

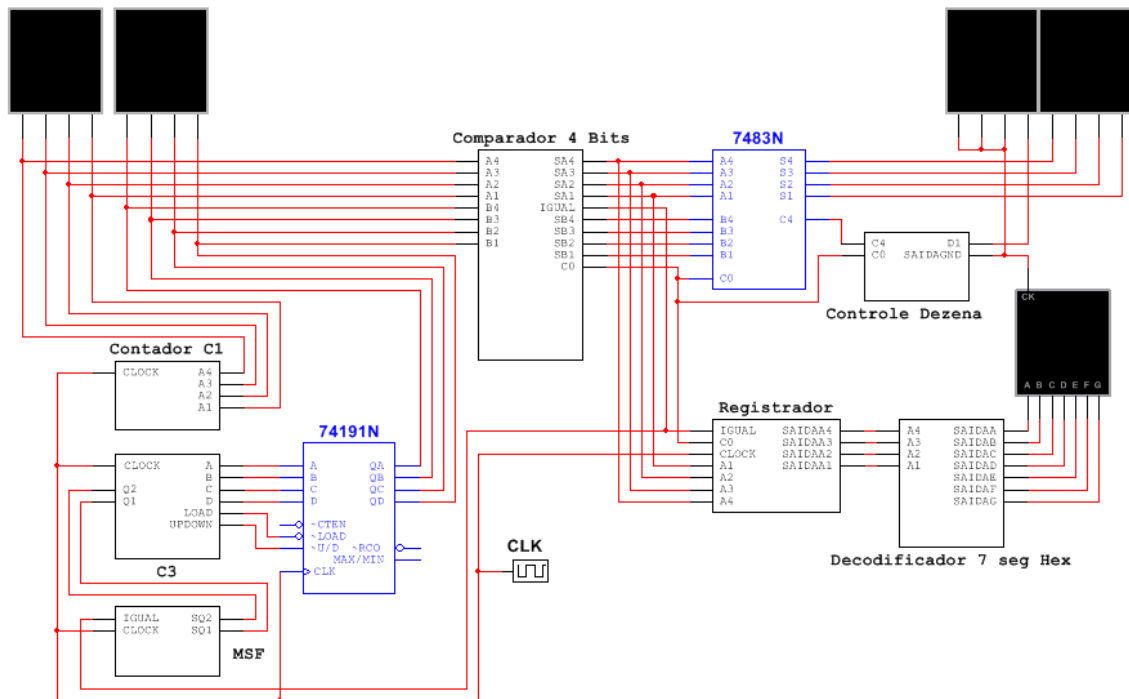
**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CÂMPUS PATO BRANCO  
ENGENHARIA DE COMPUTAÇÃO**

**RAFAEL ANDERSON DALMOLIN  
WESLEY DO NASCIMENTO ALMEIDA**

**PROJETO INTEGRADOR - PI  
PROJETO DE CIRCUITOS COMBINACIONAIS E SEQUENCIAIS**

**PATO BRANCO - 2018**

**CIRCUITO ELÉTRICO PROJETO FINAL**



## 1. Contador C1

### 1.1. Planejamento

Para o desenvolvimento do contador C1, teve como base o modelo de contadores síncronos, no qual os flip-flops são controlados simultaneamente pelo mesmo pulso de clock, garantindo assim que todas as mudanças de estados dos flip-flops aconteçam ao mesmo tempo, evitando assim a propagação de atrasos que ocorrem nos modelos de contadores assíncronos.

## 1.2. Estados do contador

O contador sequencial C1 teve como sequência de contagem cíclica os valores F-4-7-C-0-3-8-1-F, sendo utilizados 4 flip-flops para melhor representação das palavras com 4 bits.

## 1.3. Características dos flip-flops

Flip-flops tipo D.

Sensíveis à borda de subida.

Set e Reset ativos em nível lógico igual a 1.

## 1.4. Funcionamento dos Flip-flops

CLOCK	D	Qf
0	0	Qa
0	1	Qa
1	0	0
1	1	1

Tabela-verdade Flip-flop tipo D

## 1.5. Desenvolvimento

Construiu-se a tabela para C1.

		SAÍDAS ATUAIS				SAÍDAS FUTURAS				ENTRADAS ATUAIS			
ESTADO	ESTADO	Q3	Q2	Q1	Q0	Q3	Q2	Q1	Q0	D3	D2	D1	D0

ATUAL	FUTURO												
0	3	0	0	0	0	0	0	1	1	0	0	1	1
1	F	0	0	0	1	1	1	1	1	1	1	1	1
2	X	0	0	1	0					X	X	X	X
3	8	0	0	1	1	1	0	0	0	1	0	0	0
4	7	0	1	0	0	0	1	1	1	0	1	1	1
5	X	0	1	0	1					X	X	X	X
6	X	0	1	1	0					X	X	X	X
7	C	0	1	1	1	1	1	0	0	1	1	0	0
8	1	1	0	0	0	0	0	0	1	0	0	0	1
9	X	1	0	0	1					X	X	X	X
A	X	1	0	1	0					X	X	X	X
B	X	1	0	1	1					X	X	X	X
C	0	1	1	0	0	0	0	0	0	0	0	0	0
D	X	1	1	0	1					X	X	X	X
E	X	1	1	1	0					X	X	X	X
F	4	1	1	1	1	0	1	0	0	0	1	0	0

Através da ferramenta de Karnaugh Map Minimazer, obteve-se as respectivas saídas D3, D2, D1, D0.

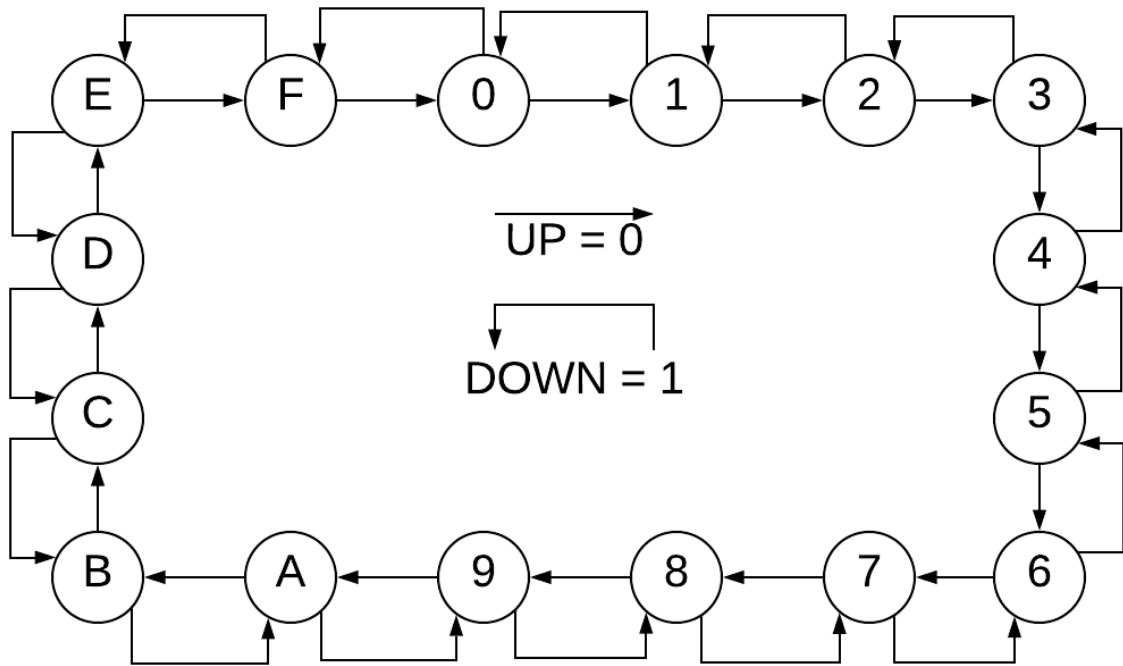
MAP D3					MAP D2				
	Q1'Q0'	Q1'Q0	Q1Q0	Q1Q0'		Q1'Q0'	Q1'Q0	Q1Q0	Q1Q0'
Q3'Q2'	0	1	1	X	Q3'Q2'	0	1	0	X
Q3'Q2	0	X	1	X	Q3'Q2	1	X	1	X
Q3Q2	0	X	0	X	Q3Q2	0	X	1	X





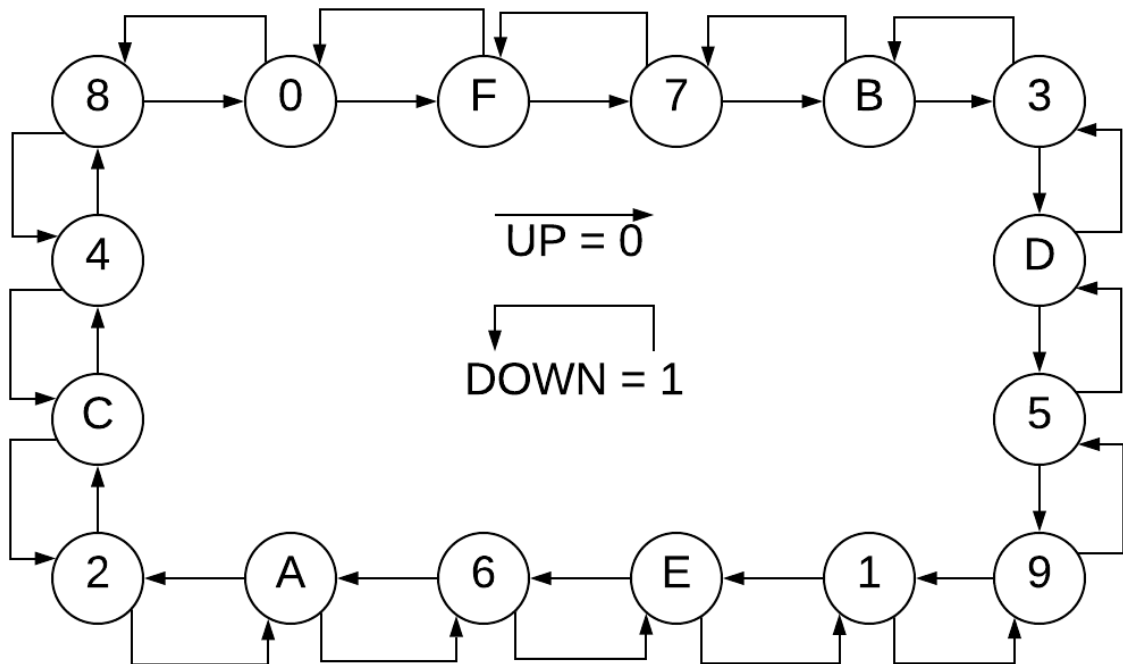
- ❑ 4 inputs Data A, B, C, D: que tem como utilidade repassar um valor para sua saída, para continuação da sua contagem a partir deste número, essa opção só tem funcionamento quando LOAD está em baixo nível;
- ❑ 1 input Enable CTEN: serve para habilitar ou desabilitar o clock do contador e o CTEN é sensível a borda de descida;
- ❑ 1 input LOAD: tem como finalidade ativar sua contagem particular;
- ❑ 1 input UP/DOWN: Essa entrada tem como finalidade alterar o sentido da contagem para Up em nível lógico baixo e para down em nível lógico alto;
- ❑ 1 input clock: pulso de controle;
- ❑ 4 outputs Data QD (MSB), QC, QB, QA(LSB) : saída dos valores de cada flip-flops;
- ❑ 1 Ripple Clock: output de clock em atraso (Ripple Clock) para ligação em cascata;
- ❑ 1 output mín/máx: serve como continuidade do clock para ligações em série. Ativo em nível lógico 1 sempre que a contagem chegar em seus números min/max, mantendo-se em nível lógico 0 no restante da contagem.

### **Diagrama de transição de estados (74191)**



Para construção do projeto utilizou-se das saídas QA como (MSB) e QD como (LSB), para fins de alterar a contagem do contador C2.

**Diagrama de transição de estados utilizado no projeto:**





### 3. Somador completo binário de 4 bits (7483)

#### 3.1. Descrição

Possui 9 portas de entrada A4, A3, A2 e A1 sendo para representar um número binário, com A4 sendo o algarismo MSB e A1 o algarismo LSB.

As entradas B4, B3, B2 e B1 para representar a outra parcela da soma, do mesmo modo B4(MSB) e B1(LSB) e c0 que (*Carry in*) é um dígito a mais a ser somado. Já para saída o CDI tem 5 portas SUM4, SUM3, SUM2, SUM1 e C4 no caso C4 é 1 quando o valor da soma é maior que 4 bits e zero se o valor da soma é menor 4 bits e as outras portas de saídas são os valores das soma respectivamente sendo SUM4(MSB) e SUM1(LSB).

### 4. Comparador de magnitude de 4 bits

#### 4.1. Planejamento

Para o desenvolvimento do comparador de magnitude foi utilizado circuitos combinacionais lógicos com a finalidade de se descobrir quando um número binário de 4 bits é maior, menor ou igual a outro número binário de 4 bits.

Como foi proposto no Projeto integrador, faz-se a soma sempre que o número do contador C2 for maior que o número do contador C1, e caso contrário faz-se a subtração. Sendo C1 o minuendo e C2 o subtraendo.

Para se fazer a subtração foi utilizado o complemento de 2, assim fazendo a inversão do número e somando mais uma unidade.

#### 4.2. Desenvolvimento

Na fabricação do comparador viu-se a necessidade de comparar dois números binários, partindo do algarismo mais significativo e seguindo gradativamente para o próximo algarismo menos significativo sempre que encontrado uma igualdade entre os dois bits, até que se determine que um número seja maior, menor ou igual ao outro.

Sendo assim foi desenvolvido um comparador de magnitude de 1 bit com três entradas A, B e AUX e três saídas S2, S1 e S0.

sendo A e B as entradas dos dois bits a serem comparados e a AUX que tem o papel de condicionar qual será o valor das saídas.

Quando o valor de entrada de AUX é zero todas as saídas são levadas para nível lógico zero, e quando o valor de AUX é 1, somente uma das três saídas fica em nível lógico 1, definindo assim S2 : A > B S1: A=B e S0 :A<B.

verifica-se na tabela abaixo:

ENTRADAS			SAÍDAS		
AUX	A	B	S2	S1	S0
0	0	0	0	0	0
0	0	1	0	0	0
	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	0	1	0

Através ferramenta de Karnaugh Map Minimazer, obteve-se as respectivas saídas S2, S1, S0.

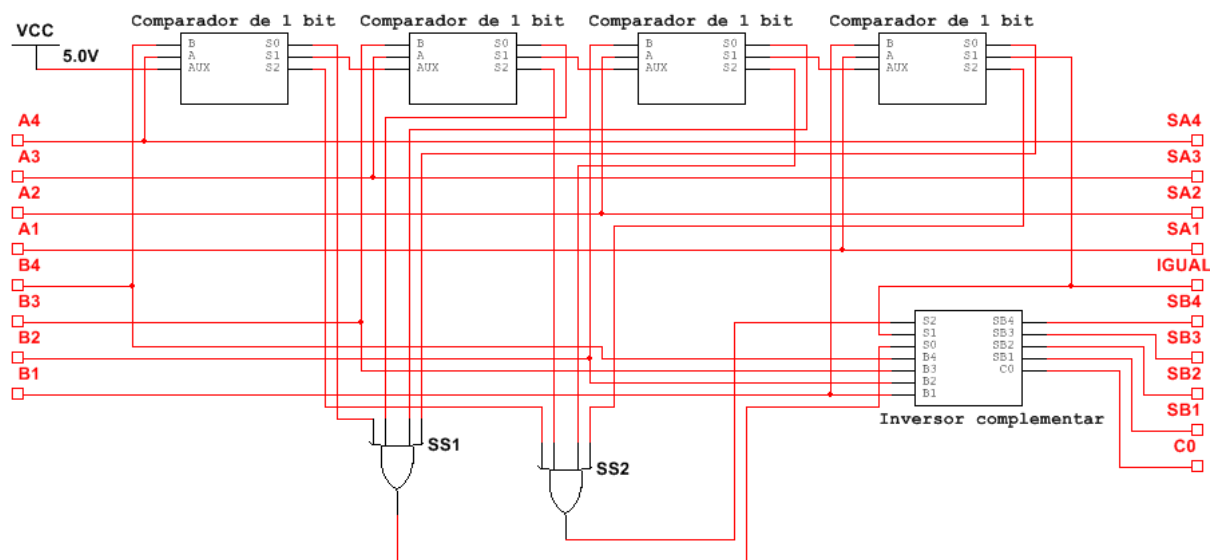
$$S2 = ABC'$$

$$S1 = AB'C' + ABC$$

$$S0 = AB'C$$

Para a comparação de palavras de 4 bits foi utilizado 4 comparadores de magnitude de 1 bit ligados em cascata, onde a saída S1 do comparador mais significativo é ligado a entrada AUX do próximo comparador assim sempre que for definido a igualdade entre dois bits, ativa-se o próximo comparador e assim por diante até definir uma resposta.

A imagem a seguir ilustra o funcionamento do comparador de 4 bits.



## 5. Decodificador BCH 7 Segmentos

### 5.1. Desenvolvimento

Como indicado no projeto integrador, foi necessário o desenvolvimento de um decodificador (BCH - 4 bits) que recebe nos pinos de entrada (E3 E2 E1 E0) um número em código binário e o decodifica para ser exibido em código hexadecimal em um display de 7 Segmentos do tipo catodo comum, que acende os segmentos quando recebe nível lógico 1.

Para construir o display utilizou-se de ferramentas como tabela verdade das possíveis entradas e saídas representadas pela tabela abaixo.

	ENTRADAS				SAÍDAS						
VALOR	E3	E2	E1	E0	a	b	c	d	e	f	g

0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	1	1	1	0	1	1	1
B	1	0	1	1	0	0	1	1	1	1	1
C	1	1	0	0	1	0	0	1	1	1	0
D	1	1	0	1	0	1	1	1	1	0	1
E	1	1	1	0	1	0	0	1	1	1	1
F	1	1	1	1	1	0	0	0	1	1	1

E para a obtenção das expressões booleana de cada saída utilizou-se do software Karnaugh Map Minimazer. como demonstrado nas tabela abaixo:

MAP a				
	C'D'	C'D	CD	CD'
A'B'	1	0	1	1
A'B	0	1	1	1
AB	0	1	0	0
AB'	1	1	0	1
MAP LAYOUT a				
	C'D'	C'D	CD	CD'
A'B'	0	1	3	2
A'B	4	5	7	6
AB	12	13	15	14
AB'	8	9	11	10
Groups				
0,2,8,10	B'D'			
2,3,6,7	A'C			
5,13	BC'D			
8,9	AB'C'			

$$a = B'D' + A'C + BC'D + AB'C'$$

Construindo-se o mapa de Karnaugh para cada saída obtém-se as seguintes expressões booleanas para se montar o circuito combinacional .

$$a = B'D' + A'C + BC + AD' + A'BD + AB'C'$$

$$b = A'B' + B'D' + A'C'D' + A'CD + AC'D$$

$$c = A'C' + A'D + C'D + A'B + AB'$$

$$d = AC' + A'B'D' + B'CD + BC'D + BCD'$$

$$e = B'D' + CD' + AC + AB$$

$$f = C'D' + BD' + AB' + AC + A'BC'$$

$$g = B'C + CD' + AB' + AD + A'BC'$$

## 6. REGISTRADOR

### 6.1. Desenvolvimento

À vários tipos de registradores estes são classificados de acordo com a forma de armazenamento e saída dos dados, neste projeto foi selecionado o modelo com entrada e saída paralela (parallel in/parallel out - PIPO) e flip-flops do tipo D sensível à borda de subida.

Para fazer a montagem utilizou-se de um circuito combinacional para tratar fatores como delay com o intuito de evitar o falso registro.

Desta maneira para que o clock dos flip-flops fiquem em nível lógico 1 e se registre o valor nas suas entradas, foi atribuído 3 variáveis responsáveis, sendo elas clock, carry out, igualdade.

Onde a única opção que quando satisfeita se registra o número é quando o clock vale 0, carry 1 e o parâmetro de igualdade 1, como demonstra a tabela abaixo.

ENTRADAS			SAÍDA
CLOCK	CARRY	IGUAL	S
0	0	0	0
0	0	1	0
0	1	0	0

0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Construindo-se o mapa de Karnaugh para a saída obtém-se a seguinte expressão booleana.

$$S = \text{CLOCK}' \cdot \text{CARRY} \cdot \text{IGUAL}$$

## 7. MÁQUINA DE ESTADOS FINITOS (FSM)

### 7.1. Desenvolvimento

Para a realização do projeto integrador foi necessário a construção de uma máquina de estado finitos que ao receber um estímulo de entrada faz a troca do estado atual para o próximo estado, onde cada um dos estados realiza mudanças distintas sobre o circuito.

De acordo com a definição do projeto foi construído uma máquina que realizasse a troca entre 4 estados distintos.

Para a construção da FSM, utilizou-se de dois flip-flops do tipo D, para criar um contador síncrono sequencial com a contagem cíclica definida 0-1-2-3 que inicia sua contagem em 0 depois indo para 1, 2, 3 e retornando para 1 novamente excluindo assim o 0 da contagem após um ciclo.

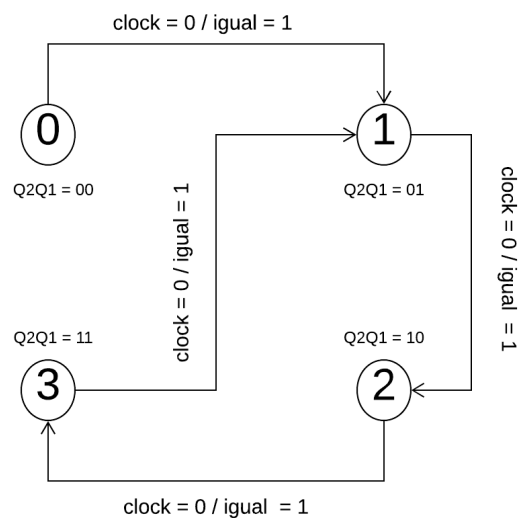
Seu desenvolvimento pode ser conferido conforme a tabela abaixo.

		SAÍDAS ATUAIS		SAÍDAS FUTURAS		ENTRADAS ATUAIS	
ESTADO ATUAL	ESTADO FUTURO	Q2	Q1	Q2	Q1	D2	D1
0	1	0	0	0	1	0	1
1	2	0	1	1	0	1	0
2	3	1	0	1	1	1	1
3	1	1	1	0	1	0	1

Através ferramenta de Karnaugh Map Minimazer, obteve-se as respectivas saídas D2, D1 para se montar o circuito combinacional.

MAP D2			MAP D1		
	Q1'	Q1		Q1'	Q1
Q2'	0	0	Q2'	0	0
Q2'	0	1	Q2'	0	1
MAP LAYOUT D2			MAP LAYOUT D1		
	Q1'	Q1		Q1'	Q1
Q2'	0	1	Q2'	0	1
Q2'	2	3	Q2'	2	3
Groups			Groups		
1		Q2' Q1	0,2		Q1'
3		Q2 Q1'	2,3		Q2
D2 = Q2'Q1 + Q2Q1			D1 = Q1' + Q2		

### Diagrama de transição de estados:



## 7.2. Descrição de cada estado

### 7.2.1. Estado zero

Estado inicial da FSM, neste estado os contadores C1 e C2 seguem suas contagem cíclicas pré definidas.

### 7.2.2. Estado um

Neste estado o contador C2 é forçado a recomençar sua contagem a partir do número E em hexadecimal.

### 7.2.3. Estado dois

Neste estado o contador C2 é forçado a recomençar sua contagem a partir do número 0(zero) em hexadecimal.

#### 7.2.4. Estado Três

Neste estado o contator C2 é forçado a alterar o sentido de contagem (de up para down ou vice-versa).

## 8. CIRCUITO C3

## 8.1. Planejamento

Com a máquina de estados gerando as saídas houve a necessidade de representar o ciclo da FMS, assim foi estabelecido um circuito combinacional que recebe como entrada as saídas da FMS e faz todo o processo de mudança que incide sobre o contador C2 como definido em cada estado anteriormente no tópico 7.2.

## 8.2. Desenvolvimento

Inicialmente para desenvolver C3, foi construído um subcircuito que é responsável por setar/resetar os 4 FFs responsáveis por transmitir as mudanças que cada estado faz sobre o contador C2, segue a tabela do mesmo abaixo:

[illegible]



Analisando as tabelas, foi verificado que os sets e resets no intervalo de 4 a 2 tem a mesma expressão como resultado de saída, havendo particularidade apenas no set 1 e reset 1.

MAP SET 4...2			MAP SET 1		
	Q1'	Q1		Q1'	Q1
CLOCK' Q2'	0	1	CLOCK' Q2'	0	0
CLOCK' Q2	0	0	CLOCK' Q2	0	0
CLOCK Q2	0	0	CLOCK Q2	0	0
CLOCK Q2'	0	0	CLOCK Q2'	0	0
MAP LAYOUT SET 4...2			MAP LAYOUT SET 1		
	Q1'	Q1		Q1'	Q1
CLOCK' Q2'	0	1	CLOCK' Q2'	0	1
CLOCK' Q2	2	3	CLOCK' Q2	2	3
CLOCK Q2	6	7	CLOCK Q2	6	7
CLOCK Q2'	4	5	CLOCK Q2'	4	5
Groups			Groups		
1		CLOCK' Q2' Q1			
SET 4...2 = CLOCK' Q2' Q1			RESET 1 = 0		
MAP RESET 4			MAP RESET 1		
	Q1'	Q1		Q1'	Q1
CLOCK' Q2'	0	0	CLOCK' Q2'	0	1
CLOCK' Q2	1	0	CLOCK' Q2	1	0
CLOCK Q2	0	0	CLOCK Q2	0	0
CLOCK Q2'	0	0	CLOCK Q2'	0	0
MAP LAYOUT RESET 4			MAP LAYOUT RESET 1		
	Q1'	Q1		Q1'	Q1
CLOCK' Q2'	0	1	CLOCK' Q2'	0	1
CLOCK' Q2	2	3	CLOCK' Q2	2	3
CLOCK Q2	6	7	CLOCK Q2	6	7
CLOCK Q2'	4	5	CLOCK Q2'	4	5
Groups			Groups		
1		CLOCK' Q2 Q1'	1		CLOCK' Q2 Q1
RESET 4 = CLOCK' Q2 Q1'			2		CLOCK' Q2 Q1'
			RESET 1 = CLOCK' Q2 Q1 + CLOCK' Q2 Q1'		

MAP LOAD			MAP UP/DOWN		
	Q1'	Q1		Q1'	Q1
CLOCK' Q2'	1	0	CLOCK' Q2'	0	0
CLOCK' Q2	0	1	CLOCK' Q2	0	1
CLOCK Q2	1	1	CLOCK Q2	0	0
CLOCK Q2'	1	1	CLOCK Q2'	0	0
MAP LAYOUT LOAD			MAP LAYOUT UP/DOWN		
	Q1'	Q1		Q1'	Q1
CLOCK' Q2'	0	1	CLOCK' Q2'	0	1
CLOCK' Q2	2	3	CLOCK' Q2	2	3
CLOCK Q2	6	7	CLOCK Q2	6	7
CLOCK Q2'	4	5	CLOCK Q2'	4	5
Groups			Groups		
4,5,6,7		CLOCK	3		CLOCK' Q2 Q1
0,4		Q2'Q1'	UP/DOWN = CLOCK' Q2 Q1		
3,7		Q2Q1			
LOAD = CLOCK + Q2'Q1' + Q2Q1					

Para que no estado três haja uma alteração no sentido de contagem de C2 (de up para down ou vice-versa), foi utilizado um flip-flop tipo T onde a cada vez que passar pelo estado três o mesmo recebe nível lógico 1, assim complementado sua saída de up para down e vice versa.