxe para o mundo, e como essa transformação moldou as expectativas em relação a outros campos tecnológicos. Embora o cenário descrito por Moore seja altamente especulativo, ele destaca o poder e o impacto da inovação tecnológica na sociedade.

Ao longo do século 21, foi possível ver uma explosão no avanço tecnológico computacional maior do que qualquer outra área. Com o avanço na capacidade do hardware, começamos a lidar com problemas cada vez mais complexos e pesados. Passamos de computadores em redes universitárias à rede mundial de computadores, a Internet e agora nossa infraestrutura é baseada em computadores distribuídos na nuvem.

# Questão #04 (3 pontos)

Considere a arquitetura do computador IAS (Modelo de John von Neumann), demonstre um exemplo de **fluxo** de processamento e os **estados dos seus registradores**, para um programa, em um linguagem hipotética, que tenha a seguinte sequencia de comandos:

* Criar variável x=20;
* Criar variável y=5;
* R = x – y;

Demonstre os estados de memória, utilizando o simulador da Universidade de Campinas (Unicamp: [*https://www.ic.unicamp.br/~edson/disciplinas/mc404/2017-2s/abef/IAS-sim/*](https://www.ic.unicamp.br/~edson/disciplinas/mc404/2017-2s/abef/IAS-sim/)).

Faça o “print” de cada ciclo (Busca/Decodificação/Execução), explicando os registradores.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| endereço | palavra | | | |
| 000 | 01 | 003 | 06 | 004 |
| 001 | 21 | 005 | 00 | 000 |
| 003 | 00 | 000 | 00 | 014 |
| 004 | 00 | 000 | 00 | 005 |
| 005 | 00 | 000 | 00 | 000 |

Cada linha representa o endereço de memória e uma palavra do programa. Cada palavra é dividida em duas instruções. Palavras que começam com 00 são interpretadas como dados numéricos, não operações.

Primeiro, o programa carrega (01) os dados do endereço 0x003, representando a variável *x* para o registrador AC. O código da operação é guardado em IR. Enquanto isso, a próxima instrução fica aguardando no registrador IBR e a palavra inteira fica em MBR.

Depois subtrai (06) disso os conteúdos do endereço 0x004, que representa a variável *y*.

Finalmente, armazena (21) o valor resultante. Executando e analisando a memória, podemos ver o valor 00F, que significa 15; e realmente. 20 – 5 = 15

# Questão #05 (2 pontos)

É fato que, frequentemente, escutamos a seguinte frase: “*Um programa é executo no sentido top-down , ele tem a característica de ser executado, de cima para baixo. Isto nos remete ao princípio da gravidade*.

Considerando que o programa abaixo foi carregado em memória, e apresenta a seguinte configuração de endereçamento :

|  |
| --- |
| **000 01 013 05 045**  **001 21 005 00 000**  **013 00 000 00 003**  **045 00 000 00 004** |

Pede-se:

Explique o que é cada linha desta sequencia acima demonstrada. Utilizar como base o simulador o simulador da (Unicamp: [*https://www.ic.unicamp.br/~edson/disciplinas/mc404/2017-2s/abef/IAS-sim/*](https://www.ic.unicamp.br/~edson/disciplinas/mc404/2017-2s/abef/IAS-sim/)).

Primeiramente, o programa carrega o valor do endereço 0x013 para AC. Depois soma-se a isso o valor presente no endereço 0x45. Finalmente, o programa armazena o resultado no endereço 0x005.

Os endereços 0x013 e 0x045 não são interpretados como palavras, mas sim como valores numéricos, pois começam com o prefixo 00, que não é mapeado para nenhum comando.