LAOC2 - Prática II
Processador

Dupla
* Arthur Severo de Souza
* Victor Le Roy Matos

Na segunda prática fomos responsáveis pela implementação de um processador simples em Verilog, incluindo o desenvolvimento de uma memória TLB.

Os seguintes comandos foram executados como teste:

Registradores inicializados com "0" (zero).

Memória inicializada na posição 3 com o valor 4 (MEM[3] = 4)

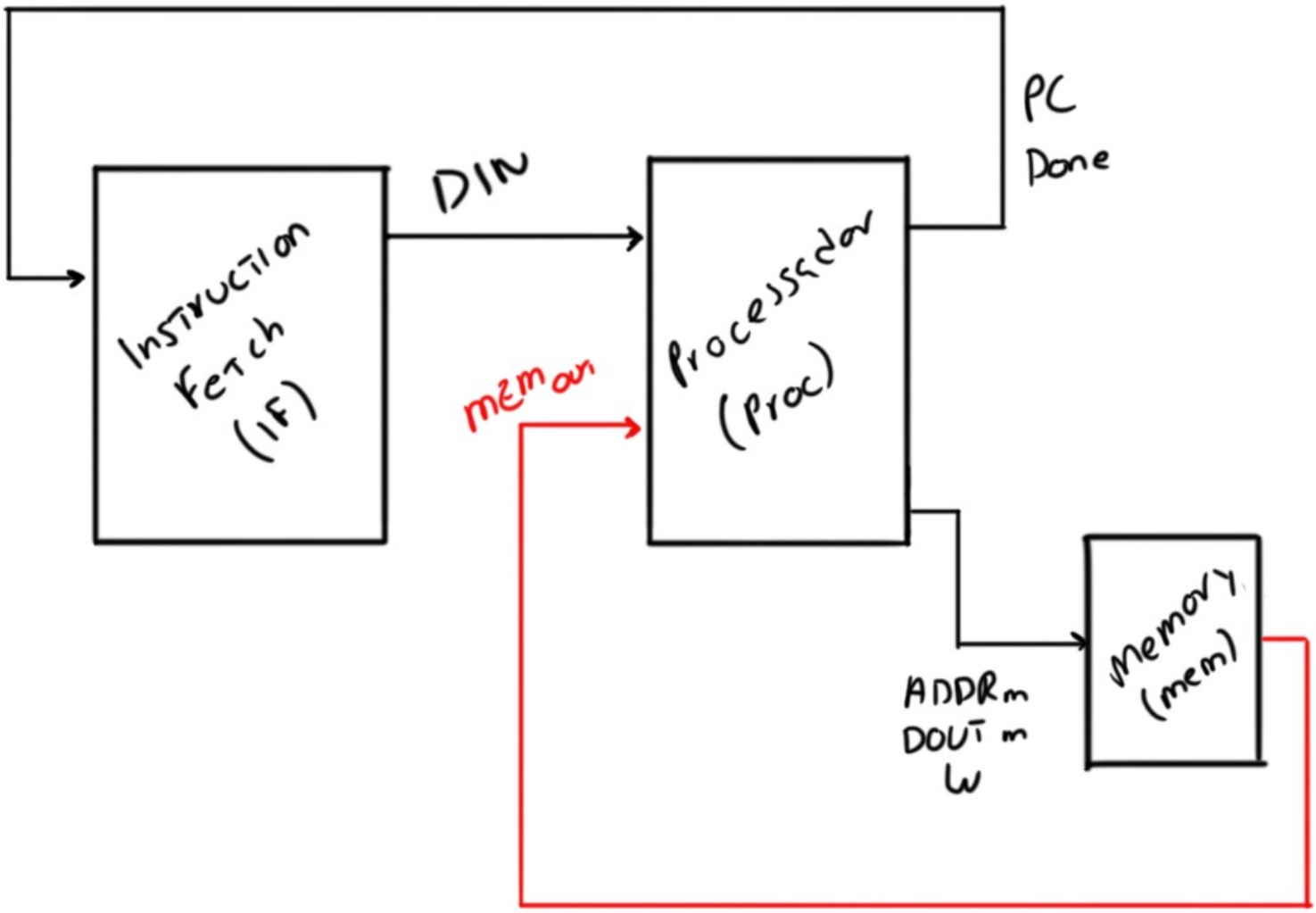
Instrução	R0	R1	R2	R3
MVI R0, #2	2	0	0	0
MVI R1, #3	2	3	0	0
ADD R1, R0	2	5	0	0
MVI R2, #6	2	5	6	0
SUB R2, R1	2	5	1	0
MV R3, R2	2	5	1	1
ADD R0,R3	3	5	1	1
OR R1,R0	3	7	1	1
SUB R1,R0	3	4	1	1
ADD R1, R3	3	5	1	1
SLL R1, R3	1	A	1	1
SRL R1, R3	1	5	1	1
MVI R0, #0	0	5	1	1
SLT R0, R1	1	5	1	1
SLT R1, R1	1	0	1	1
MVI R3, #3	1	0	1	3
MVI R1, #5	1	5	1	3
ADD R0, R3	4	5	1	3
MVI R0, #0	0	5	1	3
LD R2, R3	0	5	4	3
ADD R2, R3	0	5	7	3
SD R2, R0	0	5	7	3
LD R0, R0	7	5	7	3
SUB R0, R3	4	5	7	3
MVI R0, #0	0	5	7	3
ADD R0, R0	0	5	7	3
MVNZ R0, R2	0	5	7	3
SUB R1,R3	0	2	7	3
MVNZ R0, R2	7	2	7	3
ADD R0, R1	9	2	7	3

LOOP

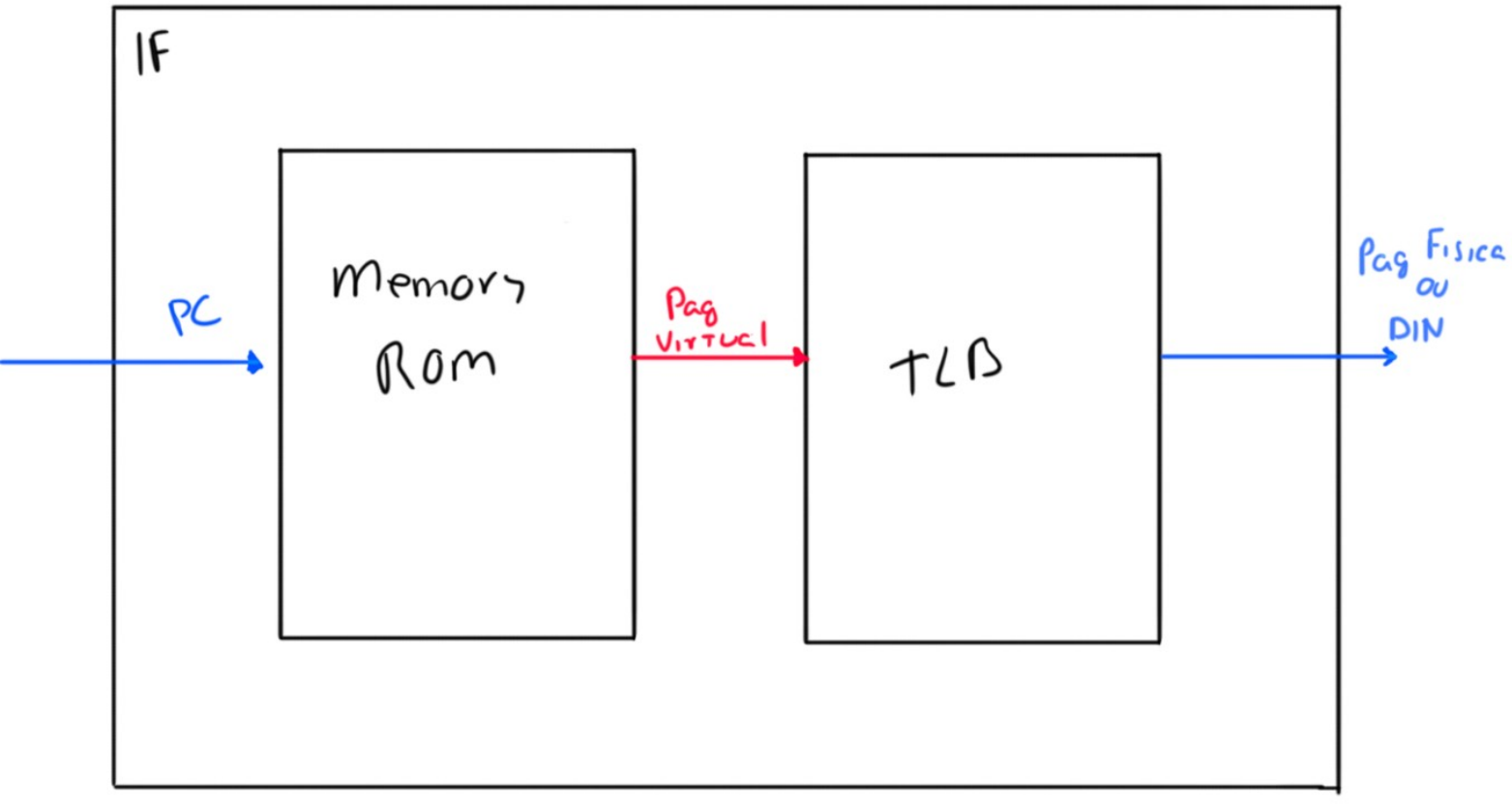
Instrução
MVI R2, #1
MVI R4, #10
MV R5,R7
SUB R4, R2
MVNZ R7,R5

Projeto

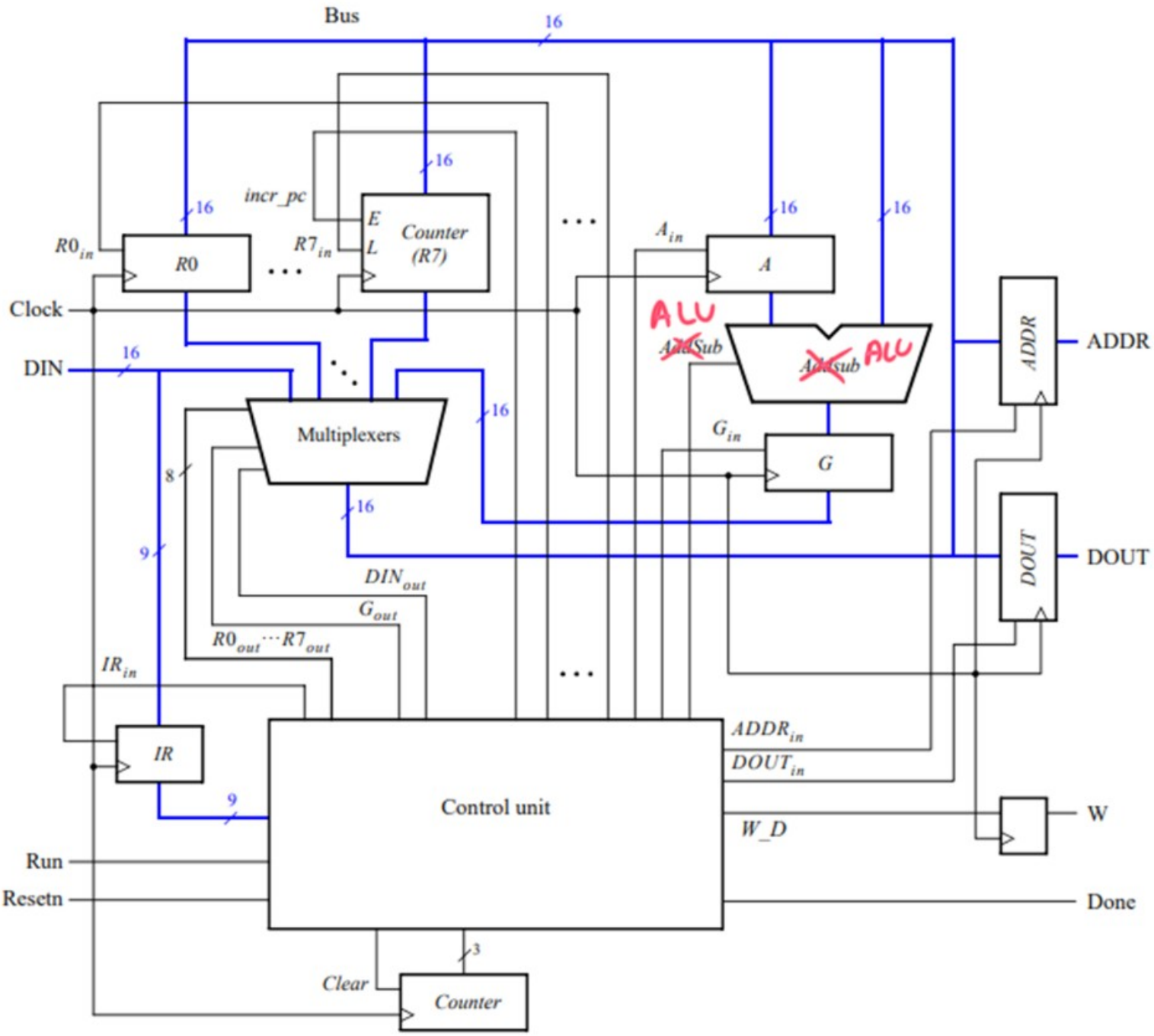
* Diagrama externo aos módulos



* Diagrama do módulo *Instruction Fetch*



* Diagrama do processador



* A ULA de subtração no guia da prática foi substituida por uma ULA geral

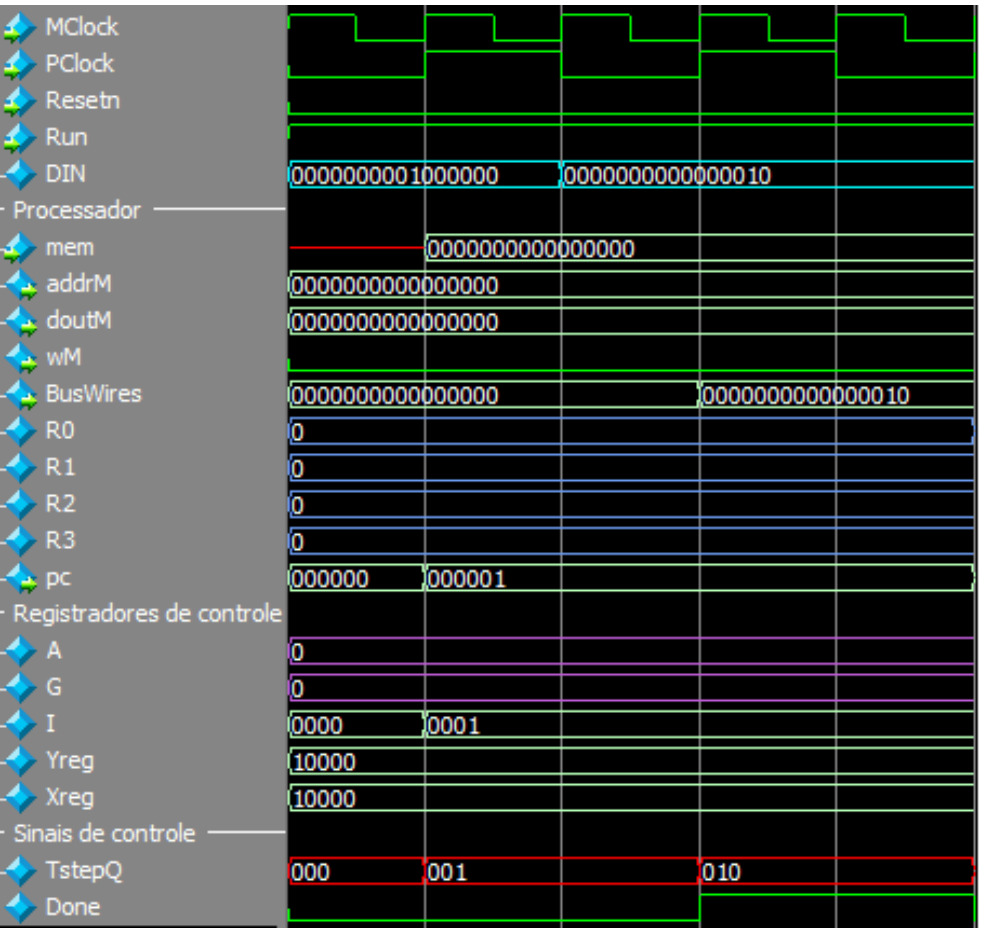
Tabela de instruções

Cod_In	I	T1	T2	T3
I (0000)	**mv**	RYout, Rxin, Done		
I (0001)	**mvi**	Stall	DINout, Rxin, Done	
I (0011)	**mvnz**	RYout, Rxin, Done		
I (0100)	**load**	RYout, ADDRin	Stall	RXin, MEMout, Done
I (0101)	**store**	RXout, DOUTin	RYout, ADDRin, Win, Done	RXin, Gout, Done
I (0110)	**add**	RXout, Ain	RYout, Gin, Alu (0000)	RXin, Gout, Done
I (0111)	**sub**	RXout, Ain	RYout, Gin, Alu (0001)	RXin, Gout, Done
I (1000)	**or**	RXout, Ain	RYout, Gin, Alu (0010)	RXin, Gout, Done
I (1001)	**slt**	RXout, Ain	RYout, Gin, Alu (0011)	RXin, Gout, Done
I (1010)	**sll**	RXout, Ain	RYout, Gin, Alu (0100)	RXin, Gout, Done
I (1011)	**srl**	RXout, Ain	RYout, Gin, Alu (0101)	RXin, Gout, Done

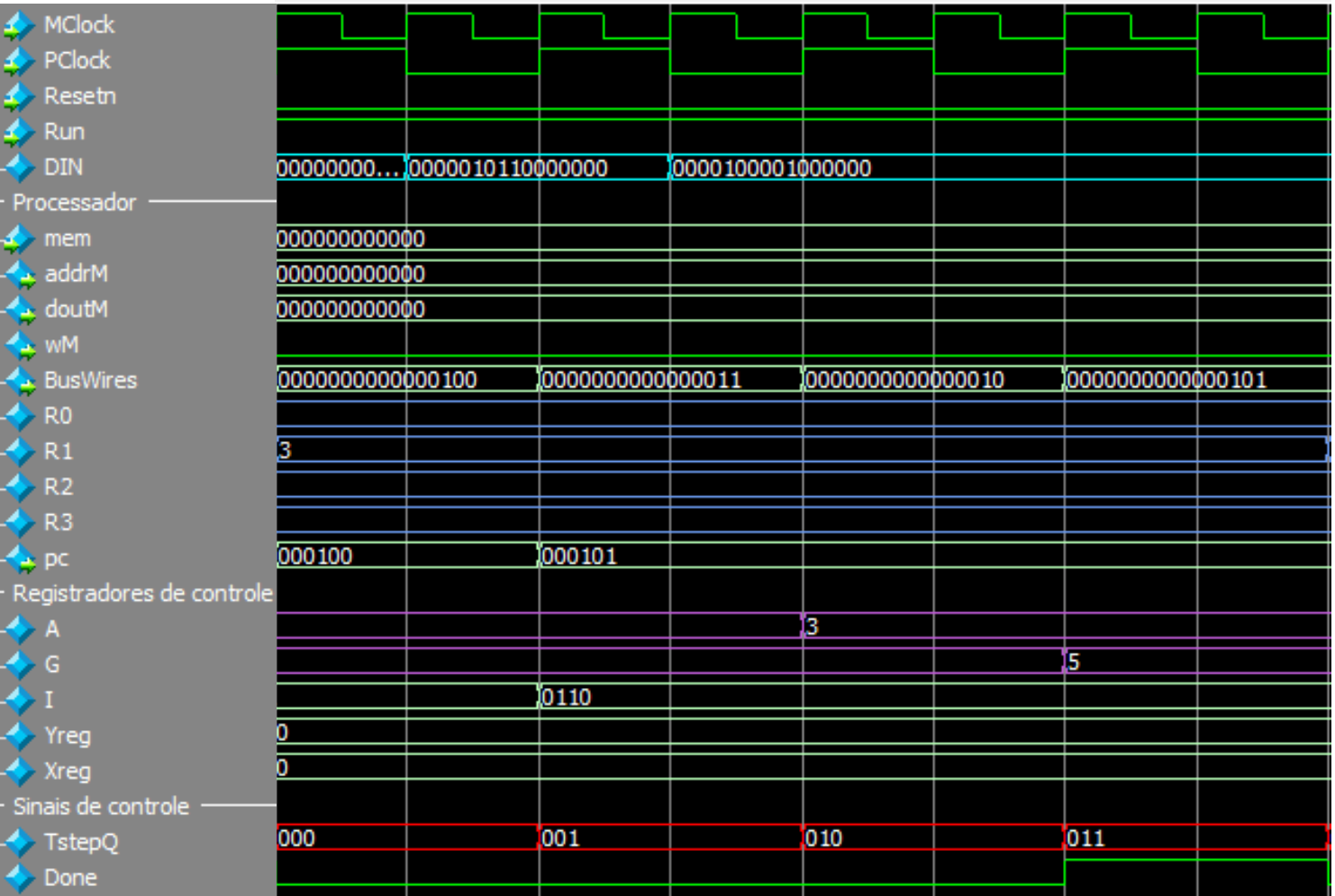
Apresentação das simulações

Serão mostrados exemplos de alguns dos comandos solicitados.

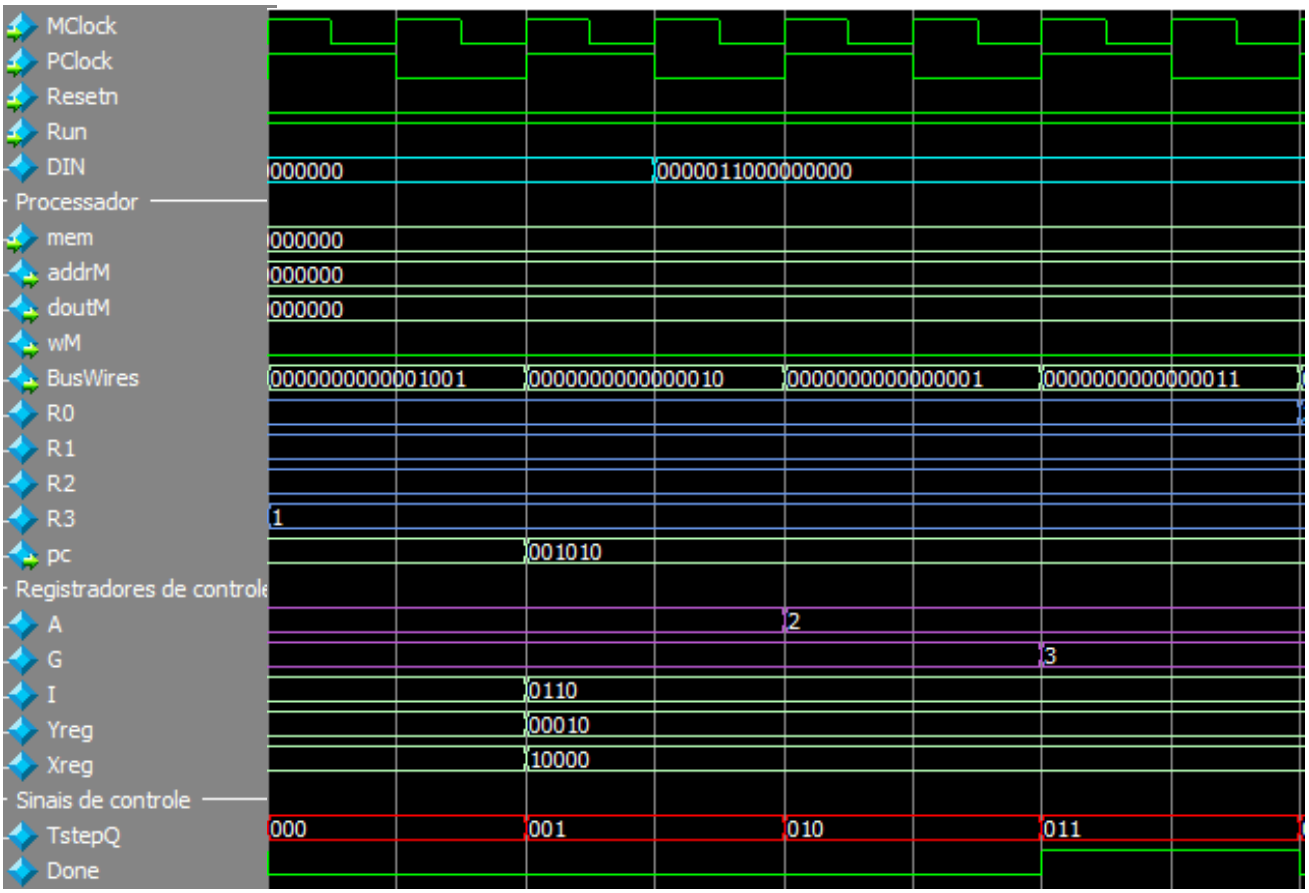
Comando MVI R0, #2



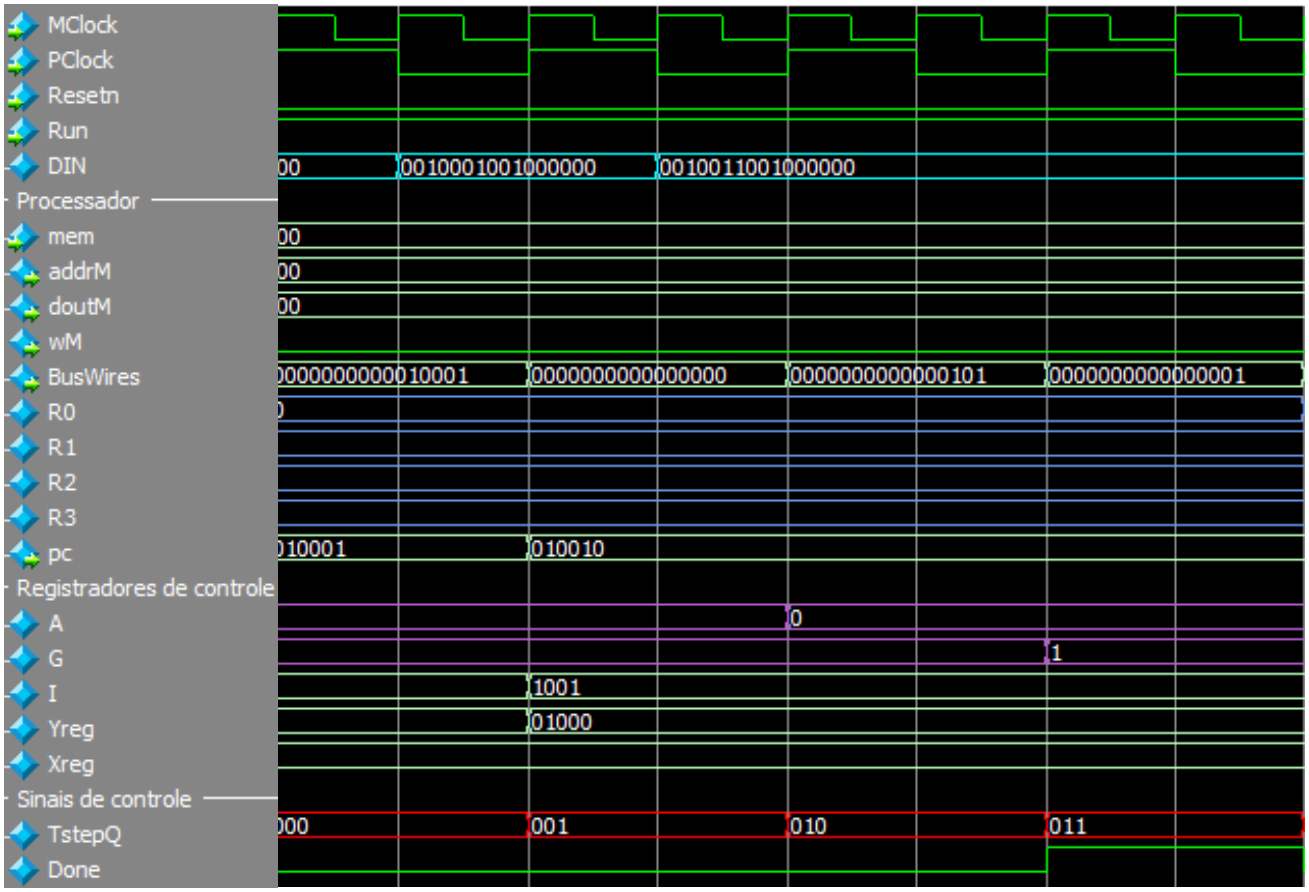
Comando ADD R1, R0



Comando OR R1, R0



Comando SLT R0, R1



* Configurações:

Clock MClock → período 20 // Borda de subida
Clock PClock → período 40 // Borde de descida
Reset → Force 0
Run → Force 1
OBS: A memória deve sempre ter a metade do período de clock do PC

Análise do código

* Módulo TLB

A TLB recebe uma página virtual contendo o endereço de uma página física. A partir de uma busca totalmente associativa e caso a página virtual seja encontrada na TLB, ela retorna a página física.

* Módulo PROC

A lógica é aplicada. Dividida em tempos, as instruções são decodificadas e os sinais necessários são aplicados, então, os comandos têm suas operações atendidas por procedimentos internos ao processador e por módulos auxiliares (a ULA e os multiplexadores, por exemplo).

* Módulo de teste

```
module pratica2 (MClock, PClock, Resetn, Run);

   input MClock, PClock, Resetn, Run;


   wire [15:0] DIN, BusWires;
   wire [5:0] addrCount;
   wire Resetn, Run, Done;

   wire [5:0] addrM;
   wire [15:0] addr, dout;
   wire w;
   wire [15:0] memOut;

   assign addrM = addr[4:0];

   ifetch ifetch (addrCount, MClock, DIN);
   ramlpm memory (addrM, MClock, dout, w, memOut);
   proc proc (DIN, memOut, Resetn, PClock, Run, Done, BusWires, addrCount, addr, dout, w);


endmodule
```