



# Projeto de Arquitetura e Organização de Computadores

## Montador e Simulador IJVM

### Professor: Danilo Alves

Equipe:

Andressa Gomes

Carlos Augusto

Stefane Adna

Cleton Moraes

# Introdução.

- Nível de Micro Arquitetura: Uma microarquitetura é a forma como um determinado número de instruções em nível ISA serão implementadas no processador.
- Toda Microarquitetura é dividida em uma Parte Operativa e uma Parte de Controle.

# Exemplo de uma Microarquitetura- MIC 1

- O **MIC-1** é uma arquitetura de processador que consiste em uma unidade de controle muito simples que executa o microcódigo criado utilizando-se instruções da IJVM.
- Essa arquitetura tem a função de interpretar instrução por instrução do nível ISA.

## Foto do MIC-1 da IJVM

Mic-1 MMV (mic1ijvm.mic1) -

File Preferences Microcode Store Assemble/Load About

**C Bus**      **B Bus**

MAR ← 00000000 →

MDR ← 00000000 →

PC ← ffffffff →

MBR ← 00 →

SP → 00008010

LV → 00008000

CPP → 00004000

TOS → 00000000

OPC → 00000000

H → 00000000

**A Bus**      **N: 0**      **Z: 0**

00000000

**Shifter**      **SLL8: 0**      **SRA1: 0**

00000000

**Method Area**

0000: 00	0008: 00	0010: 00
0001: 00	0009: 00	0011: 00
0002: 00	000a: 00	0012: 00
0003: 00	000b: 00	0013: 00
0004: 00	000c: 00	0014: 00
0005: 00	000d: 00	0015: 00
0006: 00	000e: 00	0016: 00
0007: 00	000f: 00	0017: 00

< >

**Constant Pool**

00010000: 00000000	00010008: 00000000	00010010: 00000000
00010004: 00000000	0001000c: 00000000	00010014: 00000000

< >

**Stack Area**

00020000: 00000000	00020010: 00000000	00020020: 00000000
00020004: 00000000	00020014: 00000000	00020024: 00000000
00020008: 00000000	00020018: 00000000	00020028: 00000000
0002000c: 00000000	0002001c: 00000000	0002002c: 00000000

< >

Delay: ☒ Off ☐ On    Speed: ☐ SubClock ☒ Clock ☐ IJVM ☐ Prog

Reset    ←    →    ⏸

**MPC: 0x0000: goto 0x2**

**MIR**      **ffff:**

2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	J	J	J	S	S	F	E	E	I	I	C	W	R	F	B
D	M	A	A	L	R		N	N	N	N		R	D	E	
R	P	M	M	L	A		A	B	V	C				T	
	C	N	Z	8	1	(2)			A		(9)				(4)

**Input Console**

**Output Console**

# O que é a IJVM ?

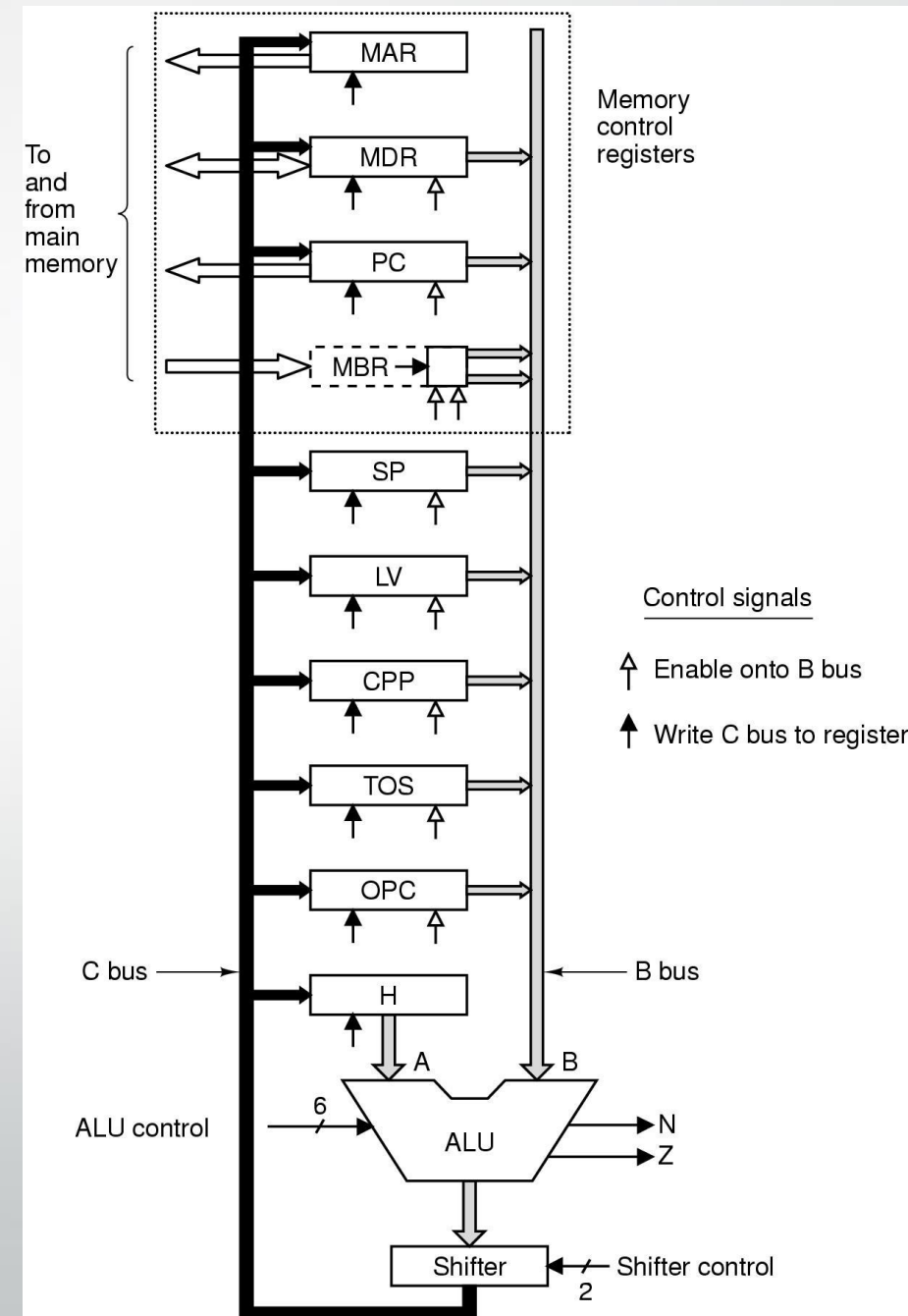
- **O IJVM** é conjunto de instruções que são usadas para ensinar noções básicas de linguagem de montagem.
- Esse conjunto de microinstruções geralmente são escritos em mnemônicos, e ao serem lidos pela microarquitetura, são então convertidas para o seu equivalente em hexadecimal, para que mais adiante possam ser executado

Mnemônico da IJVM	Equivalente em Hexadecimal	Significado
BIPUSH byte	0X10	Colocar um byte na pilha
ISTORE variável	0x36	Retirar a palavra do topo da pilha e armazenar essa palavra em uma variável local
ILOAD	0x15	Coloca uma variável no topo da pilha
IADD	0x60	Retira da pilha as duas palavras do topo; coloca no topo da pilha o resultado da soma dessas palavras
ISUB	0x64	Retira da pilha as duas palavras do topo; coloca no topo da pilha o resultado da subtração dessas palavras
DUP	0x59	Copia a palavra do topo da pilha e coloca a cópia no topo da pilha
GOTO deslocamento	0xA7	Desvio incondicional
IFLT deslocamento	0x9B	Retira da pilha a palavra do topo; desvia se ela for menor que zero
IOR	0x80	Retira da pilha as duas palavras do topo; coloca no topo da pilha o resultado da operação OR booleana dessas palavras
IAND	0x7E	Retira da pilha as duas palavras do topo; coloca no topo da pilha o resultado da operação AND booleana dessas palavras
NOP	0x00	Não faz nada

# Registadores da MIC-1

Pode ser notado que os quatro primeiros registradores (MAR, MDR, PC e MBR) são registradores que trabalham diretamente com os endereços na memória, onde:

- MAR e PC apenas enviam para a memória.
- MBR recebe e é de comunicação exclusiva com a memória.
- Já MDR trabalha tanto enviando como recebendo dados referentes à memória.



Assim, temos os seguintes registradores e suas funções:

- **MAR:** Registrador de Endereço de Memória para o barramento 32 bits.
- **MDR:** Registrador de Dados de Memória para o barramento de 32 bits, apontado por MAR.
- **MBR:** Registrador de Dados de Memória para o barramento de 8 bits, apontado por PC.
- **PC:** Contador de Programa. Aponta para a Memória em um Barramento de 8 bits, que contém instruções.
- **SP:** Apontador de Pilha.
- **LV:** Apontador para a base das Variáveis Locais, localizada na pilha.
- **CPP:** Aponta para o POOL de constantes e apontadores para outras áreas da memória.
- **TOS e OPC:** Registradores Temporários.
- **H:** Acumulador.





# Apresentação do Projeto

# Referências

- <https://jansebp.wordpress.com/2013/08/20/microarquitecturas-mic-1-e-mic-2-parte-i-apresentacao/>
- <http://www.dpi.inpe.br/~carlos/Academicos/Cursos/ArqComp/aula7.html>
- <https://en.wikipedia.org/wiki/IJVM>
- <https://pt.wikipedia.org/wiki/Microarquitectura>