



UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

PROJETO ELEVADOR

FELIPE HENRIQUE VERONES PEREIRA DOS SANTOS

Araranguá – SC

2022

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

PROJETO ELEVADOR

**Relatório apresentado à disciplina de
Linguagem de Descrição de Hardware
(DEC7555) como contextualização do projeto
apresentado e trabalho final da disciplina.**

Araranguá – SC

2022

Sumário

Introdução	4
Desenvolvimento.....	5
Descrição geral do sistema.....	5
Entradas.....	8
Saídas.....	8
Funcionamento.....	8
Conclusão	12

Introdução

Para a conclusão da disciplina de Linguagem e Descrição de Hardware, foi proposto à turma variados temas de projetos envolvendo os conhecimentos adquiridos em aula, o tema abordado por este documento refere-se ao desenvolvimento de um sistema de controle de dois elevadores em um prédio de três andares, o qual irá englobar todos os aspectos de seu funcionamento, incluindo o funcionamento dos botões internos de cada elevador, os botões referente a cada um dos três andares e o térreo, bem como os procedimentos envolvendo a otimização dos mesmos. Neste documento serão apresentadas as regras de funcionamento do sistema de elevadores, passando por cada componente e abstraindo o máximo possível para que seja de fácil entendimento.

Desenvolvimento

Descrição geral do sistema

A proposta deste trabalho consistia em desenvolver um sistema embarcado para realizar o controle de um sistema de elevadores que atende a um prédio de 4 pavimentos.

São dois elevadores que podem atender todos os andares. A figura a seguir ilustra o prédio com seus dois elevadores. Em cada andar existem dois botões (vermelho e azul na figura). O botão azul é usado quando um usuário quiser acionar o elevador para subir para um andar superior, e o vermelho é usado quando o usuário quer descer para um andar inferior. Estes botões azuis e vermelhos serão representados pelas chaves deslizantes (SWITCH0 até SWITCH7) existentes no kit DE10-Lite.

Dois displays de sete segmentos devem ser utilizados para indicar a posição de cada elevador no prédio. Ou seja, em que “andar” está cada elevador. A passagem de um andar para outro deve ser de 4 segundos. Por exemplo, para ir do Térreo para o terceiro andar (quarto pavimento) o display deve indicar o número do pavimento a cada quatro segundos, totalizando doze segundos. O botão BOTÃO 0 deverá ser usado como *reset* geral do sistema. Uma questão importante na administração de uma edificação é o consumo de energia. O objetivo principal é otimizar o uso dos elevadores, mantendo apenas um em operação (tanto quanto possível). Por exemplo, se um elevador está descendo, com um usuário a partir do terceiro andar, e outro usuário pressiona o botão para descer no primeiro pavimento; então, o elevador que já está em movimento deverá parar no primeiro andar para servir este segundo usuário.

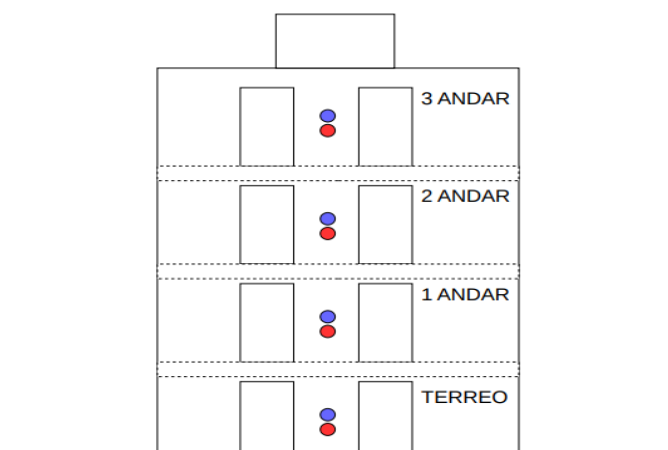


Figura 1: representação do edifício com quatro pavimentos e dois elevadores.

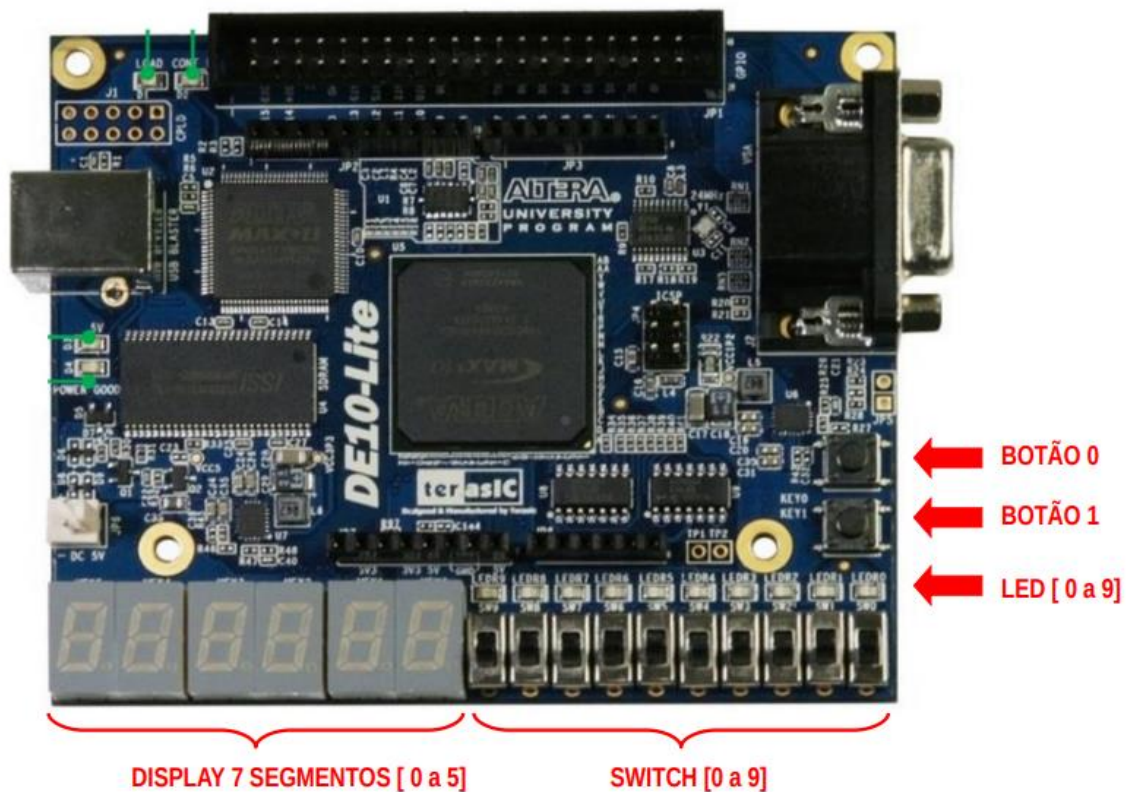
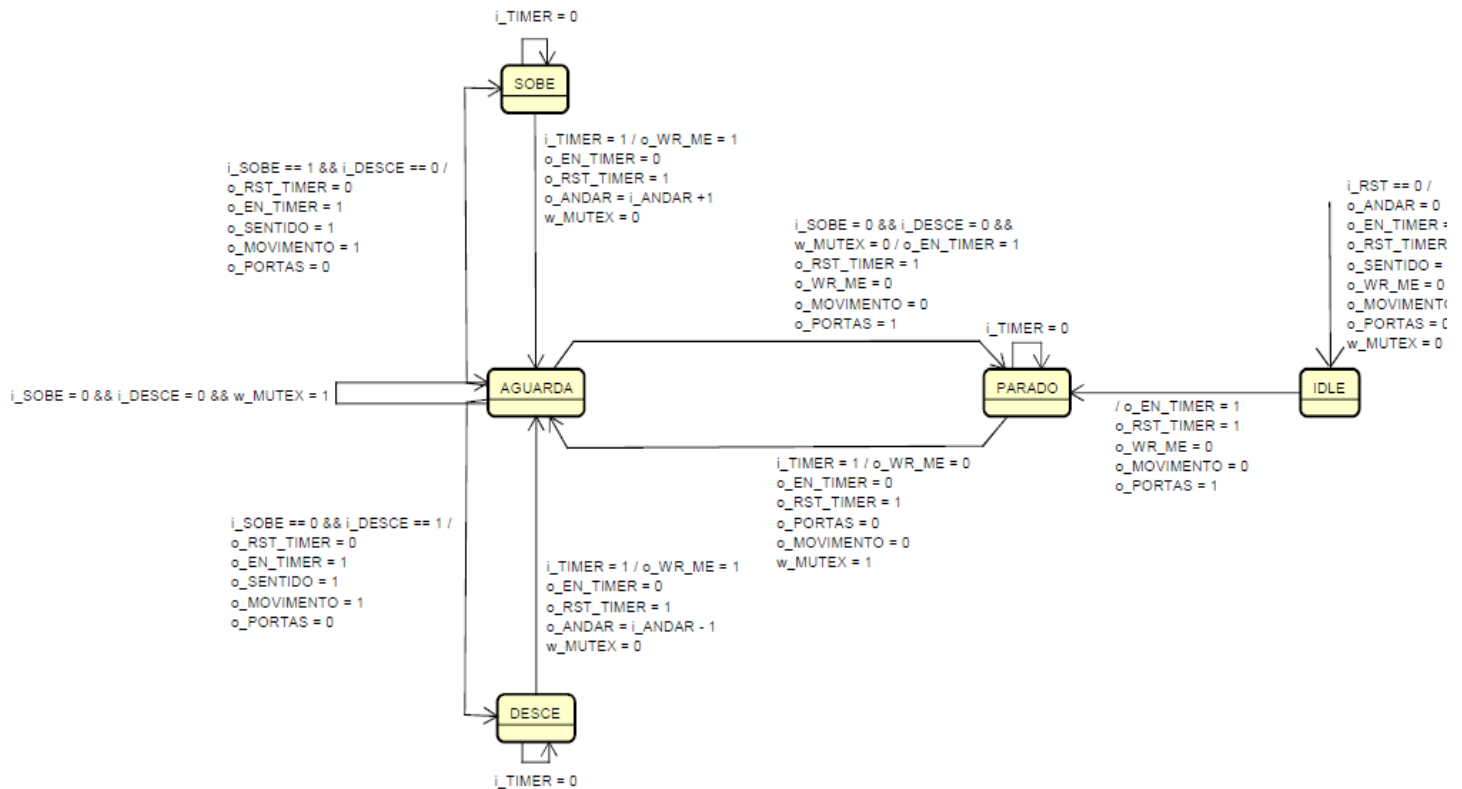


Figura 2: Kit de desenvolvimento DE10-Lite usado no desenvolvimento do trabalho.

O desenvolvimento do projeto de elevador começa em sua lógica, em quais serão suas entradas, como será seu funcionamento, qual será a otimização do sistema, dentre outras coisas. Além analisar quais serão os possíveis estados do elevador e como eles respondem ao sistema, de acordo com a entrada apresentada.

Pensando nestes aspectos, foi criada uma máquina de estados, que representaria os 5 estados do elevador, descritos e representados a seguir:

- Idle;
- Parado;
- Aguarda;
- Sobe;
- Desce.



Cada um destes estados possui suas regras de funcionamento e sua lógica particular, como visto a seguir:

O estado padrão do elevador é *Idle*, onde ele resetará o timer de 4s e o colocará em estado de espera, além de setar o andar de destino padrão para o Térreo e parar seu movimento, enquanto espera uma solicitação.

Parado, as portas se abrem e o timer é ativado. Quando o timer é zerado novamente, resetamos o mesmo ao valor de 4s, fechamos as portas e entramos no estado de Aguarda. Quando vier uma solicitação para subir ou descer, ativamos o timer de 4s, entramos em movimento e assumimos o estado Sobe ou Desce, respectivamente. Quando o elevador está em movimento, eu ativo novamente o timer, que retém um valor binário referente a 4 segundos, como descrito no problema, e sempre que o timer é zerado, incremento ou decremento o andar, a depender da solicitação feita pelo usuário, de forma que a transição entre andares demore sempre 4 segundos, a uma frequência de 50MHz, pela ação do componente Timer. O andar atual do elevador sempre é registrado em um componente Registrador, que retroalimenta a máquina de estados, atualizando os andares em sequência conforme vão sendo alcançados.

Entradas

Para as entradas, temos um vetor de 6 bits associado, bit a bit a 6 switches (SW0 A SW5), de forma a economizar na quantidade limitada de 10 switches que temos à disposição no kit, visto que não necessitamos de um botão vermelho (descida) no Térreo e nem de um botão azul (subida) no último andar, 3º pavimento.

Assim, SW0 é o switch referente ao botão de subida do Térreo, SW1 o botão de descida do primeiro andar, SW2 o botão de subida do primeiro andar, SW3 é botão de descida do segundo andar, SW4 é botão de subida do segundo andar e, finalmente, SW5 é o botão de descida do terceiro e último andar.

Nos switches restantes, temos: SW6 e SW7 como botões internos do elevador, sinalizando 2 bits que serão interpretados como os andares para os quais se deseja ir, em sistema binário, sendo SW6 o bit menos significativo; e temos SW8 e SW9 como, respectivamente, botões dos elevadores direito e esquerdo para confirmação do destino, de forma que o elevador feche as portas, e parta para o andar de sua escolha.

Usamos também o botão 0 do kit para aplicar o reset assíncrono, restaurando o circuito para o estado inicial.

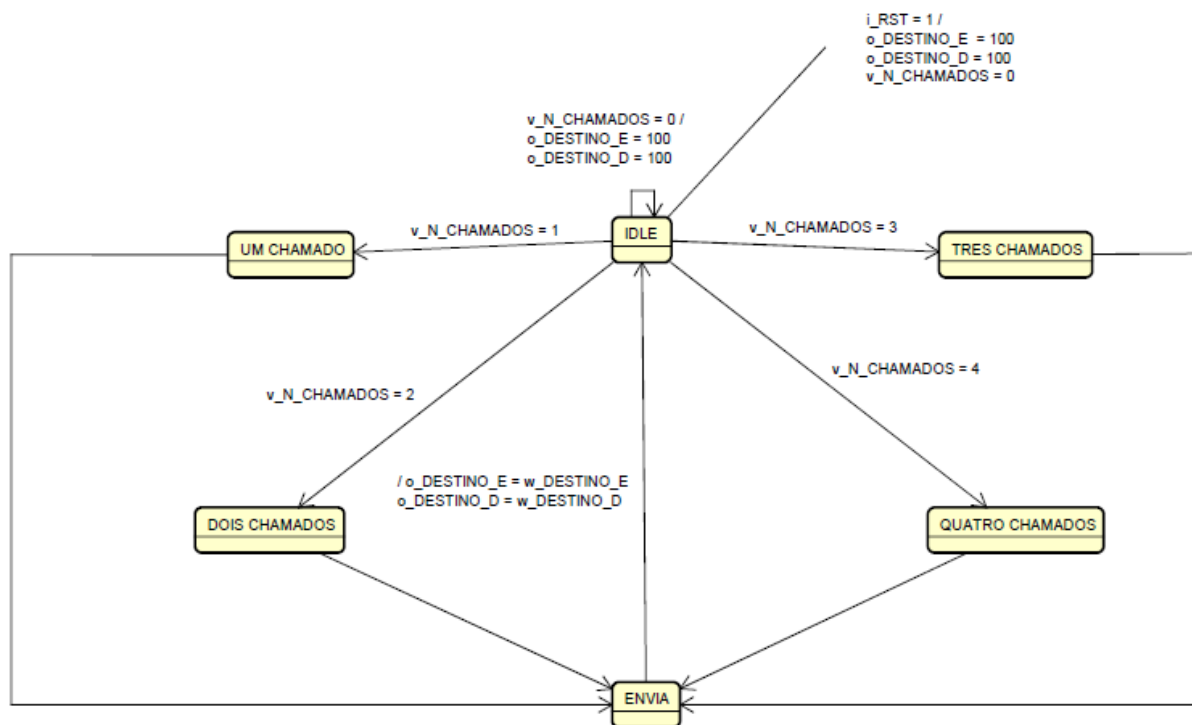
Saídas

Usamos dois displays de 7 segmentos para visualização do andar atual de cada elevador, sendo o display mais à direita referente ao elevador da direita, e o display mais à esquerda referente ao elevador da esquerda.

Também utilizamos dois LEDs, LEDR8 e LEDR9, exatamente acima dos switches de confirmação de painel direito e esquerdo, respectivamente, de forma que estes LEDs acendem indicando que a respectiva porta está abrindo.

Funcionamento

O componente **Seletor Andar** recebe os inputs dos botões de cada andar (SW0 a SW5) e os interpreta para descobrir de qual ou quais andares o elevador foi acionado, retornando como saída os sinais desses andares, que serão capturados pelo componente **Mede Distância**.



O componente **Mede Distância** é outra máquina de estados que, em seu estado *Idle*, conta os chamados em cada andar, sendo possível um máximo de 4 chamados e claro, um mínimo de zero chamados. Enquanto temos zero chamados, mantemos sempre os destinos como térreo e atribuímos como false a confirmação do painel interno, visto que não temos ninguém no elevador para confirmar o destino.

Com um chamado, descobrimos quem fez o chamado através dos sinais vindos do **Seletor de Andar** e abrimos um *request* para este andar em questão, confirmando a ida do elevador parado ou que esteja mais próximo. Caso estejam parados no mesmo andar, sempre manda o esquerdo.

Com dois chamados, novamente descobrimos quem fez os chamados e calculamos as distâncias para os dois elevadores. Se os elevadores estiverem em movimento, não atende nenhum chamado até que algum pare. Parados, calcula o destino mais próximo para cada elevador e atende os chamados.

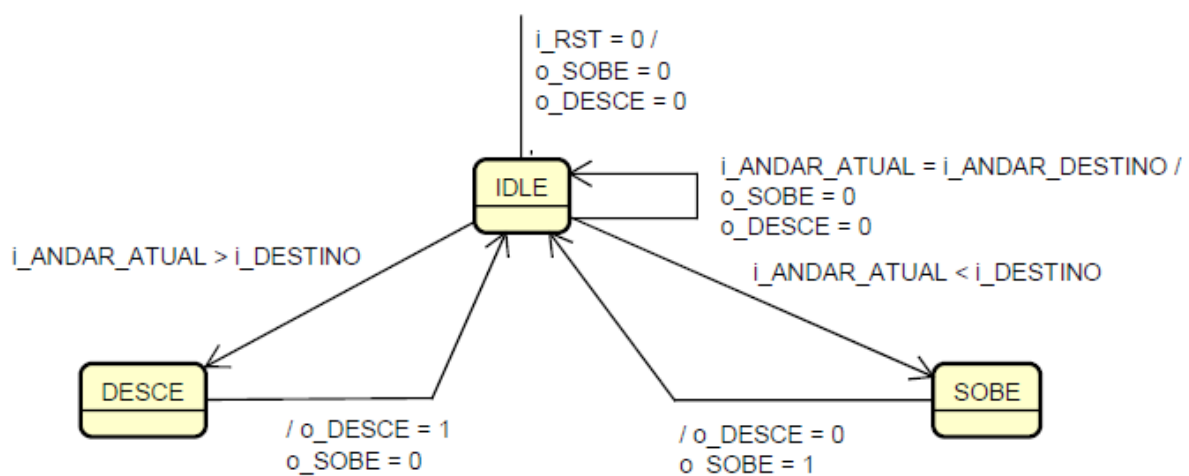
Com três chamados, verifica onde são cada um dos chamados e calcula as distâncias entre todas as chamadas e cada elevador. Verificando qual a distância mais próxima para cada um dos elevadores, manda-os até lá e deixa a terceira chamada na espera.

Com quatro chamados, apenas envia os elevadores para o andar mais próximo.

O **Mede Distância** envia esses dados de destino de cada elevador para o componente **Separador**, que simplesmente separa e interpreta o bit mais significativo do destino recebido, o bit que define a confirmação do destino, quando em nível baixo, e claro, coloca os dois bits menos significativos restantes em suas respectivas posições no vetor destino final.

Com isso, o componente **Seleciona Destino** interpreta os inputs do sistema de confirmação de painel, dados pelo SW8 e SW9, e os compara com a informação adquirida agora do **Mede Distância**, utilizando o input do painel como a checagem definitiva de confirmação de destino. Como o nome já sugere, esse componente também utiliza o input de dois bit de andar de destino, dado por SW7 e SW6, levando esse dado à saída quando o input de confirmação do painel for zero (ativo baixo).

Esses dados são enviados ao **Compara Destinos**, para que se compare o destino atual ao destino em espera, e, a partir do sinal de confirmação e do sinal interno de *wait request*, definir para qual destino irá.



Ainda, temos a máquina de estados **Sobe Desce**, esse componente recebe o destino e o andar atual, dos dois registradores do componente **Elevador**, e os interpreta para definir se o estado é de subida ou descida, assim retornando esses sinais para que a máquina de estados principal possa partir do estado *Aguarda*, já explicado, até os estados *Subir* ou *Descer*, dependentes desse sinal do componente em questão.

Por fim, um dos registradores de cada componente elevador, que armazena o andar atual, envia seu *output* de andar não só para o Mede Distância fazer suas operações, mas também para o respectivo componente decodificador do display de 7 segmentos daquele elevador, para que possamos ter na saída o andar atual de cada elevador em tempo real.

Para mais detalhes sobre diagramas de blocos e esquemas de máquinas de estado, conferir documento *Documentacao.pdf*.

Conclusão

No decorrer do semestre nos foi apresentado que a Linguagem e Descrição de Hardware é extremamente versátil e de grandiosa importância para o mundo de sistemas embarcados. Com este trabalho observou-se, de forma didática, como funciona um sistema controlador de elevador, suas complexidades, componentes necessários e as diversas técnicas essenciais para que funcione no mundo real.

Apesar do semestre atípico, com os conhecimentos adquiridos e alguma pesquisa, foi possível realizar o projeto com clareza e obter um resultado satisfatório.