



Universidade Federal do Rio Grande do Norte  
Centro de Tecnologia - CT  
Circuitos Digitais

## **REGISTRADOR DE MÚLTIPLAS FUNÇÕES**

ELE2715 - Laboratório 10

Isaac de Lyra Junior

Natal, 04 de março de 2021

Isaac de Lyra Junior

## **REGISTRADOR DE MÚLTIPLAS FUNÇÕES**

Projeto da disciplina de Circuitos Digitais do  
Departamento de Engenharia Elétrica da Universidade  
do Rio Grande do Norte para relatório das atividades.

Docente: Samaherni Moraes Dias

Natal, 04 de março de 2021

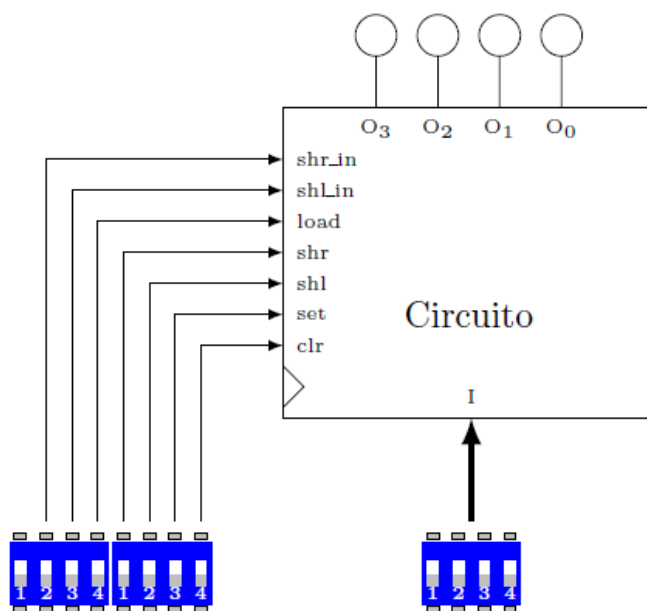
## **SUMÁRIO**

<b>1 DESENVOLVIMENTO</b>	<b>4</b>
<b>2 RESULTADOS</b>	<b>6</b>
<b>3 CONCLUSÃO</b>	<b>9</b>
<b>REFERÊNCIAS</b>	<b>10</b>
<b>ANEXO A - CIRCUITO ESQUEMÁTICO COMPLETO</b>	<b>11</b>

## 1 DESENVOLVIMENTO

A problemática do laboratório é implementar um registrador de múltiplas funções. O registrador possui ao todo seis funções distintas com ordem de prioridade entre elas, por ordem: CLEAR assíncrono, manter, carregar, deslocar à direita, deslocar à esquerda e set síncrono). A Figura 1 mostra o bloco do circuito que deve ser implementado. Se  $SHR = 1$ , o registrador deverá deslocar, após o pulso de clock, os bits da saída para a direita com o bit de entrada dado por shr\_in. Se  $SHL = 1$ , o registrador deverá delocar, após o pulso de clock, os bits da saída para a esquerda com o bit de entrada dado por shl\_in. Se  $SET = 1$ , todos os bits da saída do registrador, após o pulso de clock, devem ir para 1. Se  $LOAD = 1$ , o registrador deverá fazer com que a saída  $O[3:0]$ , após o pulso de clock, receba o valor da entrada  $I[3:0]$ . Por fim, se  $CLR=1$ , todos os bits da saída do registrador devem ir para 0 imediatamente.

**Figura 1** - Bloco do circuito do registrador de múltiplas funções.



**Fonte:** Dados do problema.

O primeiro passo foi transformar o comportamento que o circuito deverá ter em uma tabela verdade, que irá fornecer como saída uma chave  $K[2:0]$  que será utilizada para escolher as funções e precedência das mesmas do nosso circuito, a Figura 2 mostra a tabela criada.

**Figura 2** - Tabela verdade de precedência do registrador

FUNÇÕES	ENTRADAS				SAÍDAS		
	LOAD	SHR	SHL	SET	K2	K1	K0
MANTER	0	0	0	0	0	0	0
CARREGAR	1	X	X	X	0	0	1
DESLOCAR DIREITA	0	1	X	X	0	1	0
DESLOCAR ESQUERDA	0	0	1	X	0	1	1
SET	0	0	0	1	1	0	0

**Fonte:** Elaborado pelo autor.

Em seguida foi utilizada a tabela verdade da Figura anterior para definir a lógica combinacional dos bits da chave **K**. O resultado está Figura 3

**Figura 3** - Lógica combinacional dos bits da chave K

SAÍDA	LÓGICA COMBINACIONAL
K2	$\sim\text{LOAD} \sim\text{SHR} \sim\text{SHL} \text{ SET}$
K1	$\sim\text{LOAD} \text{ SHL} + \sim\text{LOAD} \text{ SHR}$
K0	$\sim\text{SHR} \text{ SHL} + \text{LOAD}$

**Fonte:** Elaborado pelo autor.

A idéia é utilizar **K** como chave de quatro multiplexadores 8x1 que vai ser responsável por selecionar qual função deve ocorrer, cada saída dos multiplexadores são conectadas a um Flip Flop do tipo D, a saída destes flip flops será a saída **O**[3:0] do circuito. Desta forma, os multiplexadores terá como primeira entrada o valor armazenado na saída dos flip flops, ou seja, uma realimentação, essa lógica será utilizada na função manter, para a função carregar, foi definido que a segunda entrada dos multiplexadores será o valor **I**[3:0], desta forma, quando a chave **K** selecionar esta saída, os flip flops irão carregar este valor na saída, após o pulso de clock.

Para a função deslocar à direita, foi colocado como entradas dos multiplexadores os bits carregados no flip flop à esquerda, com exceção do multiplexador mais à esquerda que irá carregar o SHR\_IN no flip flop responsável pelo bit mais significativo de **O**. A mesma lógica é utilizada para a função deslocar à esquerda, sendo que dessa vez carregando os bits dos flip flops à direita, com exceção do flip flop responsável pelo bit menos significativo, este carregará SHL\_IN. Por fim, para a função carregar é definido como entrada dos multiplexadores os bits da entrada **I**[3:0] do circuito, assim, após o pulso de clock é carregado este valor na saída. A função

clear, por ser assíncrona, foi definida como entrada reset destes flip flops, desta forma, assim que  $CLR = 1$ , os flip flops são zerados.

## 2 RESULTADOS

Para a implementação do circuito deste registrador de funções foi utilizado o software Proteus em sua versão 8.9. Como componentes foi utilizado CI's comerciais dados na Tabela 1.

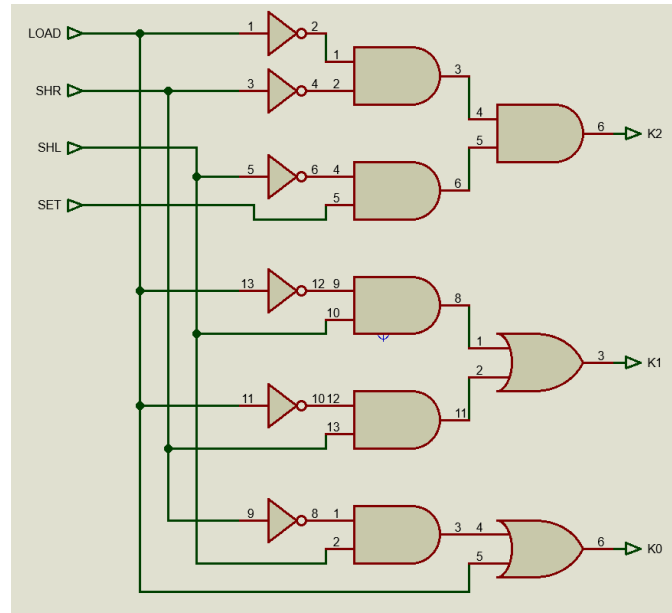
**Tabela 1** - Componentes utilizados

FUNÇÃO	CI COMERCIAL
NOT	7404
AND[2]	7408
OR[2]	7432
MULTIPLEXADOR 8X1	74151
FLIP FLOP D	4013
RESISTOR	560 $\Omega$

**Fonte:** Elaborado pelo autor.

Com toda a lógica combinacional definida, foi possível implementar o bloco responsável por definir os bits da chave **K**, que utiliza como entrada o LOAD, SHR, SHL e SET, o resultado está exposto na Figura 4.

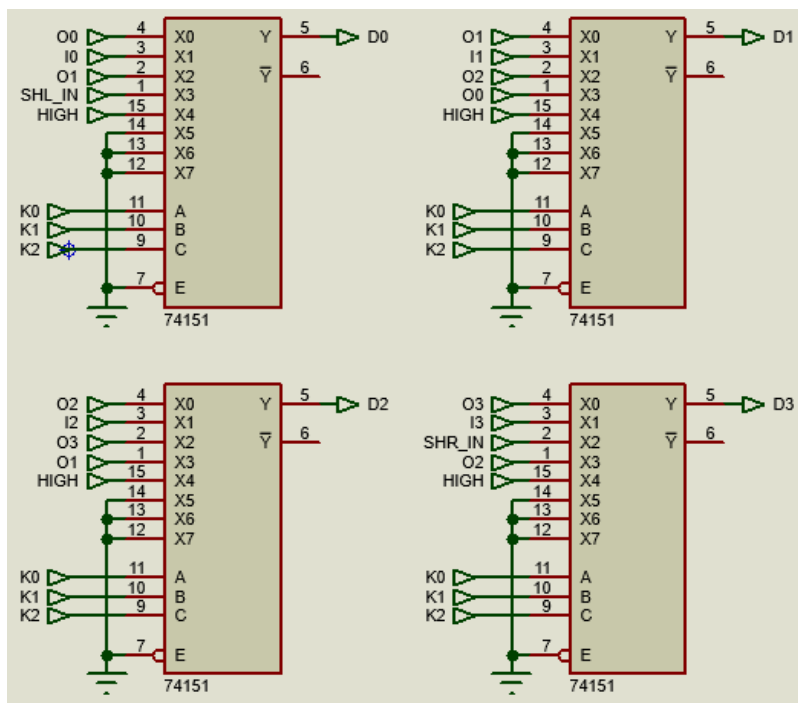
**Figura 4** - Circuito combinacional do seletor **K**



Fonte: Elaborado pelo autor.

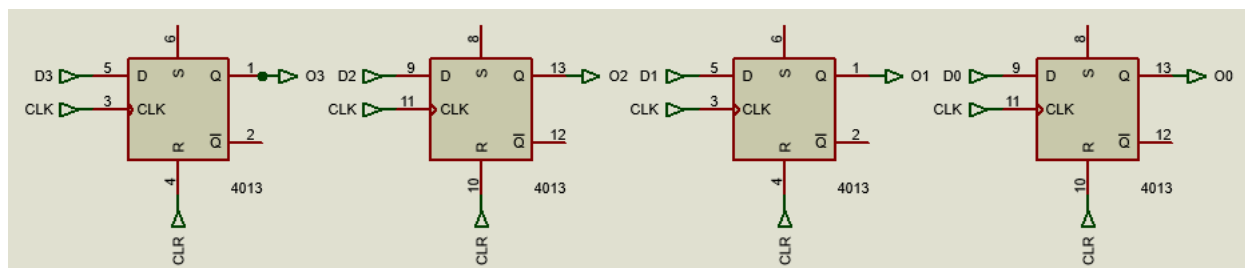
Em seguida, foi utilizado **K** como chave seletora de 4 multiplexadores 8x1, a saída de cada multiplexador é utilizada como entrada em um flip flop D, está na Figura 5 e 6 respectivamente.

Figura 5 - Conteúdo do arquivo .do



Fonte: Elaborado de autor.

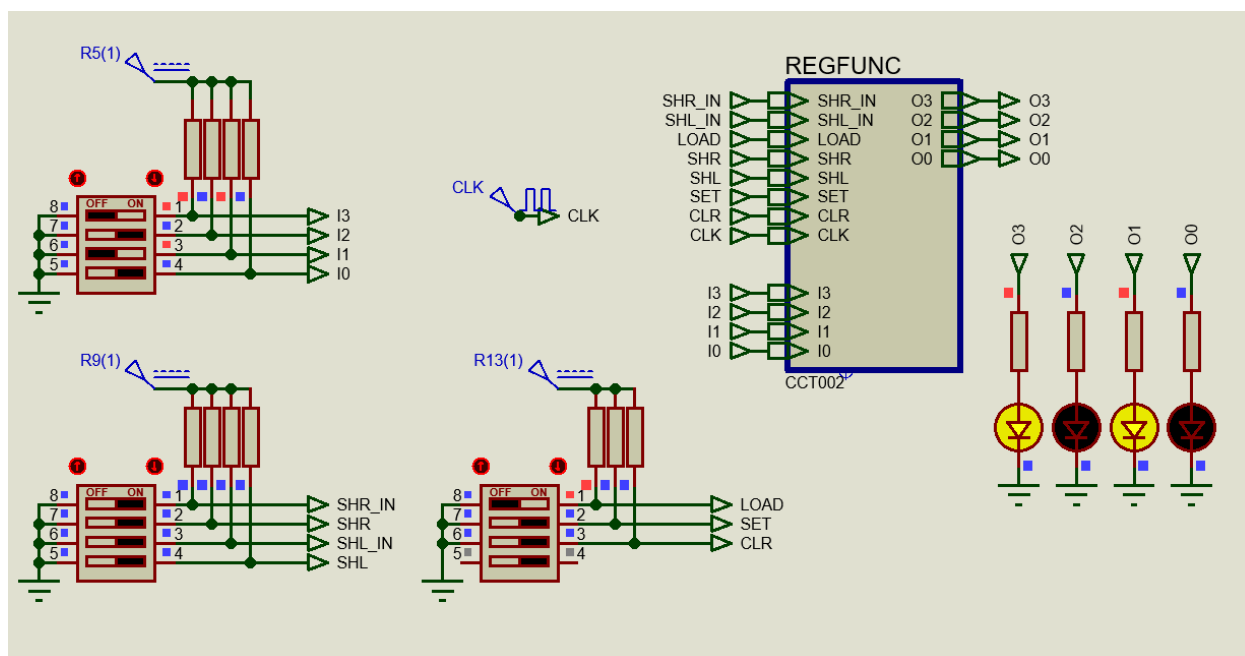
**Figura 6 -** Conteúdo do arquivo .do



**Fonte:** Elaborado de autor.

Por fim, o circuito ficou como demonstrado na Figura 7, na mesma figura também é possível ver uma simulação do registrador de funções realizando a função LOAD, carregando o número  $I[3:0]$  na saída  $O[3:0]$ . O esquemático como um todo pode ser visto no ANEXO A.

**Figura 9 -** Simulação da função LOAD



**Fonte:** Elaborado pelo autor.



### **3 CONCLUSÃO**

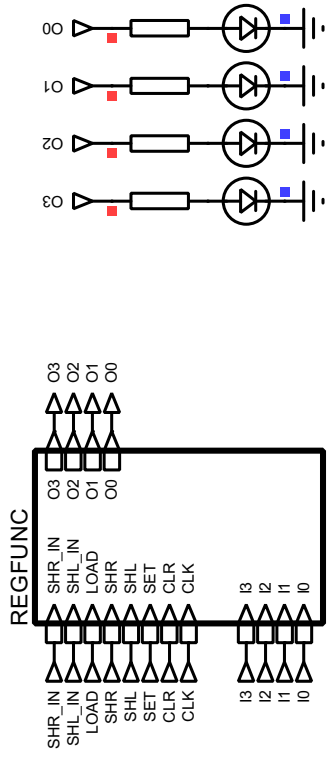
O relatório tinha como finalidade apresentar a implementação de um registrador de funções que terá 6 funções, tais funções também possuem uma ordem de prioridade. Para conseguir implementar esta ordem de prioridade foi feito uma tabela verdade e feito uma chave seletora para selecionar qual função deve ser selecionada a partir das entradas do circuito. Toda a implementação e simulação foi realizada no software Proteus.

## REFERÊNCIAS

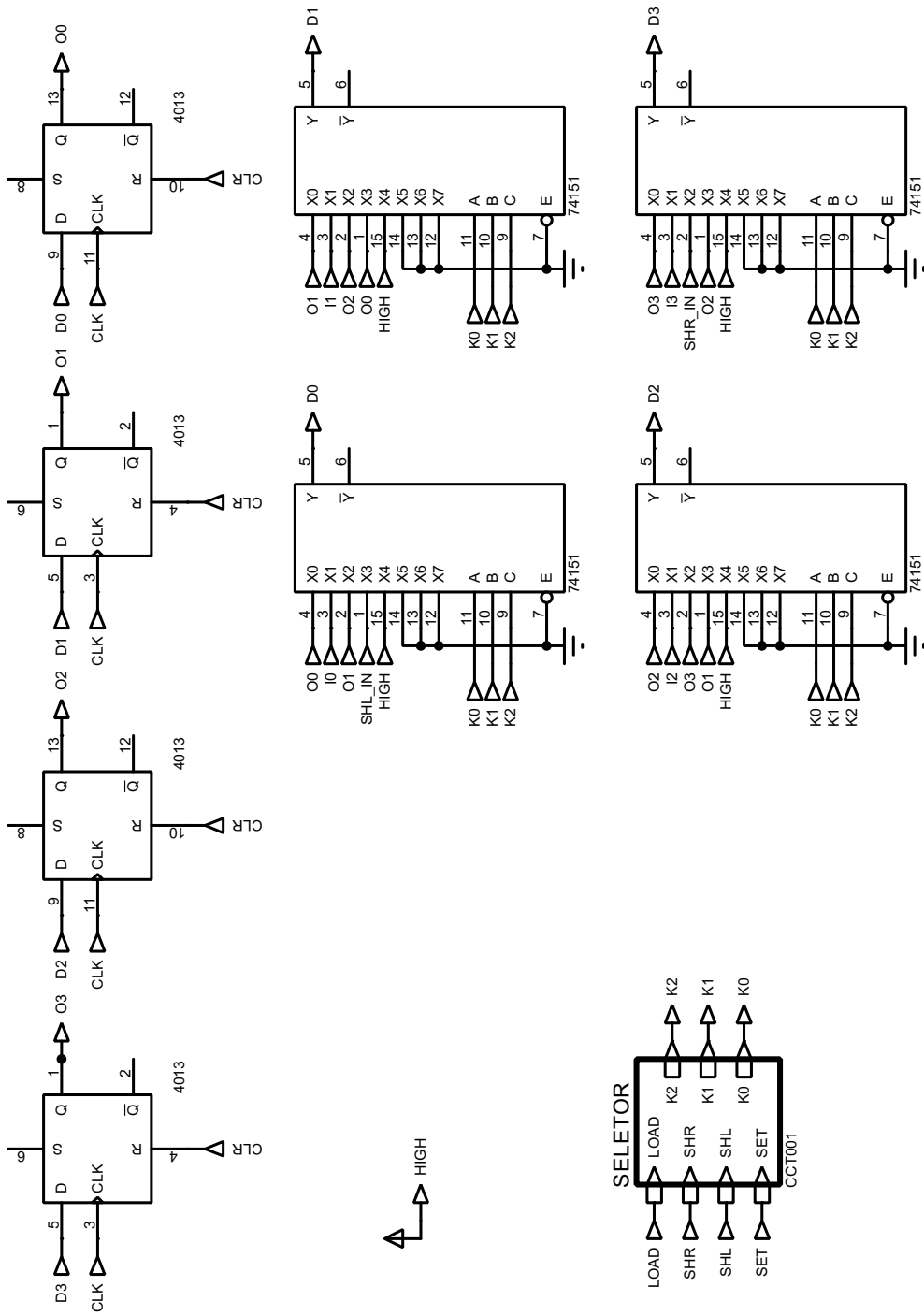
TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L.. **Sistemas Digitais: Princípios e Aplicações**. 11a ed. São Paulo: Pearson, 2011. 844 p.

VAHID, Frank. **Sistemas digitais: projeto, otimização e HDLS**. Rio Grande do Sul: Artmed Bookman, 2008. 558 p.

## ANEXO A - CIRCUITO ESQUEMÁTICO COMPLETO



# SUBCIRCUITO REGFUNC



SUBCIRCUITO SELETOR

