



Laboratório de Circuitos Digitais
Igor dos Reis Gomes
Fernando Hiroshi Murusaki

RELATÓRIO DECODIFICADOR (FLEXOWRITER)

Laboratório de Circuitos Digitais

Bauru
2024

Bauru

2024

Igor dos Reis Gomes

Fernando Hiroshi Murusaki

RELATÓRIO DECODIFICADOR (FLEXOWRITER)

Laboratório de Circuitos Digitais

Trabalho teórico-prático apresentado
como requisito para primeira avaliação
semestral de Laboratório de Circuitos
Digitais – Prof. Me. Dr. Marco Aurelio
Rocha.

INTRODUÇÃO

Neste projeto, nossa meta foi compreender e implementar um circuito que simulasse o funcionamento de uma Flexowriter de 6 bits. A Flexowriter é uma antiga máquina de escrever que utilizava um sistema de decodificação para converter caracteres e símbolos em dados que pudessem ser interpretados por máquinas. O foco principal do trabalho foi realizar a codificação e decodificação de caracteres com base em uma tabela de caracteres predefinida (Tabela 1).

O circuito foi desenvolvido para receber um código que representa um caractere e, a partir desse código, produzir uma saída correspondente, que pode ser uma letra, um número ou, em casos de erro, uma mensagem informando que o caractere fornecido não está nos intervalos válidos (de A a Z ou de 0 a 9). Essa implementação não apenas facilita o entendimento dos sistemas de codificação, mas também oferece uma visão sobre como as máquinas de escrever funcionavam e destaca a importância da validação de dados nesse tipo de sistema.

Figura 1 - Códigos de Caracteres

A = 001000	B = 000101	C = 101110	D = 110010	E = 000000	F = 110110
G = 010100	H = 000010	I = 101100	J = 111010	K = 111110	L = 010010
M = 011000	N = 100110	O = 000100	P = 001010	Q = 111100	R = 101010
S = 110100	T = 101000	U = 001100	V = 000110	W = 111000	X = 010110
Y = 011010	Z = 010000	0 = 000001	1 = 001110	2 = 000111	3 = 100111
4 = 010011	5 = 010001	6 = 010101	7 = 011110	8 = 011100	9 = 000011

Fonte: Folha de prova fornecida pelo professor

MATERIAIS USADOS

Utilizamos um total de cinco CIs (Circuitos Integrados) para concluir o projeto. O CI NOT 7404 foi utilizado para realizar a negação de todas as entradas, enquanto o CI NAND 7410 (com três entradas) foi empregado para reduzir a complexidade de algumas operações. Além disso, usamos três CIs NAND 7400 (com duas entradas), que foram responsáveis pela maior parte das operações booleanas. Para as saídas, utilizamos três LEDs vermelhos, representando os sinais A, N e E. A interligação de todos os componentes foi feita utilizando jumpers, conforme necessário.

PROCEDIMENTOS

3.1 Tabela de verdade

Tabela 1 - Tabela de verdade

a	b	c	d	e	f	A	N	E	CARACTER
0	0	0	0	0	0	1	0	0	E
0	0	0	0	0	1	0	1	0	0
0	0	0	0	1	0	1	0	0	H
0	0	0	0	1	1	0	1	0	9
0	0	0	1	0	0	1	0	0	O
0	0	0	1	0	1	1	0	0	A
0	0	0	1	1	0	1	0	0	V
0	0	0	1	1	1	0	1	0	2
0	0	1	0	0	0	1	0	0	B
0	0	1	0	0	1	X	X	1	ERRO
0	0	1	0	1	0	1	0	0	P
0	0	1	0	1	1	X	X	1	ERRO
0	0	1	1	0	0	1	0	0	U
0	0	1	1	0	1	X	X	1	ERRO
0	0	1	1	1	0	0	1	0	1
0	0	1	1	1	1	X	X	1	ERRO

0	1	0	0	0	0	1	0	0	Z
0	1	0	0	0	1	0	1	0	5
0	1	0	0	1	0	1	0	0	L
0	1	0	0	1	1	0	1	0	4
0	1	0	1	0	0	1	0	0	G
0	1	0	1	0	1	0	1	0	6
0	1	0	1	1	0	1	0	0	X
0	1	0	1	1	1	X	X	1	ERRO
0	1	1	0	0	0	1	0	0	M
0	1	1	0	0	1	X	X	1	ERRO
0	1	1	0	1	0	1	0	0	Y
0	1	1	0	1	1	X	X	1	ERRO
0	1	1	1	0	0	0	1	0	8
0	1	1	1	0	1	X	X	1	ERRO
0	1	1	1	1	0	0	1	0	7
0	1	1	1	1	1	X	X	1	ERRO
1	0	0	0	0	0	X	X	1	ERRO
1	0	0	0	0	1	X	X	1	ERRO
1	0	0	0	1	0	X	X	1	ERRO
1	0	0	0	1	1	X	X	1	ERRO
1	0	0	1	0	0	X	X	1	ERRO
1	0	0	1	0	1	X	X	1	ERRO
1	0	0	1	1	0	1	0	0	N
1	0	0	1	1	1	0	1	0	3
1	0	1	0	0	0	1	0	0	T
1	0	1	0	0	1	X	X	1	ERRO
1	0	1	0	1	0	1	0	0	R
1	0	1	0	1	1	X	X	1	ERRO
1	0	1	1	0	0	1	0	0	I
1	0	1	1	0	1	X	X	1	ERRO
1	0	1	1	1	0	1	0	0	C
1	0	1	1	1	1	X	X	1	ERRO
1	1	0	0	0	0	X	X	1	ERRO
1	1	0	0	0	1	X	X	1	ERRO

1	1	0	0	1	0	1	0	0	D
1	1	0	0	1	1	X	X	1	ERRO
1	1	0	1	0	0	1	0	0	S
1	1	0	1	0	1	X	X	1	ERRO
1	1	0	1	1	0	1	0	0	F
1	1	0	1	1	1	X	X	1	ERRO
1	1	1	0	0	0	1	0	0	W
1	1	1	0	0	1	X	X	1	ERRO
1	1	1	0	1	0	1	0	0	J
1	1	1	0	1	1	X	X	1	ERRO
1	1	1	1	0	0	1	0	0	Q
1	1	1	1	0	1	X	X	1	ERRO
1	1	1	1	1	0	1	0	0	K
1	1	1	1	1	1	X	X	1	ERRO

Fonte: Elaboração própria

3.2 Redução da tabela de verdade

Assim que começamos o planejamento do circuito entendemos que a maior dificuldade do projeto seria reduzir a tabela de verdade de forma tradicional, pois precisaríamos fazer um mapa de Karnaugh utilizando "dont cares" de forma com que diminuísse a quantidade de portas lógicas utilizadas para a realização de 3 saídas.

Inicialmente, tivemos esses resultados como redução das portas lógicas:

$$A = D'A' + C'A' + FA' + E'CB'$$

$$N = C'A + BA + EA + F'DCB + F'EDC$$

Esses resultados não somente não atingiam a meta inicial de somente 6 CI's como também tinham várias redundâncias que poderiam ser removidas com um olhar mais crítico para o circuito.

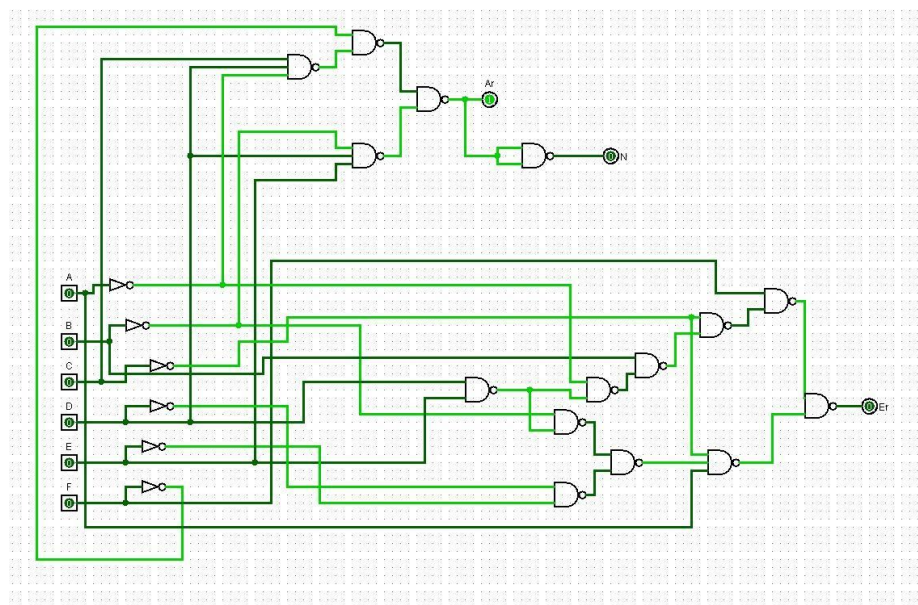
3.3 Simplificação da expressão

Ao analisar novamente a tabela de verdade percebemos alguns padrões notáveis que poderiam nos ajudar, a saída A sempre é a negação de N e a saída E é sempre verdadeira em padrões de XNOR entre A e N. Utilizamos isso como lógica para criar uma nova expressão para as saídas Ao reduzir o circuito e testar outros métodos chegamos a uma expressão que seria a escolhida para a criação do circuito.

$$A = (EDB')(CDA'F')$$

$$N = A' = (EDB')'(CDA'F')'$$

Figura 2 - Circuito final montado no aplicativo Logisim

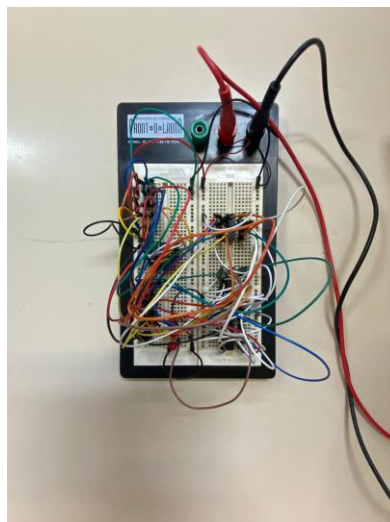


Fonte: Elaboração própria

4 MONTAGEM DO CIRCUITO

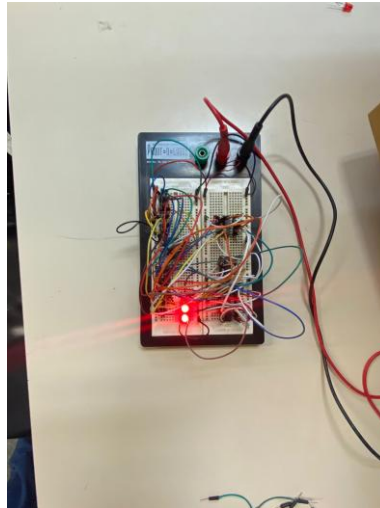
Após muita preparação, a montagem do circuito foi um sucesso, durou três horas e vinte minutos e não houve problemas.

Figura 3 - Circuito final desligado



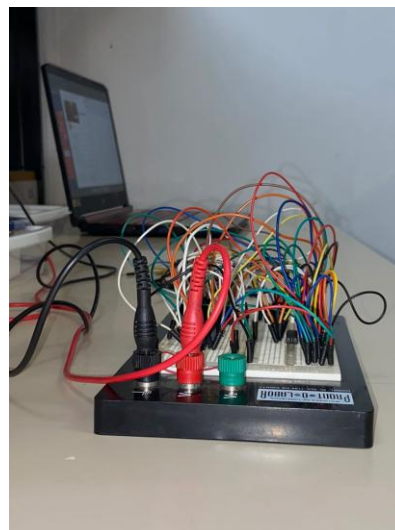
Fonte: Elaboração própria

Figura 4 - Circuito final ligado



Fonte: Elaboração própria

Figura 5 - Circuito final desligado (outro ângulo)



Fonte: Elaboração própria

5 REGISTRO DE PROJETO

DATA	TEMPO DE INÍCIO	TEMPO DE TÉRMINO	TEMPO GASTO	CATEGORIA DE TRABALHO	DESCRIÇÃO DA ATIVIDADE
20/09	15:00	15:55	1 hora	P	Construção da tabela de verdade

20/09	15:55	16:55	1 hora	P	Construção do mapa de Karnaugh
20/09	16:55	18:00	1 hora	P	Estudo das possíveis formas de implementações do circuito
27/09	14:40	17:40	3 horas	P	Redução do mapa de Karnaugh
27/09	17:40	18:00	20 minutos	P	Criação do circuito no Logisim
04/10	14:30	17:00	3 horas e 30 minutos	C	Construção do circuito
04/10	17:00	17:10	10 minutos	D	Teste de combinações de entradas no circuito
05/10	14:00	16:00	2 horas	C	Elaboração do relatório

CONCLUSÃO

Com o desenvolvimento deste projeto, tivemos a oportunidade de trabalhar com a Flexowriter de 6 bits e aprofundar nosso conhecimento a ponto de criar uma solução prática e eficiente. Aplicamos os conceitos teóricos vistos em aula de forma integrada, canalizando todo o esforço desta primeira metade do semestre em um único propósito.

Passamos por várias fases, desde a análise do problema até a implementação final. Inicialmente, trabalhamos com tabelas verdade e mapas de Karnaugh, conteúdos apresentados pelo professor Danilo Budoya. Posteriormente, utilizamos a álgebra booleana para otimizar as equações lógicas. Com essas etapas concluídas, partimos para a montagem e esquematização dos circuitos, incluindo testes e depurações. Graças à organização acumulada em projetos anteriores, conseguimos finalizar tudo com sucesso logo na primeira tentativa.

Ao final, o projeto não só atendeu aos requisitos acadêmicos, como também superou nossas expectativas. Ele nos proporcionou uma nova visão sobre como abordar e resolver problemas, além de ter sido uma experiência enriquecedora e muito gratificante.

