



Departamento de Engenharia de Computação e Automação  
DCA0212.1 – CIRCUITOS DIGITAIS  
2016011573 – Gabriel de Oliveira Barreto  
2016021177 – Layse Pereira de A. Batista  
20170031179 – Mariana Camila C. de Queiroz

## **RELATÓRIO - CONTADOR UP/DOWN**

NATAL - RN  
OUTUBRO/2018

## 1. INTRODUÇÃO

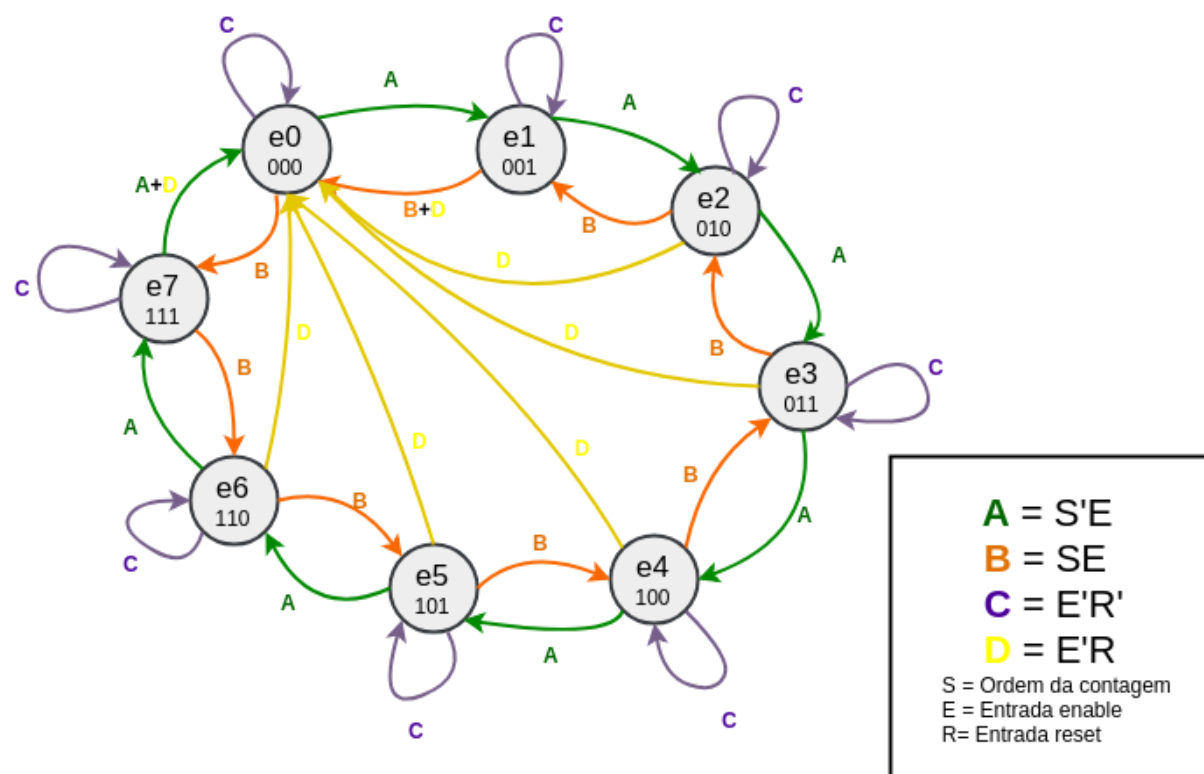
Esse projeto tem como finalidade construir e implementar um contador capaz de contar números binários de 3 bits de forma progressiva ou regressiva a cada transição de clock. A contagem, por sua vez, poderá ser visualizada por meio de um display BCD de 7 segmentos.

Para isso, usaremos, entre outros, os conceitos, de portas lógicas, flip-flop JK e máquina de estados finita.

## 2. DESENVOLVIMENTO

O primeiro passo para o desenvolvimento do projeto foi a construção da máquina de estados finita (FMS) de acordo com as regras do contador. Sendo assim, foi elaborado o diagrama de estados abaixo:

**Figura 1** - Diagrama de estados



**Fonte:** Elaborado pelo autor

Esse diagrama descreve o comportamento do contador up/down de 3 bits, o qual é capaz de contar de 0 a 7 na ordem crescente (indicado pelas setas verdes) quando a entrada  $S=0$ , e ordem decrescente (indicado pelas setas laranjas), quando  $S=1$ . Além disso, o contador recebe as entradas *enable* e *reset*, as quais habilita a contagem de acordo com a ordem selecionada e retorna para zero, respectivamente. No entanto, o *reset* só funciona quando o *enable* estiver

desabilitado (setas amarelas). Por fim, quando ambos estiverem desabilitados, a contagem permanece estática (setas roxas).

Tendo, portanto, o diagrama de estados, foi possível construir a tabela-verdade apresentada abaixo (Figura 3), a qual é baseada no comportamento do flip-flop JK (Tabela 1):

**Tabela 1:** Comportamento do Flip-Flop JK

J	K	Q
0	0	Q0 (mantém)
0	1	1
1	0	0
1	1	Q0' (alterna)

Fonte: Elaborado pelo autor

**Figura 2:** Tabela-verdade

Estado Atual			Estado Futuro (S=0)			Estado Futuro (S=1)			Crescente						Decrescente					
Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>
0	0	0	0	0	1	1	1	1	1	X	1	X	0	X	0	X	0	X	0	X
0	0	1	0	1	0	0	0	0	1	X	0	X	X	0	1	X	1	X	X	0
0	1	0	0	1	1	0	0	1	1	X	X	1	0	X	1	X	X	0	0	X
0	1	1	1	0	0	0	1	0	0	X	X	0	X	0	1	X	X	1	X	0
1	0	0	1	0	1	0	1	1	X	1	1	X	0	X	X	0	0	X	0	X
1	0	1	1	1	0	1	0	0	X	1	0	X	X	0	X	1	1	X	X	0
1	1	0	1	1	1	1	0	1	X	1	X	1	0	X	X	1	X	0	0	X
1	1	1	0	0	0	1	1	0	X	0	X	0	X	0	X	1	X	1	X	0

Fonte: Elaborado pelo autor

A partir da lógica aplicada nas tabelas acima, extraímos as equações lógicas de J<sub>2</sub>, J<sub>1</sub>, J<sub>0</sub> e K<sub>2</sub>, K<sub>1</sub>, K<sub>0</sub>, as quais estão apresentadas abaixo, considerando também as estradas *enable* e *reset*:

$$J_2 \leq ((Q_1 Q_0)'S' + (Q_1 + Q_0)'S)E + E'R$$

$$K_2 \leq ((Q_1 Q_0)'S' + (Q_1 + Q_0)'S)E + E'R'$$

$$J_1 \leq (Q_0'S' + Q_0S)E + E'R$$

$$K_1 \leq (Q_0'S' + Q_0S)E + E'R'$$

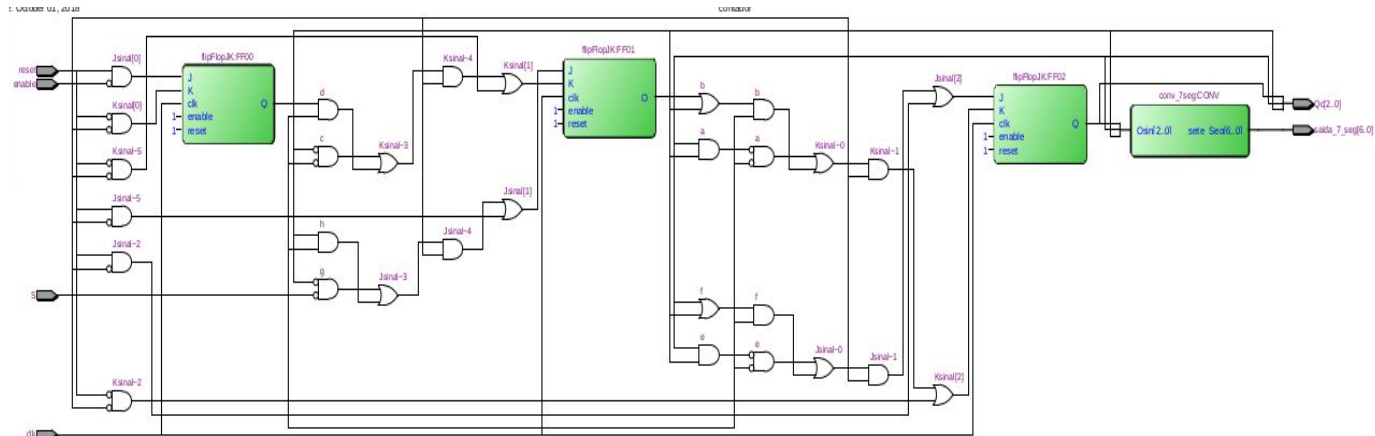
$$J_0 \leq 0 + E'R$$

$$K_0 \leq 0 + E'R'$$

A partir dessas equações, foi possível implementar na linguagem VHDL um programa que descreve o comportamento desejado para o projeto. Abaixo, pode-se

observar uma visão geral através de um esquema RLT obtido no *software* Quartus II:

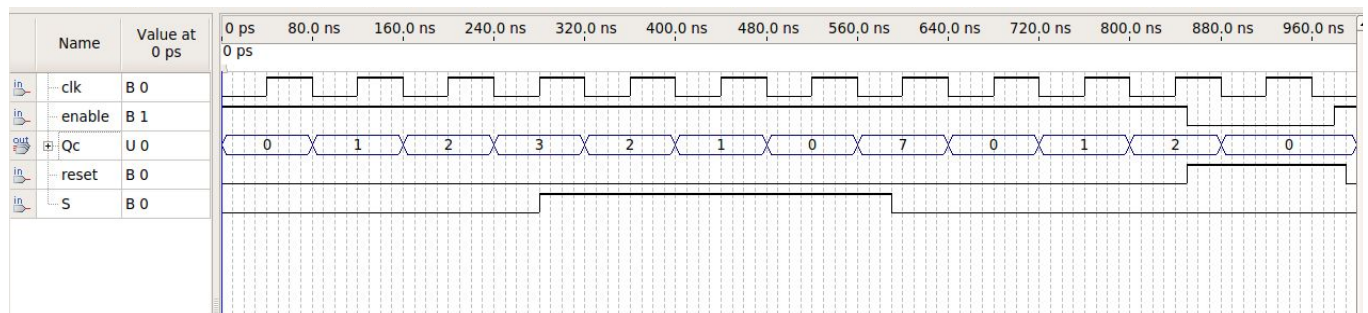
**Figura 3: Esquema RTL**



Fonte: Elaborado pelo autor

Por fim, elaboramos uma simulação em forma de onda para verificar o funcionamento do contador, assim, obtivemos os seguintes resultados apresentados na Figura 4:

**Figura 4: Teste de ondas**



Fonte: Elaborado pelo autor

A Figura 4 mostra inicialmente uma contagem na ordem crescente, quando  $S=0$ ,  $reset=0$  e  $enable=1$ . Quando há uma transição de clock, nesse caso sensível a descida, a contagem vai de 3 para 2 e continua na ordem decrescente pois o sinal de  $S$  muda para 1. Em seguida,  $S$  volta para 0 e a contagem é feita na ordem crescente novamente. Por fim, pode-se observar um sinal 0 no fim pois o reset foi ativado e o enable desativado.

### 3. CONCLUSÃO

A partir da análise do problema apresentado, foi possível realizar as etapas para a construção de um projeto, desde a projeção da máquina de estado, até a implementação do código em VHDL de acordo com os conhecimentos adquiridos

até então. Sendo assim, a escolha de utilizar o comportamento do flip-flop JK permitiu que obtivéssemos uma tabela bem mais fácil de se trabalhar, se comparada ao flip-flop D, por exemplo, o que facilitou desde a obtenção das equações até a implementação do código.