



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA - CT

CIRCUITOS DIGITAIS

## **CONTADOR INTELIGENTE**

ELE2715 - Grupo 02 - Projeto - Problema 03

Wesley Brito da Silva

Isaac de Lyra Junior

Jose Lindenberg de Andrade

Marcelo Ferreira Mota Júnior

Vinicius Souza Fonsêca

Natal, 28 de fevereiro de 2021

## RESUMO

Este relatório tem como objetivo projetar e implementar um Contador Inteligente que tem a capacidade de fazer uma contagem crescente sempre que as entradas **up/dw** estiverem no nível lógico alto, e decrescente quando as mesmas apresentarem nível lógico baixo. Onde, o usuário define o valor mínimo e o valor máximo através da entrada **mx/mi**, em conjunto com a entrada **load** e os valores definidos nas entradas de 4 bits, A2, A1 e A0, no qual a faixa de valores onde o contador irá operar depende disto. O contador também irá crescer e decrescer de acordo com um passo definido pelo usuário, pelo comando **step**, em conjunto com a **load** e o A0. A entrada **clr** estando em nível lógico alto, o contador automaticamente ajusta para contar em 0 e 999, com passo 1. Ao final, a saída LED vai para o nível lógico alto, sempre que o valor de contagem atingir o valor máximo ou o mínimo devidamente definido e as saídas de 4 bits, Q2, Q1 e Q0, representando em BCD o valor da contagem, em qualquer instante.

**Palavras-chave:** Circuitos integrados. Operações. Implementar. Simulação. BCD. VHDL.

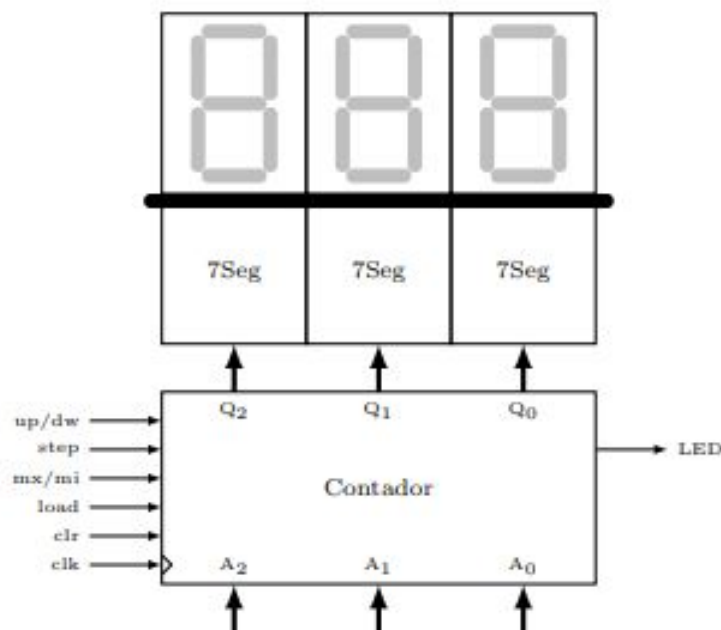
## SUMÁRIO

|   |           |
|---|-----------|
| <b>1. INTRODUÇÃO</b>  | <b>3</b>  |
| Figura 1 - Circuito do Contador Inteligente.  | 3         |
| Figura 2 - Esquemático para um flip-flop SR.  | 4         |
| Figura 3 - Circuito interno do Flip Flop JK.  | 4         |
| <b>2. DESENVOLVIMENTO</b>   | <b>5</b>  |
| 2.1 Contador  | 5         |
| Figura 4 - Contador Progressivo de 4 bits.  | 5         |
| Figura 5 - Contador Regressivo de 4 bits.   | 5         |
| Figura 6 - Registradores com Flip-flops   | 6         |
| 2.2 Funções   | 6         |
| 2.2.1 UP/DOWN   | 6         |
| Figura 7 - Exemplo de contador crescente e decrescente com UP/DOWN de 3 bits.   | 7         |
| 2.2.2 LOAD  | 7         |
| 2.2.3 MX/MI   | 8         |
| Figura 9 - Comparador de magnitude para 4 bits.   | 8         |
| Figura 10 - Comparador utilizando o 74LS85 com entrada de A=B.  | 9         |
| 2.2.4 CLR   | 9         |
| 2.2.5 STEP  | 9         |
| Figura 11 - Somador Completo de 2 bits.   | 10        |
| 2.2.6 BIN-BCD   | 11        |
| A Codificação Binária Decimal, também conhecida como BCD (Binary-coded decimal), se baseia em um sistema de numeração muito utilizado no ramo de informática e em sistemas digitais eletrônicos. Nele, é utilizado apenas dois algarismos: o 0 (zero) e o 1 (um). O BCD codifica o sistema decimal em binário, do números (decimais) 0 a 9, onde cada número é representado pelo seu equivalente binário como mostrado na Tabela 1. | 11        |
| <b>3. CONCLUSÃO</b>   | <b>11</b> |
| <b>REFERÊNCIAS</b>  | <b>12</b> |
| <b>ANEXOS</b>   | <b>13</b> |
| ANEXO A - Relato Semanal  | 14        |
| Anexo B - BIN-BCD   | 16        |

## 1. INTRODUÇÃO

A proposta desta atividade consiste em projetar um contador inteligente (figura 1) capaz de fazer uma contagem crescente sempre que as entradas **up/dw** estiverem no nível lógico alto, e decrescente quando as mesmas apresentarem nível lógico baixo. Onde, o usuário define o valor mínimo e o valor máximo através da entrada **mx/mi**, em conjunto com a entrada **load** e os valores definidos nas entradas de 4 bits, A2, A1 e A0, no qual a faixa de valores onde o contador irá operar depende disto. O contador também irá crescer e decrescer de acordo com um passo definido pelo usuário, pelo comando **step**, em conjunto com a **load** e o A0. A entrada **clr** estando em nível lógico alto, o contador automaticamente ajusta para contar entre 0 e 999, com passo 1. Ao final, a saída LED vai para o nível lógico alto, sempre que o valor de contagem atingir o valor máximo ou o mínimo devidamente definido e as saídas de 4 bits, Q2, Q1 e Q0, representando em BCD o valor da contagem, em qualquer instante.

**Figura 1** - Circuito do Contador Inteligente.



Fonte: Texto base Problema 03.

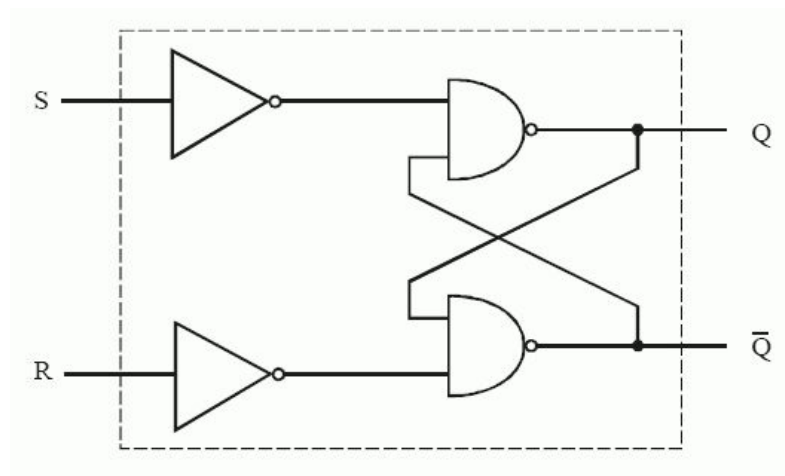
Para especificar o conceito do que são *flip-flops*, temos que ter em mente antes o conceito de lógica sequencial. De maneira simples, circuitos sequenciais são componentes que dependem de uma variável de entrada e/ou de dados anteriores que são armazenados que

são operados por uma sequência de comandos que são gerados por pulsos elétricos (clocks).

Já o *flip-flop* é um circuito de sinal que apresenta a variável de saída  $Q$  e  $Q'$ . Ao se dizer clock juntamente com circuito sequências, dá para ter uma ideia que os valores de saída só se alteram quando se tem um pulso no clock. Já podemos ter uma ideia que os flip-flops são praticamente circuitos sequenciais que tem a possibilidade de fazer inúmeras aplicações.

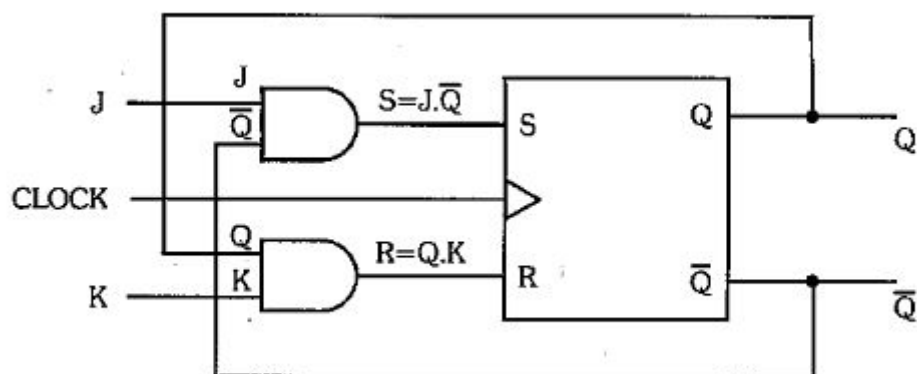
Explicado isso, no presente relatório discorre sobre uma proposta de implementação de um circuito sequencial capaz de realizar aplicações de entrada com resposta de saída em dois sinais.

**Figura 2** - Esquemático para um flip-flop SR.



Fonte: VAHID (2008).

**Figura 3** - Circuito interno do Flip Flop JK.



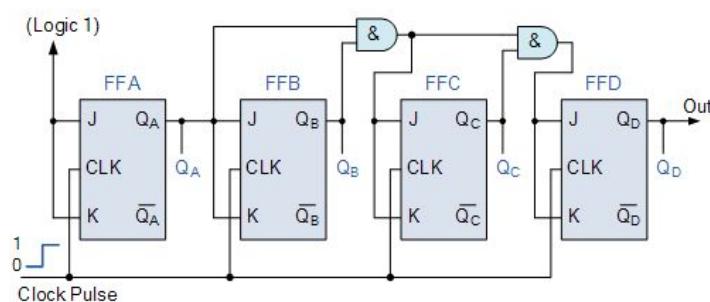
Fonte: VAHID (2008).

## 2. DESENVOLVIMENTO

### 2.1 Contador

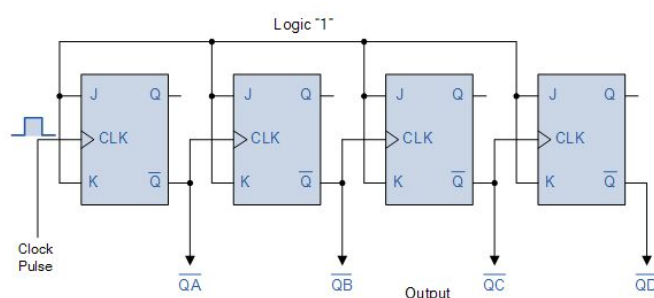
Para elaborar este circuito será necessário um conjunto de 10 flips-flops JK que receberão as entradas A2, A1 e A0. Serão 10 flips-flops pois o maior número que pode ser representado nos 3 displays é 999 e este número em binário será “0011 1110 0111”, ou seja, serão utilizados 10 bits e os bits mais significativos serão zerados devido ao seu não acionamento na entrada do BIN-BCD. Temos como demonstração para construir o contador, conforme a figura 4 e figura 5. Nelas, podemos ver que a saída de um estágio está conectada no próximo.

**Figura 4** - Contador Progressivo de 4 bits.



**Fonte:** Electronics Tutorials (2014).

**Figura 5** - Contador Regressivo de 4 bits.

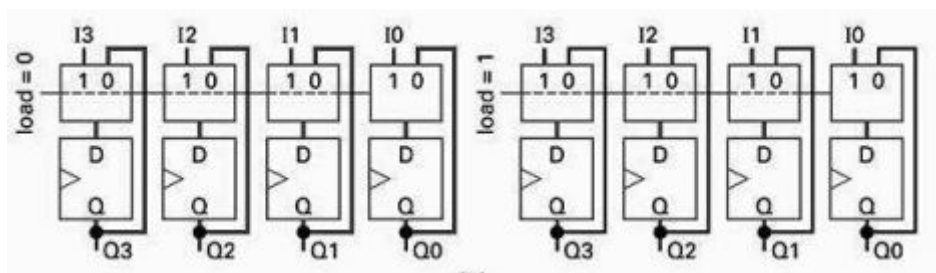


**Fonte:** Electronics Tutorials (2014).

Segundo Vahid (2008), os registradores são dispositivos que armazenam pequenas quantidades de informações. Estas informações são armazenadas no registrador mediante um sinal de carga externo podendo ser usadas por outras partes do circuito. Um

dos elementos que compõem as células básicas dos registradores são os Flip-flops conforme a Figura 6.

**Figura 6** - Registradores com Flip-flops



Fonte: VAHID (2008)

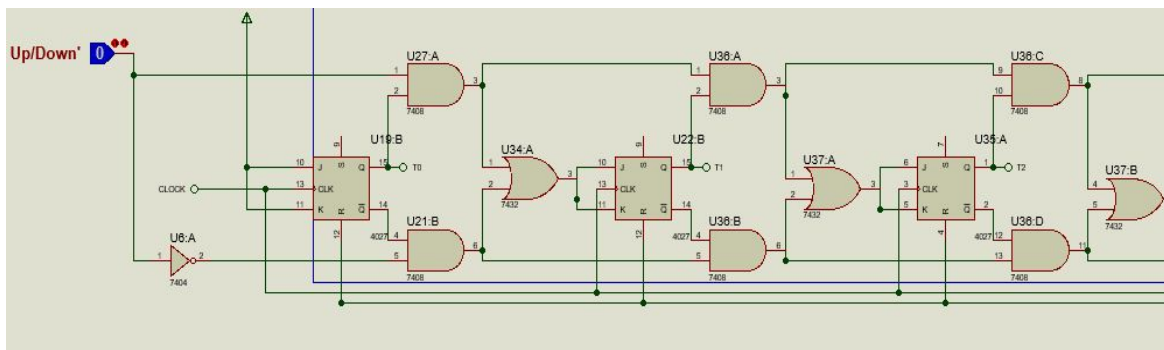
Baseado na figura 6 simplesmente é um conjunto de flip-flops que podem armazenar o dado a cada ciclo de um clock. No caso é bastante útil devido ao seu registro ser carregado em todos os ciclos do clock. No qual é possível obter o controle sobre a carga colocando um multiplexador 2x1 na linha de frente de cada flip-flop, sendo assim, Quando o sinal for 0, entende-se que quando o clock começar o ciclo em 1, o flip-flop conectado será carregado com o valor de sua saída Q, o que resulta a sua gravação interna, do contrário, quando o load for igual a 1, quando houver a mudança do clock, será carregado o valor do dado de entrada que foi feita (0 ou 1). assim em um conjunto de flip-flop poderá retornar o valor de acordo que foi pedido. Por exemplo, se tiver de entrada (1100) na hora da gravação os 4 flip-flops, nesse caso utilizado eles ficaram armazenados na saída Q de cada um.

## 2.2 Funções

### 2.2.1 UP/DOWN

Para realizar esta função foi necessário mudar um pouco da lógica de contador incrementador e o decrementador para juntá-los, além de inserir uma porta de um bit que será o seletor que definirá o tipo de operação a ser realizada como podemos ver na figura 7.

**Figura 7 -** Exemplo de contador crescente e decrescente com UP/DOWN de 3 bits.



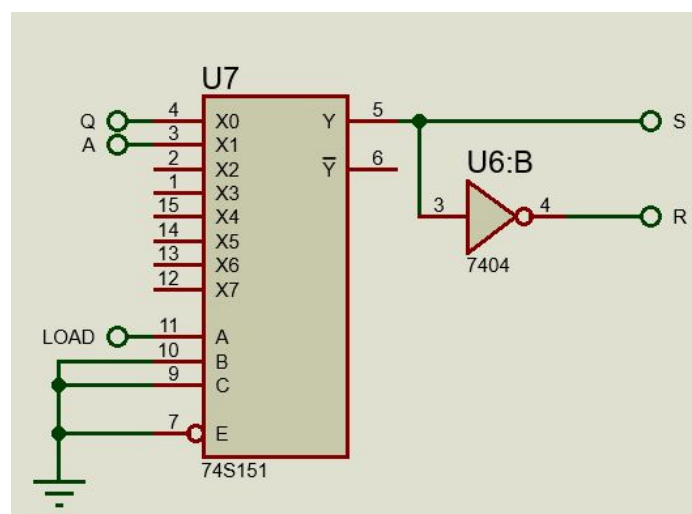
**Fonte:** Autores.

### 2.2.2 LOAD

Para esta função, é preciso entender a funcionalidade das entradas do S e R dispostas. Quando o  $S=0$  e  $R=0$ , a saída Q sofre influência apenas do clock. Quando  $S=1$  e  $R=0$ , a saída Q é igual a 1. Quando o  $S=0$  e  $R=1$ , a saída Q é igual a 0. Por fim, quando o  $S=1$  e  $R=1$ , não há uma saída possível.

Então os bits de saída dos Flips Flops JK (FFJK) passarão por um multiplexador que vai depender do “LOAD”. Sendo esta função que definirá qual a saída a ser resetada diretamente no contador. O LOAD funciona de maneira a comutar os dados da entrada que serão definidas pelo usuário, sendo estes os A’s e as saídas correspondentes dos Flips Flops JK (T0-T3). Como podemos ver na figura 8, a saída irá diretamente para o S dos registradores e será invertida no R.

**Figura 8 - Função Load.**



Fonte: Autores.



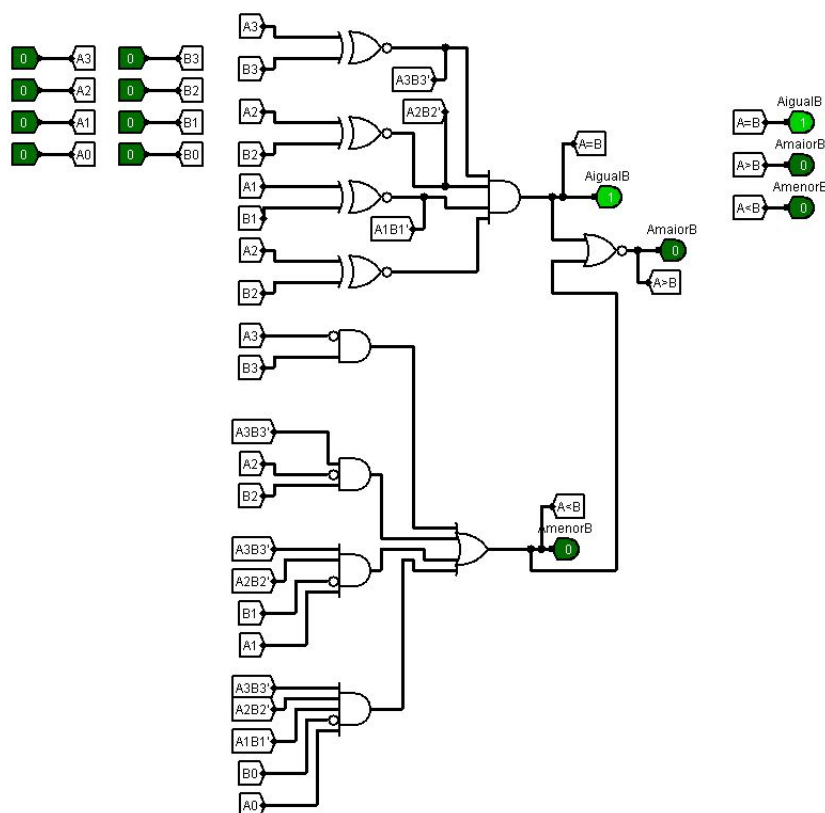
Outro encargo do LOAD é definir quantos passos (STEP) ocorrerão na contagem. Para isso, deverá ser feita uma porta AND com entrada LOAD e a CLR invertida (figura 12).

### 2.2.3 MX/MI

Percebemos também que os números em bits, ao incrementarem, vão até o “1111” que é seu número máximo, ultrapassando o limite que queremos que é “1001” (9 em binário). Para resolver esse problema deve ser implementada uma função chamada “comparador”.

Este bloco é um componente operacional que tem o objetivo de equiparar duas entradas de dois conjuntos de números binários A e B. No caso, estão sendo utilizados para comparar os valores que são dispostos pelos Flips Flops com os valores limites do contador, neste caso são o ‘9’ e o ‘0’. Podemos ver na figura 9, a lógica por trás do comparador utilizado. Vemos que AigualB é equivalente quando os valores das entradas de A e B são iguais ( $A = B$ ). Em relação a AmaiorB é quando o valor binário de A é maior que os valores de B ( $A > B$ ). De forma análoga em AmenorB é quando o valor em binário de A é menor que o valor binário de B ( $A < B$ ). Para este comparador se dá o nome de comparador de magnitude.

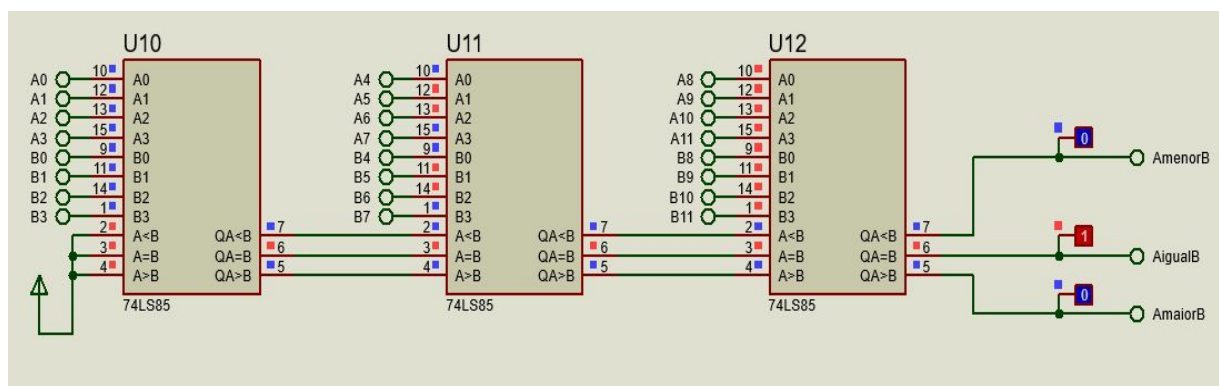
**Figura 9** - Comparador de magnitude para 4 bits.



Fonte: Autores.

Com isso, o CI mais indicado para tal função é o 74LS85, pois a estrutura de funcionamento é semelhante com a do comparador exibido na figura anterior. Temos um exemplo da entrada do A sendo "111101100000" e B "111101100000", na comparação, os dois são iguais, e isso é o que comprovamos na Figura 10.

**Figura 10** - Comparador utilizando o 74LS85 com entrada de A=B.



**Dado:** Autores.

A partir da imagem temos a entrada binário em A como sendo “11110110000” e em B temos “111101100000”. Logo como essa verificação é feita bit a bit então percebe-se que eles são iguais na sua saída delimitada em 1. Mostrando assim a operação de igualdade binária.

## 2.2.4 CLR

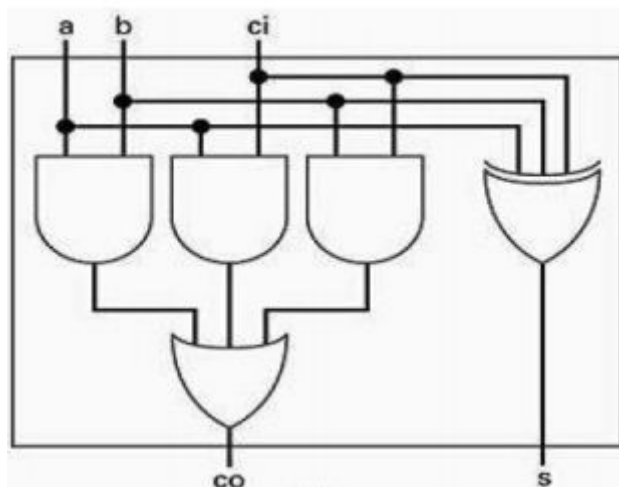
Esta função tem como objetivo de definir o contador automaticamente ajustando-o para contar entre 0 e 999 com passo de 1, mas só quando o nível lógico do CLR estiver alto. Logo, para que isto aconteça, o CLR deve ser barrado antes da operação de AND com o Load (Figura 12) e essa saída será a chave para comutar os valores selecionados pelo multiplexador.

## 2.2.5 STEP

Para a elaboração do circuito utilizando os Steps, precisa de um multiplexador que terá como entradas os valores da saída de uma somador completo (Figura 11) entre os valores de saída dos Flips Flops (T) e com o valor do passo definido na entrada (A0) e quem fará o

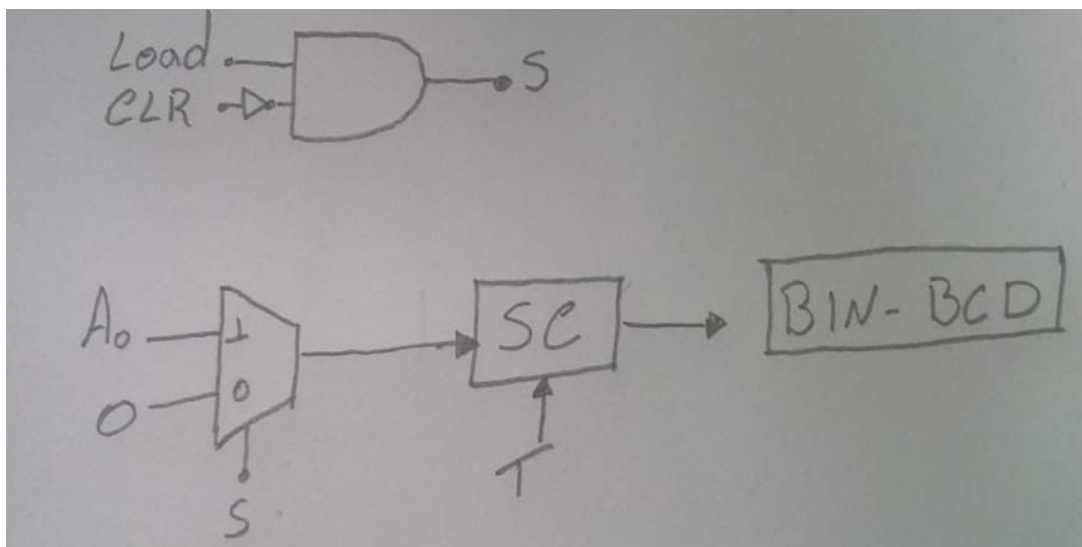
registro desses passos será a entrada “LOAD” definida pelo usuário, de acordo com a figura 12.

**Figura 11** - Somador Completo de 2 bits.



Fonte: VAHID (2008).

**Figura 12** - Entradas e saídas do multiplexador do STEP.



Fonte: Autores.

Isto é, quando o clr for 1 e o LOAD estiver ativo, a saída desse AND será 0 e a saída do multiplexador será o valor de A0 com passo 1. Do contrário será feita a quantidade de passos que o usuário digitar. Vale lembrar que isso é feito somente para 1 bit do A0. Como são 4 bits de A0 então deve ser feito tal procedimento mais três vezes antes do somador

completo. Sendo assim as entradas A1 e 0 possuem como saída do MUX, a entrada do segundo somador completo e assim por diante.

A partir daí todas as saídas do somador completo irá a entradas do BIN-BCD e este será responsável pelo acionamento dos led dos display de 7 segmentos para a operação.

### 2.2.6 BIN-BCD

A *Codificação Binária Decimal*, também conhecida como **BCD (Binary-coded decimal)**, se baseia em um sistema de numeração muito utilizado no ramo de informática e em sistemas digitais eletrônicos. Nele, é utilizado apenas dois algarismos: o 0 (zero) e o 1 (um). O BCD codifica o sistema decimal em binário, do números (decimais) 0 a 9, onde cada número é representado pelo seu equivalente binário como mostrado na Tabela 1.

**Tabela 1:** Tabela verdade do BCD

| $A_3A_2A_1A_0$ | $S_3S_2S_1S_0$ |
|----------------|----------------|
| 0000           | 0000           |
| 0001           | 0001           |
| 0010           | 0010           |
| 0011           | 0011           |
| 0100           | 0100           |
| 0101           | 1000           |
| 0110           | 1001           |
| 0111           | 1010           |
| 1000           | 1011           |
| 1001           | 1100           |

Fonte: Autores.

## 3. CONCLUSÃO

O presente trabalho propôs projetar e implementar um Contador Inteligente. Utilizando os dados encontrados após a elaboração da solução do problema, será possível criar o contador. Para construção da resolução desse problema foi implantada as respostas de problemas anteriores para um melhor entendimento, como é o caso da utilização dos multiplexadores e a operação do somador completo na função STEP, ou seja, podemos perceber que a elaboração por meio de multiplexadores facilitou a lógica de comutar os valores desejados para serem mostrados com um menor trabalho.

Foi possível ver também que é legítimo criar um contador crescente e decrescente, após entender o funcionamento de Flips Flops, principalmente o JK, utilizando de suas

diversas configurações, como a síncrona e assíncrona.

## REFERÊNCIAS

- [1] VAHID, Frank. **Sistemas digitais: projeto, otimização e HDLS**. Rio Grande do Sul: Artmed Bookman, 2008. 558 p
- [2] **A figura acima representa um flip-flop RS básico**. Estude Grátis, 2021. Disponível em <https://www.estudegratis.com.br/questao-de-concurso/418656>. Acesso em 27 de fevereiro de 2021.
- [3] TUTORIALS, Eletronics. **Synchronous Counter**. 2014. Disponível em: [https://www.electronics-tutorials.ws/counter/count\\_3.html](https://www.electronics-tutorials.ws/counter/count_3.html). Acesso em: 26 fev. 2021.
- [4] ENDIGITAL. **Sequencial**. Disponível em: <http://endigital.orgfree.com/sequencial/Image77.gif>. Acesso em: 28 fev. 2021.

## **ANEXOS**

## ANEXO A - Relato Semanal

**Líder:** Vinicius Souza Fonseca

### A.1 Equipe

**Tabela 1:** Identificação da equipe.

| <b>Função do grupo:</b> | <b>Discente</b>              |
|-------------------------|------------------------------|
| Redator                 | Wesley Brito da Silva        |
| Debatedor               | Isaac Lyra de Junior         |
| Videomaker              | Jose Lindenberg de Andrade   |
| Auxiliar                | Marcelo Ferreira Mota Junior |

### A.2 Defina o problema

O problema consiste em projetar um circuito lógico para implementar uma Unidade de conta aritmética de 12 bits que tem a capacidade de realizar a contagem de 0 até 999 em número decimal que deverá ser demonstrado nos 3 displays de 7 segmentos, no qual deve ser selecionadas através de uma chave seletora (número binário) de 4 bits.

### A.3 Registro do *brainstorming*

Inicialmente pensamos em como o contador inteligente funciona, usando o proteus com os CIs específicos para ter noção e entendimento do que precisamos fazer para o desenvolvimento da solução da questão, de acordo, da não utilização de circuitos integrados prontos para o funcionamento. Antes do estudo, após a aula de circuito digital separamos os horários disponíveis de cada membro e após isso planejamos o que devemos pesquisar, para ter o desenvolvimento do circuito eletrônico.

Em um segundo momento discutimos sobre a adaptação de algumas partes do projeto, o que acarretou o problema de implementação ao ser usado os flip flops de maneira que possa fazer a contagem sem precisar no final um bcd para o display, o que ocasionava contagem em hexadecimal após o valor do número 9 “1001”, após muito esforço e erros tivemos um esclarecimento de um colega da turma e foi decidido refazer novamente do zero, só que consciente do que estávamos errando e aplicando os consertos em cima dos erros cometidos.



Por fim, com todos os circuitos operando adequadamente, partimos para a finalização das atividades, sendo estas: desenvolvimento de um relatório de atividades, criação de vídeo explicativo e preparação para debates.

#### **A.4 Pontos-chaves**

Para o desenvolvimento do contador inteligente, precisamos previamente entender o conceito e funcionamento do *flip-flop* ( para desenvolvimento de contador). Além disso, é necessário entender que para o circuito máximo e mínimo, precisamos entender o funcionamento do comparador que pode haver 4 limites, que é definido o máximo e mínimo pelo usuário, outra parte sendo o load e a função clear no circuito lógico e no final poder usar conhecimento adquirido para a realização da atividade e fazer a montagem do circuito até chegar na demonstração esperada do resultado no display de 7 segmentos.

#### **A.5 Questões de pesquisa**

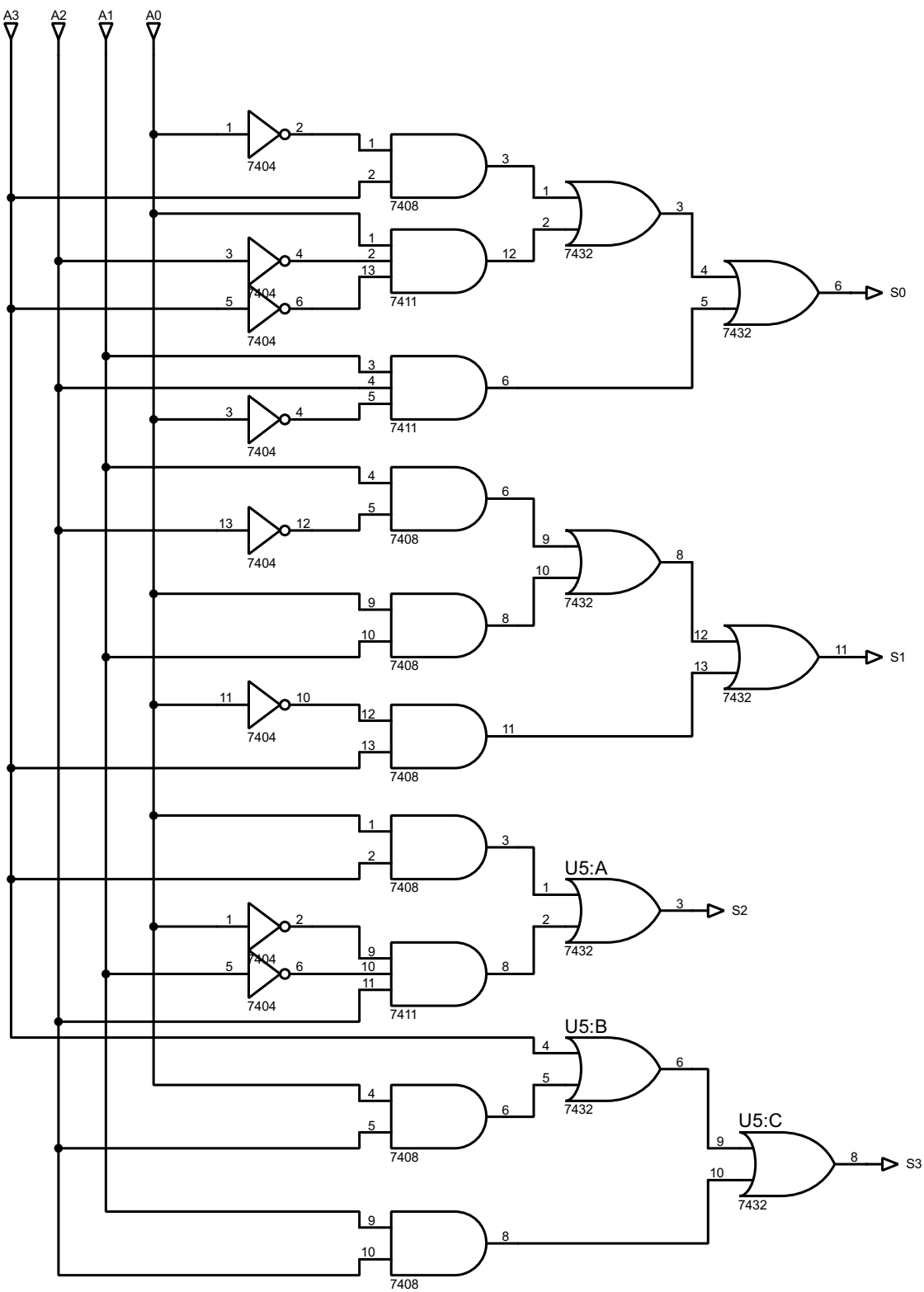
Algumas questões de pesquisas foram:

- Funcionamento de Flip-Flop
- *Contador*
- Comparadores
- Registradores

#### **A.6 Planejamento da pesquisa**

Inicialmente foi tomada a decisão de deixar a quarta-feira dia 24 de fevereiro como dia de estudo, devido aos compromissos com relação a prova e trabalhos na terça-feira dia 23 de fevereiro, cada integrante do grupo realizou uma pesquisa sobre as operações para que no brainstorm todos estivessem tendo conhecimento da problemática. E no dia seguinte foi feito o brainstorm, a partir de então, foram realizadas pesquisas em conjunto de acordo com as ideias adquiridas e feito todo no simulador.

**Anexo B - BIN-BCD**



NOME PROJETO: Unidade logica do BIN-BCD  
AUTOR: Grupo 02

DATA: 28/02/2021  
PAGINA: 01 DE 02

Conversor BIN-BCD 12-bits

