

# Laboratório de Arquitetura e Organização de Computadores I

## Processador Simplificado

### Introdução

Este documento tem como propósito apresentar um processador multi-ciclo simplificado que será usado como base para a construção das tarefas que serão propostas até o fim do curso.

O processador executa 9 tipos de instruções (apresentadas nas Tabelas 1 e 2) e tem uma latência máxima de 4 ciclos. Esta implementação multi-ciclo tem sua arquitetura interna apresentada na Figura 1.

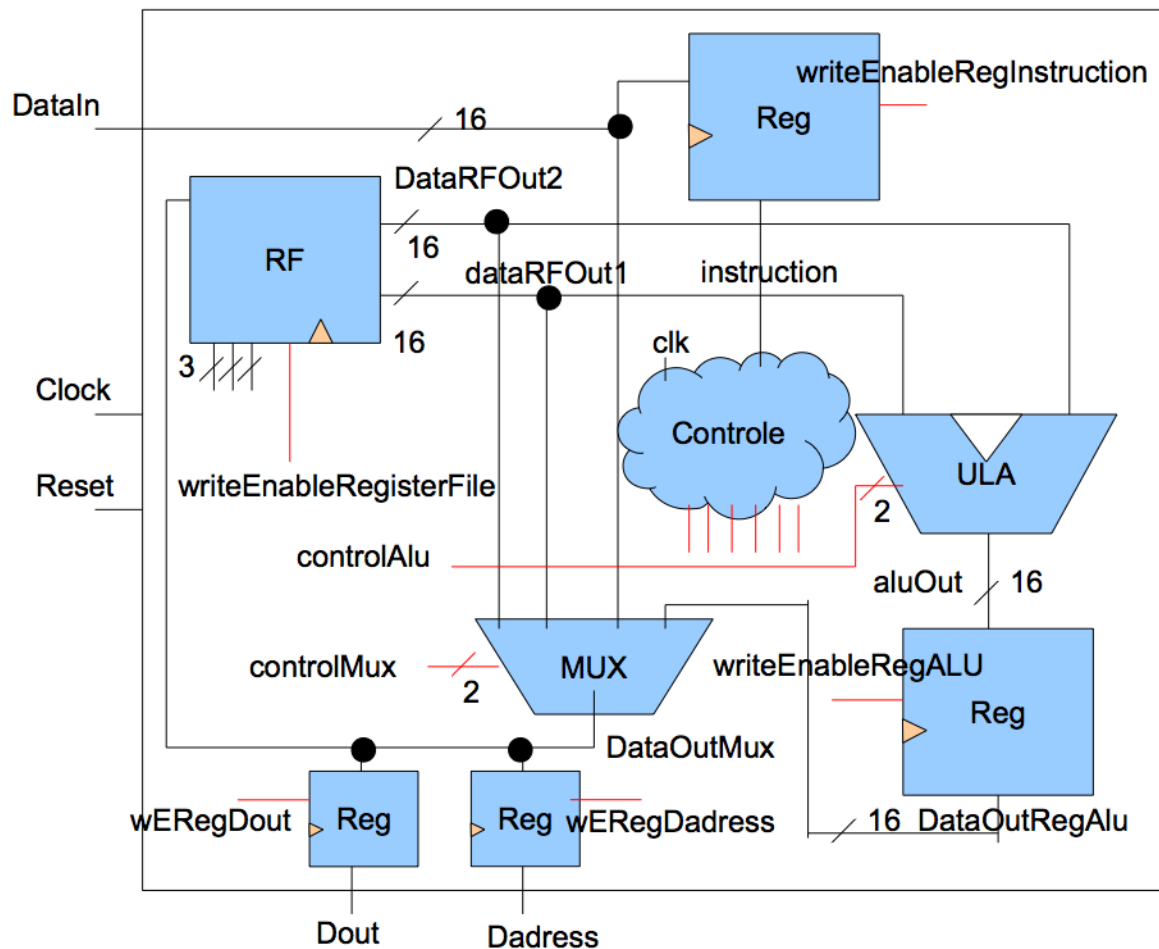


Figura 1: Arquitetura interna do Processador

O primeiro registrador neste processador, no topo da figura e alimentado pela entrada DataIn, tem a função de armazenar a instrução, o que só ocorre quando o sinal writeEnableRegInstruction é ativado. RF é o banco de registradores com 8 posições, sendo que

o ultimo endereço desta estrutura é o PC. Os dois sinais de saída, Dout e Dadress, têm funções de comunicação com a memória. O primeiro sinal, Dout, representa o valor a ser escrito na memória, quando for o caso. O segundo, Dadress, por sua vez, indica o endereço de leitura ou escrita da memória. Esta comunicação ocorre da forma apresentada na Figura 2, o que implica que a memória é compartilhada entre instruções e dados.

<b>Indice</b>	15	11	8	5	0
<b>Significado</b>	OPCODE	Endereço x	Endereço y	Endereço z	-
<b>Comprimento do campo</b>	4 bits	3 bits	3 bits	3 bits	3 bits

Tabela 1: Formato da Instrução

Instrução	Opcode	Descrição
copy	1110	Copia o dado do registrador de endereço y para o registrador de endereço x
copy input	1111	Copia o dado da entrada DataIn do processador para o registrador de endereço x
conditional copy	1011	Se o dado no registrador alimentado pela ALU for 0, copia o dado do registrador y para o registrador x
load	1101	Copia para o registrador x o dado encontrado na memória cujo endereço está no registrador y
store	1100	Copia para a posição de memória armazenada no registrador y o dado armazenado no registrador x
add	0000	Armazena no registrador x o resultado da soma dos dados nos registradores y e z
or	0001	Armazena no registrador x o resultado da operação OR dos dados nos registradores y e z

and	0010	Armazena no registrador x o resultado da operação AND dos dados nos registradores y e z
not	0011	Armazena no registrador x o resultado da operação NOT do dado no registrador y

Tabela 2: Instruções

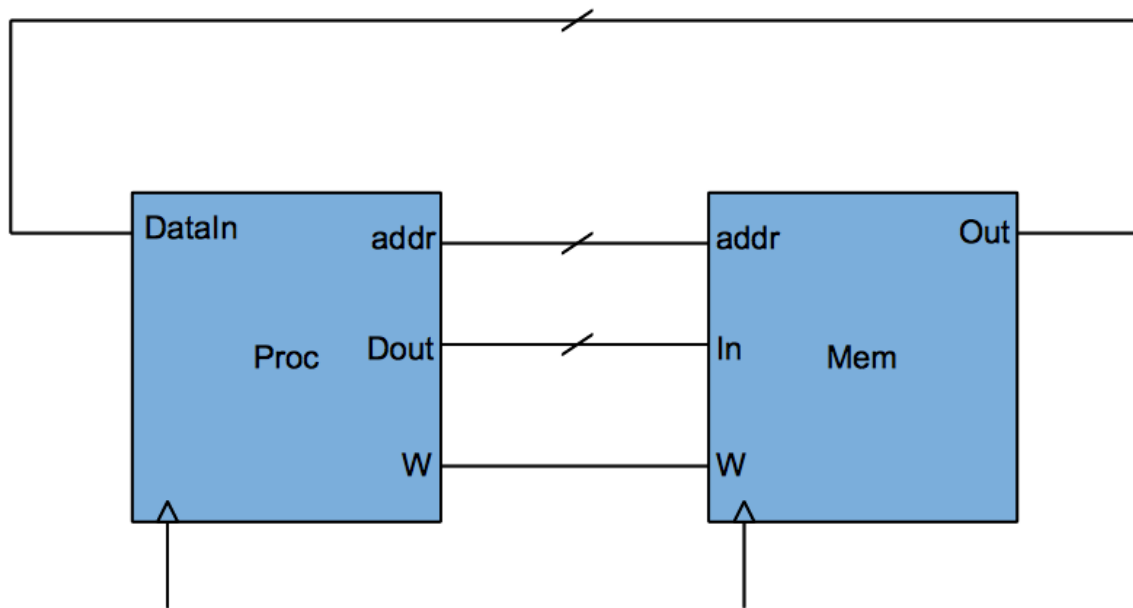
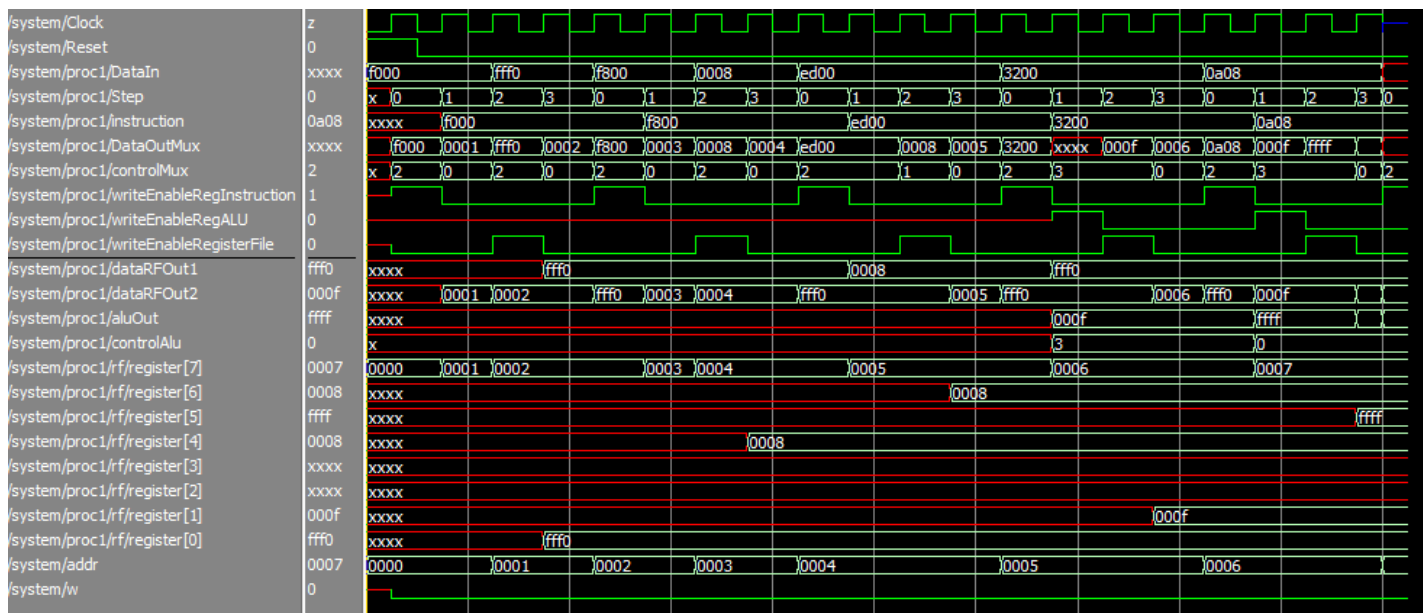


Figura 2: Comunicação entre Processador e Memória



Esta simulação apresenta a dinâmica do processador proposto. O módulo em questão tem a seguinte interface:

```
module processor (DataIn, Reset, Clock, Dout, Daddress, W);
input [15:0] DataIn;
input Reset, Clock;
output [15:0] Dout, Daddress;
output reg W;
```

Observa-se que, tão logo a entrada Reset é desativada aos 100 ns, a primeira instrução  $f000_{16}$  é aplicada na entrada DataIn. Fazendo a conversão do hexadecimal (todos os sinais de 16 bits estão neste formato) e considerando a especificação apresentada em sala conclui-se que essa instrução (opcode  $1111_2$ ,  $f_{16}$ ), realizará a cópia de um dado da entrada para o registrador de endereço 000. O valor do dado, aplicado no ciclo de clock seguinte (iniciado em 250 ns), é  $fff0_{16}$ . A segunda instrução executada,  $f800_{16}$ , é do mesmo tipo da primeira, diferindo apenas no endereço do destino, registrador 4 ( $100_2$ ), e no valor do dado, 8.

A terceira instrução (850 ns) realiza a cópia do dado armazenado no registrador 000 para o registrador  $110_2$  ( $6_{10}$ ). Aos 1250 ns da simulação, é aplicada na entrada a instrução  $3200_{16}$  que representa uma operação de negação sobre o dado do registrador 000 e o resultado é armazenado no registrador 1. Finalmente, a última instrução realiza uma soma dos dados que estão nos registradores 000 e 001 e armazena o resultado no registrador 101.