Departamento de Engenharia Elétrica e de Computação

SEL 455 – Lab. de Sistemas Digitais
PRÁTICA N º 9
"Dispositivos de Lógica Programável Complexa CPLDs ("Complex Programmable Logic Devices")"

"Controle de um Servomotor"

Objetivo

Desenvolver um projeto para controlar um mini servomotor, mostrado na Figura 1, e sintetizá-lo no FPGA EP4CE30F23C7 da família Cyclone IV-E da Altera, Módulo de desenvolvimento Mercúrio IV –Macnica DHW.

Funcionamento do servomotor

O controle da posição do servomotor (Figura 1) é realizado por uma modulação de largura de pulso, com período total da onda de **20ms**. O tempo em nível lógico alto varia conforme o modelo do servomotor e define o ângulo em que o eixo deve permanecer. Para o modelo utilizado no laboratório, este tempo (em nível lógico alto) da onda deve variar entre **1ms** (0º) e **2ms** (180º). Note que o tempo utilizado está relacionado com o ângulo em que o eixo deve permanecer, e não com uma variação em relação à posição atual, portanto, um tempo de nível lógico alto igual a 1ms sempre levará o eixo do servomotor ao ângulo 0°, independente do ângulo em que ele estava quando recebeu o comando. Os ângulos intermediários estão linearmente relacionados com os tempos intermediários de nível lógico alto da onda, como exemplificado na Figura 2. Os sinais de alimentação e de controle são listados na Tabela I.

Tabela I Sinais de Alimentação e Controle

Conexões		
Função	**Cor do Fio**	
Alimentação (GND)	Marrom	Preto
Alimentação (+5V)	Vermelho	Vermelho
Sinal de Controle	Amarelo	Branco

Atenção: Os limites de curso do eixo do servomotor são garantidos por uma trava mecânica. Provocar esforços excessivos sobre esta trava pode danificar as engrenagens ou o motor do equipamento, portanto **o primeiro passo deve ser estabelecer os limites de variação do pulso, para que os valores extremos sejam respeitados**. Pode-se utilizar o osciloscópio para garantir uma onda adequada antes de aplicá-la ao servomotor. **NÃO MOVIMENTAR MANUALMENTE A POSIÇÂO DO EIXO DO SERVOMOTOR, POIS PODE ACARRETAR DANOS NO MESMO!!!**

Profª Luiza Maria Romeiro Codá

Descrição do Projeto:

O projeto deve controlar o eixo do servomotor utilizando dois *Push Button* para gerar um pulso no pino de saída nomeado PULSO_INV, da seguinte maneira:

- Quando nenhum botão for pressionado o servomotor deve permanecer na sua posição central (90º, Figura 2), sendo gerado nessa situação um pulso de **1,5ms** na saída PULSO**;**

- Quando o *Push Button* nomeado como chave1(*KEY[2])* for pressionado o servomotor deve ser posicionado em 0º (Figura 2), sendo gerado nessa situação um pulso de **1,0ms** na saída PULSO**;**
Observação: as chaves push buttom KEY[0] a KEY[11] quando acionadas geram sinal '1'.

- Quando o *Push Button* nomeado como chave2(*KEY[11])* for pressionado o servomotor deve ser posicionado em 180º (Figura 2), sendo gerado nessa situação um pulso de **2,0ms** na saída PULSO**;**

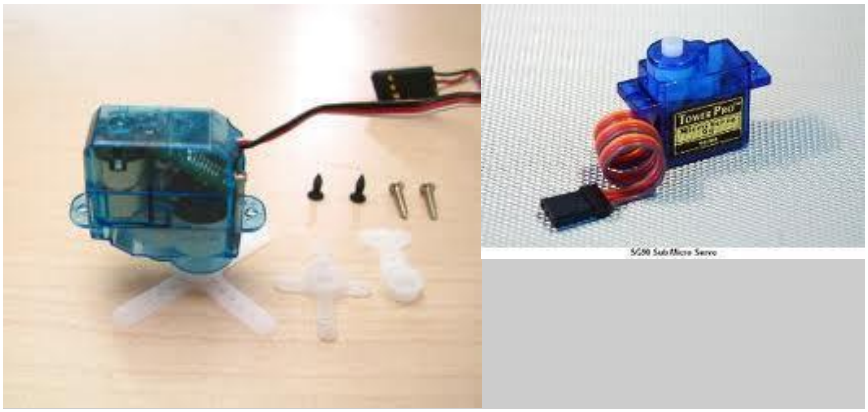


Figura 1. Mini Servomotor.

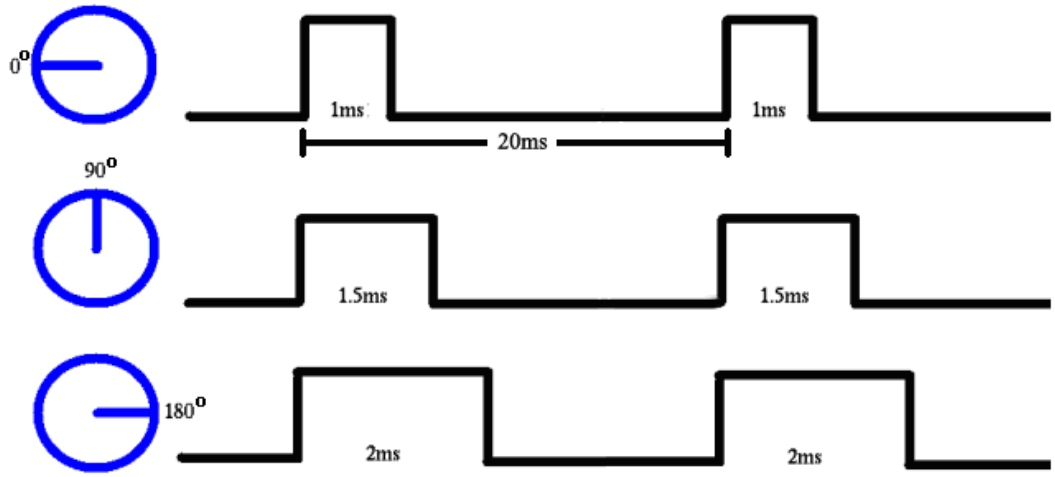


Figura 2. Exemplo de onda usada para posicionamento de um servomotor (http://www.lirtex.com/robotics/servo-motors-information-and-control/**).**

Implementação do projeto

1º passo: gerar o sinal da base de tempo, com período de 20 ms utilizando o clock da placa de 50MHz nomeado como *CLOCK* ;

2º passo: determinar o número de períodos do clock da placa contidos em 1 ms, 1.5 ms e 2 ms. Essas três constantes serão usadas para gerar os pulsos desejados.

3.º passo: com os valores do 2º passo, criar as três constantes no projeto usando o componente **lpm_constant**

Atenção: no componente lpm_constant o número de bits para as três constantes deve ser o mesmo, nivelado pela constante de maior valor(2ms).

4º passo: utilize dois multiplex (componente BUSMUX da biblioteca megafunctions) de duas entradas para selecionar as constantes do 3.º passo. O **primeiro multiplex** usa a chave1 para selecionar uma das constantes relacionadas aos pulsos de 1ms e 1,5ms. O **segundo multiplex** usa a chave2 para selecionar entre a saída do primeiro multiplex ou a constante relacionada ao pulso de 2ms. Observação: as chaves push buttom quando acionadas fornecem nível lógico alto.

5.º passo: utilizando o componente **lpm_counter** crie um contador que conte no modo decrescente desde o valor de uma das constantes até zero usando como sinal de clock do contador a entrada *CLOCK de* 50MHz. No final da contagem deve ser recarregado novamente com o valor da constante presente em sua entrada. A contagem é habilitada pelo sinal PULSO. Para isso no lpm_counter devem ser habilitados os seguintes sinais:

- Entrada de dados **DATA,** que recebe a saída do multiplex2;
- Entrada habilitadora de clock **cnt_en** controlada pela saída Q do FF do 6º passo**;**
- Entrada **aload** para habilitar a recarga da constante;
- Saída **cout** a qual indica o fim da contagem.

A operação do contador é ilustrada pela Figura 3.

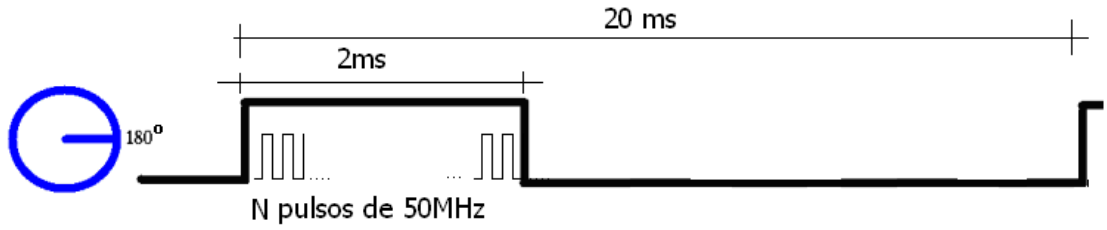


Figura 3 Exemplo de operação do contador decrescente na geração de pulso de 2ms.

6.º passo: Utilize um Flip flop T no modo Toggle para gerar a saída PULSO da seguinte maneira:

- Sua entrada clock deve ser alimentada com o sinal de 20ms (Figura 3) gerado no 1º passo.

- Sua entrada clear deve receber o pulso cout **invertido** para zerar a saída Q deste FF quando a contagem do contador do 5º passo tiver sido concluída. Dessa forma, é gerado um pulso como o ilustrado na Figura 3.
- A saída Q deve ser invertida por uma porta NOT para gerar a saída PULSO_INV.

7º passo: Para atribuir a pinagem no projeto utilize a Tabela II e escolha na barra de ferramenta do *software* Quartus II a opção ASSIGNMENT/PIN PLANNER cuja tela é apresentada na Figura 4. Associe a número da pinagem aos nomes das entradas e saídas conforme a Tabela II. A figura 5 mostra o conector da interface GPIO1 e a tabela com a pinagem do FPGA ligada à da interface GPIO1

Tabela II associação dos pinos do projeto

Nome do pino no projeto	**Nome do pino**	**Número do pino**	**Pino da Interface**
chave1	*KEY[2]*	PIN_U22	
chave2	*KEY[11]*	PIN_Y17	
PULSO_INV	*GPIO1_D[0]*	PIN_E11	**Pin 2**
CLOCK	*CLOCK_50MHz*	PIN_T1	
	GND	**Não associar**	**Pin 12**

Obs: Os botões PB KEY [11..0} quando não pressionados, mantém na entrada do FPGA sinal '0', portanto, quando algum botão for pressionado,

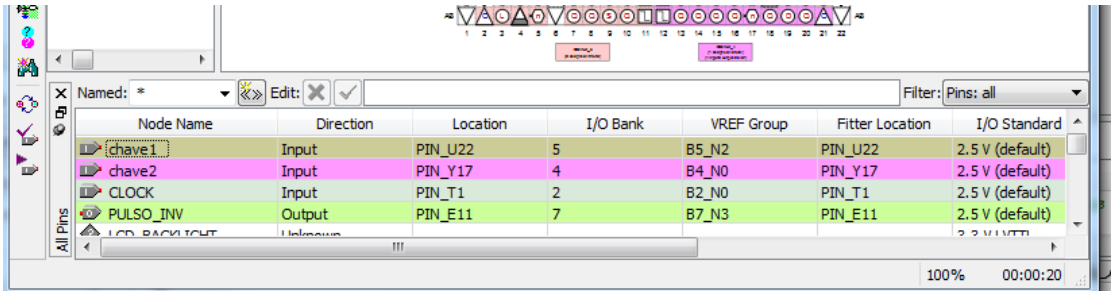
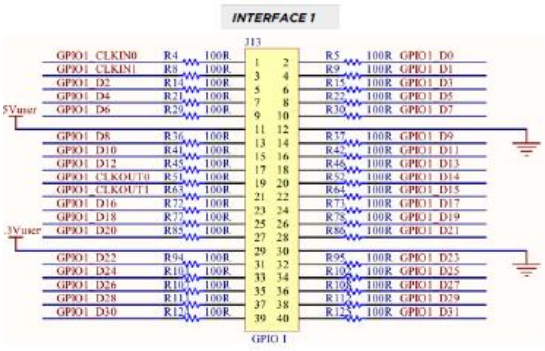


Figura 4 Tela do software Quartus II para associação da pinagem



INTERFACE 1			
GPIO1_CLKIN[0]	A11	1	Entrada de clock 0 do GPIO1
GPIO1_D[0]	E11	2	Pino 0 de dado, entrada ou saída, do GPIO1
GPIO1_CLKIN[1]	B11	3	Entrada de clock 1 do GPIO1
GPIO1_D[1]	F11	4	Pino 1 de dado, entrada ou saída, do GPIO1
GPIO1_D[2]	E12	5	Pino 2 de dado, entrada ou saída, do GPIO1
GPIO1_D[3]	D13	6	Pino 3 de dado, entrada ou saída, do GPIO1
GPIO1_D[4]	E13	7	Pino 4 de dado, entrada ou saída, do GPIO1
GPIO1_D[5]	F13	8	Pino 5 de dado, entrada ou saída, do GPIO1
GPIO1_D[6]	G13	9	Pino 6 de dado, entrada ou saída, do GPIO1
GPIO1_D[7]	F14	10	Pino 7 de dado, entrada ou saída, do GPIO1
VCC	-	11	+5V
GND	-	12	GND
GPIO1_D[8]	E14	13	Pino 8 de dado, entrada ou saída, do GPIO1
GPIO1_D[9]	C15	14	Pino 9 de dado, entrada ou saída, do GPIO1
GPIO1_D[10]	D15	15	Pino 10 de dado, entrada ou saída, do GPIO1
GPIO1_D[11]	E15	16	Pino 11 de dado, entrada ou saída, do GPIO1
GPIO1_D[12]	C17	17	Pino 12 de dado, entrada ou saída, do GPIO1
GPIO1_D[13]	D17	18	Pino 13 de dado, entrada ou saída, do GPIO1
GPIO1_CLKOUT[0]	F15	19	Saída de clock 0 do GPIO1
GPIO1_D[14]	C19	20	Pino 14 de dado, entrada ou saída, do GPIO1
GPIO1_CLKOUT[1]	D19	21	Saída de clock 1 do GPIO1
GPIO1_D[15]	D20	22	Pino 15 de dado, entrada ou saída, do GPIO1

Figura 5 a)Conector GPIO 1 e a descrição de seus pinos
b)Pinagem do FPGA relacionada às interfaces GPIO1

8º passo: Utilize um protoboard para testar o funcionamento do servomotor,. O pulso gerado pelo circuito FPGA (PULSO_INV) não fornece corrente suficiente para controlar o servomotor, que pode requerer até 500mA, sendo necessário o uso de um driver de corrente que eleve a corrente fornecida pelo FPGA(I_{DC} por pino de I/O= 300µ A). Utilize o *driver* de corrente ULN2003 (Figura 5) e interligue a saída PULSO_INV com o pino 1 desse *driver.* A saída correspondente (pino 16) alimenta o pino de entrada do servomotor. Alimente os outros dois pinos do servomotor com a fonte de alimentação em +5V de acordo com a Tabela I. E interligue o pino 6 (GND) da placa mercúrio IV (tabela II) com o GND da fonte de alimentação.

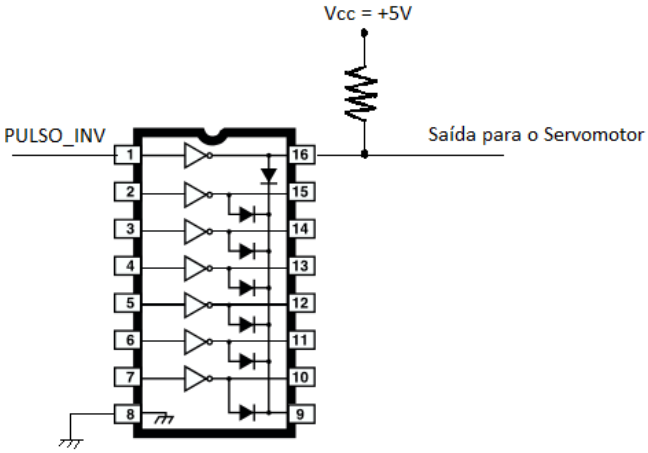


Figura 5 Interligação do driver ULN2003 com a saída PUlSO do projeto

OBS: Como relatório entregue a figura do esquemático do projeto documentada

Informações adicionais sobre o servomotor

O servomotor é uma máquina, mecânica ou eletromecânica, que apresenta movimento proporcional a um comando, em vez de girar ou se mover livremente sem um controle mais efetivo

de posição como a maioria dos motores; servomotores são dispositivos de malha fechada, ou seja: recebem um sinal de controle; verificam a posição atual; atuam no sistema indo para a posição desejada, o diagrama em blocos é mostrado na Figura 4. Em contraste com os motores contínuos que giram indefinidamente, o eixo dos servomotores possui a liberdade de girar cerca de 180 graus, mas são precisos quanto à posição (ref: Wikipédia). Disponíveis em diversos tamanhos, os servomotores são utilizados em muitas aplicações diferentes, como aeromodelismo.

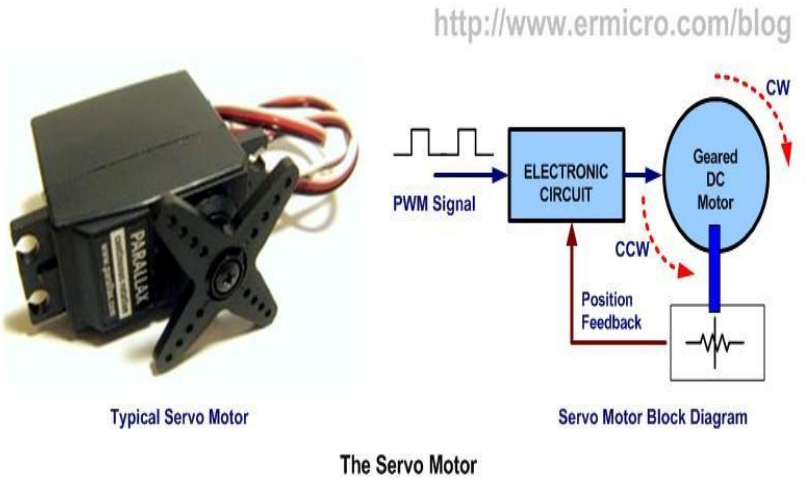


Figura 4. Diagrama em blocos do circuito interno do servomotor (http://www.ermicro.com/blog/).

Especificações dos componentes utilizados

- **Servomotor:**

O servomotor utilizado é o ElectriFly ES50 Nano usado em aeromodelismo
Velocidade: 0,13 seg/60°
Tensão: 4.8V - 6.0V

- **FPGA:**

Nome: EP4CE30F23C7
Categoria: Circuito Integrado(CI)
Família: *Embedded - FPGAs (Field Programmable Gate Array*)
Série: Cyclone IV E
Número de Blocos Lógicos Configuráveis(CLBs ou LABs**):** 1803
Número de bits da RAM: 608256
Número de portas de Entrada/Saída(I/O): 328
Tensão de Alimentação: 1,15 V ~ 1,25 V
Tipo de Montagem: Montagem de superfície

Significado dos Campos do nome do FPGA:

- EP4CE: Cyclone IV –FPGA de baixo custo
- 30 : quantidade de elementos lógicos: 28848 (aproximadamente 30 mil)
- F23: Encapsulamento: Fineline BGA de 23 x 23 mm e 484 pinos
- C: temperatura de operação, 0°C a 85°C
- 7: tempo de atraso da porta: 7ns