**Disciplina**: Performance em Sistemas Ciberfisicos

**Professor:** Guilherme Schnirmann

**Nome Estudante:**

**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)

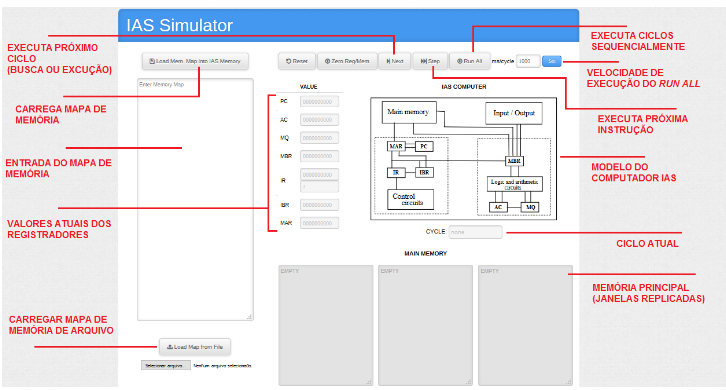
- **01**: instrução **LOAD** em hexadecimal;

- **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:

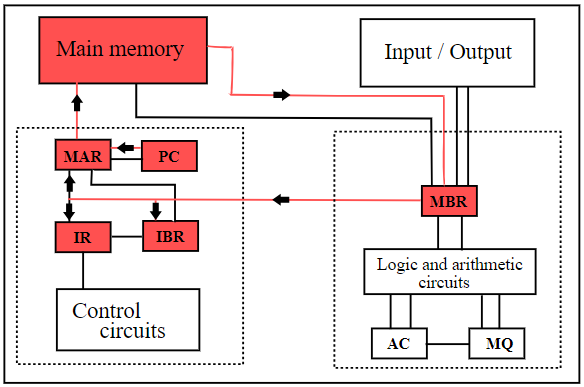
000 01 008 05 009

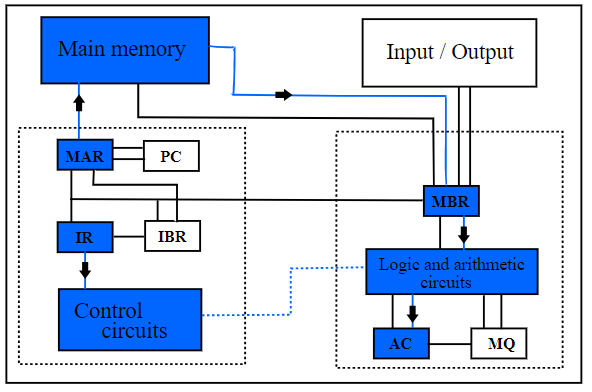
008 00 000 00 002

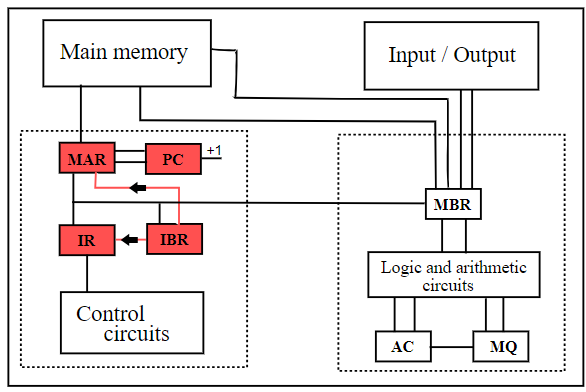
009 00 000 00 003

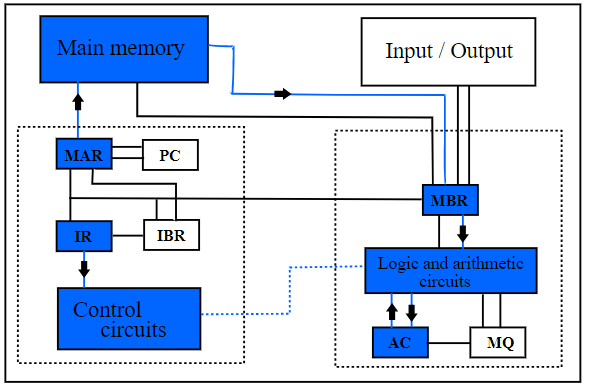
Copie o código em “Enter Memory Map” e clique em “load Mem. Map into IAS Memory”. **Sempre que fizer uma alteração no seu código, você vai precisar resetar os registradores e carregar novamente o mapa de memória.**

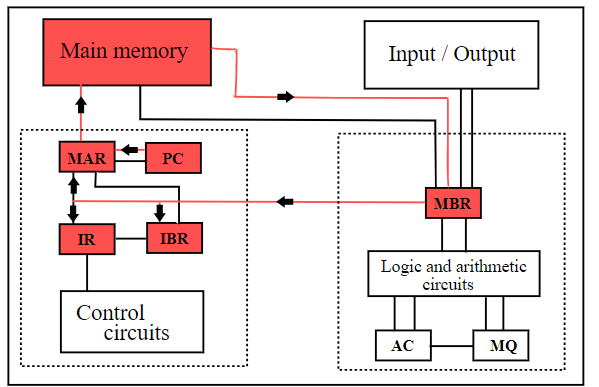
1. Execute passo-a-passo (clicando em **next**) e coloque um print de cada passo executado.
   1. Explique e registre o que cada registrador está recebendo/acessando em cada passo.



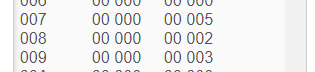








* 1. Adicione uma linha no programa que armazene o resultado da soma no endereço 007. Apresente o print da memória.



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?

- no primeiro comando ele multiplica as posições 5 e 6 depois ele armazena no endereço 7, as linhas de baixo são os valores para o 5 e o 6

* 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 )

- o código não funciona pois a instrução01 pega um valor AC que não funciona na multiplicação por ele precisa de um valor MQ

* 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

- Utilizando o comando 09 a multiplicação funciona pois ele recebe o valor como mq

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

000 01 1f4 05 501

001 21 1f4 00 000

1f4 00 000 00 002

1f5 00 000 00 005



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

000 01 005 0C 006

001 0A 000 21 007

005 00 000 00 006

006 00 000 00 002

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

000 01 005 0C 006

001 0A 000 21 007

005 00 000 00 006

006 00 000 00 002

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

000 01 005 0C 006

001 0A 000 21 007

005 00 000 00 007

006 00 000 00 002

* 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 01 007 0C 008

001 0A 000 21 009

002 01 009 05 006

003 01 000 00 00A

006 00 000 00 004

007 00 000 00 006

008 00 000 00 002

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).

No computador IAS, os registradores desempenham papéis específicos no processamento de dados e instruções:

Registrador de Memória (M): Armazena temporariamente dados da memória principal.

Acumulador (A): Armazena resultados de operações aritméticas e lógicas.

Registrador de Instrução (IR): Contém a instrução atual em execução.

Contador de Instrução (PC): Mantém o endereço da próxima instrução a ser executada.

Registrador de Endereço (MAR): Armazena o endereço de memória sendo acessado.

O fluxo de dados e instruções segue este padrão:

Instruções são buscadas na memória pelo endereço no PC.

A instrução é decodificada e executada usando operandos armazenados nos registradores.

O resultado é armazenado de volta na memória ou em registradores conforme necessário.