

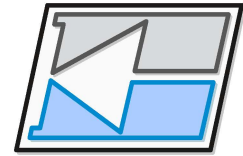


INSTITUTO FEDERAL  
SANTA CATARINA

INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DE SANTA CATARINA  
CAMPUS FLORIANÓPOLIS

UNIDADE CURRICULAR: Eletrônica Digital 2 - EDB  
PROFESSOR(A): Matheus Leitzke Pinto

ALUNO: Leonardo Henrique Dill Bruxel



# Painel de Controle para Elevadores

## Implementado em Máquinas de Estado

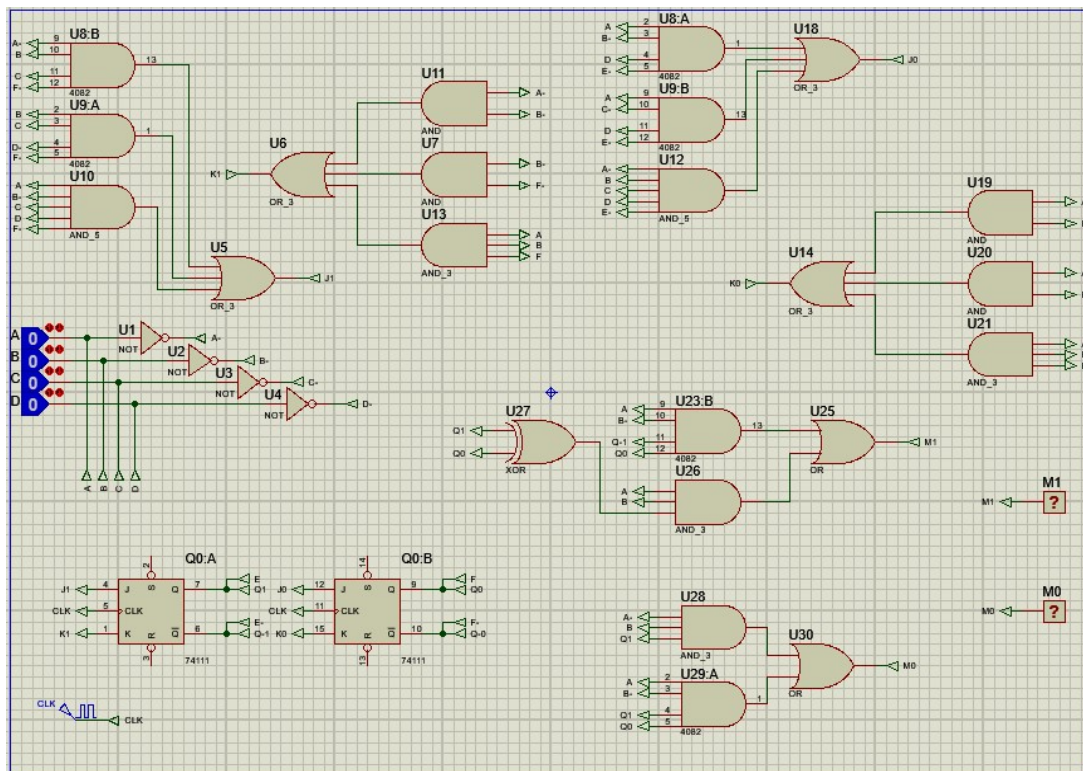


Figura 1: Diagrama esquemático do circuito, feito usando o software Proteus®.

# 1. Introdução.

Uma Máquina de Estados é um conceito abstrato usado para facilitar a compreensão e desenvolvimento de circuitos lógicos digitais. Inúmeras simplificações e vantagens são acrescentadas ao circuito pela implementação do método de construção de uma máquina de estados. Entre as vantagens, há por exemplo a redução do número de componentes utilizados e, portanto, a diminuição do preço final do produto. A complexidade do projeto também é diminuída e torna-se mais acessível para a compreensão por usuários do circuito.

Existem, na verdade, dois principais gêneros de máquinas de estado: as máquinas de Mealy<sup>1</sup> e de Moore<sup>2</sup>. As máquinas de Mealy são diferentes das de Moore por algumas razões, sendo a principal delas a interferência da entrada na saída, ou seja, a lógica da saída envolve a entrada. A máquina de Moore, no entanto, segue uma lógica mais linear.

Este projeto usa de um sistema de Mealy para a realização das tarefas impostas à máquina de estados. O método de Mealy foi escolhido por conta da definição do circuito, que depende de sensores para o funcionamento correto dos motores. Esses sensores são entradas que interferem na saída e, portanto, definem o circuito como uma máquina de estados de Mealy.

Uma maneira comum de representar máquinas de estado é por meio de um Diagrama de Estados, que também pode ser chamado de *Flowchart*. O diagrama de estados deste projeto pode ser encontrado online<sup>3</sup>.

A escolha do tópico (Painéis de Elevadores) foi devida a dificuldade de encontrar um tema diversificado, específico, incomum e complexo o suficiente para a realização do projeto. No entanto, a escolha causou problemas visto à unfamiliaridade com o circuito e o funcionamento de um elevador. Diversos erros aconteceram e mais de uma versão do circuito foi descartada por completo. Diversos diagramas de estados foram feitos para atender as especificações do projeto e diversos componentes foram analisados para a escolha da plataforma FPGA perfeita para a implementação do circuito.

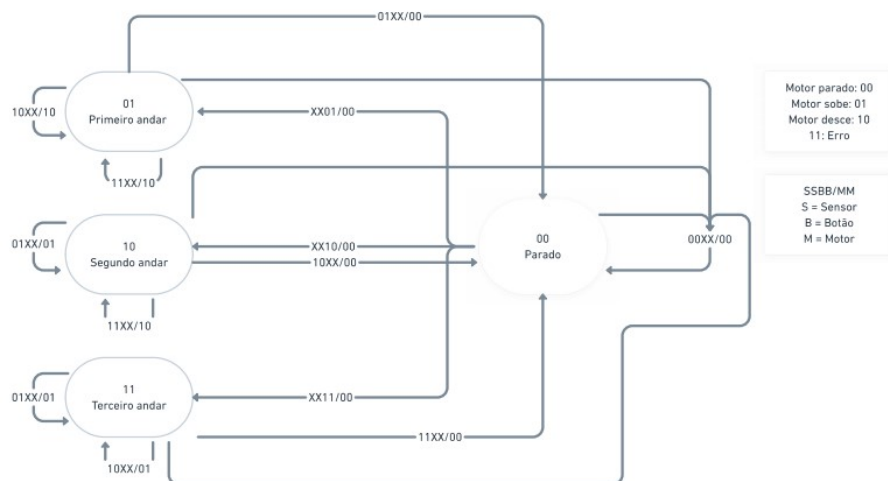


Figura 2: Diagrama de Estados do projeto, feito usando a plataforma Whimsical®.

1. G. H. Mealy: Matemático, nascido em 1927, em Massachusetts, EUA
2. E. F. Moore: Matemático e cientista da computação, nascido em 1925, em Maryland, EUA.
3. Acesso em: <https://whimsical.com/C1KT8q8zPravGLpSe9PkXL@2Ux7TurymPKTNTRtP9Mz>

## 2.1. Escolha da plataforma FPGA.

Uma parte vital do processo de desenvolvimento deste projeto é a escolha dos componentes para a implementação do mesmo. Diversas variáveis devem ser consideradas para evitar problemas, e por isso, o mais minucioso cuidado foi tomado para a escolha da plataforma perfeita. A plataforma escolhida deve ser, conforme as especificações do projeto, um componente FPGA ou outra forma de dispositivo lógico programável (PLD), deve conter uma interface analógica (entrada ou saída) e algum tipo de conversão entre sinal analógico e digital.

Diversos FPGAs foram considerados para a realização do trabalho, porém três se destacaram e se tornaram as principais opções. A placa Papilio® One 250K, a placa Altera® Cyclone® II e a mais recente Altera® Cyclone® IV. As informações sobre as placas podem ser encontradas na tabela abaixo.

Informação	Papilio One 250K	Altera Cyclone II	Altera Cyclone IV
Clock	32 MHz	50 MHz	50 MHz
Tensão de operação	6,5 - 15 V	5 V	5 V
Memória	4 Mb	4 Mb	6.3 Mb
Preço	R\$ 419,90	R\$ 139,90	R\$ 190,24
Vantagens	Permite módulos semelhante aos Shields de Arduino.	Plataforma de entrada para iniciantes.	Funcionalidades Wireless e Ethernet.

Tabela 1: Informações a respeito das plataformas para a escolha do FPGA.

Um clock de cerca de 100 Hz é suficiente para que o circuito funcione, logo, as três placas atendem essa especificação. A tensão de operação deve ser a menor possível, para a melhor eficiência energética do circuito, que não deve consumir muita potência. A memória utilizada depende da programação, porém não deve exceder os 4Mb das duas primeiras plataformas e muito menos os 6.3 Mb da terceira. A plataforma escolhida, portanto, foi a placa Altera Cyclone II, visto o valor do componente. As diversas funcionalidades permitidas pelas outras placas não são essenciais e tampouco seriam utilizadas no projeto.

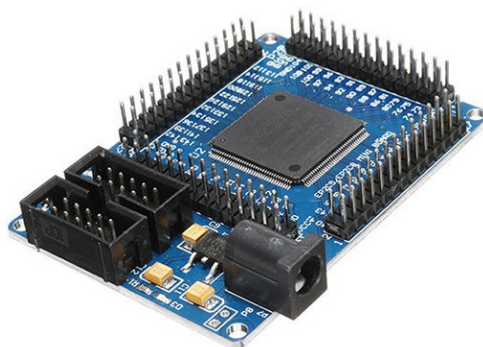


Figura 3: Placa FPGA Altera® Cyclone® II.

## 2.2. Implementação de uma interface analógica.

Como descrito nas especificações do projeto, o circuito deve conter algum tipo de interface analógica, sendo entrada ou saída. Como um painel de elevador não é o exemplo de um circuito em que se haja uma interface analógica, foi criada uma solução. No vão do elevador + deve ser implementada uma fita resistiva fixa, que por sua vez fará contato com uma peça móvel no corpo do elevador. Esse circuito deve funcionar quase como um potenciômetro, e o pino móvel deve ser a entrada analógica do conversor ADC que por sua vez enviará o sinal S, para representar o sensor. Apenas 2 bits são necessários, porém, nenhum dos conversores disponíveis comercialmente encontrados possuem uma resolução tão baixa. O problema pôde ser solucionado apenas com a elaboração de um componente ADC feito especialmente para o circuito, usando a formulação padrão de conversores ADC demonstrada nos slides e nas aulas teóricas da disciplina. O circuito é demonstrado abaixo:

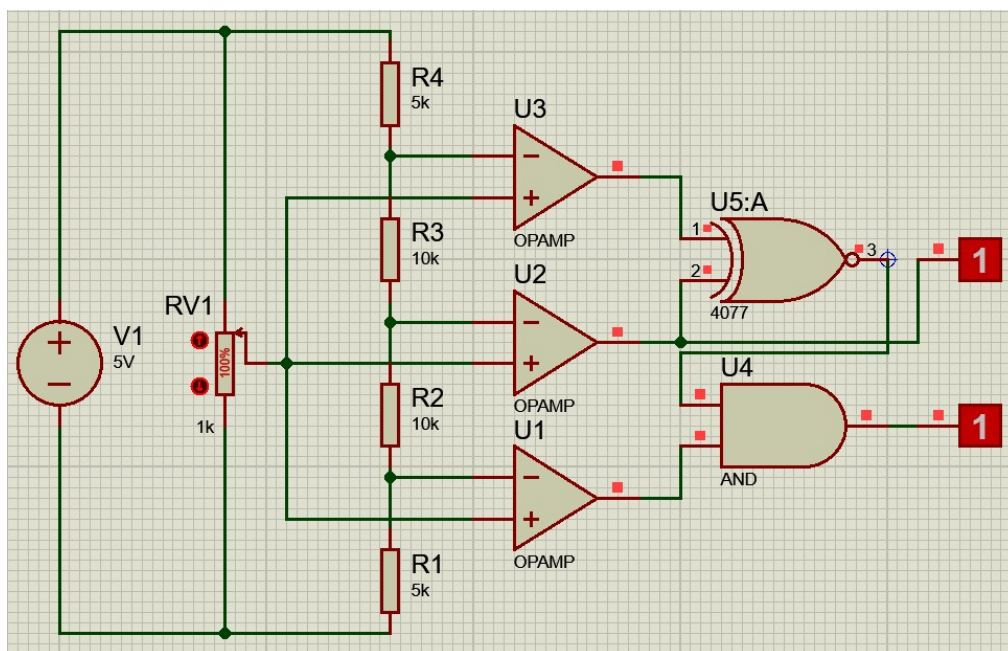


Figura 4: Conversor ADC projetado especialmente para os sensores do projeto.

RV1, no diagrama, representa o sensor, a fita resistiva que funciona como um potenciômetro. U1, U2 e U3 são Amplificadores Operacionais<sup>1</sup> (AmpOps) em modo comparador, onde as tensões na entrada são comparadas e, se a tensão no pino positivo for maior que a do pino negativo, a saída será 5 V, que pode ser interpretada como um bit 1. U5 e U4 são componentes para realizar a lógica de saída, convertendo os 3 bits saídos dos AmpOps em 2 bits no padrão normalmente usado para dados binários.

1. Foi usado um componente de simulação genérico do software Proteus®. No entanto, para uma implementação física do projeto, o ideal seria a busca por um componente real.



### 3. Desenvolvimento do circuito.

A máquina de estados foi elaborada usando o diagrama de estados. As tabelas de excitação e da verdade foram montadas em um arquivo de planilhas. O arquivo está disponível na pasta do projeto sob o nome de “Painel de elevador - Planilhas 2.0”.

A	B	C	D	E	F						
S1	S0	B1	B0	Q1 ATUAL	Q0 ATUAL	Q1 PROX	Q0 PROX	J1	K1	J0	K0
0	0	0	0	0	0	0	0	0	X	0	X
0	0	0	0	0	1	0	0	0	X	X	1
0	0	0	0	1	0	0	0	X	1	0	X
0	0	0	0	1	1	0	0	X	1	X	1
0	0	0	1	0	0	0	0	0	X	0	X
0	0	0	1	0	1	0	0	0	X	X	1
0	0	0	1	1	0	0	0	X	1	0	X
0	0	0	1	1	1	0	0	X	1	X	1
0	0	1	0	0	0	0	0	0	X	0	X
0	0	1	0	0	1	0	0	0	X	X	1
0	0	1	0	1	0	0	0	X	1	0	X
0	0	1	0	1	1	0	0	X	1	X	1
0	0	1	1	0	0	0	0	0	X	0	X
0	0	1	1	0	1	0	0	0	X	X	1
0	0	1	1	1	0	0	0	X	1	0	X
0	0	1	1	1	1	0	0	X	1	X	1
0	1	0	0	0	0	0	0	0	X	0	X

$$J1 = \underline{A}BC\underline{F} + BC\underline{D}\underline{F} + \underline{A}BCDF$$

$$K1 = \underline{A}\underline{B} + \underline{B}\underline{F} + ABF$$

$$J0 = \underline{A}BD\underline{E} + A\underline{C}DE + \underline{A}BCDE$$

$$K0 = \underline{A}\underline{B} + \underline{A}\underline{E} + ABE$$

Figura 5: Tabela de excitação (incompleta, pois é muito longa para ser inserida neste arquivo)

A	B	E	F		
S1	S0	Q1	Q0	M1	M0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	0	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	0

M1 = S1.S0.Q1.Q0 + S1.S0.(Q1⊕Q0)

M0 = S1.S0.Q1 + S1.S0.Q1.Q0

Figura 6: Tabela da lógica de saída do circuito, que representa a interpretação do sinal para o funcionamento dos motores.

Usando as tabelas, no entanto, erros foram encontrados e o funcionamento do circuito estava comprometido. Incongruências e inconformidades foram encontradas, no entanto, e corrigidas, e o circuito então passou a funcionar. Na primeira versão da tabela, a lógica de acionamento dos estados era diferente da lógica apresentada no diagrama de estados, e portanto, erros decorrentes disso apareceram. Na segunda versão da tabela, com os erros corrigidos, o funcionamento ocorreu.

Na simulação, há 4 entradas controladas diretamente pelo usuário. Dois bits representam o sinal do sensor, e outros dois bits representam o sinal dos botões. Usando essas entradas e o estado atual em que se encontram os flip-flops, uma operação lógica é feita para definir as entradas J e K dos dois flip-flops que armazenam o estado do circuito. A saída desses flip-flops, então, servem de entradas para a lógica de saída, que por sua vez controla os motores.

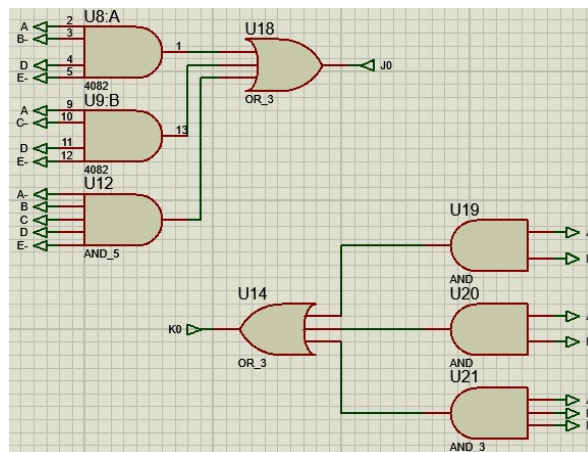


Figura 7: Lógica das entradas do Flip-Flop 1.

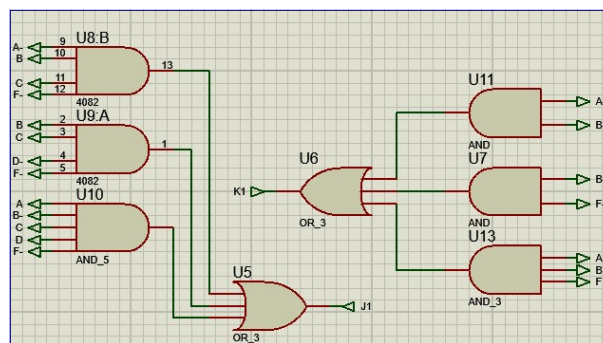


Figura 8: Lógica das entradas do Flip-Flop 0.

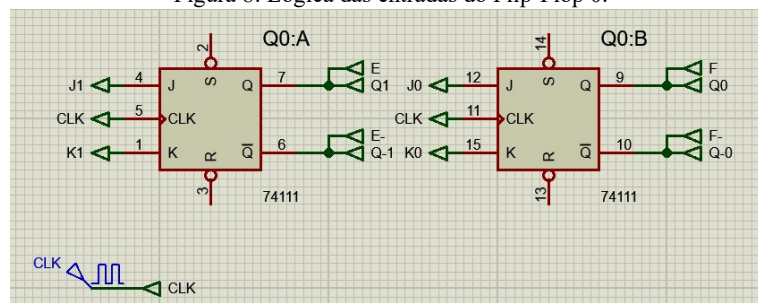


Figura 9: Flip-Flops 1 e 0.

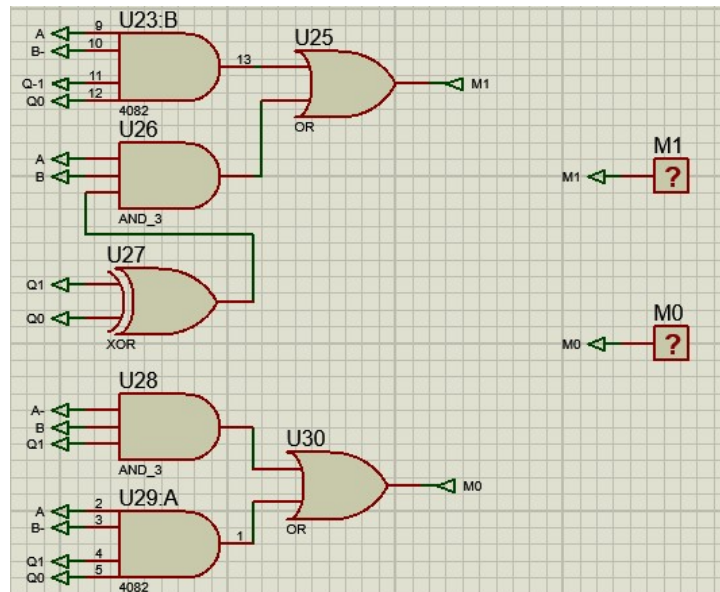


Figura 10: Diagrama da lógica de saída e controle dos motores.

Com essa implementação, a tabela da verdade foi respeitada e o circuito funcionou perfeitamente. Os arquivos de simulação podem ser encontrados no arquivo do projeto, sob a pasta “Simulações”.

## 4. Conclusão.

Com este projeto, muito foi aprendido sobre máquinas de estados, um conceito extremamente importante para a idealização de circuitos digitais complexos, e também sobre os outros conteúdos lecionados na disciplina, como conversores (ADC e DAC), flip-flops, registradores de deslocamento (que ficaram de fora do projeto final porém que foram cogitados na primeira representação do circuito), contadores (que passaram pelo mesmo processo) e latches. O conteúdo se mostrou dominado e portanto, concluído.

## 5. Agradecimentos.

Agradeço primeiramente ao professor Matheus Pinto, pois sem sua didática, eu não seria capaz de realizar este projeto.

Agradeço também a meu colega Geovane, que ofereceu a ajuda para a correção de erros, bugs e glitches do projeto. Sem Geovane, esse projeto não seria realizado com sucesso.

Agradeço também a meus colegas Valter, Pedro Henrique e Brayan, que deram-me o suporte emocional para lidar com os pesados conteúdos e projetos desse semestre.

Também agradeço a minha namorada Laura Hobold, novamente pelo suporte emocional para lidar com o duro semestre vivido à distância.

Agradeço às bandas Novos Baianos, Demônios da Garoa, Seroma, Rubel, Masha y el bloque depresivo, Joy Division e Legião Urbana, que serviram de inspiração para escrever o relatório final e realizar a solução das inúmeras das quais o circuito depende.

Agradeço aos músicos Kanye West, Tim Maia, Jorge Ben, Jorge Mautner, Seu Jorge, Clarice Falcão e Roberto Carlos, pelos mesmos motivos das bandas citadas anteriormente.

Agradeço novamente ao professor Matheus, que deu assistência para a resolução de um erro na reta final do projeto, que comprometia o funcionamento do circuito.