

Prática 2

Lucas Loscheider Reis Muniz

João Guilherme Monteiro Guimarães

1 – Registrador: responsável por simular um flip-flop do tipo D.

O flip-flop do tipo D possui apenas uma entrada, um terminal de clock e duas saídas (Q e $\sim Q$).

Module registrador (p_Input, p_Clock, p_WriteOn, p_Output);

```
Input p_Clock, p_WriteOn; // clock e variável p_WriteOn responsável por autorizar ou
//o registro da entrada na saída.
```

```
Input [15:0] p_Input; // a entrada
```

```
Output reg [15:0] p_Output; // saída
```

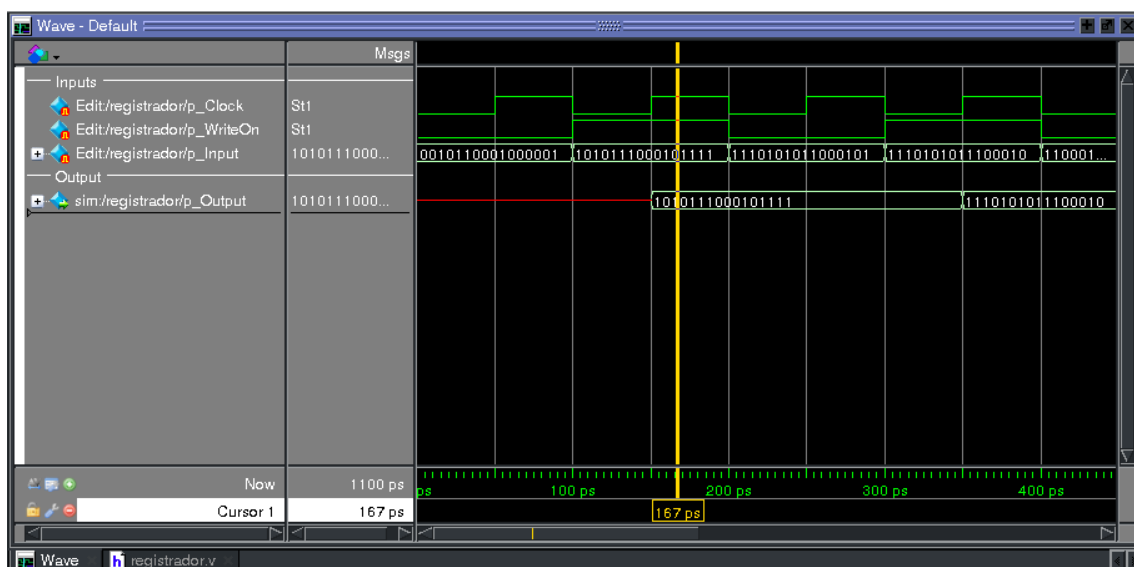
```
always@(posedge p_Clock) // Analisa se houve alteração do clock do valor zero ao um
//(borda de subida).
```

```
if (p_WriteOn) // caso p_WriteOn seja um, a saída recebera o valor da entrada.
```

```
p_Output = p_Input;
```

```
endmodule
```

Segue abaixo uma foto das instruções simuladas no modelsim:



Podemos ver que toda vez que o clock muda de zero para um, caso writeOn seja de valor um, a entrada é registrada na saída.

2 – Somador e subtrator

//Responsável por realizar a soma e subtração das entradas p_A e p_B, dependendo do valor
//de controle p_Control.

```
Module somador (p_A, p_B, p_Control, p_Output);
```

```
    Input p_Control; //identifica se será soma ou subtração
```

```
    Input [15:0] p_A, p_B; // duas entradas de 16 bits
```

```
    Output reg [15:0] p_Output; // resultado da soma ou subtração
```

```
always@(p_A, p_B, p_Control) // checa a mudança nas variáveis
```

```
    if(p_Control) // caso o controle seja de valor 1, será feita a soma
```

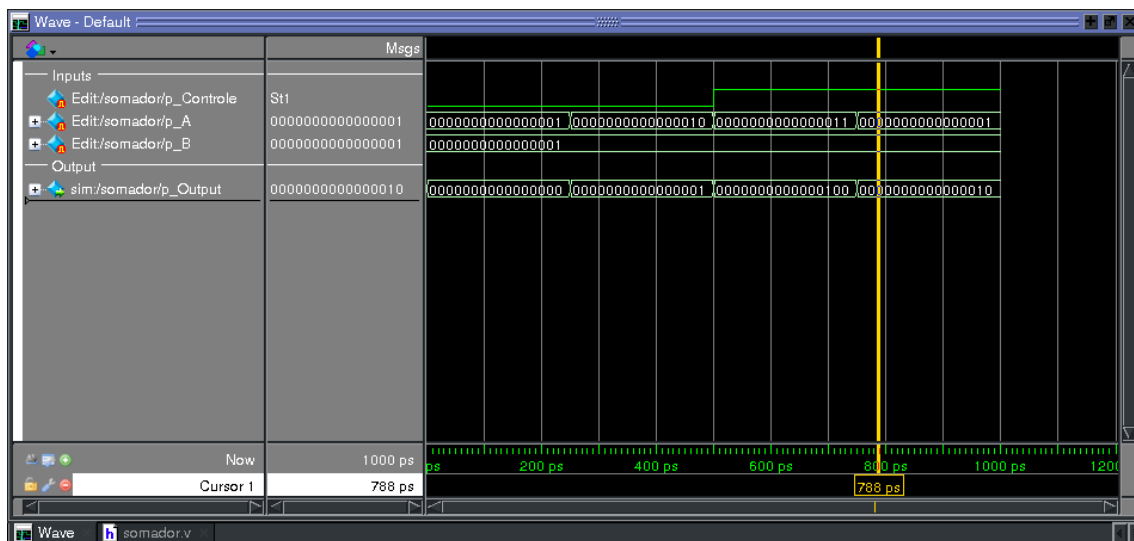
```
        p_Output = p_A + p_B;
```

```
    else // caso o controle seja de valor 0, será a subtração
```

```
        p_Output = p_A - p_B;
```

```
endmodule
```

segue a imagem abaixo



Observamos que, determinado um valor constante de p_B, com p_A variando temos na saída o resultado dessa subtração. Quando Controle passa para um, obtemos o resultado da soma.