

UNIVERSIDADE FEDERAL DA BAHIA - UFBA
ESCOLA POLITÉCNICA / DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
ENGG56 – PROJETO DE CIRCUITOS INTEGRADOS DIGITAIS – TRABALHO PRÁTICO

Processadores podem ser classificados de acordo com o número de operandos utilizados em suas instruções, assim como a forma de acesso aos mesmos (em memória ou em registradores). Assim, tem-se máquinas de 0, 1, 2, 3 endereços e arquiteturas LOAD/STORE.

Máquinas de 0-endereços são as mais simples de se implementar. Por esta razão, este padrão foi adotado na especificação de JVM (*Java Virtual Machine*), de forma a flexibilizar a implementação da mesma em diferentes plataformas computacionais, envolvendo processadores de diferentes arquiteturas (Intel, ARM, por exemplo) e sistemas operacionais (Windows, Linux, Android, entre outros).

O conjunto de instruções JVM (os chamados *bytecodes*) é bastante extenso e trata, basicamente, de operações lógicas e aritméticas executadas sobre uma pilha e *arrays* de memória.

Como forma de introduzi-los neste universo, vocês deverão implementar uma máquina de 0-endereços com um conjunto de instruções bem simples (Tabela 1), assim como fazer sua validação em termos de temporização e funcionalidade.

Produtos:

Organizem-se em grupos de no máximo 4 alunos. Cada grupo deverá entregar os seguintes artefatos:

1. A pasta de projeto **C:\ENGG56\Quartus\Processador** compactada, contendo:
 - a implementação do sistema completo (processador/memórias) no ambiente Altera Quartus Prime, com prototipação na placa DE2-115;
 - um programa teste simples (em memória) que comprove o funcionamento das instruções do processador, com exibição de dados na placa DE2-115;
 - análise temporal estática realizada no Timing Analyzer, com o correspondente arquivo .SDC.
2. A pasta de projeto **C:\ENGG56\ModelSim\Processador** compactada, contendo o *testbench* desenvolvido.
3. Um relatório em *.pdf*, contendo:
 - a descrição da implementação desenvolvida (obrigatório a arquitetura do processador especificado e a FSM referente à sua unidade de controle);
 - os cenários de teste realizados;
 - a descrição dos resultados da análise temporal estática;
 - a participação individual e aprendizado adquirido por cada membro.

Entrega: 06/12

Via link do Google Drive compartilhado com o e-mail oliveira.wagner@ufba.br.

Software:

Dois softwares devem ser baixados: Intel Quartus Prime Lite Edition (versão 18.1) e ModelSim, considerando a família de dispositivos FPGA Cyclone IV.

<https://www.intel.com/content/www/us/en/software-kit/665990/intel-quartus-prime-lite-edition-design-software-version-18-1-for-windows.html?>

<https://www.intel.com/content/www/us/en/software-kit/665988/intel-quartus-prime-lite-edition-design-software-version-18-1-for-linux.html?>

Em qualquer dos casos, recomenda-se a opção “Individual Files”. Além disso, selecionem apenas o Quartus, o ModelSim e o suporte ao dispositivo Cyclone IV (arquivo .qdz).

Caso você deixe os arquivos baixados na mesma pasta, basta fazer a instalação do Quartus que o ModelSim e o dispositivo Cyclone IV serão instalados conjuntamente. Caso contrário, após a instalação do Quartus e do ModelSim, deve-se fazer a instalação do arquivo referente ao dispositivo Cyclone IV (arquivo .qdz). Para tal, use o menu do Quartus: “Tools -> Install Devices...”.

Para fazer o mapeamento para entradas/saídas disponíveis na placa DE2-115, veja o manual da placa a seguir ou use o arquivo de pinagem *DE2_115.qsf* disponibilizado no Moodle.

https://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=English&No=502&FID=cd9c7c1feaa2467c58c9aa4cc02131af

IMPORTANTE: *Não implemente testbenches lineares.*

Pede-se que os testbenches não se limitem a exibir apenas waveforms. Faça uso de mensagens descritivas, exibidas no console.

Definições referentes às memórias utilizadas ficarão sob a responsabilidade de cada grupo.

Soluções muito parecidas serão consideradas cópias, com nota zerada para os grupos envolvidos.

Tabela 1. Conjunto de instruções para o processador proposto.

Tipo	Instrução	Operando	Descrição
Transferência de Dados	PUSH	M	$TOS \leftarrow [M]$
	PUSH_I	Imm	$TOS \leftarrow \text{Immediate}$
	PUSH_T		$TOS \leftarrow Temp_1$
	POP	M	$[M] \leftarrow TOS$
Aritmética	ADD		$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow Temp_1 + Temp_2$
	SUB		$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow Temp_1 - Temp_2$
	MUL		$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow Temp_1 * Temp_2$
	DIV		$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow Temp_1 / Temp_2$
Lógica	AND		$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow Temp_1 \text{ AND } Temp_2$
	NAND		$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow Temp_1 \text{ NAND } Temp_2$
	OR		$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow Temp_1 \text{ OR } Temp_2$
	XOR		$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow Temp_1 \text{ XOR } Temp_2$
	CMP		$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow 1 \text{ (se } Temp_1 > Temp_2)$
			$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow 0 \text{ (se } Temp_1 = Temp_2)$
			$Temp_1 \leftarrow TOS; Temp_2 \leftarrow TOS; TOS \leftarrow -1 \text{ (se } Temp_1 < Temp_2)$
Transferência de Controle	NOT		$Temp_1 \leftarrow TOS; TOS \leftarrow \text{NOT } Temp_1$
	GOTO	M	desvio incondicional para o endereço [M]
	IF_EQ	M	$Temp_1 \leftarrow TOS$; desvio para o endereço [M] se $Temp_1 = 0$
	IF_GT	M	$Temp_1 \leftarrow TOS$; desvio para o endereço [M] se $Temp_1 > 0$
	IF_LT	M	$Temp_1 \leftarrow TOS$; desvio para o endereço [M] se $Temp_1 < 0$
	IF_GE	M	$Temp_1 \leftarrow TOS$; desvio para o endereço [M] se $Temp_1 \geq 0$
	IF_LE	M	$Temp_1 \leftarrow TOS$; desvio para o endereço [M] se $Temp_1 \leq 0$
	CALL	M	chamada de subrotina no endereço [M]
	RET		retorno de subrotina

Observações:

1. TOS (*Top of Stack*): representa o topo da pilha de dados;
2. M: representa um endereço de memória;
3. [M]: representa o conteúdo de um endereço de memória M;
4. Imm: representa um valor imediato (constante);
5. Temp_x: representa um registrador interno;
6. Atribuição sobre TOS: representa inserção no topo da pilha de dados, com consequente atualização de TOS; e
7. Atribuição proveniente de TOS: representa remoção do dado presente no topo da pilha de dados, com consequente atualização de TOS.