



UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
EEL7020 – SISTEMAS DIGITAIS

PROJETO

Transmissão de dados com detecção e correção de erros usando Código de Hamming
--

Equipe:

Diogo Feltrin Pereira – 07141073
Gesiel Antônio Martins – 06241063
Michell Luiz Flesch – 06241058

Florianópolis, 26 de novembro de 2007.

1 INTRODUÇÃO

Neste trabalho será implementado um circuito digital que realize a transmissão serial de dados de um circuito transmissor para um circuito receptor. Estes dados serão codificados utilizando Código de Hamming, pois, este método proporciona a correção dos dados recebidos pelo circuito receptor caso ocorra algum erro num qualquer bit destes dados.

Descreveremos os métodos que foram utilizados para o desenvolvimento de todo circuito sua implementação e fase de testes.

O conteúdo deste projeto foi proposto pela disciplina de Sistemas Digitais com o propósito de fixar os conhecimentos adquiridos neste semestre.

2 DESCRIÇÃO DO PROJETO

Implementar um circuito para transmissão e recepção de dados com detecção e correção de erros usando Código de Hamming. O diagrama funcional básico pode ser observado na figura 2.1.

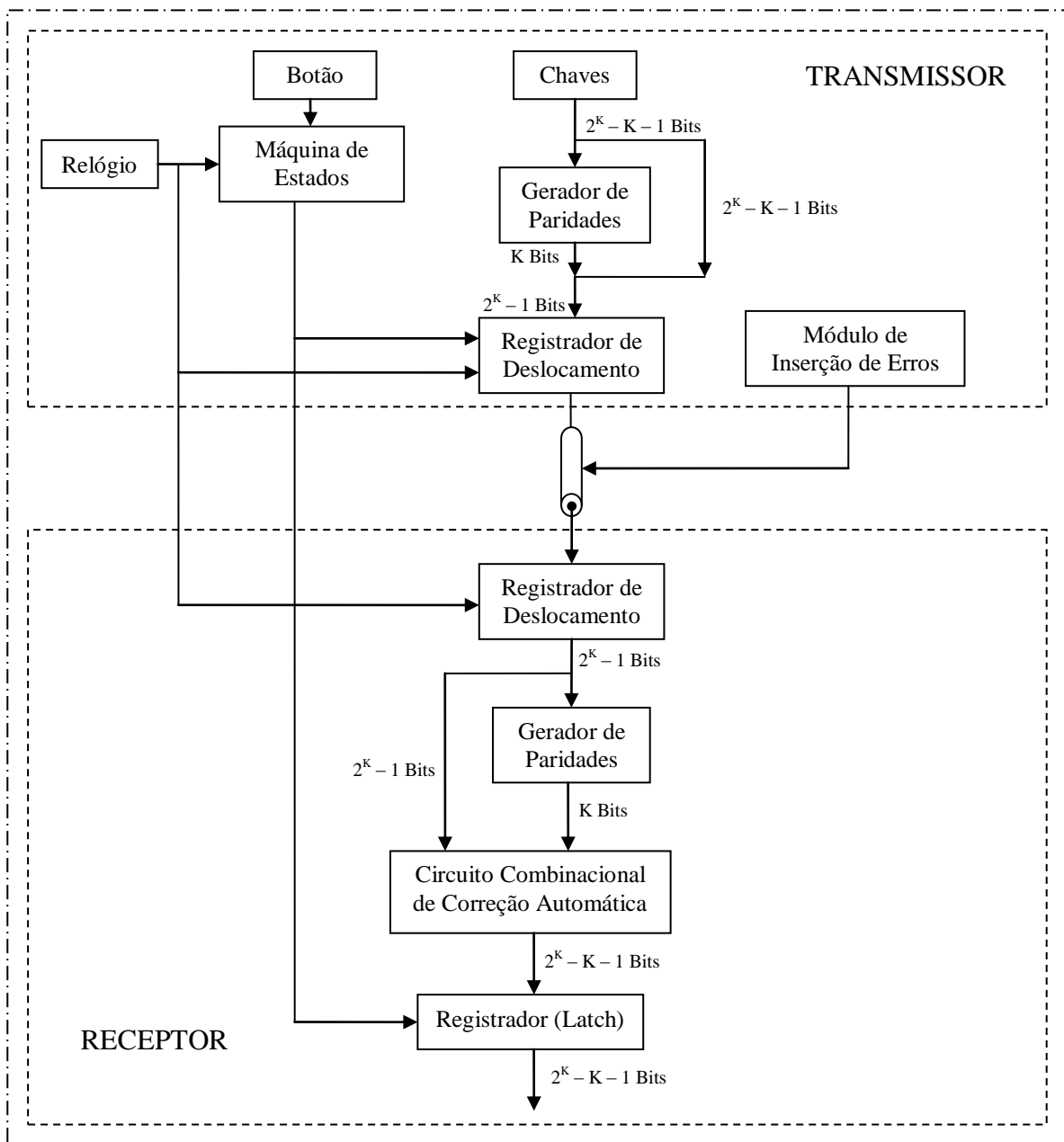


Figura 2.1 – Diagrama funcional básico do projeto

2.1 Código de Hamming

Este método consiste em adicionar bits de paridade a uma palavra de dados. Esses bits formam um código que será utilizado para correção da palavra de dados caso ocorra algum erro num bit destes dados.

A geração dos bits de paridade é realizada utilizando-se portas lógicas OU-EXCLUSIVO (XOR).

Todo processo de codificação e decodificação será demonstrada nos circuitos de transmissão e recepção dos dados.

2.2 Especificações iniciais do projeto

1. Considerar um sistema de transmissão que usa um circuito que geram paridades baseadas em código de Hamming.
2. No lado do transmissor a informação de quatro bits gerada a partir de chaves é usada para gerar os bits de paridade que serão concatenados a esta mesma informação formando a mensagem que será transmitida.
3. Um circuito para inserir erros deve ser construído. Os erros se resumirão em alterar só um bit da mensagem.
4. Os circuitos usados na conversão paralelo-série e série-paralelo que fazem a transmissão serão registros de deslocamento.
5. No lado do receptor a mensagem gera a partir do gerador de paridades bits que dão a posição do bit da mensagem que sofreu alteração.
6. Circuito combinacional de correção automática deve retornar a informação original de quatro bits que foi definida para a transmissão.
7. Um relógio projetado a partir de um 555 deverá gerar o sincronismo a nível de bit, transmitido em uma linha separada da de dados.
8. Uma máquina de estados deverá ser projetada definindo sinais de controle que também serão transmitidos em uma linha separada tanto para o transmissor como para o receptor.

3 PARTE EXPERIMENTAL

3.1 Descrição do Circuito

Demonstraremos a seguir o desenvolvimento de todo projeto. Para melhor entendimento, descrevemos em sequência os seguintes circuitos: transmissor, receptor, máquina de estados, relógio e gerador de erros.

3.1.1 Transmissor

O circuito transmissor (Figura 3.1) possui um registrador paralelo-serial de acordo com a especificação inicial do projeto. Neste módulo do circuito os quatro bits de entrada são usados para gerar uma combinação de bits de paridade. Essa combinação representa uma codificação e será importante no circuito receptor para a correção dos bits recebidos.

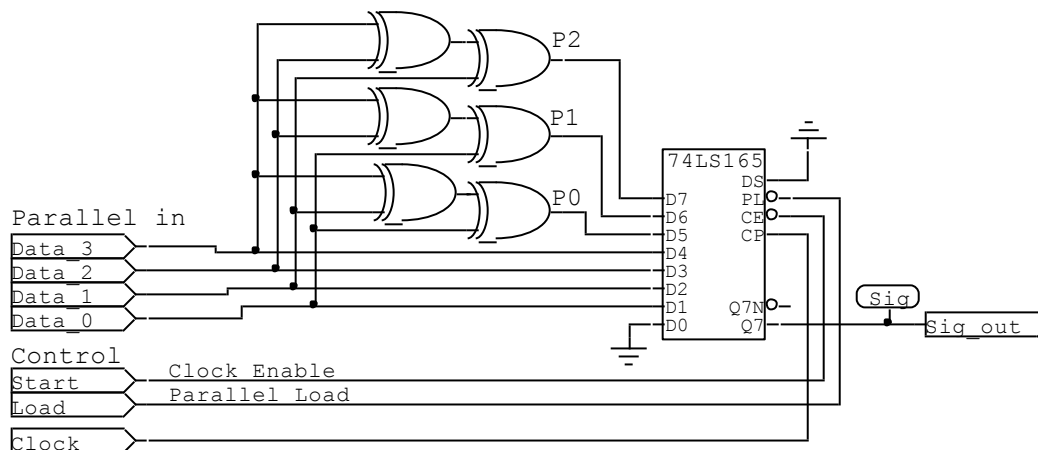


Figura 3.1 – Circuito transmissor paralelo-serial.

A combinação gerada possui a seguinte expressão booleana:

- $P0 = d0 \oplus d1 \oplus d3$
- $P1 = d0 \oplus d2 \oplus d3$
- $P2 = d1 \oplus d2 \oplus d3$

Obs: O símbolo \oplus representa a expressão OU-EXCLUSIVO (XOR).

Análise do funcionamento do 74LS165:

O esquema do circuito transmissor utiliza o CI 74LS165. Observe que as entradas Ds e D_0 não estão sendo utilizadas e por isso, foram conectadas ao GND. Os bits de dados juntamente com os de paridade estão conectados as entradas D_1 a D_7 .

As tabelas seguintes, 3.1 e 3.2, apresentam a função de cada pino e o funcionamento interno do CI.

PIN NO.	SYMBOL	NAME AND FUNCTION
1	/PL	asynchronous parallel load input (active LOW)
7	/Q ₇	complementary output from the last stage
9	Q ₇	serial output from the last stage
2	CP	clock input (LOW-to-HIGH edge-triggered)
8	GND	ground (0 V)
10	Ds	serial data input
11, 12, 13, 14, 3, 4, 5, 6	D ₀ to D ₇	parallel data inputs
15	/CE	clock enable input (active LOW)
16	Vcc	positive supply voltage

Tabela 3.1 – Descrição dos pinos do CI 74LS165.

OPERATING MODES	INPUTS					Qn REGISTERS		OUTPUTS	
	/PL	/CE	CP	DS	D0-D7	Q0	Q1-Q6	Q7	/Q7
parallel load	L	X	X	X	L	L	L - L	L	H
	L	X	X	X	H	H	H - H	H	L
serial shift	H	L	↑	l	X	L	q0-q5	q6	/q6
	H	L	↑	h	X	H	q0-q5	q6	/q6
Hold "do nothing"	H	H	X	X	X	q0	q0-q5	q7	/q7

Tabela 3.2 – Tabela-verdade do funcionamento do CI 74LS165.

Observando a tabela anterior, retiramos algumas informações que serão importantes para o desenvolvimento do circuito responsável pelo controle de todo sistema.

A tabela de funcionamento nos mostra que a linha $/PL$ estando em nível zero (Low), os estados dos registradores internos $Q0-Q7$ possuem os respectivos valores de suas entradas $D0-D7$, ou seja, os registradores são constantemente atualizados pelos dados de entrada.

O deslocamento dos dados acontece a cada subida do clock, mas para que isso aconteça, a linha $/PL$ deve estar em nível 1 (High) e $/CE$ em nível zero. Assim, concluímos que a entrada $/PL$ pode ser considerada como o complemento da entrada $/CE$.

Como este bloco do circuito está conectado a outros blocos, utilizamos os rótulos *Load* e *Start* para identificar as entradas $/PL$ e $/CE$.

3.1.2 Receptor

O circuito receptor (Figura 3.2) é composto por quatro partes: entrada de dados, verificador de paridade, correção automática e saída dos dados.

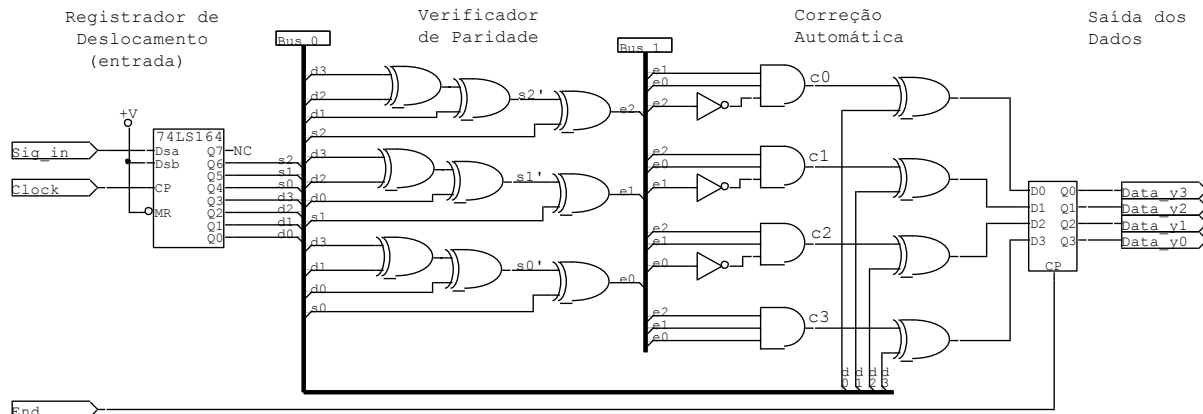


Figura 3.2 – Circuito receptor.

Neste módulo os dados chegam de forma serial pela entrada *Sig_in* e ficam disponíveis nas saídas do registrador serial-paralelo 74LS164. Em seguida, passam pelo circuito verificador de paridade que testa se a paridade recebida pelas linhas $s0$, $s1$ e $s2$ são idênticas às paridades $s0'$, $s1'$ e $s2'$. No caso de ocorrer um erro nos dados, ou seja, um dos bits $d0$ a $d3$ tenha o seu valor invertido, o circuito de paridade gera uma combinação (Tabela 3.3) de código indicando o bit que está invertido.

ENTRADAS			SAÍDA
e0	e1	e2	Bit invertido
0	0	0	nenhum
1	1	0	d0
1	0	1	d1
0	1	1	d2
1	1	1	d3

Tabela 3.3 – Indica quais bits de dados foram invertidos.

A seguir, a expressão booleana das linhas $e0$, $e1$ e $e2$.

- $e0 = s0 \oplus d0 \oplus d1 \oplus d3$
- $e1 = s1 \oplus d0 \oplus d2 \oplus d3$
- $e2 = s2 \oplus d1 \oplus d2 \oplus d3$

Obs: Este circuito não admite a inversão de mais de um bit.

No próximo estágio ocorre a correção dos dados. As linhas $c0$, $c1$, $c2$ e $c3$ quando estiverem em nível lógico 1 indicam que os bits $d0$, $d1$, $d2$ e $d3$ foram invertidos, respectivamente. Se isso ocorrer o bit será invertido novamente. A seguir, estão as expressões booleanas das linhas $c0$, $c1$, $c2$ e $c3$.

- $c0 = e0 \cdot e1 \cdot \overline{e2}$
- $c1 = e0 \cdot \overline{e1} \cdot e2$
- $c2 = \overline{e0} \cdot e1 \cdot e2$
- $c3 = e0 \cdot e1 \cdot e2$

Obs: O símbolo \cdot representa a expressão E (AND).

Ao final, um registrador paralelo-paralelo (Latch) transfere os quatro bits de dados para a saída. Isso ocorre através de um pulso positivo recebido pela linha *End*. A diante, abortaremos o tipo de registrador paralelo-paralelo que será usado.

Análise do funcionamento do 74LS164:

Neste circuito utilizamos o registrador serial-paralelo 74LS164. As duas próximas tabelas (tabelas 3.4 e 3.5) mostram as funções dos pinos e os modos de operação do CI.

PIN	SYMBOL	DESCRIPTION
1, 2	Dsa, Dsb	data input
3, 4, 5, 6, 10, 11, 12, 13	Q0 to Q7	outputs
7	GND	ground (0 V)
8	CP	clock input (LOW-to-HIGH, edge-triggered)
9	/MR	master reset input (active LOW)
14	VCC	DC supply voltage

Tabela 3.4 – Descrição dos pinos do CI 74LS164.

OPERATING MODES	INPUTS				OUTPUTS	
	MR	CP	Dsa	Dsb	Q0	Q1-Q7
Reset (clear)	L	X	X	X	L	L - L
shift	H	↑	l	l	L	q0-q6
	H	↑	l	h	L	q0-q6
	H	↑	h	l	L	q0-q6
	H	↑	h	h	H	q0-q6

Tabela 3.5 – Tabela-verdade de funcionamento do CI 74LS164.

O CI possui duas entradas de dados como pode ser visto na tabela 3.4. E observando o funcionamento (Tabela 3.5), colocamos uma das entradas em nível 1. Assim, o valor da outra entrada será transmitido para a saída $Q0$. Conforme mostrado na figura 3.2 a entrada Dsb foi conectada ao Vcc .

A entrada $/MR$ foi conectada a Vcc , já que não havia a necessidade de reinicializar as saídas do registrador em todo processo.

3.1.3 Máquina de Estados

Responsável pelo funcionamento de todo o processo de transmissão, a Máquina de Estados é uma das partes mais importantes de todo sistema, além de ser a fase mais crítica no desenvolvimento do projeto. Este módulo tem como funções: iniciar, monitorar o tempo e finalizar o processo de transmissão.

Descreveremos agora toda fase de projeto relacionada com a Máquina de Estados. Primeiramente foram definidas algumas características básicas:

- i. O início da transmissão será acionado por um botão de pressão (Push-Button);
- ii. O botão de pressão fica inativo enquanto os dados estiverem sendo transmitidos;

- iii. Enviar sinais ao circuito transmissor, habilitando-o e controlando o tempo de transição dos sete bits da palavra de dados;
- iv. Monitorar o tempo necessário para a transmissão de uma palavra de dados;
- v. Após esgotar o tempo, desabilita o circuito transmissor e envia um pulso ao circuito receptor indicando fim de transmissão;
- vi. Retorna ao seu estado inicial;

Juntamente com as características citadas anteriormente, foi desenvolvido um modelo para a implementação da Máquina de Estados. Este modelo representa na forma de blocos: a Máquina de Estados, o módulo de entrada e as ligações com os módulos transmissor e receptor. A figura 3.3 apresenta o modelo desenvolvido.

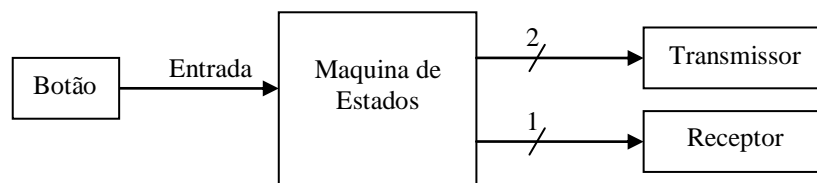


Figura 3.3 – Representação em blocos das ligações da Máquina de Estados.

Definimos a quantidade de linhas de controle (Saída) conforme as figuras 3.1 e 3.2.

Para facilitar o projeto deste módulo vamos separá-lo em módulos menores. Essa técnica torna-se útil devido à complexidade inicial envolvida no projeto da Máquina de Estado (Figura 3.4).

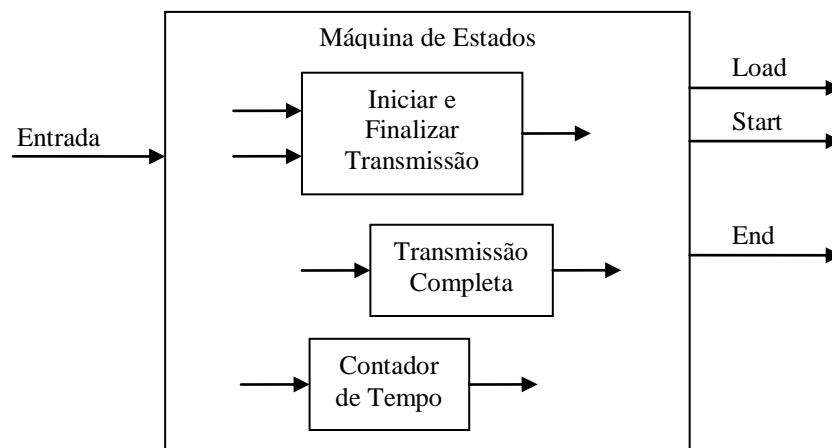


Figura 3.4 – Blocos da Máquina de Estados.

Após separar a Máquina de Estados em blocos menores, foram definidas as linhas de entrada e saída desses blocos conforme as seguintes características:

- Iniciar e Finalizar a Transmissão: uma entrada referente ao início da transmissão, uma entrada interna referente ao fim de contagem e uma saída para controlar os blocos externos.

- Transmissão Completa: uma entrada indicando fim de contagem e uma saída para controlar a linha *End*.
- Contador de Tempo: a entrada inicia o contador e a saída indica fim de contagem.

Será demonstrado o projeto destes circuitos e em seguida definiremos suas ligações internas e externas.

➤ Bloco Iniciar e Finalizar a Transmissão

Desenvolvemos um diagrama de estados (figura 3.5(b)) para representar seu funcionamento temporal. Para uma análise mais genérica utilizamos A, B e Z como rótulos de entrada e saída (figura 3.5(a)).

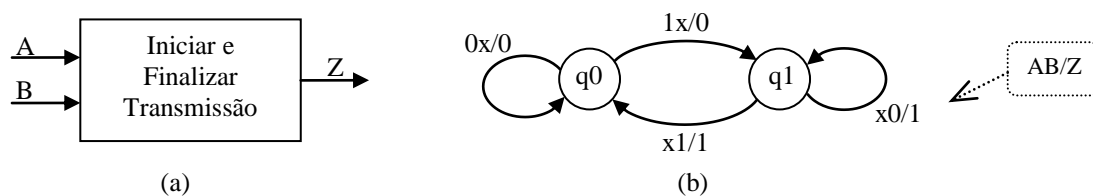


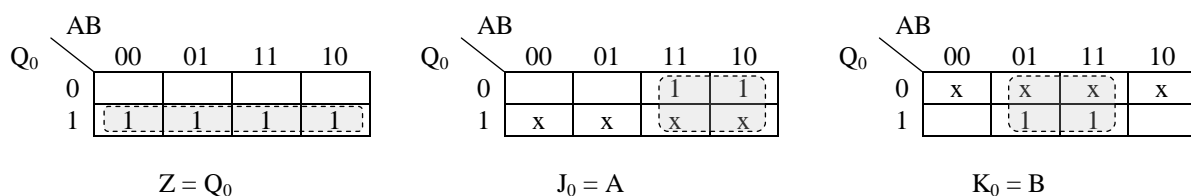
Figura 3.5 – (a) Bloco Iniciar e Finalizar Transmissão; (b) diagrama de estados.

De acordo com o diagrama de estados, foi implementada a tabela de transição de estados (figura 3.6) utilizando flip-flops (FF) tipo J-K.

Entradas		Estado		Saída	Prox. Estado		FF 0	
A	B	Q ₀		Z	Q ₀ ⁺		J ₀	K ₀
0	0	q0	0	0	q0	0	0	x
0	1	q0	0	0	q0	0	0	x
1	0	q0	0	0	q1	1	1	x
1	1	q0	0	0	q1	1	1	x
0	0	q1	1	1	q1	1	x	0
0	1	q1	1	1	q0	0	x	1
1	0	q1	1	1	q1	1	x	0
1	1	q1	1	1	q0	0	x	1

Tabela 3.6 – tabela de transição de estados.

Obtemos a função lógica da saída S e das entradas J_0 e K_0 utilizando mapa de Karnaugh. A implementação do circuito resultante está na figura 3.6.



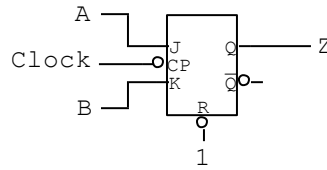


Figura 3.6 – Implementação do bloco Iniciar e Finalizar Transmissão.

➤ Bloco Transmissão Completa

Aplicamos o mesmo processo descrito anteriormente, para a implementação deste bloco. A seguir estão o diagrama de estados, tabela de transição, funções lógicas obtidas e o circuito implementado.

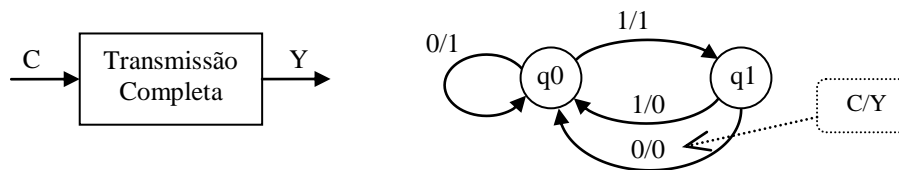


Figura 3.7 – Diagrama de estados.

Entrada	Estado		Saída	Prox. Estado		FF 0
C	Q_1		Y	Q_1^+		$J_1 \ K_1$
0	q0	0	1	q0	0	0 x
1	q0	0	1	q1	1	1 x
0	q1	1	0	q0	0	x 1
1	q1	1	0	q0	0	x 1

Tabela 3.7 – tabela de transição de estados.

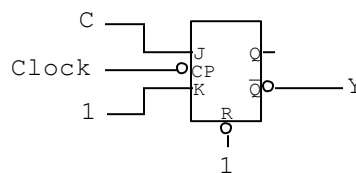
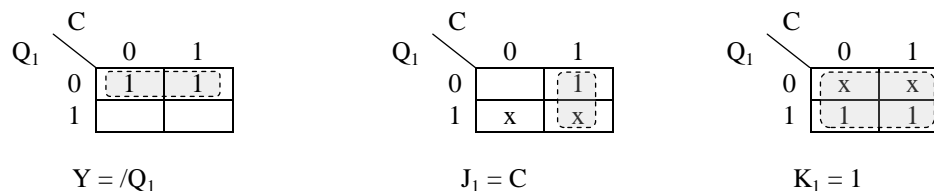


Figura 3.8 – Implementação do bloco Transição Completa

➤ Bloco Contador de Tempo

Observando as especificações dos CI's (apostila da disciplina), foi decidido utilizar o contador de década 74LS90. O próximo passo consistiu em adaptar este CI ao circuito desejado e para isto utilizamos a tabela-verdade (tabela 3.8(a)) de sua configuração Bi-Quinary (conectar a saída Q_3 com a entrada CP_0 para esta configuração) e tabela de reset do mesmo.

COUNT	OUTPUT			
	Q ₀	Q ₃	Q ₂	Q ₁
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

(a)

RESET INPUTS				OUTPUTS			
MR ₁	MR ₂	MS ₁	MS ₂	Q ₃	Q ₂	Q ₁	Q ₀
H	H	L	x	L	L	L	L
H	H	x	L	L	L	L	L
x	x	H	H	H	L	L	H
x	L	x	L	COUNT			
L	x	L	x	COUNT			
L	x	x	L	COUNT			
x	L	L	x	COUNT			

(b)

Tabela 3.8 – (a) Tabela de configuração Bi-Quinary; (b) tabela de reset.

A figura 3.9 mostra o circuito completo. Tomando por base as tabelas anteriores, colocamos as duas entradas MS_0 e MS_1 em nível lógico zero e a entrada MR_1 em nível 1. Assim, temos apenas uma entrada como foi especificado no modelo desenvolvido. E de acordo com a tabela de reset, o contador iniciará sua contagem se sua entrada estiver em nível zero.

Já para definir quando a saída ficará ativa, devemos observar o comportamento dos blocos Iniciar e Finalizar a Transmissão e Transmissão Completa. Primeiramente, vamos relembrar o fato de que todo circuito está em sincronia com o sinal de clock, logo, se o contador ativar sua saída em um tempo t , esse sinal estaria nas saídas dos outros dois blocos no tempo $t + 1$. Deste modo, como temos sete bits para serem transmitidos, o contador deverá acionar sua saída no tempo sete menos um, ou seja, deverá contar até seis. Observando a tabela 3.8(a), concluímos que utilizando uma porta lógica E (AND) nas saídas Q_0 e Q_1 , teremos a saída ativa em após seis pulsos de clock.

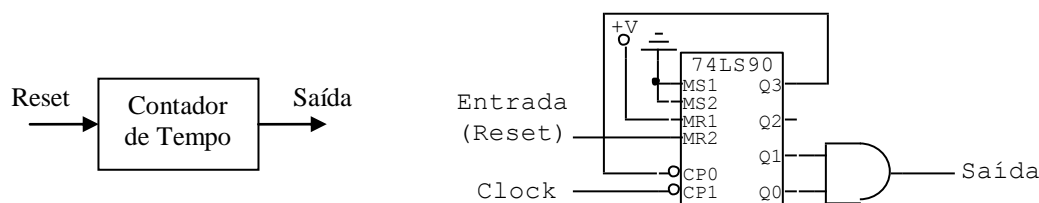


Figura 3.9 – Implementação do bloco Contador de Tempo.

➤ Unindo os blocos

Analisando o comportamento dos três blocos anteriores e as especificações iniciais, foram realizadas as seguintes ligações entre os mesmos (figura 3.10). A tecla de pressão foi conectada a entrada J_0 , a saída do contador nas entradas K_0 e J_1 (Reset) e a entrada do contador na saída $/Q_0$. Também analisando o circuito transmissor (seção 3.1.1) e o receptor (seção 3.1.2) e suas respectivas especificações de controle, foram definidas as saídas Q_0 e $/Q_0$ com os rótulos *Load* e *Start* e a saída $/Q_1$ com o rótulo *End*.

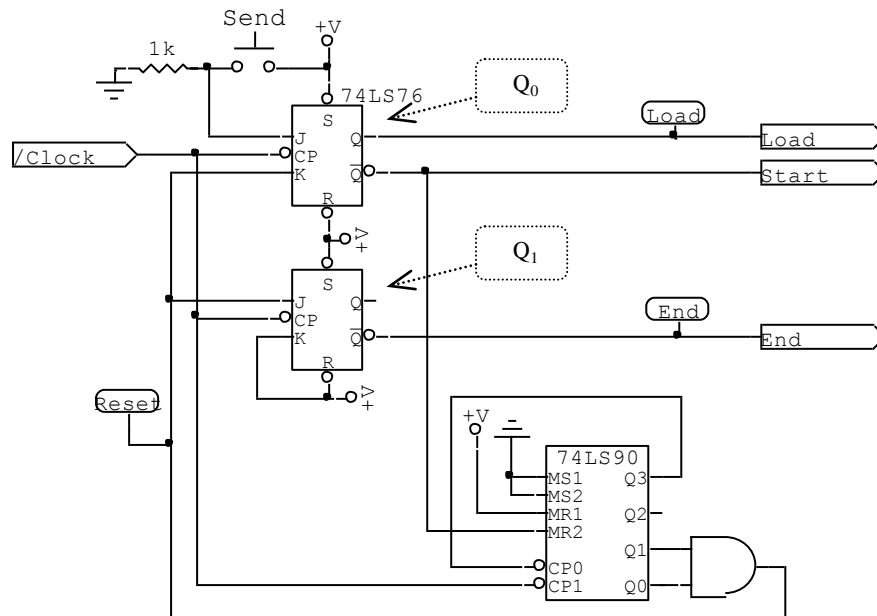


Figura 3.10 – Máquina de Estado

O gráfico dos sinais gerados pela Máquina de Estados pode ser observado na figura 3.11.

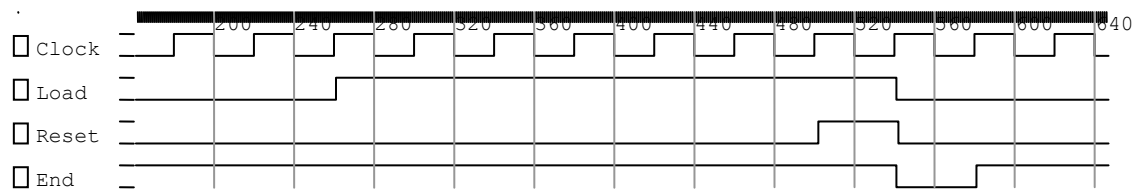


Figura 3.11 – Gráfico dos sinais da Máquina de Estado.

3.1.4 Relógio (Clock)

Em projetos de circuitos digitais síncronos existe a necessidade de utilizar um circuito multivibrador astável. Neste projeto, conforme foi especificado, utilizou-se o CI temporizador 555 de 8 pinos juntamente com outros componentes passivos para gerar uma forma de onda retangular. A figura 3.12 mostra o circuito projetado para a parte experimental. Para obter uma melhor transição entre os níveis lógicos TTL no sinal de clock, foi utilizado uma porta lógica 74LS32 (OR) e uma porta 74LS04 (NOT) para gerar o complemento do clock.

Obs: Todos os circuitos deste projeto são acionados em relação à borda de subida do clock.

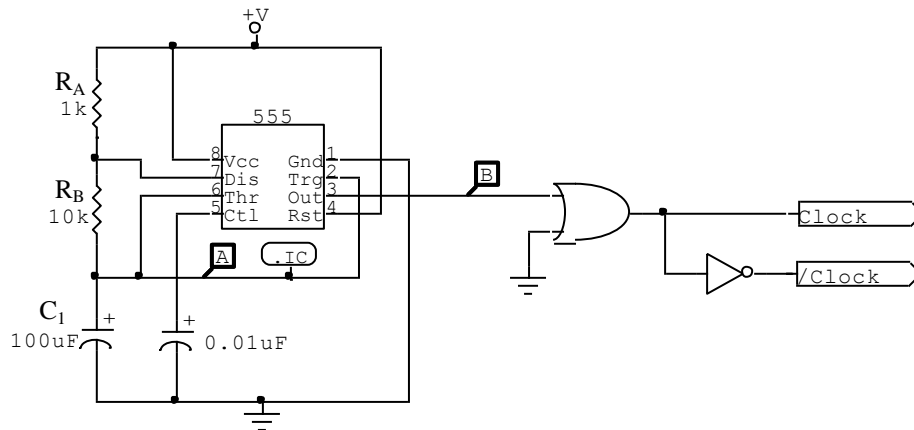


Figura 3.12 – Circuito de Clock

Para melhor visualização dos resultados, o circuito de clock foi projetado para gerar uma forma de onda com uma frequência mínima de 1 Hz (a frequência real obtida foi de 0,68 Hz). O gráfico da figura 3.13 mostra as formas de onda de carga do capacitor C_1 (A) e pino *out* (B) do 555.

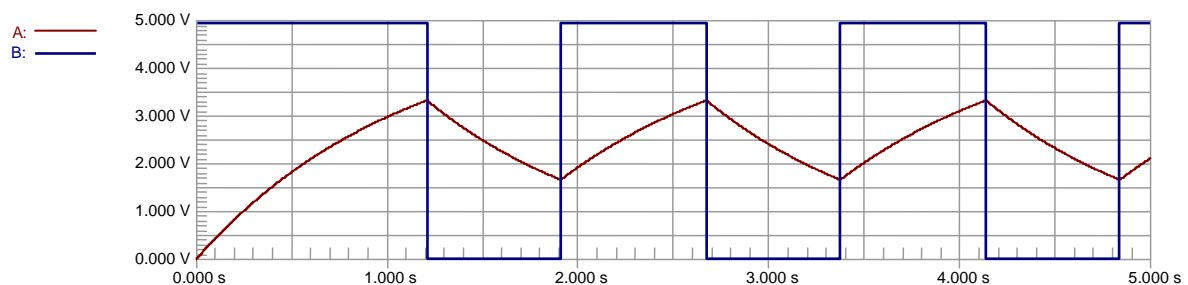


Figura 3.13 – Forma de onda do circuito de Clock

3.1.5 Gerador de Erros

Um circuito para gerar erros na linha de transmissão foi utilizado. A figura 3.14 demonstra uma configuração básica para o Gerador de Erros. Consiste em usar uma porta lógica OU-EXCLUSIVO (XOR) com uma de suas entradas ligada em um botão de pressão. A outra entrada é conectada na saída do circuito transmissor (linha *Sig_out*) e a saída na entrada serial do circuito receptor (linha *Sig_in*).

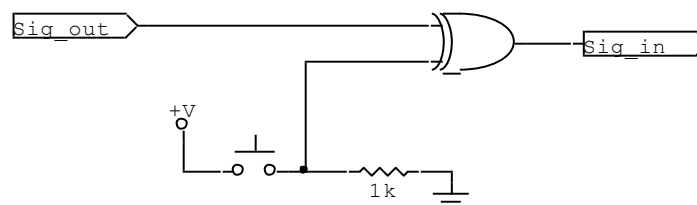


Figura 3.14 – Circuito para inserir erro

Podemos observar que ao pressionar o botão, o bit atual que está sendo transmitido terá seu valor invertido.

3.2 Montagem

3.2.1 Chaves

Na montagem realizada utilizamos um conector com cinco chaves para inserir os bits na entrada do circuito transmissor, além de quatro resistores (figura 3.15).

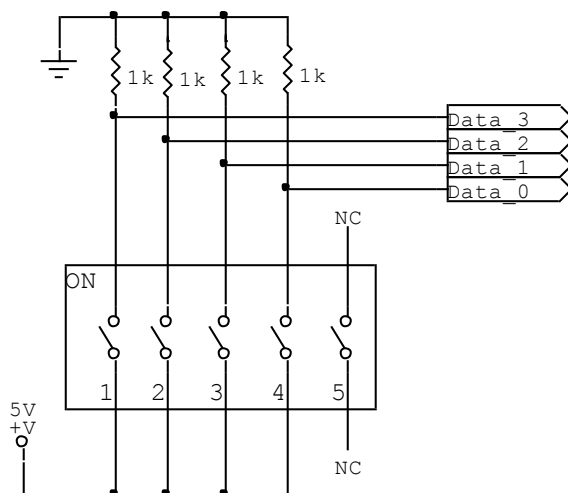


Figura 3.15 – Chaves que representam os bits de entrada

3.2.2 Circuito Auxiliar de Geração de Erros

Utilizamos um circuito para auxiliar a inserção de erros nos quatro bits relativos aos dados. Consiste em garantir que somente os quatro bits de dados poderão ser invertidos, além de emitir um sinal visual no momento em que a alteração pode ser realizada. Utilizamos a tabela de configuração do Contador de Tempo, para gerar a função lógica deste circuito (figura 3.16).

		Q ₀ Q ₁			
		0 0	0 1	1 1	1 0
Q ₂ Q ₃	0 0			1	1
	0 1	1	x	x	x
	1 1	x	x	x	x
	1 0		1	x	x

$$\text{Saída} = Q_0 + Q_3 + Q_1 Q_2$$

COUNT	OUTPUT				Saída
	Q ₀	Q ₃	Q ₂	Q ₁	
0	L	L	L	L	0
1	L	L	L	H	0
2	L	L	H	L	0
3	L	L	H	H	1
4	L	H	L	L	1
5	H	L	L	L	1
6	H	L	L	H	1
7	H	L	H	L	x
8	H	L	H	H	x
9	H	H	L	L	x

Figura 3.16 – Função lógica de circuito auxiliar de geração de erros e tabela de configuração do Contador de Tempo.

A figura 3.17 apresenta o circuito auxiliar de Geração de Erros. Adicionamos uma porta E (END) em série com o botão de pressão para completar o circuito. Assim, os bits só poderão ser alterados quando a linha *Error* estiver em nível lógico 1.

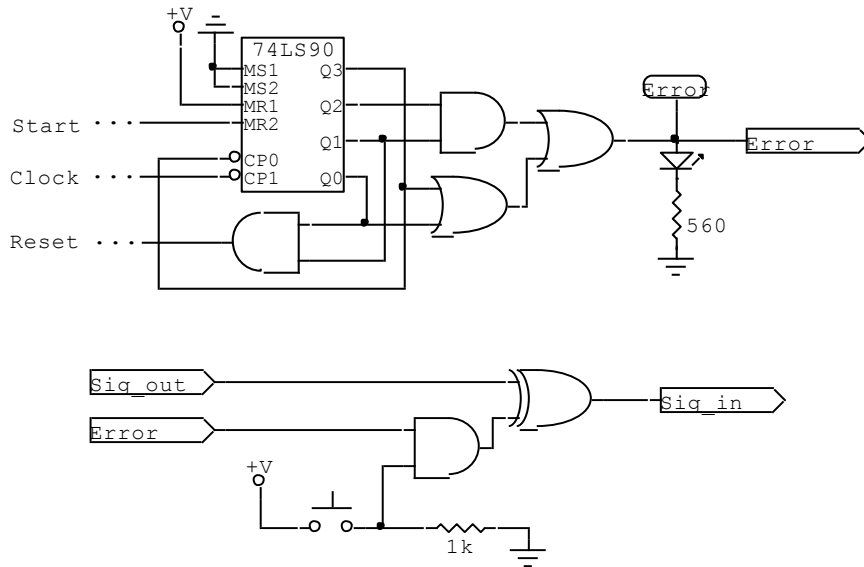


Figura 3.17 – Circuito

3.2.3 Registrador para saída de dados

Utilizamos o registrador paralelo-paralelo 74LS374 na montagem realizada, já que o mesmo possui oito entradas e oito saídas. O motivo foi a necessidade de visualizar também os bits que foram invertidos e deste modo colocamos LEDs na saída deste registrador (figura 3.18).

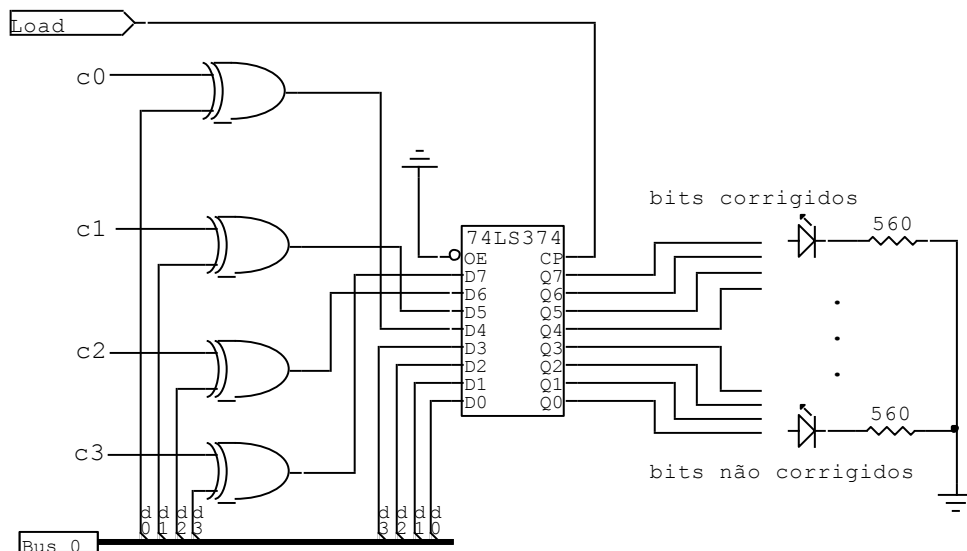


Figura 3.18 – Circuito

Como o registrador anterior foi utilizado somente para testes. Outro registrador poderia ser utilizado para a implementação final do circuito depois de realizados os testes. O CI 74LS173, como mostrado na figura 3.19 poderia ser utilizado.

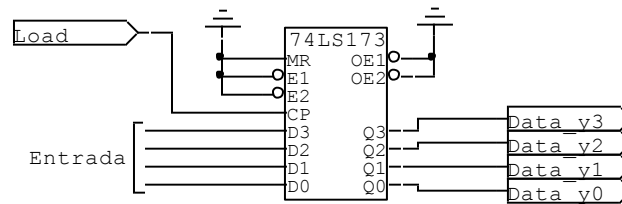


Figura 3.19 – Circuito

4 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] <http://www.ti.com/>
- [2] <http://www.nxp.com/>
- [3] Apostila da Disciplina

5 CONCLUSÃO

Com este trabalho foram revisados e implementados os conhecimentos adquiridos na disciplina de Sistemas Digitais. Todo o projeto foi desenvolvido com sucesso, obtivemos os resultados esperados com o projeto, a simulação e teste que foram realizados.