



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO  
DCA0212.0 – CIRCUITOS DIGITAIS – TEORIA

## **PROJETO MARCAPASSO**

### **DISCENTES:**

EFRAIN MARCELO PULGAR PANTALEON  
FERNANDO LUCAS SOUSA SILVA  
MATHEUS GOMES DINIZ ANDRADE  
TEÓFILO VITOR DE CARVALHO CLEMENTE

### **DOCENTES:**

TIAGO BARROS  
SAMUEL XAVIER

## 1. Introdução

O marcapasso atrioventricular é um dispositivo implantável que se conecta por eletrodos ao coração, sendo capaz de monitorar o ritmo cardíaco e estimular o mesmo, evitando que os batimentos fiquem abaixo do nível normal (WECOR, 2020)<sup>1</sup>. Com objetivo de projetar um marcapasso para consolidar os assuntos teóricos vistos em aula, o presente relatório abordará a lógica combinacional, equações para formar o bloco de controle e máquinas de estado que consistem na modelagem de um aparelho marcapasso.

## 2. Processo de Design do Controlador

O processo de desenvolvimento do design de um controlador digital, consiste em 5 etapas mostradas abaixo para o controlador de um marcapasso atrioventricular.

### 2.1. Máquina de Estados Finitos (FSM)

Essa representação, apresentada na figura 1, foi disponibilizada na orientação do projeto e consiste na etapa de captura de como ocorre o funcionamento do bloco de controle, a partir dela é possível observar os instantes no qual cada processo ocorre e assim entender seu funcionamento para aplicação da lógica.

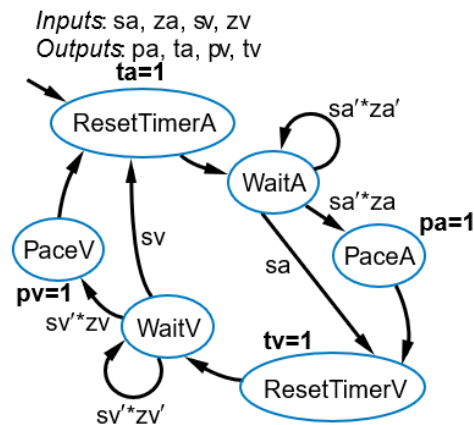


Figura 1 - Máquina de estados finitos (FSM).

### 2.2. Bloco de Controle

O bloco de controle é uma arquitetura criada para o entendimento do projeto, nesse caso, para o marcapasso atrioventricular, ela é composta por um registrador de 3 bits que representa os 6 estados, mostrados na etapa anterior, e pela lógica combinacional que possui 7 entradas: 4 externas (sa, sv, za, zv) e 3 do próprio registrador (s0, s1, s2), e 7 saídas: 4 externas (pa, ta, pv, tv) e 3 que são entradas para o registrador (n0, n1, n2). Como demonstrado na figura 2.

<sup>1</sup> <https://www.wecor.com.br/o-que-e-marcapasso-e-quando-e-recomendado>

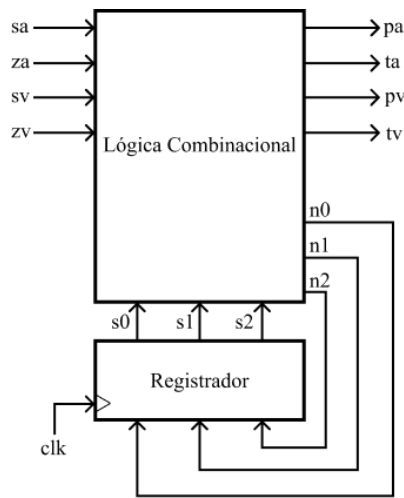


Figura 2 - Bloco de controle para marcapasso atrioventricular.

### 2.3. Codificar Resultados

A codificação foi realizada pensando na quantidade de estados vistos na etapa anterior, sendo um total de 6. Por isso, precisou-se de 3 bits que é capaz de representar 8 estados, no entanto, apenas 6 serão utilizados para a representação dos estados de 0 a 5 (com exceção do 6 e 7 que não serão necessários), dispostos da seguinte forma: ResetTimerA = 000; WaitA = 001; PaceA = 010; ResetTimerV = 011; WaitV = 100; PaceV = 101. Como mostrado na figura 3, a seguir.

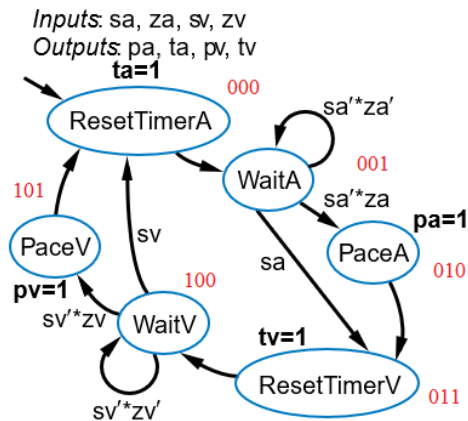


Figura 3 - Codificação de estados.

### 2.4. Tabela Verdade

Baseado nas informações coletadas pelos processos anteriores, foi possível construir a tabela verdade para a máquina de estados finitos, no entanto, tendo em vista que a tabela completa teria 128 ( $2^7$ ) linhas, se fez necessário simplificar para obter uma tabela verdade mais específica. Com isso, analisando os sinais de entrada e saída, percebe-se que para alguns casos os sinais: sa, za, sv e zv, não fazem diferença para o valor resultante das saídas: n0, n1 e n2, que são as saídas que importam para o registrador de estados. Portanto, a tabela resultante está mostrada na figura 4.

	INPUTS							OUTPUTS						
	sa	za	sv	zv	s2	s1	s0	pa	ta	pv	tv	n0	n1	n2
ResetTimerA	x	x	x	x	0	0	0	0	1	0	0	0	0	1
WaitA	0	0	x	x	0	0	1	0	0	0	0	0	0	1
	0	1	x	x	0	0	1	0	0	0	0	0	1	0
	1	x	x	x	0	0	1	0	0	0	0	0	1	1
PaceA	x	x	x	x	0	1	0	1	0	0	0	0	1	1
ResetTimerV	x	x	x	x	0	1	1	0	0	0	1	1	0	0
WaitV	x	x	0	0	1	0	0	0	0	0	0	1	0	0
	x	x	0	1	1	0	0	0	0	0	0	1	0	1
	x	x	1	x	1	0	0	0	0	0	0	0	0	0
PaceV	x	x	x	x	1	0	1	0	0	1	0	0	0	0
---					1	1	0							
					1	1	1							

Figura 4 – Tabela verdade dos estados.

## 2.5. Lógica Combinacional

Para criar o circuito em portas lógicas faz-se necessário entender como as saídas dependem das estradas. Para isso, são utilizados os minitermos que permitem retirar informações da tabela verdade e representar em forma de equações.

$$n0 = (s2' \cdot s1 \cdot s0) + (sv' \cdot zv' \cdot s2 \cdot s1' \cdot s0') + (sv' \cdot zv \cdot s2 \cdot s1' \cdot s0')$$

$$n1 = (sa' \cdot za \cdot s2' \cdot s1' \cdot s0) + (sa \cdot s2' \cdot s1' \cdot s0) + (s2' \cdot s1 \cdot s0')$$

$$n2 = (s2' \cdot s1' \cdot s0') + (sa' \cdot za' \cdot s2' \cdot s1' \cdot s0) + (sa \cdot s2' \cdot s1' \cdot s0) + (s2' \cdot s1 \cdot s0') + (sv' \cdot zv \cdot s2 \cdot s1' \cdot s0')$$

$$pa = s2' \cdot s1 \cdot s0'$$

$$ta = s2' \cdot s1' \cdot s0'$$

$$pv = s2 \cdot s1' \cdot s0$$

$$tv = s2' \cdot s1 \cdot s0$$

Após isso, foi constatado que as equações de n0, n1 e n2 eram passíveis de simplificações e assim foi feito e com isso obtemos o seguinte resultado:

$$n0 = s2' \cdot s1 \cdot s0 + sv' \cdot s2 \cdot s1' \cdot s0'$$

$$n1 = s2' \cdot (s1' \cdot s0 \cdot (sa + za) + s1 \cdot s0')$$

$$n2 = s0' \cdot s2' + s0' \cdot sv' \cdot zv \cdot s2 \cdot s1' + s2' \cdot s1' \cdot s0 \cdot (sa + za')$$

Após a obtenção das equações simplificadas foi então possível construir o que seria a lógica combinacional juntamente com o registrador de estados e assim formar o bloco de controle como veremos a seguir na figura 5.

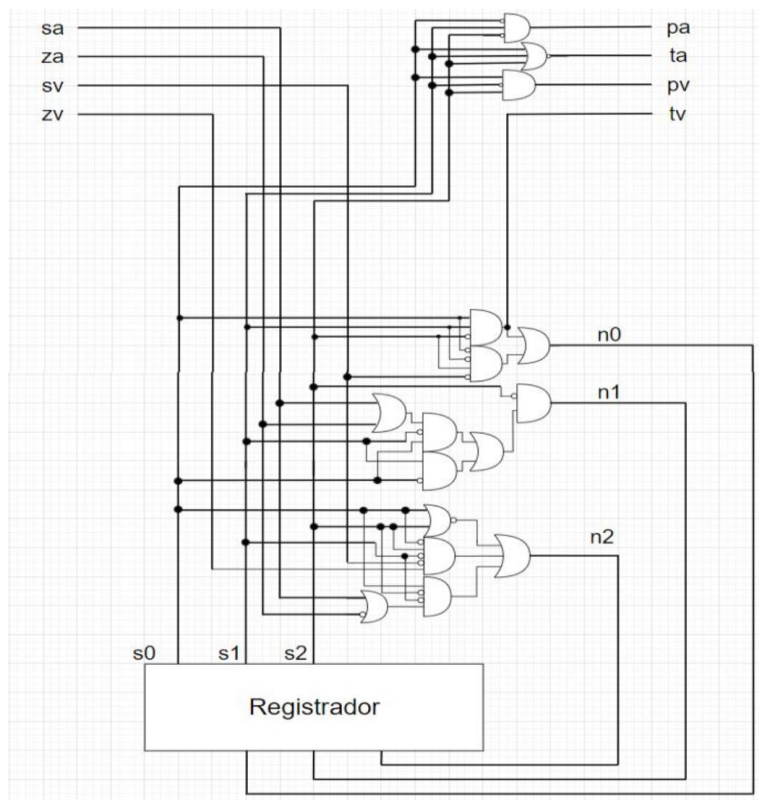


Figura 5 - Circuito em portas lógicas.

### 3. Dificuldades e Soluções

Dentre as dificuldades e dúvidas encontradas durante o desenvolvimento do projeto, as mais importantes foram durante a criação da tabela verdade e a implementação da lógica combinacional. A tabela verdade era muito extensa então, após reuniões em grupo, conseguimos solucionar esse problema percebendo que algumas saídas não dependiam de determinadas entradas, com isso, foi possível diminuir o tamanho da tabela aos poucos.

Além disso, tendo em vista que os projetos anteriores não eram tão extensos, a implementação da lógica combinacional causou dificuldade pelo fato das equações serem maiores do que esperávamos resultando em muitas portas lógicas, mas simplificando-as utilizando as propriedades e postulados vistos em teoria, conseguimos com bastante cautela concluir essa etapa e assim permitir a construção de circuitos lógicos mais compatíveis com a realidade.

### 4. Conclusões

A realização do projeto do marcapasso permitiu a aplicação da teoria vista em sala de aula em um modelo real e utilizado no cenário comercial, para tal foi de grande importância a compreensão dos elementos que compõem o componente, para que assim fosse possível o desenvolvimento. Além disso, foi de suma importância o conhecimento da manipulação de equações lógicas, visto que, foi através das propriedades, postulados e conceitos que permitiram-nos reduzir equações muito extensas a um tamanho menor. Outro ponto importante, foi a aprendizagem da simplificação da tabela verdade, pois a tabela verdade inicial possuía 128 linhas e consequentemente demandava um circuito lógico muito grande e com a observação de que alguns termos não influenciaram na resposta final foi possível obter uma tabela verdade de apenas 12 linhas, o que demonstra que esse procedimento é muito importante, principalmente no cenário comercial no qual as grandes empresas sempre precisam otimizar seus produtos e sistemas.

### 5. Referências bibliográficas

SAIBA o que é marcapasso e quando é recomendado. **WeCor**, 2020. Disponível em: <https://www.wecor.com.br/o-que-e-marcapasso-e-quando-e-recomendado>. Acesso em: 24 nov. 2021.

Vahid, Frank. Digital Design with RTL Design, VHDL, and Verilog Solution Manual. 2º Edição. 2010.