

Unidade de Processamento Baseado em Pilha

Fundamentos de sistemas digitais – 2016.2

O presente trabalho tem por objetivo explorar os conceitos apresentados em sala de aula ao longo do semestre. Para tanto, deverão ser desenvolvidos módulos VHDL mesclando descrições estruturais e comportamentais, bem como deve ser implementado um módulo para validação do circuito (testbench).

1 Descrição funcional do sistema

O objetivo do trabalho é desenvolver uma unidade de processamento composta por uma (a) memória ROM e (b) uma unidade de processamento baseada em pilha. Na ROM deverá estar disponível o programa a ser executado pela unidade de processamento. A unidade de processamento deverá ser composta por um bloco de controle e um bloco de dados. O bloco de controle será responsável por implementar a máquina de estados de controle do sistema, enquanto o bloco de dados será responsável por realizar o conjunto de passos indicados pelo bloco de controle.

O funcionamento da unidade de processamento é apresentado a seguir:

- A memória ROM conterà o programa a ser executado pela unidade de processamento
- Um programa é composto por um conjunto de indicações de operandos e operações
- A memória ROM
 - Endereça palavras de 4 bits
 - O bit mais significativo indica se a palavra de 4 bits é um dado ou uma instrução
 - 0 → Dado e 1 → Instrução
 - As instruções listadas na Tabela 1 são suportadas
 - Os dados permitem a sinalização de valores que vão de 0 a 7
 - Não se representa valores negativos na memória ROM

Tabela 1 - Lista de instruções aceitas pela unidade de processamento

Instrução	Operação (3bits)	Descrição
ADD	000	OP1 + OP2
SUB	001	OP1 – OP2
INC	010	OP1+1
DEC	011	OP1-1
AND	100	OP1 & OP2
OR	101	OP1 OP2
INV	110	! OP1
STOP	111	HALT

Conforme apresentado na Tabela 1, são aceitos um conjunto básico de operações aritméticas e lógicas. Tais instruções operam sobre apenas um operando (i.e. Instrução NOT), sobre nenhum operando (i.e. Instrução STOP) ou ainda sobre 2 operandos (i.e. todas as demais instruções). As instruções lógicas e aritméticas seguem o funcionamento descrito na Tabela 1, sendo equivalente as operações exploradas ao longo do semestre. A instrução STOP deve finalizar a execução do programa existente na memória ROM.

Para o funcionamento da unidade de processamento baseada em pilha, define-se que o seguinte procedimento deve ocorrer. A memória ROM deve conter a lista de dados e instruções a serem executadas. Conforme dito anteriormente, caso a palavra presente na ROM tenha o seu bit mais significativo em 0, trata-se de um dado, enquanto o valor 1 indica tratar-se de uma das operações listadas previamente. Sempre que um dado for listado, este deve ser adicionado em uma memória RAM, interna a unidade de processamento. Tal armazenamento deve seguir o funcionamento de uma pilha. Neste tipo de recurso, o primeiro dado a ser armazenado é o último a ser capturado, enquanto os últimos dados a serem inseridos devem ser os primeiros a serem acessados, conforme ilustrado na Figura 1.

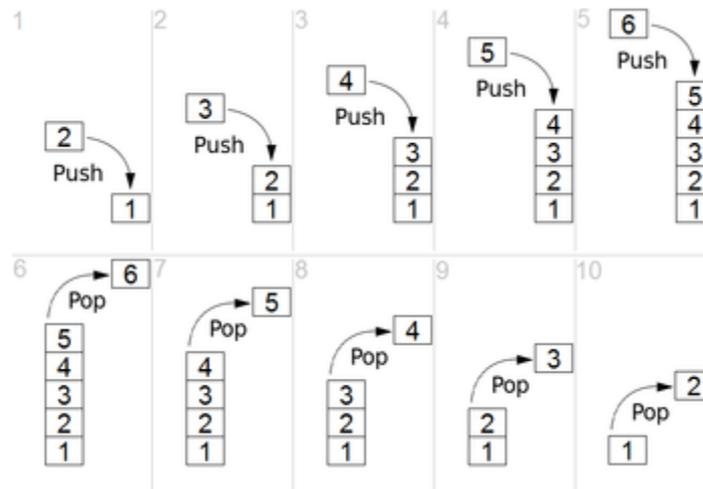


Figura 1 - Funcionamento de uma memória baseada em pilha.

Conforme ilustrado na Figura 1, os *pushs* (i.e. empilhamentos de dados) fazem com que os dados inseridos por primeiro sejam os últimos a serem acessados. Neste contexto, toda vez que um dado for detectado na memória ROM, este deve ser *empilhado* em uma memória RAM interna a unidade de processamento. Por outro lado, toda vez que uma instrução for detectada, a mesma deve operar sobre os dados empilhados na memória RAM interna, e seu resultado deve ser empilhado na memória RAM.

Um exemplo de funcionamento deste sistema deve ser conforme detalhado a seguir. Assumindo a seguinte expressão:

$$(7 + 4) - (2 + 3)$$

A mesma poderia estar representada da seguinte forma na memória ROM:

Endereço	Instrução	Significado
----------	-----------	-------------

0	0010	3
1	0011	2
2	1000	+
3	0111	7
4	0100	4
5	1000	+
6	1001	-
7	1111	STOP

A execução deste programa deveria ter o reflexo na memória RAM interna da unidade de processamento baseada em pilha, conforme apresentado na Tabela 2:

Tabela 2 - Fluxo de execução da equação

Endereço da RAM	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5	Passo 6	Passo 7
3							
2					4		
1		3		7	7	11	
0	2	2	5	5	5	5	6

No **passo 1** a primeira posição da memória ROM é lida e interpretada como um dado. Como reflexo, este dado (i.e. 2) é armazenado na posição 0 (zero) da memória RAM. No **passo 2**, a segunda posição da ROM é lida e um novo dado é interpretado. Como resultado, este segundo dado (i.e. 3) é armazenado na posição 1 da memória RAM. No **passo 3** a instrução de soma é decodificada. Neste caso, o conteúdo da posição 1 é assumido como operando 1 e o conteúdo da posição 0 é assumido como operando 2. Ambos são desempilhados e armazenados em registradores que representem suas posições na entrada da ULA. O resultado desta operação (i.e. 5) é então empilhado na PILHA, conforme representado no **passo 3** da Tabela 2. No **passo 4** um novo dado é detectado (i.e. 7) e empilhado na RAM. A seguir, no **passo 5**, um novo dado (i.e. 4) é decodificado na ROM e empilhado na RAM. No **passo 6**, uma nova instrução de soma é detectada e os dois primeiros operandos da PILHA são removidos da pilha e utilizados como operandos da ULA. O resultado da operação (i.e. 11) é então armazenado na posição 1 da RAM. No **passo 7**, a operação de subtração é decodificada, e os dois valores contidos na memória ROM são subtraídos, sendo o resultado armazenado na PILHA. Finalmente, a instrução STOP é detectada e o programa para sua execução.

2 Descrição arquitetural do sistema

Conforme descrito anteriormente, a unidade de processamento baseada em pilha será composta por um bloco de dados e outro de controle. Na unidade de controle, deverá ser implementada a máquina de estados que será responsável por:

1. Gerenciar o acesso a memória ROM
2. Identificar dados/instruções
3. Solicitar o empilhamento de dados, quando interpretados
4. Solicitar a realização das instruções quando decodificadas, desimpilhando os operandos da memória RAM e armazenando o resultado na mesma memória.
5. Sinalizar
 - a. O final de execução de um programa

b. Operações ilegais (e.g. solicitação de operações sem a presença de operandos)

A máquina de estados a ser implementada é ilustrada na Figura 2.

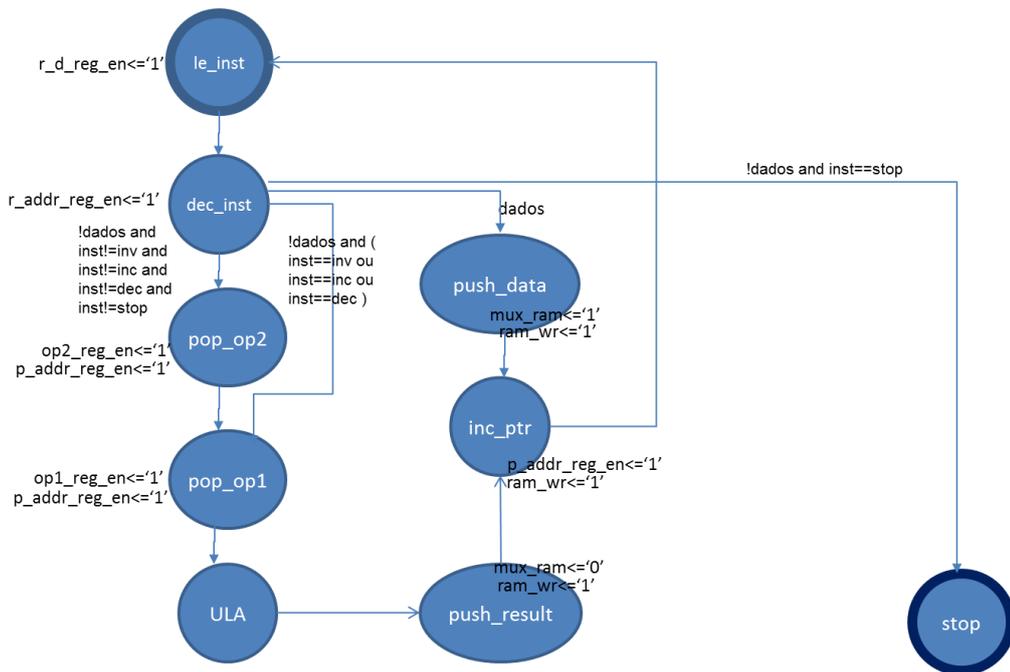


Figura 2 - Máquina de estados da unidade de processamento baseado em pilha

Além do bloco de controle, a unidade de processamento deverá possuir um bloco de dados, onde as operações são efetivamente realizadas, bem como o uso da pilha de dados. Neste contexto, a Figura 3 ilustra o bloco de dados que deverá ser implementado.

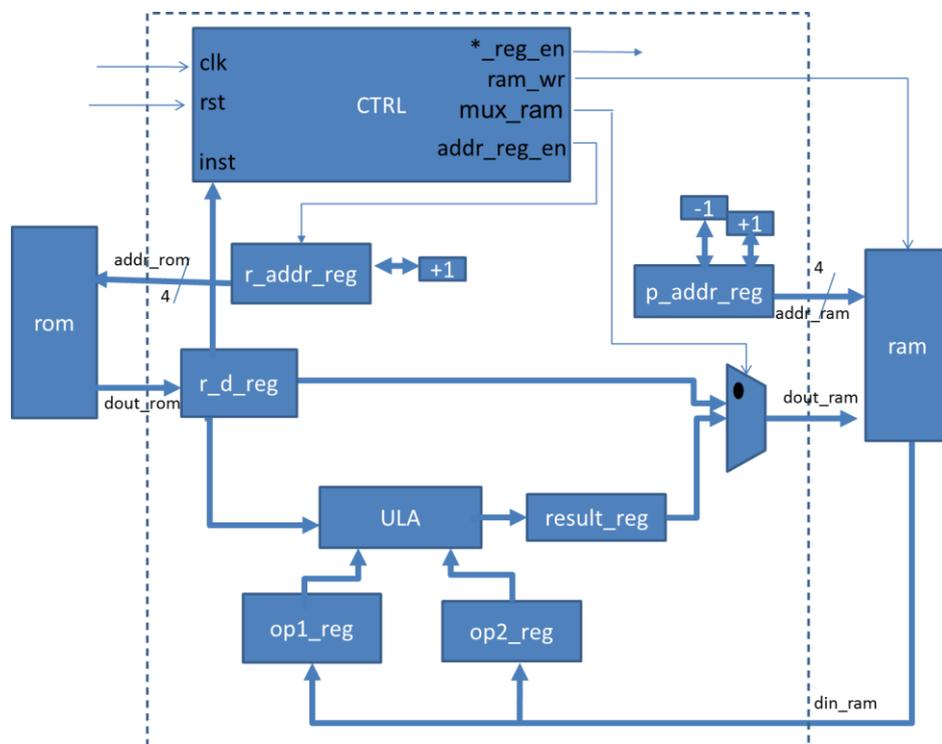


Figura 3 - Bloco de dados da unidade de processamento baseada em pilha.

Na Figura 3, o módulo na parte superior, identificado como CTRL, representa o bloco de controle. Os demais blocos representam as memórias ROM (a esquerda) e RAM (a direita), enquanto os demais blocos representam os registradores, apontadores das memórias e uma ULA. O uso destes recursos deve seguir aquilo que é especificado na máquina de estados.

Um ponto importante é entender que o registrador utilizado para apontar a memória ROM é sempre incrementado em uma posição, enquanto o registrador utilizado para apontar a memória RAM pode ser tanto incrementado quanto decrementado.

3 Validação e documentação

Para a validação do módulo que deverá ser implementado um testbench. A memória ROM deverá estar pré preenchida para garantir a execução. A escolha dos padrões de estímulos a serem gerados deverá ser documentada em relatório, deixando claro o que se espera como resultado daquele estímulo. Adicionalmente, algumas imagens da simulação deverão ser apresentadas em relatório com o objetivo de comprovar se o que se esperava como resultado foi efetivamente alcançado. Para cada imagem postada no relatório, espera-se a inserção de comentários.

4 Considerações finais

O trabalho pode ser realizado em grupos de até dois (2) alunos. Este deve ser entregue nos dias determinados na agenda da disciplina, via moodle, até o final da aula.

A entrega deve ser feita com um arquivo compactado (.zip) onde deverão estar contidos os arquivos VHDL do projeto e o relatório (em PDF). O nome do arquivo deve conter nome e sobrenome de todos os alunos que fizeram parte do trabalho. Ex.: AnaMattos_CarlosBrandao_JanainaMirtten.zip.

Para o desenvolvimento deste trabalho, são previstas três entregas, as quais serão avaliadas da seguinte forma:

1. 16/11/2016 – projeto da FSM (30% da nota)
2. 23/11/2016 – projeto da ULA + bloco de dados (30% da nota)
3. 30/11/2016 – final: integração, testbench e relatório (40% da nota)

O material postado no moodle é de inteira responsabilidade do aluno. A presença de arquivos corrompidos, que impeçam a avaliação do trabalho pelo professor será considerada como a não entrega do trabalho. A detecção de plágio acarretará grau zero a todos os alunos envolvidos, tanto quem plagiou quanto quem foi plagiado. Sendo assim, proteja seu trabalho para evitar problemas.