TRABALHO PRÁTICO - Parte I
Projeto de um Emulador da Arquitetura de um Computador hipotético baseado na arquitetura do livro do Calingaert

Introdução

O trabalho descrito a seguir consiste em implementar um Emulador para um Computador Hipotético, conforme apresentado no livro do Calingaert (livro texto), com alterações e complementos de algumas funções. Tal sistema será composto de **dois** módulos que deverão operar de forma integrada: o **executor** (emulador propriamente dito) e uma **interface visual**.
O trabalho deverá ser realizado **em duplas**, utilizando a **escolhida em aula** e algum de seus compiladores disponíveis nos Laboratórios.
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) **até** a data e hora limite de **12/05/2010 às 24:00 h**, e apresentado em sala de aula, conforme solicitado pelo professor.
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,**
- **observação do prazo e dos itens para entrega, etc**.

Descrição Geral

1. Memória

A memória do computador é definida pelos seguintes atributos:

Tamanho da memória	Indefinido (não menor que 1 KB)
Palavra de memória	16 bits
Unidade de endereçamento	Palavra
Bit de paridade	<NA>
Cache	<NA>
Observações adicionais: <NA> significa "Não se aplica".	

2. Registradores

Esta arquitetura apresenta um conjunto reduzido de 6 registradores, cujos tamanhos são de 16 bits e 8 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	Tamanho (bits)	Descrição
PC	Contador de Instruções (*Program Counter*)	**16**	Mantém o endereço da próxima instrução a ser executada
SP	Ponteiro de Pilha (*Stack Pointer*)	**16**	Aponta para o topo da pilha do sistema; tem incremento/decremento automático (*push*/*pop*)

Demais Registradores:

A lista seguinte mostra os demais registradores implementados no computador hipotético e sua descrição.

Ident.	Registrador	Tamanho (bits)	Descrição
ACC	Acumulador	**16**	Armazena os dados (carregados e resultantes) das operações da Unid. de Lógica e Aritmética
MOP	Modo de Operação	**8**	Armazena o indicador do modo de operação, que é alterado apenas por painel de operação (via console de operação - interface visual)
RI	Registrador de Instrução	**16**	Mantém o *opcode* da instrução em execução (registrador interno)
RE	Registrador de Endereço de Memória	**16**	Mantém o endereço de acesso à memória de dados (registrador interno)

3. Modos de Endereçamento

Os vários modos de endereçamento disponíveis nessa arquitetura estão listados a seguir, juntamente com suas utilizações:

Modos de Endereçamento	Identificação e Utilização
Direto	Trivial
Indireto	Bit 5 e/ou 6 do código da instrução ligado (código da instrução = código básico [+ 32] [+ 64], conforme seja o primeiro e/ou o segundo operando); o operando indica onde está o endereço do dado ou da instrução
Imediato	Bit 7 do código da instrução ligado (código da instrução = código básico + 128); o operando é o próprio dado
Indexado	<NA>
Observações adicionais:	

4. 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 uma palavra de memória. As ações dizem respeito aos registradores, conforme identificação definida na tabela de registradores e endereços de memória referenciados. As observações sinalizadas se são descritas na legenda abaixo do quadro.

Mnemônico (Sugerido)	**Cód. de Máq.* (opcode)**	**Tam. da Instrução (palavras)**	**N° de Operan-dos**	**Ação (comentário) ****	**Modos de endereçamento (D/In/Im)**	**Obser-vações**
ADD	02	2	1	ACC← ACC + opd1	D/In/Im	(#)
BR	00	2	1	PC← opd1	D/In	
BRNEG	05	2	1	PC← opd1, se ACC < 0	D/In	
BRPOS	01	2	1	PC← opd1, se ACC >0	D/In	
BRZERO	04	2	1	PC← opd1, se ACC = 0	D/In	
CALL	15	2	1	[SP]← PC; PC← opd1 (desvio para sub-rotina opd1)	D/In	(%)
COPY	13	3	2	opd1← opd2	opd1: D/In opd2: D/In/Im	(#)
DIVIDE	10	2	1	ACC← ACC / opd1	D/In/Im	(#)
LOAD	03	2	1	ACC← opd1	D/In/Im	(#)
MULT	14	2	1	ACC← ACC * opd1	D/In/Im	(#)
READ	12	2	1	opd1← *input stream*	D/In	
RET	16	1	0	PC← [SP] (retorno de sub-rotina)	-	(%)
STOP	11	1	0	término (fim) de execução	-	
STORE	07	2	1	opd1← ACC	D/In	
SUB	06	2	1	ACC← ACC - opd1	D/In/Im	(#)
WRITE	08	2	1	*Output stream*← opd1	D/In/Im	(#)

Legenda

* O código de máquina (*opcode*) é alterado pelo modo de endereçamento indireto ou imediato.

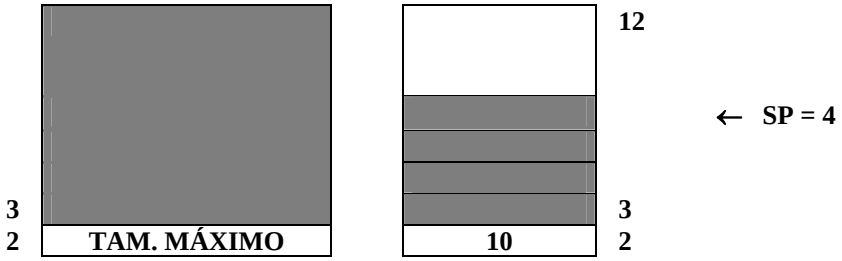
** A referência "[SP]" representa um elemento a ser retirado da pilha (operação ***pop***) e a referência "[SP]←" representa uma operação ***push*** na pilha.

(#) Instruções que podem ter endereçamento imediato.

(%) As instruções CALL e RET utilizam a "Pilha do Sistema" para tratamento dos endereços de retorno, conforme descrito no item específico sobre este tema.

5. Pilha do Sistema

Uma pilha é utilizada pelo sistema para armazenar os endereços de retornos de sub-rotinas, conforme indicado na seção sobre o "Conjunto de Instruções". Esta pilha do sistema é endereçada (acessada) através do registrador **SP** (ponteiro da pilha).
A pilha do sistema está localizada no início da memória física, a partir do **endereço 2** (**endereço base da pilha**), cujo conteúdo não pode se desempilhado e deve manter o seu tamanho máximo (*Stack Limit*). O valor inicial do SP é implicitamente carregado com zero ao "ligar a máquina virtual". O ponteiro da pilha somente pode crescer incrementando até seu limite, causando um desvio para o endereço 0 (zero), caracterizada como uma exceção de *"Stack Overflow"*, caso haja uma tentativa de empilhar com a pilha cheia.
A estrutura da pilha é a seguinte:



6. Modos de operação e instruções privilegiadas

Não há instruções privilegiadas, nem diferenciação de modos de operação da CPU, quanto a privilégios.
O Emulador deve ser definido para trabalhar em três modos de operação alternativos:
- (0) modo contínuo sem interação com a interface (visual) de operação;
- (1) modo contínuo interagindo com a interface de operação a cada ciclo de instrução e
- (2) modo de depuração (passo a passo) interagindo com a interface de operação, que proporciona a execução de apenas uma instrução a cada comando da interface.

O modo de operação é definido quando a máquina é reiniciada, mas também pode ser alterado durante a operação no modo (2), ou no modo (1), se a interface proporcionar uma interrupção no processamento ("*break*" ou passagem temporária para o modo (2)).

7. Periféricos (I/O) - operações, interrupções

A técnica utilizada para as operações de I/O é descrita a seguir:
- as operações de entrada simplesmente recebem um número inteiro resultante do tratamento da seqüência de caracteres anteriores a um CR (código 13; ENTER) introduzida no console (teclado);
- as operações de saída mostram na posição corrente do console (tela) os caracteres numéricos correspondente a um número inteiro;
- não há instruções para verificação de status de periféricos;
- o papel de controlador do periférico pode ser desempenhado pelo módulo de implementação da interface visual .

Observações

O produto gerado por este trabalho é pré-requisito e será fundamental para a viabilização do próximo trabalho (Parte II), a ser desenvolvido coletivamente, pois representará a plataforma de execução de um sistema de programação.

Bibliografia

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.