

UNIVERSIDADE DE SÃO PAULO

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO - POLI USP

PCS3645 - Laboratório Digital II



Gabriel Yugo Nascimento Kishida	11257647
Gustavo Azevedo Corrêa	11257693

Turma 1 - Bancada A4

PLANEJAMENTO X

Prof. Edson Midorikawa

Prof. Paulo Cugnasca

Prof. Reginaldo Arakaki

SÃO PAULO - SP

2021

SUMÁRIO

1	INTRODUÇÃO	3
1.1	Visão Geral: Servomotor	3
1.2	Funcionalidade	5
1.3	Não Funcionalidade	5
2	SOLUÇÃO TÉCNICA	6
2.1	Descrição Geral	6
3	ESTRATÉGIA DE MONTAGEM E AFERIÇÃO DE QUALIDADE	7
3.1	Montagem	7
4	RESULTADOS	10
5	APÊNDICE	11
5.1	Código Base	11
5.2	Código da Placa de Controle	12
6	REFERÊNCIAS BIBLIOGRÁFICAS	13

1 INTRODUÇÃO

1.1 Visão Geral: Servomotor

Um servomotor é um motor elétrico cuja principal diferença com outros motores é seu controle de posicionamento. É muito usado em aplicações, como automodelismo e aeromodelismo, onde o controle preciso de posicionamento de partes do equipamento é essencial para seu funcionamento.

Internamente, um servo motor é composto de diversos elementos:

- **Motor** - responsável pelo acionamento das engrenagens e movimentação do eixo principal do servomotor;
- **Engrenagens** - responsáveis pela redução do motor e aumento do torque;
- **Encaixe de saída** - conexão da saída para o controle;
- **Potenciômetro** - usado para monitorar a posição do servomotor;
- **Circuito de controle** - é a base do funcionamento do servomotor: monitora a saída do potenciômetro e a ativação do motor interno para manter a posição determinada pela entrada.

O controle do servomotor é obtido por um sinal de entrada que especifica sua posição. O formato deste sinal segue a modulação PWM (Pulse Width Modulation): onde uma informação é passada através da largura do pulso em nível alto em relação ao período total de oscilação, ou seja através do seu fator de forma (duty cycle).

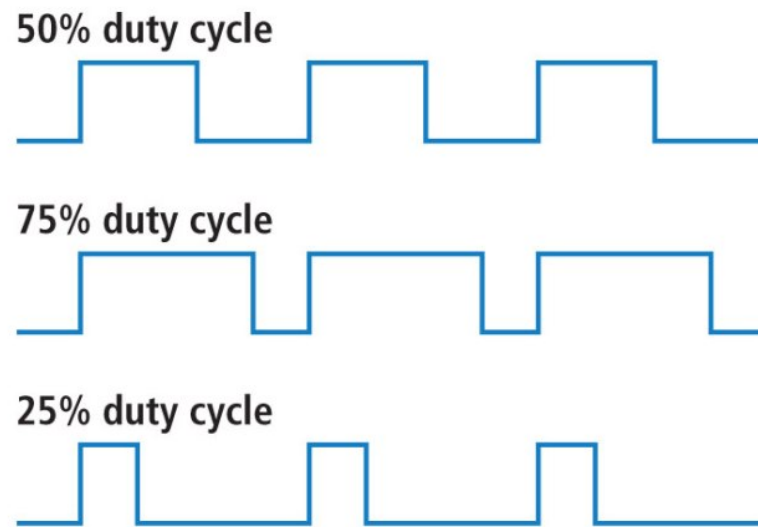


Figura 1: Exemplos de informações passadas por meio de sinais PWM

Em um **servomotor**, a posição é controlada com um sinal com frequência de 50 Hz e pulsos com larguras de 1 até 2ms, conforme a seguinte figura:

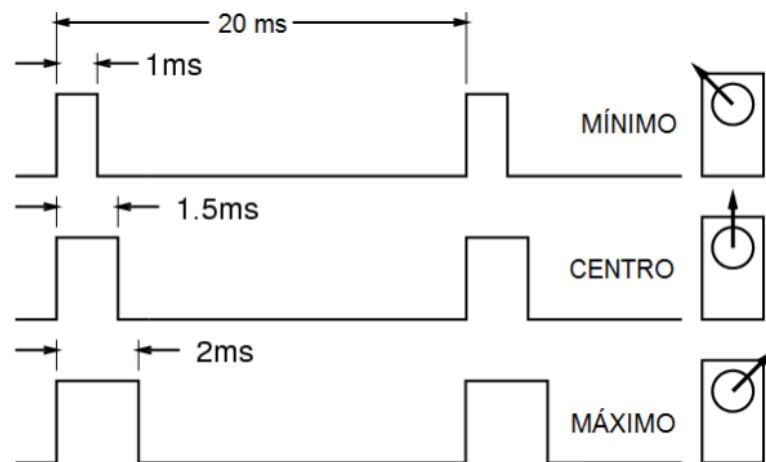


Figura 2: Exemplos de sinais passados para um servomotor (sinal mínimo, central e máximo)

Variando a largura do sinal, o motor pode variar sua posição em até 180° .

Todas as conexões de servomotores apresentam pelo menos três sinais:

- Terminal negativo (ou TERRA);
- Terminal positivo (ou VCC 5V);
- Sinal de controle: sinal PWM com nível de tensão compatível com TTL;

1.2 Funcionalidade

O circuito funciona com um sinal *clock* de 50MHz que a cada período amostra o sinal *posicao* para determinar a posição atual do servomotor e devolver o sinal *pwd* para efetivamente controlá-lo.

1.3 Não Funcionalidade

O controle poderia servir para controlar mais de um servo, gerando diversos sinais *pwd* de saída. Ademais, poderia ser implementado uma barramento para o controle de vários servos em uma ordem de necessidade.

2 SOLUÇÃO TÉCNICA

2.1 Descrição Geral

Neste experimento da disciplina de Laboratório Digital II, nos familiarizaremos com o desenvolvimento de um circuito modulador de sinais PWM, utilizaremos o que foi desenvolvido para controlar um servomotor (posicionando-o a partir do sinal). No entanto, para um desenvolvimento completo e organizado, o projeto deve ser acompanhado dos seguintes cuidados:

Aferir o circuito fornecido pelo docente em *.vhd*, estudando o funcionamento do circuito e possíveis melhorias a serem aplicadas na lógica do sistema. Isto também inclui a documentação do funcionamento no relatório, de forma extensa e compreensiva para auxiliar os estudantes e os leitores a melhor entenderem o desenvolvimento do projeto.

Verificar se o circuito desenvolvido foi aplicado de maneira correta, analisando suas entradas e saídas no arquivo *.vhd*, procurando erros no código e verificando se o mesmo compila no *Quartus Prime*.

Simular o circuito desenvolvido para diversas entradas, estudando as saídas obtidas no *Modelsim* e analisando se os resultados obtidos são condizentes com o que era esperado.

3 ESTRATÉGIA DE MONTAGEM E AFERIÇÃO DE QUALIDADE

3.1 Montagem

Simulando o funcionamento do circuito no software Quartus, é gerado a carta de tempos com as saídas do circuito pelo ModelSim e a partir disso verifica-se as larguras dos pulsos com os valores de entrada.

Chegou-se em larguras de pulsos e período do sinal *PWM* diferentes do esperado, tendo isso em mente modificou-se o arquivo *VHDL* com a descrição do circuito para que o sinal se encaixasse com o esperado de períodos de pulso de 20ms e as larguras de 1, 1,5 e 2 ms.

Isso pode ser verificado no código base (5.1), com os valores de contagem máxima e de duração de pulso que estão errados (considerando valores de *clock* de entrada de 50MHz).

Para obter os valores de contagem corretos, realizaram-se cálculos para obter a largura e período desejado. Com os dados, preencheu-se a tabela proposta no documento de descrição do circuito.

Assim, para obter um pulso de período de 20ms, é necessária uma contagem de 1000000 ciclos de *clock* (de frequência de 50MHz). Desta forma, a contagem máxima é 1000000.

Além disso, a relação entre a largura e quantidade de ciclos de *clock* é:

Largura (ms)	Ciclos de <i>clock</i>
1	50000
1,5	75000
2	100000

Feitos os ajustes necessários testou-se o circuito novamente no Modelsim e foram obtidos os seguintes padrões de onda.

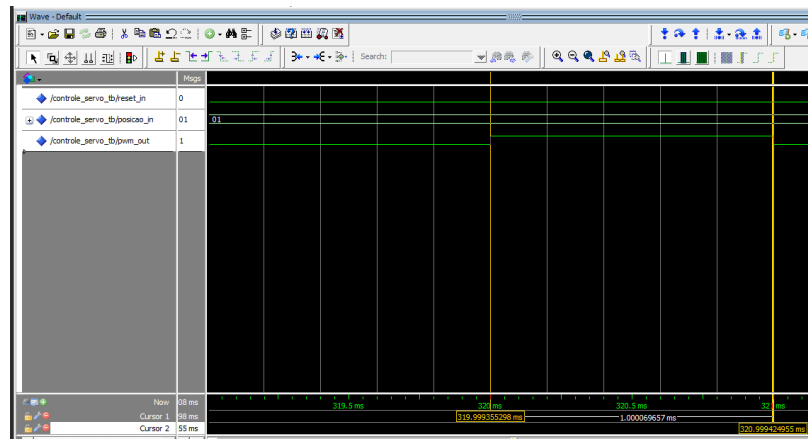


Figura 3: Pulso de posição mínima

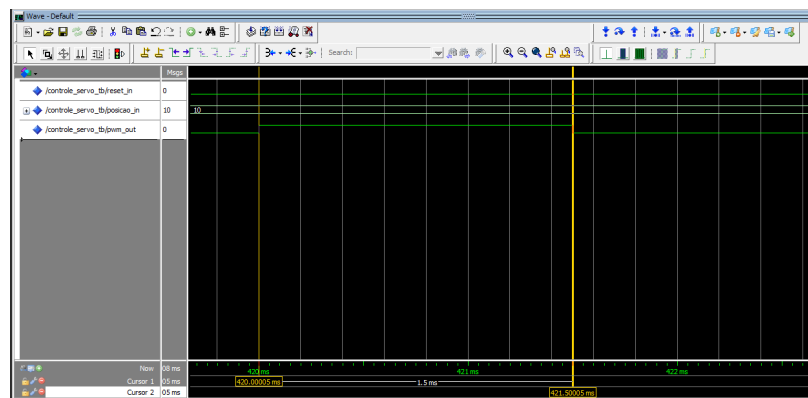


Figura 4: Pulso de posição central

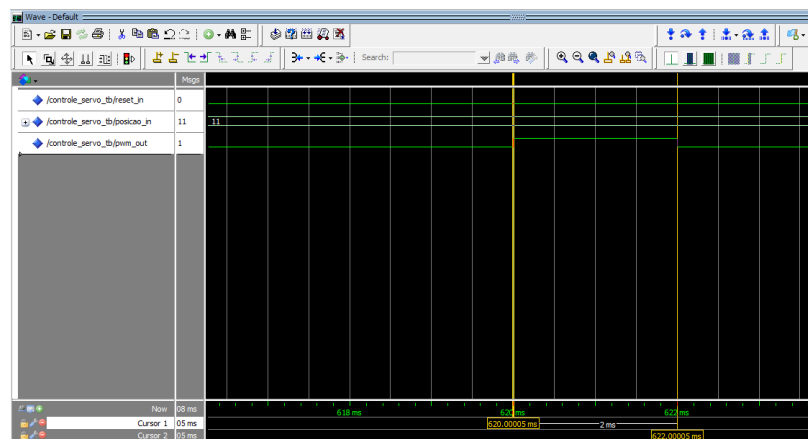


Figura 5: Pulso de posição máxima

Tendo o funcionamento do circuito de acordo com os padrões esperados, segue-se a tabela de pinagem para passar o projeto para a placa FPGA DE0-CV.

Sinal	Ligação na placa FPGA	Pino na FPGA	Analog Discovery
clock	CLK_50	PIN_M9	-
reset	GPIO_0_D27	PIN_P18	DIO0
largura[0]	GPIO_0_D29	PIN_R17	DIO1
largura[1]	GPIO_0_D31	PIN_T20	DIO2
pwm	GPIO_1_D27	PIN_F15	CH1+

Figura 6: Pinagem do circuito para a placa FPGA DE0-CV

Em seguida, durante a aula de laboratório, será aplicado o circuito desenvolvido na placa na bancada disponibilizada pelo docente, e demonstrar-se-á o funcionamento do circuito.

Após a demonstração, será realizado o desafio da aula.

4 RESULTADOS

A ser preenchido após o experimento.

5 APÊNDICE

5.1 Código Base

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity circuito_pwm is
6  port (
7      clock      : in  std_logic;
8      reset      : in  std_logic;
9      largura     : in  std_logic_vector(1 downto 0);
10     pwm         : out std_logic );
11 end circuito_pwm;
12
13 architecture rtl of circuito_pwm is
14     constant CONTAGEMMAXIMA : integer := 1250;
15     signal contagem         : integer range 0 to CONTAGEMMAXIMA-1;
16     signal largura_pwm      : integer range 0 to CONTAGEMMAXIMA-1;
17     signal s_largura        : integer range 0 to CONTAGEMMAXIMA-1;
18 begin
19     process(clock, reset, largura)
20     begin
21         if(reset='1') then
22             contagem <= 0;
23             pwm <= '0';
24             largura_pwm <= s_largura;
25         elsif(rising_edge(clock)) then
26             if(contagem < largura_pwm) then
27                 pwm <= '1';
28             else
29                 pwm <= '0';
30             end if;
31             if(contagem=CONTAGEMMAXIMA-1) then
32                 contagem <= 0;
33                 largura_pwm <= s_largura;
34             else
35                 contagem <= contagem + 1;
36             end if;
37         end if;
38     end process;
39     process(largura)
40     begin
41         case largura is
42             when "01" => s_largura <= 50;  — pulso de 1 us
43             when "10" => s_largura <= 500; — pulso de 10 us
44             when "11" => s_largura <= 1000; — pulso de 20 us
45             when others => s_largura <= 0;  — nulo   saida 0
46         end case;
47     end process;
48
49 end rtl;

```

Pseudocódigo 5.1: Código base fornecido

5.2 Código da Placa de Controle

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity controle_servo is
6  port (
7      clock      : in  std_logic; — 50MHz
8      reset      : in  std_logic;
9      posicao     : in  std_logic_vector(1 downto 0); — 00=0ms, 01=1ms 10=1.5ms 11=2ms
10     pwm        : out std_logic );
11 end controle_servo;
12
13 architecture rtl of controle_servo is
14     constant CONTAGEMMAXIMA : integer := 1000000;
15     signal contagem         : integer range 0 to CONTAGEMMAXIMA-1;
16     signal posicao_pwm       : integer range 0 to CONTAGEMMAXIMA-1;
17     signal s_posicao         : integer range 0 to CONTAGEMMAXIMA-1;
18 begin
19     process(clock, reset, posicao)
20     begin
21         — inicia contagem e posicao
22         if(reset='1') then
23             contagem <= 0;
24             pwm <= '0';
25             posicao_pwm <= s_posicao;
26         elsif(rising_edge(clock)) then
27             — saida
28             if(contagem < posicao_pwm) then
29                 pwm <= '1';
30             else
31                 pwm <= '0';
32             end if;
33             — atualiza contagem e posicao
34             if(contagem=CONTAGEMMAXIMA-1) then
35                 contagem <= 0;
36                 posicao_pwm <= s_posicao;
37             else
38                 contagem <= contagem + 1;
39             end if;
40         end if;
41     end process;
42
43     process(posicao)
44     begin
45         case posicao is
46             when "01" => s_posicao <= 50000; — pulso de 1ms
47             when "10" => s_posicao <= 75000; — pulso de 1.5 ms
48             when "11" => s_posicao <= 100000; — pulso de 2 ms
49             when others => s_posicao <= 0; — nulo saida 0
50         end case;
51     end process;
52
53 end rtl;

```

Pseudocódigo 5.2: Código da Placa de Controle

6 REFERÊNCIAS BIBLIOGRÁFICAS

- (1) ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. Tutorial para criação de circuitos digitais em VHDL no Quartus Prime 16.1. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2017.
- (2) ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. Tutorial para criação de circuitos digitais hierárquicos em VHDL no Quartus Prime 16.1. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2017.
- (3) ALTERA / Intel. DE0-CV User Manual. 2015.
- (4) ALTERA / Intel. Quartus Prime Introduction Using VHDL Designs. 2016.
- (5) ALTERA / Intel. Quartus Prime Introduction to Simulation of VHDL Designs. 2016.
- (6) D'AMORE, R. VHDL - descrição e síntese de circuitos digitais. 2a edição, LTC, 2012.
- (7) WAKERLY, John F. Digital Design Principles & Practices. 4th edition, Prentice Hall, 2006.