UNIVERSIDADE FEDERAL DA PARAÍBA
DEPARTAMENTO DE INFORMÁTICA
CONCEPÇÃO ESTRUTURADA
PROF: DR. ANTONIO CARLOS CAVALCANTI

DECODIFICADOR

Edimar Bezerra da Silva Neto
Matrícula 11409565

João Pessoa / 2018

Introdução

O decodificador é um circuito que recebe uma entrada de N bits e obtém uma saída igual a 2^N, o nosso decodificador possui 5 entradas de 1 bit e 32 saídas de 1 bit, que será utilizado para selecionar o registrador do banco de registradores.

Implementação do bloco em C.

A função abaixo recebe como parâmetro o vetor de tamanho 5 e uma string com as 32 saídas.

```
void decodificador(int *s, char* vetor){
    int i,registrador;
    registrador = 16*s[0]+8*s[1]+4*s[2]+2*s[3]+s[4];
    for(i = 0; i < 32;i++){
        vetor[i] = '0';
    }
    vetor[31-registrador] = '1';
}
```

Obtemos um arquivo .tv com todas as 32 possíveis entradas e suas saídas.

decod.tv

```
00000000000000000000000000000001
00001000000000000000000000000010
00010000000000000000000000000100
00011000000000000000000000001000
00100000000000000000000000010000
00101000000000000000000000100000
00110000000000000000000001000000
00111000000000000000000010000000
01000000000000000000000100000000
01001000000000000000001000000000
01010000000000000000010000000000
01011000000000000000100000000000
01100000000000000001000000000000
01101000000000000010000000000000
01110000000000000100000000000000
01111000000000001000000000000000
10000000000000010000000000000000
10001000000000100000000000000000
10010000000001000000000000000000
10011000000010000000000000000000
10100000000100000000000000000000
10101000001000000000000000000000
10110000010000000000000000000000
10111000100000000000000000000000
11000001000000000000000000000000
11001000010000000000000000000000
11010000100000000000000000000000
11011000010000000000000000000000
11100001000000000000000000000000
11101001000000000000000000000000
11110010000000000000000000000000
11111100000000000000000000000000
```

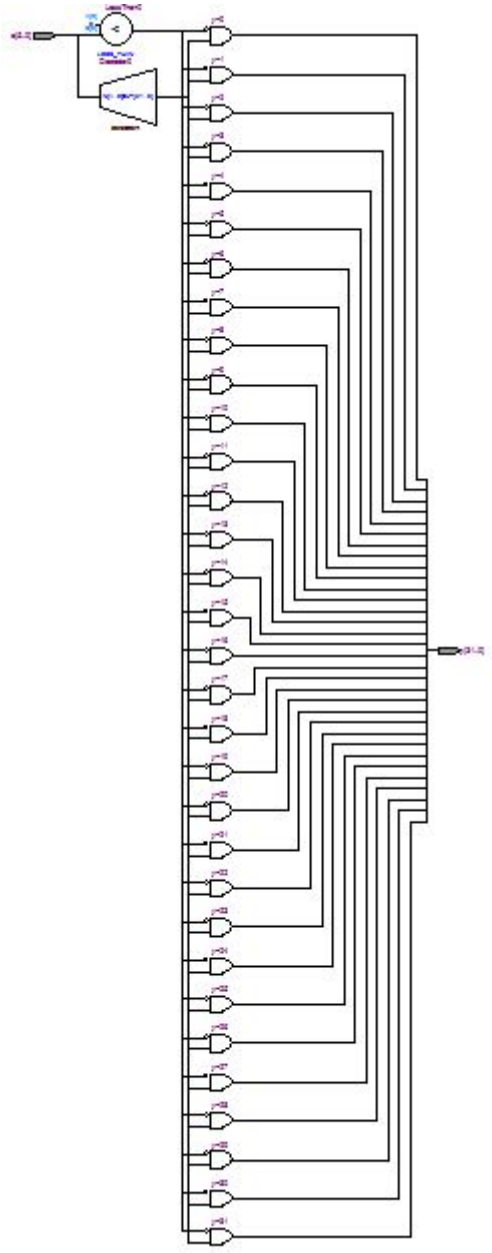
a(5 entradas) e y(32 saídas).

Implementação no Quartus

```
module decoder(input logic [5:0] a,output logic [31:0] y);

always_comb begin
    y = 0;
    y[a] = 1;
    end
endmodule
```

RTL Viewer



TestBench

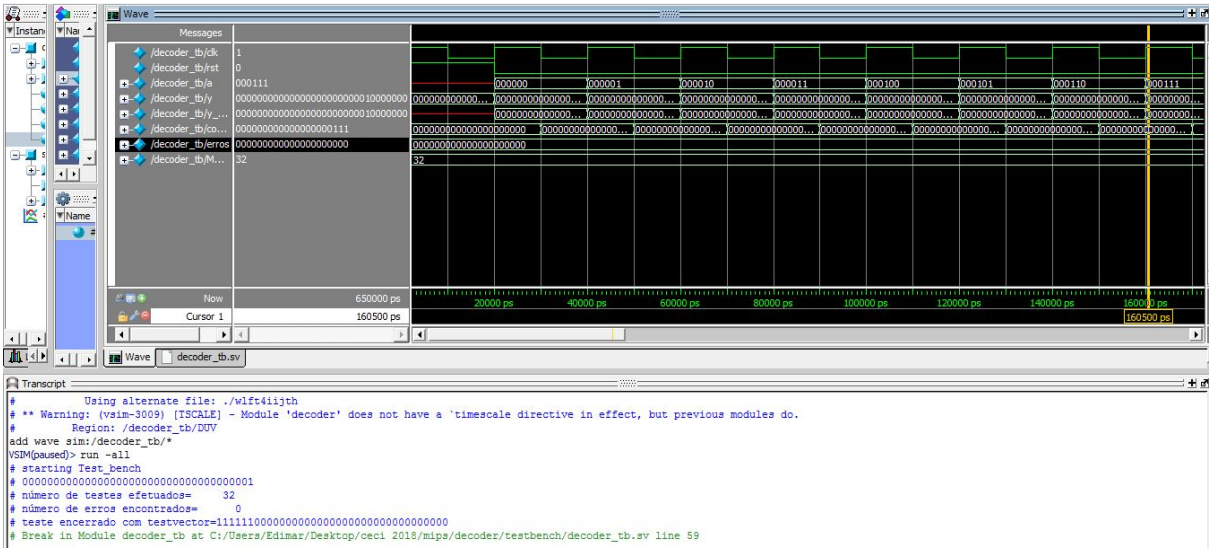
```
`timescale 1ns / 100ps module decoder_tb;

    logic clk, rst;
    logic [5:0] a;
    logic [31:0] y, y_esperado;
    logic [36:0] entrace [0:31];
    logic [0:20] counter,erros;

const int MEMSIZE = 32;
        decoder DUV(
        .a(a),
        .y(y)
        );
    always begin
        clk = 1;
        #10;
        clk = 0;
        #10;
    end
    initial
      begin
      counter = 0;
      erros=0;
        $display("starting Test_bench");
        $readmemb("decod.tv",entrace);
        $display("%b",entrace[counter]);
        rst = 1;
        #20;
        rst = 0;
        end

    always @(posedge clk)
    if (~rst)
    begin // skip during reset
            a[5:0]= entrace[counter][36:32];
            y_esperado[31:0] = entrace[counter][31:0];
        end
```

Simulação RTL Level



Simulação Gate Level