

UNIVERSIDADE FEDERAL DE PELOTAS – UFPEL  
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO (CDTec)  
CURSOS DE CIÊNCIA DA COMPUTAÇÃO E ENGENHARIA DE COMPUTAÇÃO  
DISCIPLINA DE PROGRAMAÇÃO DE SISTEMAS

PROFs.: Me. ANDERSON PRIEBE FERRUGEM

PRIMEIRO TRABALHO  
TÓPICO: SIMULADOR DE MÁQUINA.

O TRABALHO SERÁ UMA APRESENTAÇÃO EM VÍDEO DO GRUPO COM TODOS PARTICIPANTES COM CÓDIGO DISPONIBILIZADO VIA GITHUB; O ENVIO É FEITO APENAS POR COMPONENTE DO GRUPO; A DURAÇÃO MÁXIMA DO VÍDEO DEVERÁ SER DE **20 MIN** COM **TOLERÂNCIA DE 5 MIN. (15-25)** ;

**A APRESENTAÇÃO DEVERÁ MOSTRAR:**

- 1) INTERAÇÃO ENTRE OS COMPONENTES;**
- 2) ARGUIÇÃO DO FUNCIONAMENTO E DAS TÉCNICAS USADAS.**

**A APRESENTAÇÃO NÃO DEVERÁ SER APENAS:**

- 1) APRESENTAÇÃO DE SLIDES;**
- 2) APRESENTAÇÕES INDIVIDUAIS DOS COMPONENTES DO GRUPO.**

EM CASO DE DÚVIDAS SOBRE A APRESENTAÇÃO PROCUREM POSTAR NO E-AULAS (DESTA FORMA A RESPOSTA FICA DISPONÍVEL A TODOS).

## FERRAMENTAS:

## SOFTWARE:

JAVA

Apresentação gráfica da execução !!!

---

# Projeto de um simulador de sistema computacional hipotético

---

## Parte I

### Projeto de um simulador SIC/XE, computador hipotético baseado na arquitetura do livro :Sytems Software: An Introduction to System Programming; Leland L. Beck

---

## Introdução

O trabalho descrito a seguir consiste em implementar um **simulador** para um Computador Hipotético (SIC/XE), conforme apresentado no livro **Sytems Software: An Introduction to System Programming** de **Leland L. Beck**, com alterações e complementos de algumas funções. Tal sistema será composto de **dois** módulos que deverão operar de forma integrada: o **executor** (**simulador** propriamente dito) e uma **interface visual**.

O resultado do trabalho deverá ser entregue com toda a documentação (programas fontes, programa executável, documentação formal sucinta das estruturas de dados definidas, das funções desenvolvidas e estratégias adotadas) pelo Github e

A avaliação do trabalho será realizada com base nos seguintes aspectos:

- **correção do programa,**
- **adequação das definições adotadas,**
- **uso das técnicas básicas de programação,**
- **autenticidade e domínio sobre o produto gerado,**

## Descrição Geral

### 1. Memória

A memória do computador é definida pelos seguintes atributos:

<b>Tamanho da memória</b>	Indefinido (não menor que 1 KB)
<b>Palavra de memória</b>	24 bits (3 bytes – cada byte com 8 bits)
<b>Unidade de endereçamento</b>	<u>Palavra</u>
<b>Bit de paridade</b>	<NA>
<b>Cache</b>	<NA>
<b>Observações adicionais:</b> <NA> significa "Não se aplica".	

## 2. Registradores

Esta arquitetura apresenta um conjunto reduzido de 7 registradores, cujos tamanhos são de 24 bits e 48 bits, conforme especificado a seguir.

### Registradores Básicos:

Os registradores de propósito pré-definido, que dão suporte às funcionalidades da arquitetura e são acessados apenas pela unidade de controle, estão listados a seguir em tabela de registradores básicos (primários).

Ident.	Registrador	Número	Tamanho (bits)	Descrição
<b>A</b>	Acumulador	<b>0</b>	<b>24</b>	Armazena os dados (carregados e resultantes) das operações da Unid. de Lógica e Aritmética
<b>X</b>	Registrador de índice	<b>1</b>	<b>24</b>	Usado para endereçamento.
<b>L</b>	Registrador de ligação	<b>2</b>	<b>24</b>	A instrução Jump to Subroutine (JSUB) armazena o endereço de retorno nesse registrador.
<b>B</b>	Registrador Base	<b>3</b>	<b>24</b>	Usado para endereçamento.
<b>S</b>	Registrador de uso geral	<b>4</b>	<b>24</b>	<NA>
<b>T</b>	Registrador de uso geral	<b>5</b>	<b>24</b>	<NA>
<b>F</b>	Acumulador de ponto flutuante	<b>6</b>	<b>48</b>	Armazena os dados (carregados e resultantes) das operações da Unid. de Lógica e Aritmética em ponto flutuante.
<b>PC</b>	Contador de Instruções ( <i>Program Counter</i> )	<b>8</b>	<b>24</b>	Mantém o endereço da próxima instrução a ser executada
<b>SW</b>	Palavra de <i>status</i>	<b>9</b>	<b>24</b>	Contém várias informações, incluindo código condicional (CC)

### 3. Modos de Endereçamento

- direto o valor do operando está no endereço de destino determinado diretamente usando as informações de deslocamento / endereço
- indireto o valor do operando está no endereço de destino armazenado no endereço fornecido pelo cálculo do endereço direto
- imediato Valor do operando é o valor do cálculo do endereço direto (o endereço de destino é o registro de instrução na CPU)

Instruções de 4-byte estão em azul (e=1)

Instruções simples do SIC estão em verde (n =0, i =0)

Modo	Flag bits n i x b p e	Descrição
<hr/>		
<b>Direto</b>		
	1 1 0 0 0 0	Deslocamento de 12 bit é o endereço destino.
	1 1 0 0 0 1	Endereço de 20 bit é o endereço destino.
	1 1 0 0 1 0	Deslocamento de 12 bit complemento de 2 a partir do PC.
	1 1 0 1 0 0	Deslocamento de 12 bit base sem sinal a frente de B (deslocamento base)
	1 1 1 0 0 0	Registrador de índice X somado ao endereço direto para apontar o endereço destino.
	1 1 1 0 0 1	Registrador de índice X somado ao endereço direto para apontar o endereço destino.
	1 1 1 0 1 0	Registrador de índice X somado ao PC para apontar o endereço destino.
	1 1 1 1 0 0	Registrador de índice X somado ao deslocamento base computado aponta o endereço destino.
	0 0 0 - - -	Instrução do SIC. Últimos 15 bits são o endereço
	0 0 1 - - -	Registrador de índice X somado ao endereço direto para apontar o endereço destino.
<b>Indireto</b>		
	1 0 0 0 0 0	Endereço de memória computado contém o endereço de destino.
	1 0 0 0 0 1	Endereço de memória computado contém o endereço de destino.
	1 0 0 0 1 0	Endereço de memória computado contém o endereço de destino.
	1 0 0 1 0 0	Endereço de memória computado contém o endereço de destino.
<b>Imediato</b>		
	0 1 0 0 0 0	Endereço de memória computado é o operando.(endereço destino está na instrução)
	0 1 0 0 0 1	Endereço de memória computado é o operando.(endereço destino está na instrução)
	0 1 0 0 1 0	Endereço de memória computado é o operando.(endereço destino está na instrução)
	0 1 0 1 0 0	Endereço de memória computado é o operando.(endereço destino está na instrução)

4. Formato de instruções **SIC/XE**

1 byte format

op (8 bits)
-------------

2 byte format

r1 (8 bits)	r2 (8 bits)
-------------	-------------

3 byte format

op (6 bits)	n	i	x	b	p	e	disp (12 bits)
-------------	---	---	---	---	---	---	----------------

4 byte format

op (6 bits)	n	i	x	b	p	e	address (20 bits)
-------------	---	---	---	---	---	---	-------------------

## 5. Conjunto de Instruções

A seguir está definido o conjunto de instruções reconhecido pelo computador, acompanhado de todas as informações necessárias para sua implementação.

Cada código de instrução (*opcode*) e operando (opd1 ou opd2) ocupa o espaço associado ao formato da instrução. As ações dizem respeito aos registradores, conforme identificação definida na tabela de registradores e endereços de memória referenciados.

**As instruções em vermelho não devem ser implementadas.**

P = privilegiado, C = conjunto CC (<, =, >), F = ponto flutuante

Consulte o Apêndice A do livro para obter informações sobre formatos de instrução e modos de endereçamento.

Mnemônico	Operandos	Formato	Cód. de Máq. (opcode)	Ação	Observações	Tipo
ADD	m	$\frac{3}{4}$	18	$A \leftarrow (A) + (m..m+2)$		
ADDF	m	$\frac{3}{4}$	58	$F \leftarrow (F) + (m..m+5)$		F
ADDR	r1,r2	2	90	$r2 \leftarrow (r2) + (r1)$		
AND	m	$\frac{3}{4}$	40	$A \leftarrow (A) \& (m..m+2)$		
CLEAR	r1	2	4	$r1 \leftarrow 0$		
COMP	m	$\frac{3}{4}$	28	$A : (m..m+2)$		C
COMPF	m	$\frac{3}{4}$	88	$F : (m..m+5)$		CF
COMPR	r1,r2	2	A0	$(r1) : (r2)$		C
DIV	m	$\frac{3}{4}$	24	$A : (A) / (m..m+2)$		
DIVF	m	$\frac{3}{4}$	64	$F : (F) / (m..m+5)$		F
DIVR	r1,r2	2	9C	$(r2) \leftarrow (r2) / (r1)$		
FIX		1	C4	$A \leftarrow (F)$	converte para inteiro	
FLOAT		1	C0	$F \leftarrow (A)$ [convert to floating]	converte para flutuante	F
HIO		1	F4	Halt I/O	canal número (A)	P
J	m	$\frac{3}{4}$	3C	$PC \leftarrow m$		
JEQ	m	$\frac{3}{4}$	30	$PC \leftarrow m$ if CC set to =		
JGT	m	$\frac{3}{4}$	34	$PC \leftarrow m$ if CC set to >		
JLT	m	$\frac{3}{4}$	38	$PC \leftarrow m$ if CC set to <		
JSUB	m	$\frac{3}{4}$	48	$L \leftarrow (PC); PC \leftarrow m <$		
LDA	m	$\frac{3}{4}$	0	$A \leftarrow (m..m+2)$		
LDB	m	$\frac{3}{4}$	68	$B \leftarrow (m..m+2)$		
LDCH	m	$\frac{3}{4}$	50	$A$ [byte mais a direita] $\leftarrow (m)$		
LDF	m	$\frac{3}{4}$	70	$F \leftarrow (m..m+5)$		F
LDL	m	$\frac{3}{4}$	8	$L \leftarrow (m..m+2)$		
LDS	m	$\frac{3}{4}$	6C	$S \leftarrow (m..m+2)$		
LDT	m	$\frac{3}{4}$	74	$T \leftarrow (m..m+2)$		
LDX	m	$\frac{3}{4}$	4	$X \leftarrow (m..m+2)$		
LPS	m	$\frac{3}{4}$	D0		Carrega o status do processador a partir das informação no endereço m	P
MUL	m	$\frac{3}{4}$	20	$A \leftarrow (A) * (m..m+2)$		
MULF	m	$\frac{3}{4}$	60	$F \leftarrow (F) * (m..m+5)$		
MULR	r1,r2	2	98	$r2 \leftarrow (r2) * (r1)$		
NORM		1	C8	$F \leftarrow (F)$	normalizado	
OR	m	$\frac{3}{4}$	44	$A \leftarrow (A)   (m..m+2)$		
RD	m	$\frac{3}{4}$	D8	$A$ [byte mais a direita] $\leftarrow$ dado de dispositivo especi		P
RMO	r1,r2	2	AC	$r2 \leftarrow (r1)$		
RSUB		$\frac{3}{4}$	4C	$PC \leftarrow (L)$		
SHIFTL	r1,n	2	A4	$r1 \leftarrow (r1)$	Deslocamento a esquerda de n bit	
SHIFTR		2	A8	$r1 \leftarrow (r1)$	Deslocamento a direita de n bit	
SIO		1	F0	Start I/O channel number (A)	address of channel program is given by (S)	P
SSK	m	$\frac{3}{4}$	EC	Protection key for address m		P
STA	m	$\frac{3}{4}$	0C	$m..m+2 \leftarrow (A)$		
STB	m	$\frac{3}{4}$	78	$m..m+2 \leftarrow (B)$		
STCH	m	$\frac{3}{4}$	54	$m \leftarrow (A)$	byte mais a direita	
STF	m	$\frac{3}{4}$	80	$m..m+5 \leftarrow (F)$		F
STI	m	$\frac{3}{4}$	D4	Interval timer value $\leftarrow (m..m+2)$		P
STL	m	$\frac{3}{4}$	14	$m..m+2 \leftarrow (L)$		
STS	m	$\frac{3}{4}$	7C	$m..m+2 \leftarrow (S)$		
STSW	m	$\frac{3}{4}$	E8	$m..m+2 \leftarrow (SW)$		P
STT	m	$\frac{3}{4}$	84	$m..m+2 \leftarrow (T)$		
STX	m	$\frac{3}{4}$	10	$m..m+2 \leftarrow (X)$		
SUB	m	$\frac{3}{4}$	1C	$A \leftarrow (A) - (m..m+2)$		
SUBF	m	$\frac{3}{4}$	5C	$F \leftarrow (F) - (m..m+5)$		F
SUBR	r1,r2	2	94	$r2 \leftarrow (r2) - (r1)$		
SVC	n	2	B0	Generate SVC interrupt. [for asse		
TD	m	$\frac{3}{4}$	E0	Test device specified by (m)		PC
TIO		1	F8	Test I/O channel number (A)		PC
TIX	m	$\frac{3}{4}$	2C	$X \leftarrow (X) + 1; (X) : (m..m+2)$		C
TIXR	r1	2	B8	$X \leftarrow (X) + 1; (X) : (r1)$		C
WD	m	$\frac{3}{4}$	DC	Dispositivo (m) $\leftarrow (A)$	byte mais a direita especificado por m	P

## **Bibliografia**

BECK, Leland. **System Software: An Introduction to Systems Programming.** Addison-Wesley, 1997

CALINGAERT, Peter. **Assemblers, Compilers, and Program Translation.** Potomac: Computer Science Press, Inc, 1979.

STALLINGS, Willian. **Computer Organization and Architecture.** 5.ed. New Jersey: Prentice Hall, 1999.

TANENBAUM, Andrew. **Structured Computer Organization.** 4.ed. New Jersey: Prentice Hall, 1999.