**Disciplina**: Performance em Sistemas Ciberfisicos

**Professor:** Guilherme Schnirmann

**Nome Estudante: Ary Felipe Farah e Silva**

**Atividade Prática / Relatório**

**Computador IAS**

# **Descrição da Atividade:**

O objetivo dessa atividade é entender como funciona o computador IAS. Esse computador é o primórdio dos computadores atuais, ou seja, é um computador Von Neumann, ainda que com estrutura arcaica é uma excelente ferramenta para entender os fundamentos e características do processador.

A memória do computador IAS é dividida em 4096 palavras (4k = 2¹²). Ou seja, temos uma memória com 12 bits de endereçamento. No nosso simulador o endereçamento está sendo feito em hexa, ou seja, 4 bits para cada dígito. Exemplo:

Posição 0 = 0000 0000 0000 = 000

Posição 10 = 0000 0000 1010 = 00A

Posição 1000 = 0011 1110 1011 = 3EB

Vamos utilizar um simulador desenvolvido na UNICAMP:

Estamos no nível mais baixo da arquitetura, ou seja, aqui as instruções são codificadas em linguagem de máquina. O formato da instrução da arquitetura do computador IAS (em hexadecimal):

**000** **01** **001** **05** **002**

**12 bits 20 bits 20 bits**

**Endereço instrução instrução**

**na memória**

Repare que temos os 3 primeiros dígitos representando o endereço em que as 2 próximas instruções serão armazenadas ao mapear em memória. **Cada dígito é um hexa e representa 4 bits.**

**000** **01** **001** **05** **002**

= **0000 0000 0000** **0000 0001** **0000 0000 0001** **0000 0101** **0000 0000 0010**

**0 0 0** **0 1** **0 0**  **1** **0 5** **0 0 2**

**Mapa memória opcode endereço opcode endereço**

**Atenção: os 3 primeiros dígitos não fazem parte da instrução! Lembre-se que a instrução tem 40 bits (começa no primeiro opcode).**

A seguir algumas instruções (opcodes) básicas:

**LOAD (01): carrega valor do endereço de memória no AC: AC <-M(X)**

**STOR (21); escreve valor do AC no endereço de memória M(X) <- AC**

**ADD (05); soma valor do endereço de memória no AC: AC <- AC +M(X)**

**SUB (06); subtrai valor do endereço de memória no AC: AC <- AC -M(X)**

**MUL (0B); multiplica valor do endereço de memória no MQ: <- MQ\*M(X)**

**LOAD MQ (mem.) (09); Carrega valor da memória para MQ: MQ <-M(X)**

**LOAD MQ AC (0A); Carrega valor de MQ para AC: AC<-MQ**

**DIV (0C). Divide valor de AC por valor de endereço da memória e resultado vai para MQ e resto para AC: MQ<-AC/M(X)**

**M(X) é o endereço que será o “parâmetro” na instrução do opcode.**

Para utilizar o simulador, deve-se atribuir na memória as instruções em hexadecimal. **Exemplo**:

**005 00 000 00 002** (valor atribuído em memória no endereço 005) **-DADO**

**006 3** (valor atribuído em memória no endereço 006) **-DADO**

**000 01 005 05 006;**

**- 000**: endereço de atribuição em memória (mapeamento)

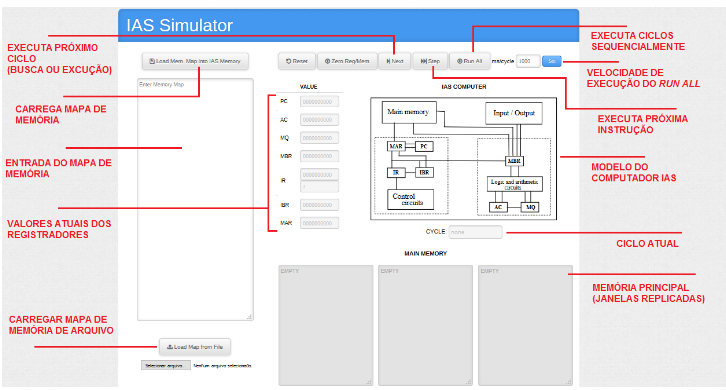
- **01**: instrução **LOAD** em hexadecimal;

- **005**: Endereço de memória de que vai ser feito o LOAD

-**05**: instrução ADD em hexadecimal

- **006**: endereço de memória de que vai ser feito o ADD

**Simulador:**



**Entrega:**

Esta atividade deverá ser entregue até o final da aula no Canvas.

O estudante deverá entregar um arquivo “.pdf” contendo as respostas da atividade proposta no roteiro.

**Roteiro da Atividade:**

1. Acesse o simulador IAS: <https://www.ic.unicamp.br/~edson/disciplinas/mc404/2017-2s/abef/IAS-sim/>
2. Considerando o exemplo inicial:

000 01 008 05 009

008 00 000 00 002

009 00 000 00 003

Copie o código em “Enter Memory Map” e clique em “load Mem. Map into IAS Memory”. **Sempre que fizer uma alteração no seu código, você vai precisar resetar os registradores e carregar novamente o mapa de memória.**

1. Execute passo-a-passo (clicando em **next**) e coloque um print de cada passo executado.
   1. Explique e registre o que cada registrador está recebendo/acessando em cada passo.

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama, Esquemático

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

* 1. Adicione uma linha no programa que armazene o resultado da soma no endereço 007. Apresente o print da memória.

R: Uma imagem contendo Ícone

Descrição gerada automaticamente

1. Considere o seguinte exemplo:

000 01 005 0b 006

001 21 007 00 000

005 00 000 00 002

006 00 000 00 003

* 1. Antes de simular, traduza esse código de máquina, o que está sendo feito e qual o resultado esperado?

Está carregando no registrador AC o valor presente na posição 005 e multiplicando com o valor que está na posição 006

Depois está guardando o resultado na posição 007.

As duas outras linhas servem para mostrar o valor em cada posição (005 = 2 e 006 = 3)

* 1. Execute passo-a-passo e explique o que está acontecendo. Você chegou no resultado esperado? Dica: Tanto a soma como a subtração são realizadas no registrador AC; Na divisão e multiplicação utilizamos o registrador MQ (de formas diferentes)

Não deu certo, pois o registrador utilizado (AC) não realiza as operações de multiplicação e divisão. Para funcionar, deveríamos usar o comando 09 (LOAD MQ) ao invés de 01 (LOAD) para usar o registrador MQ para calcular a multiplicação. Depois, para carregar o valor do MQ no AC e conseguir armazená-lo em algum espaço de memória, utilizar o comando 0A.

Por fim, utilizar o comando 21 para conseguir guardar o valor na célula 007.

000 09 005 0b 006

001 0A 000 21 007

* 1. Agora execute o seguinte código – compare com o código inicial e explique.

000 09 005 0b 006

001 0a 000 21 007

005 00 000 00 002

006 00 000 00 003

Funcionou, explicação no item b.

1. Traduza em linguagem de máquina e execute no simulador (coloque os prints).

0: LOAD 500, ADD 501

1: STOR 500

500: 2

501: 5

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama, Esquemático

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama, Esquemático

Descrição gerada automaticamente



1. Divisão:
   1. Faça a divisão de 6 por 2. O resultado fica em qual registrador?

Se não for utilizado o comando 0A, o resultado fica no registrador MQ

* 1. Armazene o resultado anterior em algum endereço de memória. Que linha de instruções foi adicionada?

000 01 020 0C 021

001 0A 000 21 022

020 00 000 00 006

021 00 000 00 002

* 1. Faça a divisão de 7 por 2. Como fica o resultado agora? Onde fica o resto?

O resultado fica armazenado no MQ e o resto em AC.

* 1. Agora faça a conta (6/2) + 4 e armazene o resultado em algum endereço de memória. O que é preciso fazer para realizar a soma? (Dica: exercício 4) Execute passo-a-passo e coloque os prints explicando cada um.

000 01 020 0C 021

001 0A 000 21 023

002 01 023 05 022

003 21 024 00 000

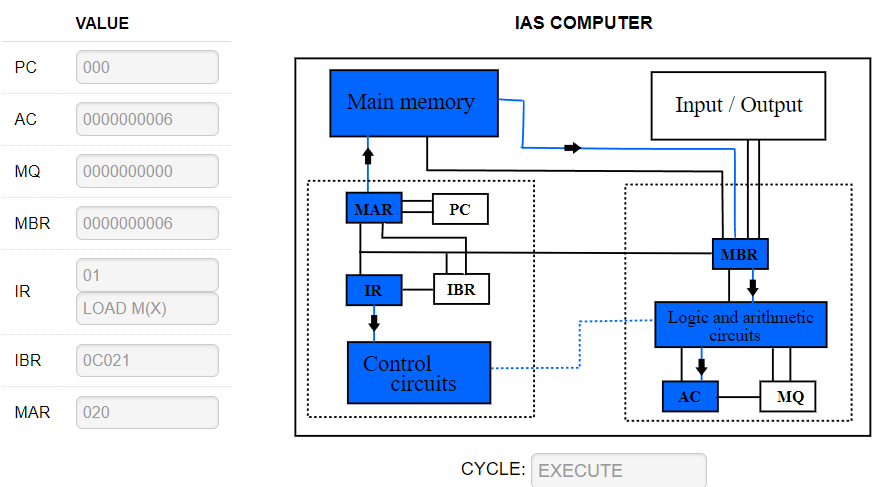
020 00 000 00 006

021 00 000 00 002

022 00 000 00 004

Diagrama

Descrição gerada automaticamente



Diagrama, Esquemático

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama, Esquemático

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama, Esquemático

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama, Esquemático

Descrição gerada automaticamente

Uma imagem contendo Logotipo

Descrição gerada automaticamente

1. Conclusão: explique o que é cada um dos registradores e qual o fluxo de movimentação de dados e instruções no computador IAS. Existem diferença em operadores utilizados para soma/subtração e divisão/multiplicação? Se você já explicou em outros itens, replique aqui ou cite onde foi explicado).

AC: registrador utilizado para soma e subtração

MQ: registrador utilizado para multiplicação e divisão

MBR: registrador de dado na memória

PC -> MBR -> IR (Codifica instrução) / IBR (Próxima Instrução) -> MAR (Identifica o valor) -> AC/MQ -> Main Memory

Os operadores para carregar os valores em AC ou MQ mudam (01 ou 09)