



Trabalho

Objetivo: O trabalho aqui proposto tem como objetivo a implementação de um simulador para uma arquitetura simplificada com memória cache, composta por um conjunto de instruções aritméticas, de desvio e de movimentação de dados entre registradores e memória.

Instruções

- O trabalho poderá ser feito em dupla.
- O trabalho poderá ser implementado em uma das seguintes linguagens:
 - C/C++;
 - Python;
 - Pascal;
 - Javascript;
 - Java;

Obs: caso a dupla deseje utilizar outra linguagem deverá comunicar o professor para verificar se a linguagem será aceita.

- O trabalho vale de 0,0 a 10,0 e corresponde a 3ª avaliação periódica.
- O trabalho deverá ser entregue via classroom na data combinada em sala.
- O nome do arquivo enviado deve seguir o padrão:
NomeAluno1_RAXXXXXX_NomeAluno2_RAXXXXXX.zip;
 - O formato para submissão deve ser .zip, .rar, ou .tar;

Descrição: Deve-se implementar um simulador para uma arquitetura simples. As descrições para o hardware são as seguintes:

Processador:

- A arquitetura deve ter 32 registradores de uso geral de 64 bits cada (8 bytes);
 - Os registradores devem ser nomeados de r0 até r31;
 - Inicialmente todos eles devem conter o valor 0;
 - É obrigatório o uso dos registradores de estado/uso específico;
 - PC – Contador de programa;
 - RSP – Ponteiro de pilha;
 - RA – Endereço de retorno;
 - OF – Registrador de 1 bit para indicar se houve overflow na última operação;

Conjunto de Instruções:

- As instruções aritméticas e de desvio só podem ter seus operandos endereçados de duas maneiras:



Disciplina: Arquitetura e Organização de Computadores I

- Endereçamento direto por registrador;
- Endereçamento por imediato;
- As instruções de acesso à memória deverão ter um operando com endereçamento direto por registrador e um operando com endereçamento por deslocamento;
- O formato e significado das instruções que deverão ser implementadas são exibidos na Tabela 1;
- Todas as instruções deverão operar somente sobre os registradores de uso geral (r0, r1, ..., r31), o ponteiro de pilha (RSP) e o registrador de endereço de retorno (RA);
 - Ou seja, os registradores de Overflow (OF) e contador de programa PC não devem ser manipulados diretamente pelo programador assembly (algumas instruções irão realizar a manipulação de tais registradores);
- As operações devem ser realizadas somente com valores inteiros;
- As instruções deverão ser de tamanho fixo e deverão ter 64 bits cada (ou seja, 8 bytes);

Tipo de instrução	Representação da instrução	Significado
Aritméticas	add rd, rs, rt	Atribui à rd a soma de rs e rt $rd \leftarrow rs + rt$
	addi rd, rs, imm	Atribui à rd a subtração entre rs e um valor imediato $rd \leftarrow rs + imm$
	sub rd, rs, rt	Atribui à rd a soma de rs e rt $rd \leftarrow rs - rt$
	subi rd, rs, imm	Atribui à rd a subtração entre rs e um valor imediato $rd \leftarrow rs - imm$
	mul rd, rs, rt	Atribui à rd o produto entre rs e rt $rd \leftarrow rs * rt$
	div rd, rs, rt	Atribui à rd o quociente da divisão de rs por rt $rd \leftarrow rs \text{ div } rt$
Lógicas	not rd, rs	Atribui à rd a negação bit a bit de rs $rd \leftarrow \sim rs$
	or rd, rs, rt	Atribui à rd a disjunção (ou lógico) bit a bit entre rs e rt $rd \leftarrow rs rt$
	and rd, rs, rt	Atribui à rd a conjunção (e lógico) bit a bit entre rs e rt $rd \leftarrow rs \& rt$
Desvios	blti rs, rt, imm	Salta caso rs seja maior que rt Se $rs < rt$ então $pc \leftarrow imm$
	bgti rs, rt, imm	Salta caso rs seja menor que rt



Disciplina: Arquitetura e Organização de Computadores I

Desvios			Se $rs > rt$ então $pc \leftarrow imm$
	beqi	rs, rt, imm	Salta caso rs e rt sejam iguais Se $rs = rt$ então $pc \leftarrow imm$
	blt	rs, rt, rd	Salta para rd caso rs seja maior que rt Se $rs < rt$ então $pc \leftarrow rd$
	bgt	rs, rt, rd	Salta caso rs seja menor que rt Se $rs > rt$ então $pc \leftarrow imm$
	beq	rs, rt, imm	Salta caso rs e rt sejam iguais Se $rs = rt$ então $PC \leftarrow imm$
	jr	rd	Salto incondicional para rd $PC \leftarrow rd$
	jof	rd	Salto em caso de overflow Se $OF == 1$ então $pc \leftarrow rd$
	jal	imm	Salto usado para chamada de função com endereço inicial em imm $RA \leftarrow PC+8$ (próxima instrução) e $PC \leftarrow imm$
	ret		Salto usado para voltar de uma chamada de função $PC \leftarrow RA$
Memória	lw	rd, imm(rs)	Carrega da memória para o registrador rd $rd \leftarrow M[imm+rs]$
	sw	rs, imm(rt)	Armazena o valor de rs na memória $M[imm+rt] \leftarrow rs$
Movimentação	mov	rd, rs	Movimentação de registrador para registrador $rd \leftarrow rs$
	movi	rd, imm	Movimentação de imediato para registrador $rd \leftarrow imm$

Tabela 1: Conjunto de instruções da arquitetura a ser simulada.

Obs: nas instruções imm é um valor constante inteiro, e $M[x]$ representa o valor de memória no endereço x.

Memória:

- A arquitetura deve ter memória estruturada em dois níveis, uma memória cache, separada em dados e instruções, e uma memória principal;
- Cada endereço de memória (tanto da principal, quanto da cache de dados ou da cache de instruções) deverá armazenar 8 bytes (ou seja, uma instrução ou um inteiro).

Obs: A limitação de 8 bytes não impede que as estruturas usadas para representar cada célula de memória possua mais informações

- A cache deve ser associativa por conjunto:
 - O número de bytes por linha, linhas por conjunto e número total de conjuntos deve ser configurável;



Disciplina: Arquitetura e Organização de Computadores I

- O tamanho total da cache é dado pelo produto desses 3 elementos;
- Use os mesmos valores para as caches de dados e instruções;
- Para facilidade com a manipulação dos dados da memória, o número de bytes por linha deve ser múltiplo de 8;
 - Caso a dupla prefira, pode utilizar o número de palavras por linha.
 - O valor máximo para bytes por linha deve ser de 1024 (no caso de palavras por linha, o valor máximo de palavras por linha deve ser 128);
Obs: A limitação de 1024 bytes por linha é referente aos dados armazenados na cache. Na implementação do simulador, estruturas auxiliares podem ser modeladas para auxiliar o controle da cache;
- A memória principal deverá ter seu tamanho configurável em número de bytes (também deve ser múltiplo de 8) e deverá ser maior que a memória cache;
 - Caso a dupla prefira, a memória principal também pode ser configurada em número de palavras da memória.
 - A medida adotada deve ser a mesma para cache e memória principal
 - Se a cache é configurada por palavras, a memória principal também deve ser medida em bytes;
 - Se a cache é configurada por bytes, a memória principal deverá ser medida em bytes
 - Caso a dupla deseje, esta opção pode ser um argumento do programa, porém, isso não é obrigatório;

Configuração e interação com o usuário:

Entradas

- A entrada principal do programa deverá ser um arquivo de texto contendo um conjunto de instruções a serem executadas, onde cada instrução ocupa uma linha do arquivo.
- Alguns exemplos estão disponíveis no mesmo tópico desta especificação.
- As configurações tais como memória cache e memória principal devem ser feitas preferencialmente por argumentos passados na linha de comando, ou então um arquivo de configuração que informa esses valores

Saídas:

- A cada ciclo de instrução deverão ser exibidos na tela:
 - Os valores armazenados na memória cache;
 - Não é necessário mostrar endereços que não estejam ocupados com dados que façam parte do programa;
 - Os valores exibidos devem ser separados por dados e instruções;
 - As instruções podem ser mostradas na sua forma textual;



Disciplina: Arquitetura e Organização de Computadores I

- Os valores armazenados em cada endereço da memória;
 - As instruções podem ser mostradas na sua forma textual;
- Os valores armazenados em cada um dos registradores de uso geral;
- Os valores armazenados em cada um dos registradores de controle de estado;

Problemas com Trabalhos COPIADOS:

Quem copiar terá o trabalho anulado (zerado), seja de outra dupla ou da internet.

Quem fornecer a cópia também terá o trabalho anulado.

Trabalhos feitos com inteligência artificial generativa (Chat GPT, Gemini, etc) também serão anulados.