



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA - CT

CIRCUITOS DIGITAIS

Isaac de Lyra Junior

**CONTADOR COM LIMITE**

ELE2715.1 - Laboratório 08

Natal, 21 de março de 2021

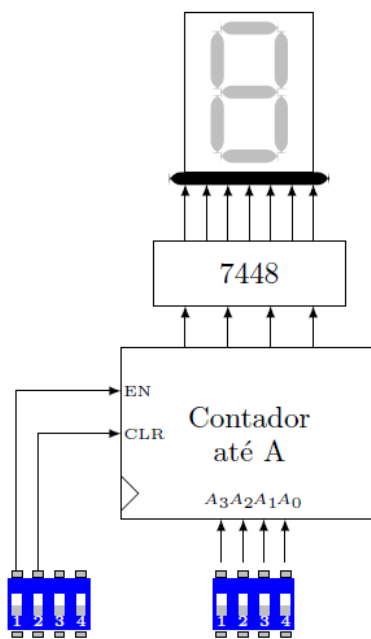
## **SUMÁRIO**

<b>1 DESENVOLVIMENTO</b>	3
1.1 CONTADORES SÍNCRONOS CRESCENTES/DECRESCENTES	3
1.2 COMPARADOR DE MAGNITUDE	4
1.3 MÁQUINA DE ESTADOS	5
<b>2 RESULTADOS</b>	8
<b>REFERÊNCIAS</b>	12
<b>ANEXO A - ESQUEMÁTICO</b>	13

## 1 DESENVOLVIMENTO

O presente relatório tem por fim apresentar a implementação de um contador com limite superior. O limite é inserido pelo usuário através de um vetor de 4 bits denominado de A, o contador deverá contar em ordem crescente até atingir o valor inserido A, ao atingir tal valor, o contador deverá iniciar a contagem decrescente até o valor 0 e repetir este procedimento. O contador deve repetir este procedimento enquanto a entrada EN estiver em nível lógico alto, caso EN esteja em nível lógico baixo o contador deverá parar a contagem. Além disso o contador terá a entrada CLR que em nível lógico alto deverá resetar o contador para 0. O fluxograma do circuito lógico do problema pode ser visto na Figura 1.

**Figura 1** - Fluxograma do problema



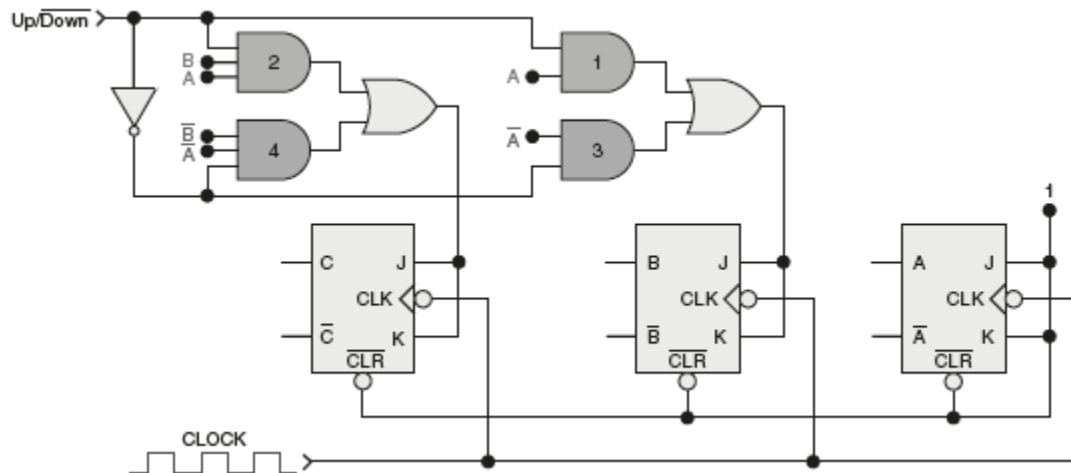
**Fonte:** Dados do problema

### 1.1 CONTADORES SÍNCRONOS CRESCENTES/DECRESCENTES

Para solução do problema proposto, foi necessário pesquisar primeiramente sobre contadores crescentes/decrescentes. A Figura 2 mostra como fazer um contador crescente/decrescente, a entrada de controle Up/Down controla se as entradas J e K dos FFs seguintes serão acionadas pelas saídas ou pelas saídas invertidas dos FFS. Quando Up/Down for mantida em nível ALTO, as portas AND no 1 e no 2 estarão habilitadas, enquanto as portas AND

no 3 e no 4 estarão desabilitadas. Isso permite que as saídas A e B passem pelas portas no 1 e no 2 para as entradas J e K dos FFs B e C. Quando Up/Down for mantida em nível baixo, as portas *AND* no 1 e no 2 estarão desabilitadas, enquanto as portas *AND* no 3 e no 4 estarão habilitadas. Isso permite que as saídas A e B passem pelas portas no 3 e no 4 para as entradas J e K dos FFs B e C (TOCCI, 2011).

**Figura 2** - Implementação de um contador crescente/decrescente



Fonte: TOCCI (2011).

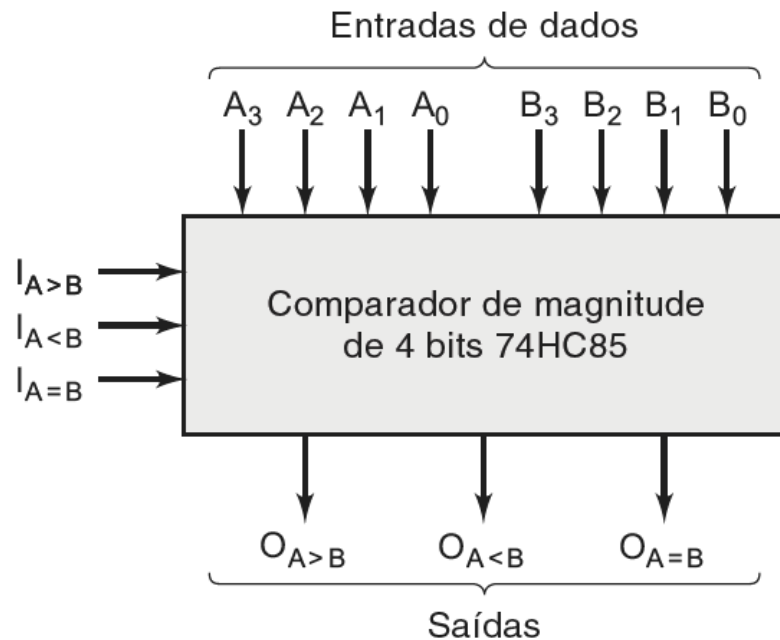
Para desenvolver a solução do problema a mesma lógica acima foi utilizada. Onde a entrada Up/Down da Figura 2 será definida pela saída da máquina de estados.

## 1.2 COMPARADOR DE MAGNITUDE

Outro componente importante para solução do problema são os comparadores de magnitude. Este circuito lógico combinacional compara duas quantidades binárias e gera saídas para indicar qual tem a maior magnitude (TOCCI, 2011).

A Figura 3 mostra o CI comparador de magnitude 74HC85, que recebe duas quantidades binárias A e B de 4 bits e tem três saídas ativas em nível ALTO. A saída  $O_{A>B}$  estará em nível ALTO quando a magnitude da quantidade A for maior que a da palavra B. A saída  $O_{A<B}$ , quando a magnitude da palavra A for menor que a magnitude da palavra B. A saída  $O_{A=B}$ , quando a palavra A e a palavra B forem idênticas (TOCCI, 2011).

**Figura 3** - Comparador de magnitude 74HC85



Fonte: TOCCI (2011).

Para solução do problema foram utilizados dois comparadores de magnitude. O primeiro para comparar a saída dos FFs com o limite A inserido pelo usuário quando o contador estiver em estado crescente. O segundo para comparar com o valor 0 quando o contador estiver no estado decrescente.

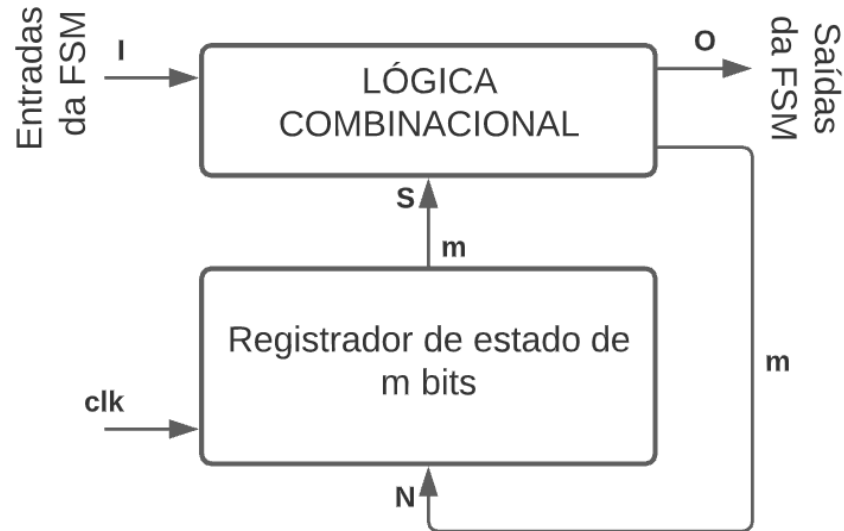
### 1.3 MÁQUINA DE ESTADOS

O termo máquina de estado se refere a um circuito que sequencia um conjunto de estados predeterminados controlados por um clock e outros sinais de entrada (TOCCI, 2011).

A arquitetura de um bloco de controle padrão para uma máquina de estados consiste em um registrador de estado e uma lógica combinacional. O registrador de estado é um registrador de  $m$  bits que contém um número binário que representa o estado atual (ou seja, poderá haver  $2^m$  estados). As entradas da lógica combinacional são as entradas da máquina de estados e também as saídas do registrador de estado. As saídas da lógica combinacional são as saídas da máquina de estado e também os bits do próximo estado que serão carregados no registrador de estado. Os detalhes da lógica combinacional determinam o comportamento do circuito. A Figura

4 mostra uma visão geral da arquitetura do bloco de controle padrão para uma máquina de estados de  $m$  bits de largura (VAHID, 2008).

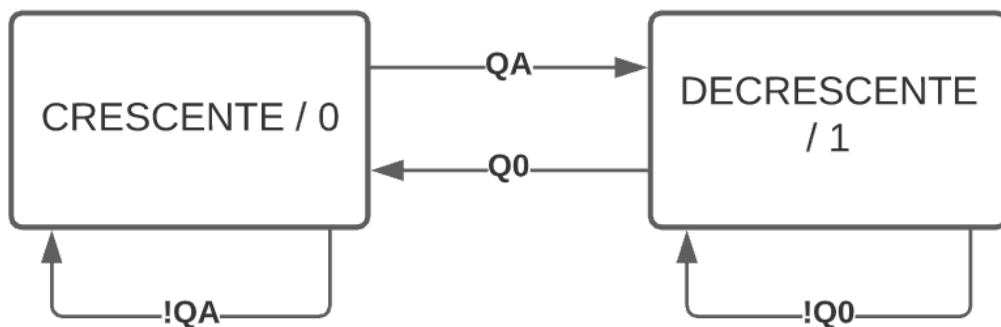
**Figura 4** - Arquitetura de um bloco de controle padrão para uma FSM.



**Fonte:** Adaptado de VAHID (2008).

O primeiro passo para a criação de uma máquina de estados é fazer o diagrama de estados. Para nosso problema foi pensada a solução de apenas dois estados possíveis, crescente e decrescente. O estado crescente dependerá apenas da comparação entre  $Q$ , que representa as saídas dos FFs, e de  $A$ , que representa o limite inserido pelo usuário. A mudança do estado crescente para decrescente acontecerá quando  $Q = A$ . O mesmo procedimento é feito durante o estado decrescente, sendo que a comparação será feita com 0 invés de  $A$ . A Figura 5 mostra o diagrama de estados da solução descrita, é possível perceber pelo diagrama que o modelo utilizado foi o de Moore, ou seja, a saída será completamente síncrona e não terá interferências de entradas externas para definição da saída.

**Figura 5** - Diagrama de estados



Fonte: Autor.

O próximo passo para a solução foi criar a tabela de transição de estados, que será uma representação alternativa do diagrama de estados visto na Figura 5. A Figura 6 mostra a tabela de transição de estados feita para a solução do problema. No estado crescente, a saída S só mudará para 0 (que representa o estado decrescente) quando o comparador retornar que  $Q = A$ , como foi dito anteriormente, e no estado decrescente a saída S só mudará de estado quando  $Q = 0$ .

**Figura 6** - Tabela de transição de estados

ESTADO ATUAL	REG (REGISTRADOR)	QA (Q=A)	Q0 (Q=0)	ESTADO FUTURO	S
CRESCENTE	1	0	X	CRESCENTE	1
	1	1	X	DECRESCENTE	0
DECRESCENTE	0	X	0	DECRESCENTE	0
	0	X	1	CRESCENTE	1

Fonte: Autor.

Por fim, foi utilizado o conceito de Mapa de Karnaugh para transformar a tabela de transição de estados acima em uma lógica combinacional para a saída S, o que resultou na equação booleana abaixo.

$$S = (REG \text{ AND } QA') \text{ OR } (REG' \text{ AND } Q0)$$

## 2 RESULTADOS

Diante do que foi apresentado na seção 1, esta seção tem por objetivo apresentar a implementação da solução discutida até então. Os componentes utilizados na implementação estão expostos na tabela 1.

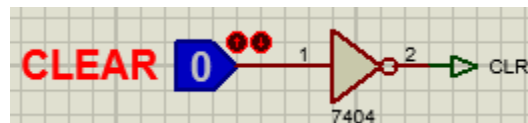
**Tabela 1** - Componentes utilizados na implementação

Função Lógica	CI Comercial
Decodificador BCD-7SEG	7448
Decodificador Bin-BCD 16 bits	
NOT	7404
AND[2]	7408
OR [2]	7432
Comparador 4 bits	74HC85
Flip-flop JK	7476

Fonte: Autor

A lógica para a entrada *CLEAR* que deverá zerar o contador quando estiver em nível lógico foi feita apenas utilizando o CI comercial 7404 que corresponde a porta *NOT*. a saída denominada de CLR foi introduzida em todas as entradas R dos FFs, isso fará com que todos os FFs resetem quando a entrada *CLEAR* estiver em nível lógico alto. A Figura 7 mostra a lógica desta entrada.

**Figura 7** - Lógica da entrada *CLEAR*

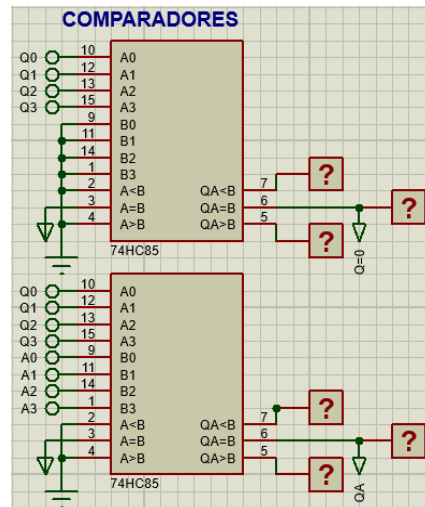


Fonte: Autor



Para implementar os dois comparadores foi utilizado o CI comparador 74HC85, no primeiro e no segundo foram introduzidas as 4 saídas dos FFs, a diferença se dá pelo número a ser comparado, para um será o valor A inserido e para o outro será o valor 0. A Figura 8 mostra o resultado da implementação da comparação, é importante observar que as saídas que serão utilizadas dos dois comparadores de magnitude é a de igualdade.

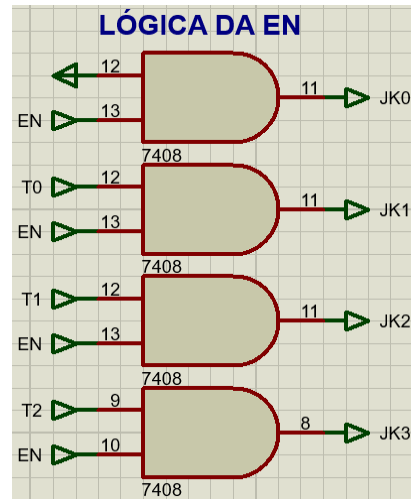
**Figura 8** - Comparadores de magnitude 74HC85



**Fonte:** Autor.

A lógica para a entrada EN que permite a contagem do contador possui uma lógica bem simples e foi implementada com portas *AND* que recebem as saídas da lógica Up/Down do FFs e também a entrada EN do contador, caso a EN estiver em nível lógico baixo os FFs receberão 0 em todas as entradas J e K, isso fará com que os FFs mantenham os valores armazenados sem alterá-los em todos os pulsos de clock. A Figura 9 mostra a implementação da lógica da entrada EN.

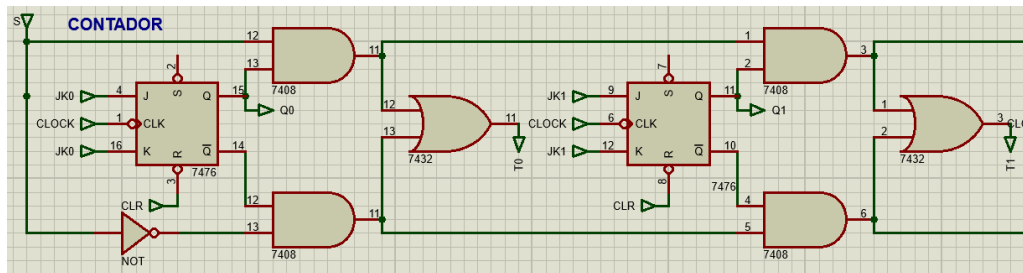
**Figura 9** - Lógica da entrada EN do contador



**Fonte:** Autor.

O contador crescente/decrescente foi implementado utilizando a lógica vista na seção 1.1, a entrada *Up/Down* do contador foi alterada para ser a saída S da lógica combinacional da máquina de estados, a entrada R dos FFs recebem a saída CLR da lógica *CLEAR* e as entradas J e K recebem a saída da lógica EN. A Figura 10 mostra a implementação do contador crescente/decrescente para os dois primeiros FFs, o resto do circuito lógico do contador pode ser visto no Anexo A.

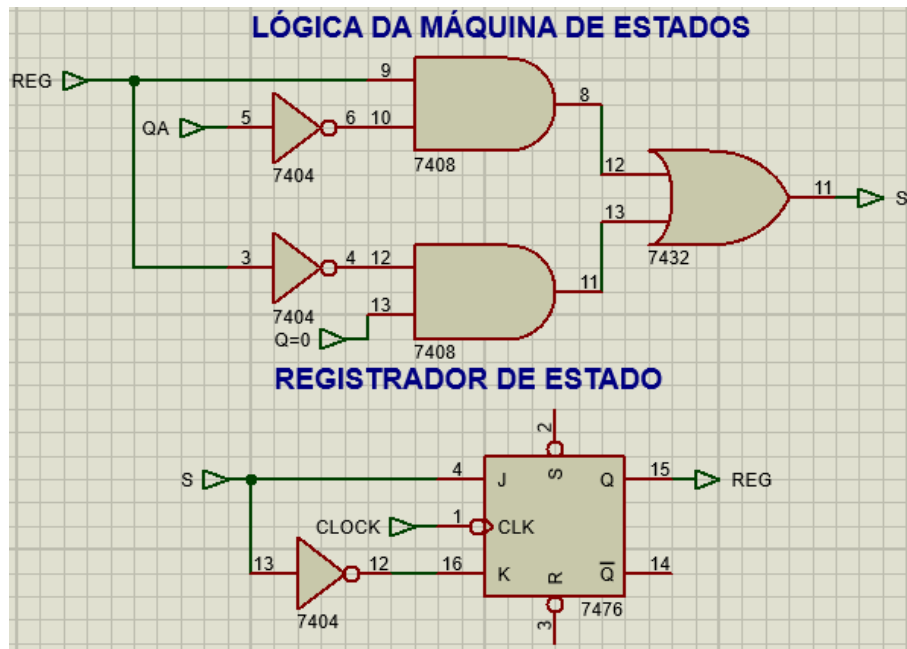
**Figura 10** - Contador crescente/decrescente



**Fonte:** Autor.

A Figura 11 mostra o bloco de controle para a máquina de estados do problema. Foi utilizada a lógica combinacional tratada na seção 1.3, que utiliza as saídas dos comparadores e o estado armazenado no registrador de estado para gerar a saída S que alimenta o contador crescente/decrescente.

**Figura 11** - Bloco de controle para a máquina de estados



**Fonte:** Autor.

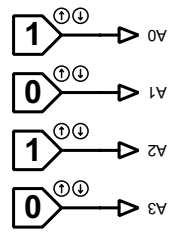
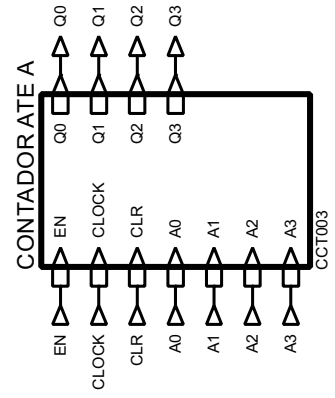
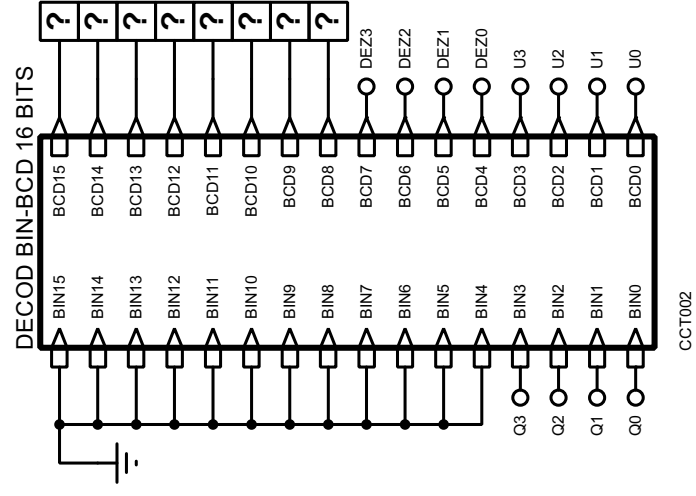
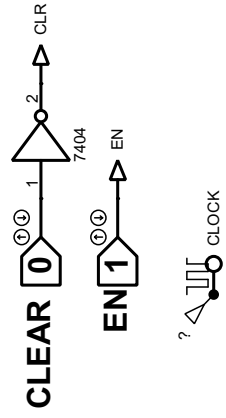
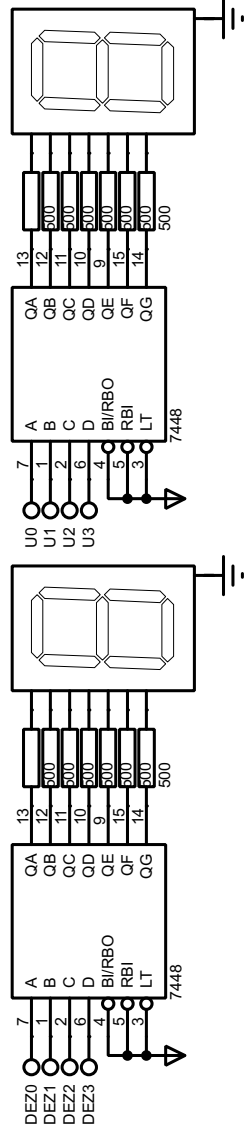
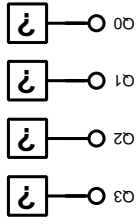
O esquemático do circuito lógico combinacional utilizado para a solução do problema pode ser visto por completo no Anexo A.

## REFERÊNCIAS

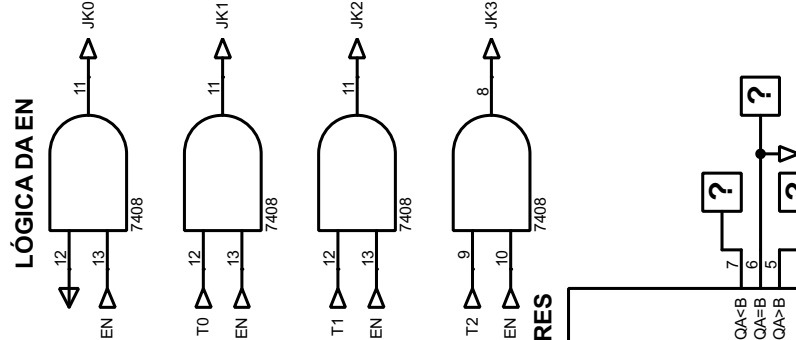
TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L.. **Sistemas Digitais: Princípios e Aplicações**. 11a ed. São Paulo: Pearson, 2011. 844 p

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

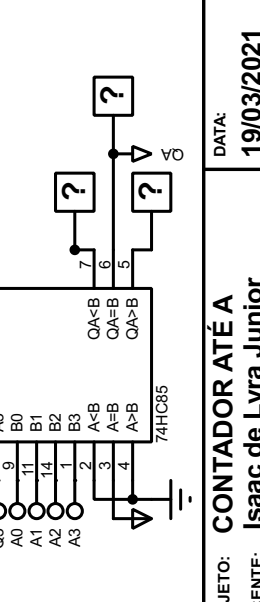
## **ANEXO A - ESQUEMÁTICO**



## LÓGICA DA EN



---



DATA: 19/03/2021

PÁGINAS: 2 de 23