

**Escola Politécnica da Universidade de  
São Paulo**



**PCS3645 - Laboratório Digital II**

**Planejamento 5**

**Alunos:** Henrique Ramos de Godoy      NUSP: 13683192

Marcelo Takayama Russo      NUSP: 13680164

Victor Pedreira Santos Pepe      NUSP: 13679565

**Professor:** Edson Toshimi Midorikawa

**Bancada B1**

São Paulo, 29 de Setembro de 2024

# Sumário

<b>Sistema de Sonar.....</b>	<b>3</b>
1. Refatoração dos Códigos.....	3
1.1 Controle Servo.....	3
Tabela 1: Ondas pwm esperadas e obtidas em simulação.....	4
Figura 1: tamanho do conf_periodo de 20ms.....	4
Figura 2: largura do PWM para posição 6 do Servomotor.....	5
1.2 Transmissão Serial 7O1.....	5
Figura 3: funcionamento da Transmissão Serial 7O1.....	6
2. Projeto do Sistema de Sonar.....	6
3.1 - Transmissão serial de 8 dígitos.....	6
Figura 4: Entradas e saídas do componente transmissor_ascii.....	7
Figura 5: RTL Viewer do transmissor_ascii.....	8
Figura 6: Transição de estados da UC do circuito transmissor_ascii.....	8
Figura 7: RTL Viewer do fluxo de dados do transmissor_ascii.....	9
Figura 8: Forma de onda inicial do testbench do transmissor ascii.....	10
Figura 9: Forma de onda final do testbench do transmissor ascii.....	10
2.1 Projeto do Circuito - Unidade de Controle.....	10
figura 10: Diagrama de estados do Sonar.....	11
2.2 Projeto do Circuito - Fluxo de Dados.....	13
figura 11: RTL Viewer do sonar.....	13
2.3 Teste do Sistema Sonar.....	14
Figura 12: Forma de onda inicial do testbench do sonar.....	15
Figura 13: Forma de onda intermediária do sonar (transmissão serial)....	15
Figura 14: Forma de onda inicial do testbench do sonar.....	16
3. Plano de execução.....	17
Figura 15: Relação entre GPIO e pinos da FPGA.....	17
Figura 16: entradas e saídas da GPIO.....	18
Tabela 2: Tabela com pinagem da FPGA e Analog Discovery.....	19
Figura 17: Tabela com pinagem removida do Pin Planner.....	19
Figura 18: Esquema de montagem.....	20
3.1 - 1 <sup>a</sup> Montagem: Servomotor e Sensor HC-SR04.....	20
Figura 19: Pulso de Trigger esperado.....	21
Figura 20: Exemplo de onda PWM.....	22
Figura 21: Exemplo de medida do sensor HC-SR04.....	22
3.2 - 2 <sup>a</sup> Montagem: Sensor HC-SR04 e transmissão Serial.....	23
3.3 Protocolo para a correção de erros.....	23
3.4 - Montagem do sistema completo.....	24
Figura 22: Exemplo de medida do sensor HC-SR04.....	24

## Sistema de Sonar

Como objetivo dessa experiência, espera-se a criação de um Sistema de Sonar, que consiste em um circuito capaz de realizar medições a cada certo período de tempo e que possa variar sua angulação, buscando medir não só apenas o que está a frente de sua posição inicial, mas ao redor. Esse componente consolida de maneira bastante interessante diversos componentes que vêm sendo trabalhados nas experiências anteriores, mas de maneira separada. Até o momento, foram trabalhados principalmente a comunicação serial, o sensor de distância e o servomotor. O Sonar permite usar esses três componentes, permitindo a criação de um circuito complexo.

## 1. Refatoração dos Códigos

### 1.1 Controle Servo

Para a realização do sistema sonar, não houve a necessidade de incluir novos componentes base além dos já vistos anteriormente. Porém para as funcionalidades propostas houve a necessidade de modificar alguns componentes e o controle servo foi um deles. Anteriormente funcionando com 3 posições apenas, para o experimento 5 seriam necessárias 8 posições, com variações de  $20^\circ$  cada, medindo de  $20^\circ$  até  $160^\circ$ . Abaixo, na tabela 1 estão indicados os períodos de onda ALTA no sinal pwm necessários para cada ângulo, em seguida o valor obtido em simulação com o módulo pwm refatorado.

Servo_3	pwm_out Valor Esperado (ns)	pwm_out Valor Obtido (ns)	Ângulo
Teste 1	700 000	700 000	20°
Teste 2	914 000	914 000	40°
Teste 3	1 129 000	1 129 000	60°
Teste 4	1 343 000	1 343 000	80°
Teste 5	1 557 000	1 557 000	100°
Teste 6	1 771 000	1 771 000	120°
Teste 7	1 986 000	1 986 000	140°
Teste 8	2 200 000	2 200 000	160°

Tabela 1: Ondas pwm esperadas e obtidas em simulação

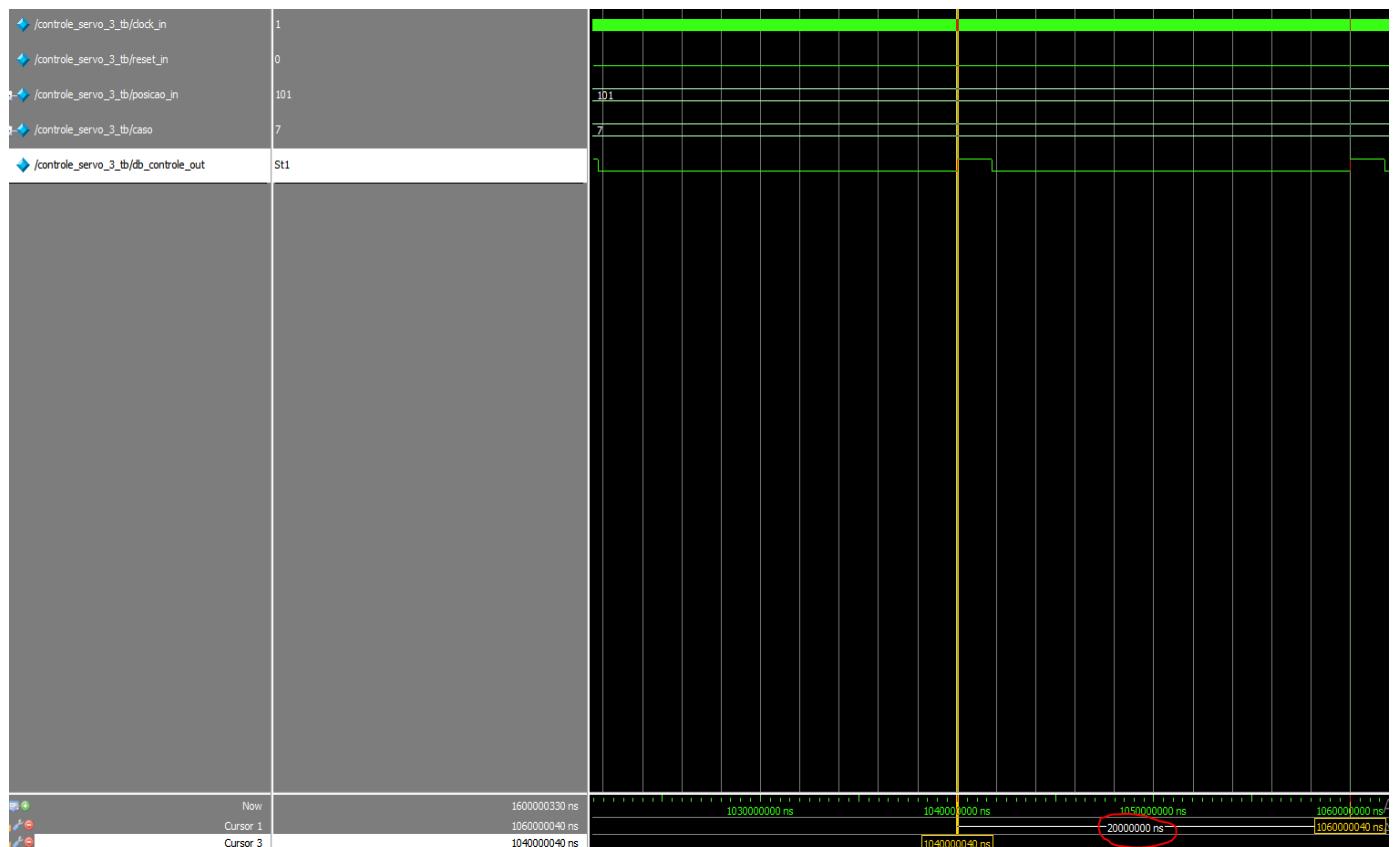


Figura 1: tamanho do conf\_período de 20ms

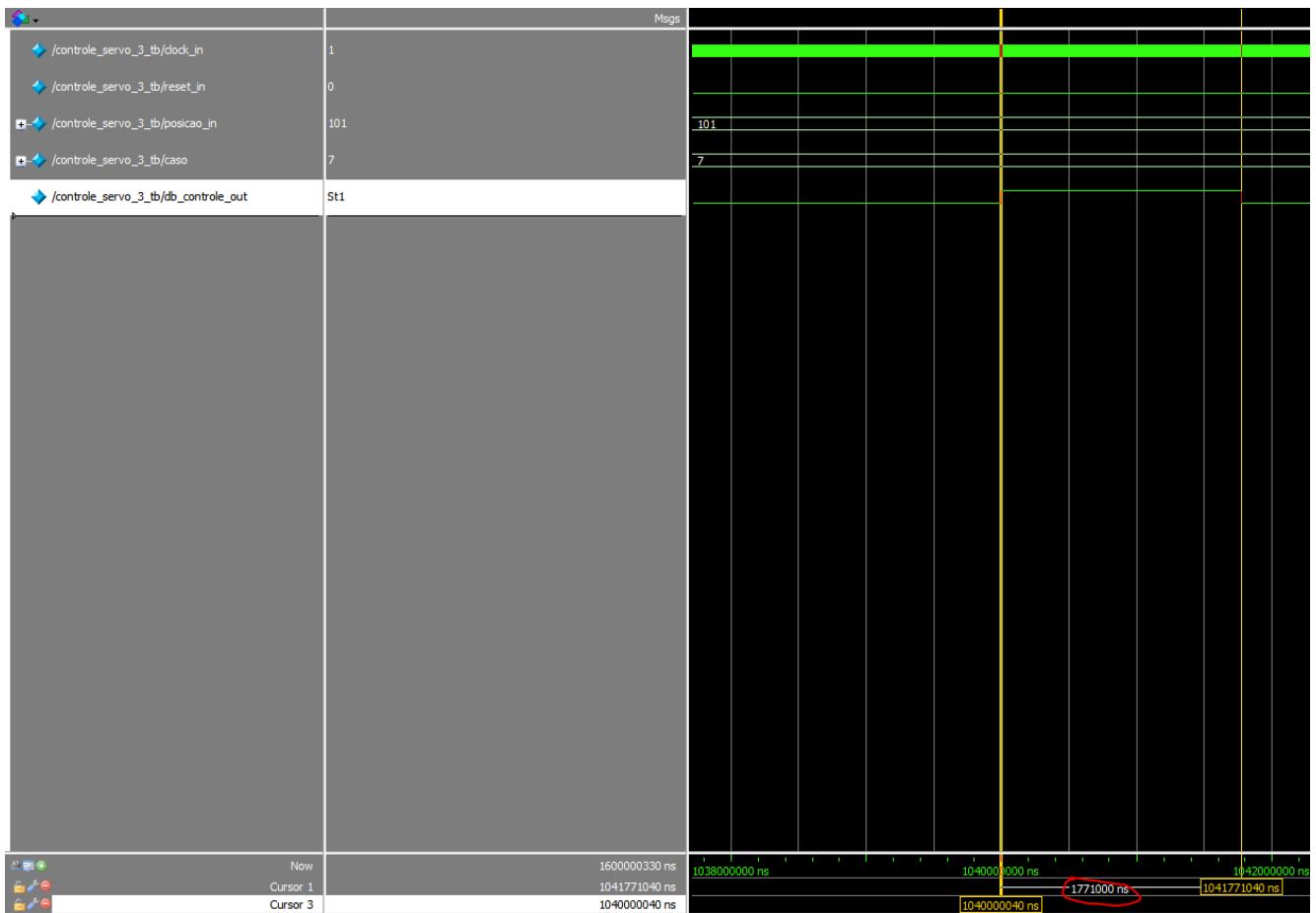


Figura 2: largura do PWM para posição 6 do Servomotor

Pelas duas imagens anteriores, é possível garantir o funcionamento do Controle servo, após sua refatoração. Na primeira imagem, buscou garantir que o período entre pulsos do PWM vale 20ms. Na segunda imagem, buscou-se garantir que o período de alto do PWM vale 1,771ms, que equivale a posição 6 do servomotor - ou seja, 120°. Com esses dois valores, o servomotor consegue interpretar qual deve ser sua posição, uma vez que se baseia no duty cicle, que é a relação entre o período em alto do PWM e o período entre os envios de sinais.

## 1.2 Transmissão Serial 7O1

Para o componente de Transmissão Serial, as mudanças não envolviam adicionar novas funcionalidades, mas remover alguns meios de comunicação externos do circuito, como os display de 7 segmentos e o detector de bordas,

uma vez que não haverá utilização de um botão para ativar esse componente. Dessa forma, foram realizados apenas alguns testes no ModelSim, garantindo que nenhum fio foi deixado em aberto etc.

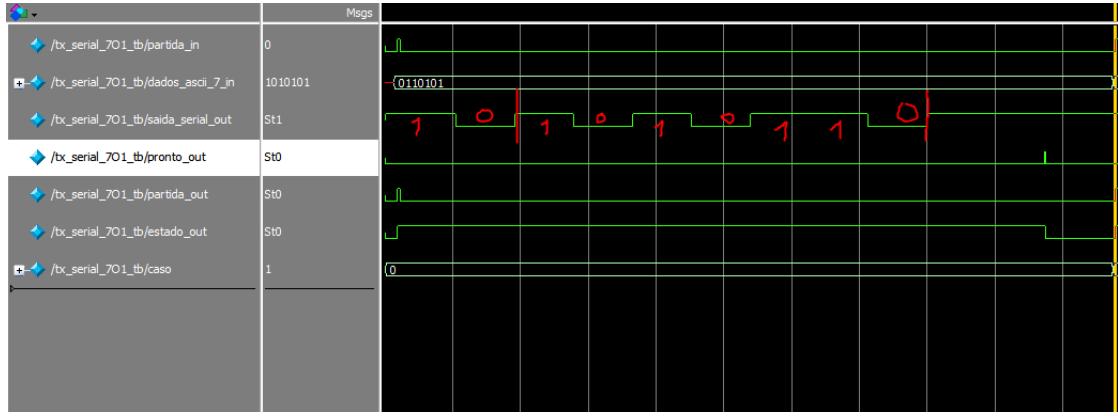


Figura 3: funcionamento da Transmissão Serial 701

Para auxiliar no processo de designação dos pinos, utilizamos as seguintes imagens, que servirão também como Plano de Montagem Experimental.

## 2. Projeto do Sistema de Sonar

Para desenvolver o sistema de sonar, foi preciso tanto utilizar os componentes refatorados quanto criar novos componentes e, por fim, uní-los de forma sistemática para atingir o objetivo desejado. Em cada seção, vamos falar um pouco sobre cada uma das etapas.

### 3.1 - Transmissão serial de 8 dígitos

No sonar, após a medição de distância ser feita, é preciso executar a transmissão de todos os 8 caracteres ASCII: centena, dezena e unidade do ângulo do servomotor, caractere de vírgula (“,”), centena, dezena e unidade da distância e, finalmente, o caractere de hashtag (“#”). Para não poluir a unidade de controle principal com excesso de estados (2 para cada caractere transmitido, como feito na experiência anterior), o procedimento de transmissão foi

encapsulado num novo componente *transmissor\_ascii*. Ele possui as seguintes entradas e saídas:

```
module transmissor_ascii
(
    input clock,
    input reset,
    input iniciar,
    input [6:0] centena_angulo,
    input [6:0] dezena_angulo,
    input [6:0] unidade_angulo,
    input [6:0] caractere_final_angulo,
    input [6:0] centena_distancia,
    input [6:0] dezena_distancia,
    input [6:0] unidade_distancia,
    input [6:0] caractere_final_distancia,
    output pronto,
    output dado_serial
);

```

Figura 4: Entradas e saídas do componente *transmissor\_ascii*

Nesse novo componente, a entrada ASCII de 7 bits de cada caractere deve ser inserida na respectiva entrada e, em seguida, acionar o sinal *iniciar*. A partir desse momento, ocorrerá a transmissão serial por meio da saída *dado\_serial*, utilizando a taxa de bauds correta (115.200) com o circuito de transmissão 7O1, já estudado anteriormente. A transmissão dos caracteres ASCII segue a mesma ordem das entradas: *centena\_angulo*, *dezena\_angulo*, *unidade\_angulo*, e assim por diante. Esse componente foi segmentado em unidade de controle e fluxo de dados. O circuito completo se encontra na figura abaixo.

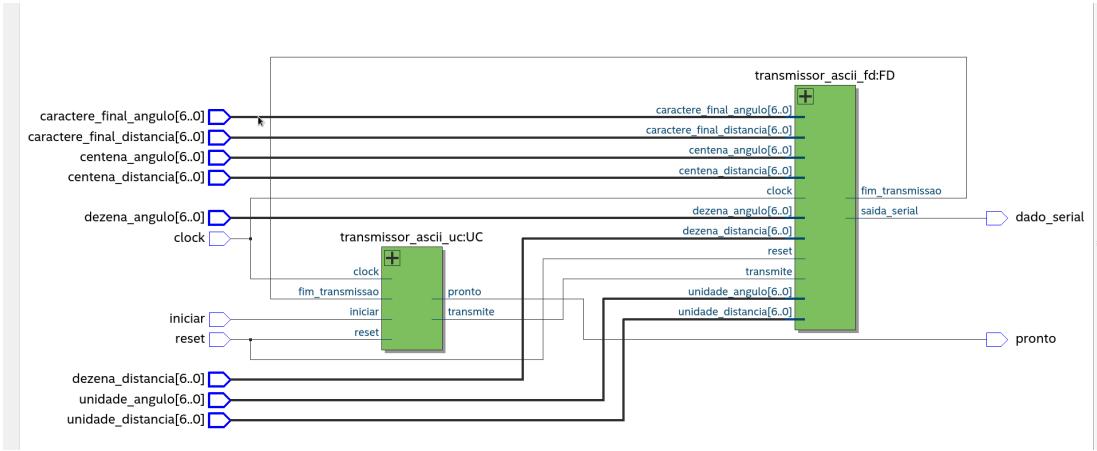


Figura 5: RTL Viewer do transmissor\_ascii

A UC é de fácil compreensão, como mostrado no diagrama de transição de estados a seguir:

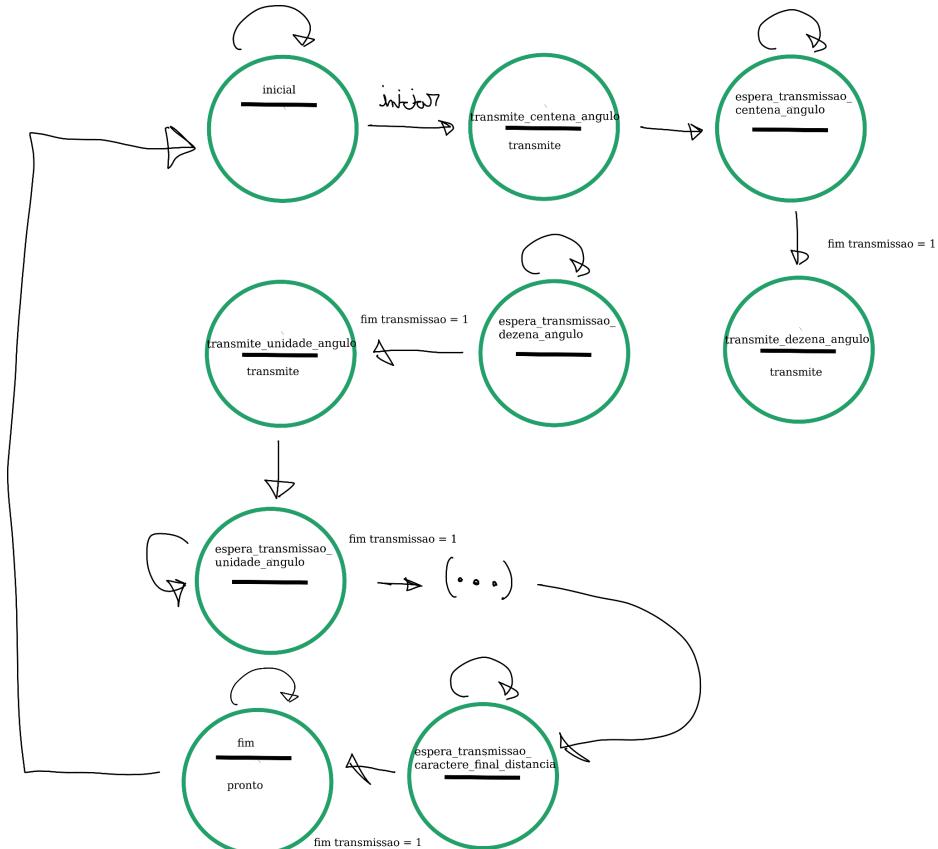


Figura 6: Transição de estados da UC do circuito transmissor\_ascii

Por fim, o fluxo de dados, obtido com o RTL Viewer, é escrito como na figura abaixo.

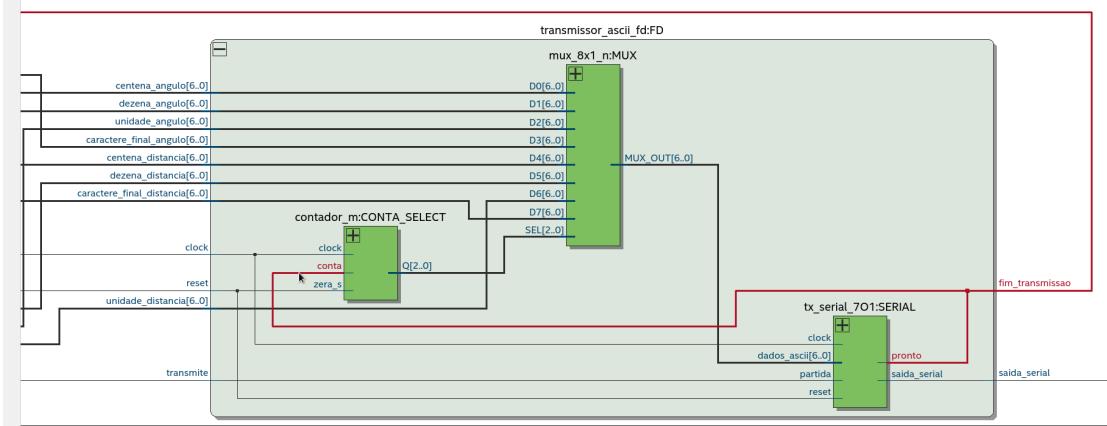


Figura 7: RTL Viewer do fluxo de dados do transmissor\_ascii

No fluxo de dados, temos 3 componentes: um multiplexador com 8 entradas de 7 bits, um *contador\_m* de 3 bits que vai de 0 a 7 e, por fim, um circuito *tx\_serial\_701*. Inicialmente, o contador está zerado e, portanto, a primeira entrada do MUX é selecionada. Esse valor será usado como dado ASCII a ser transmitido pelo serial 701. O sinal *transmite* será enviado pela UC e irá ativar o processo de transmissão. Quando este chegar ao fim, o *contador\_m* será incrementado (observe o fio destacado de vermelho). Dessa forma, os caracteres serão transmitidos 1 por 1, similarmente à trena digital desenvolvida na experiência anterior. Foi feito um testbench, no qual o ângulo vale 147 graus e a distância 297 cm. Foram obtidas as seguintes formas de onda (inicial e final).

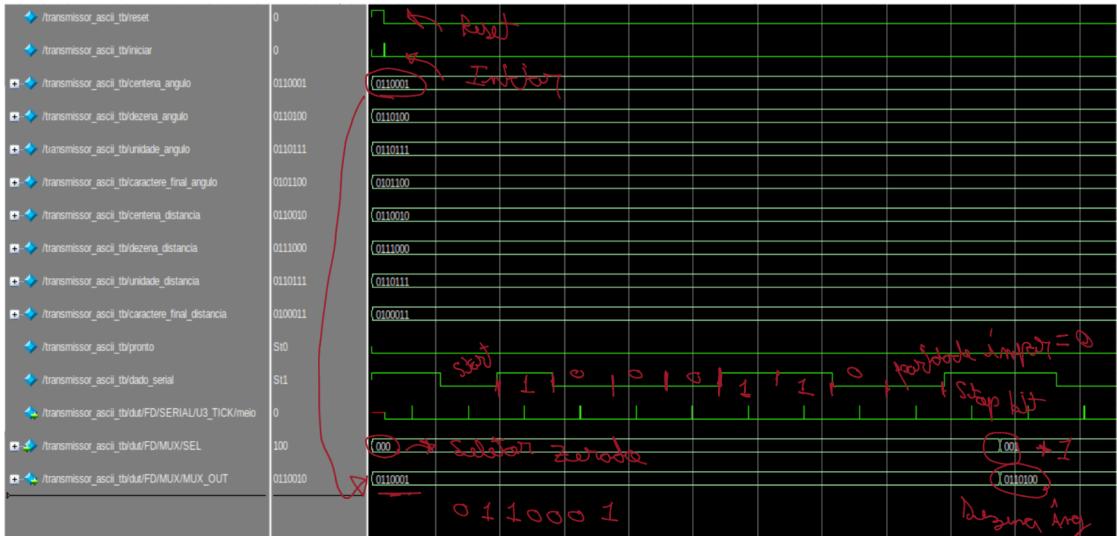


Figura 8: Forma de onda inicial do testbench do transmissor ascii

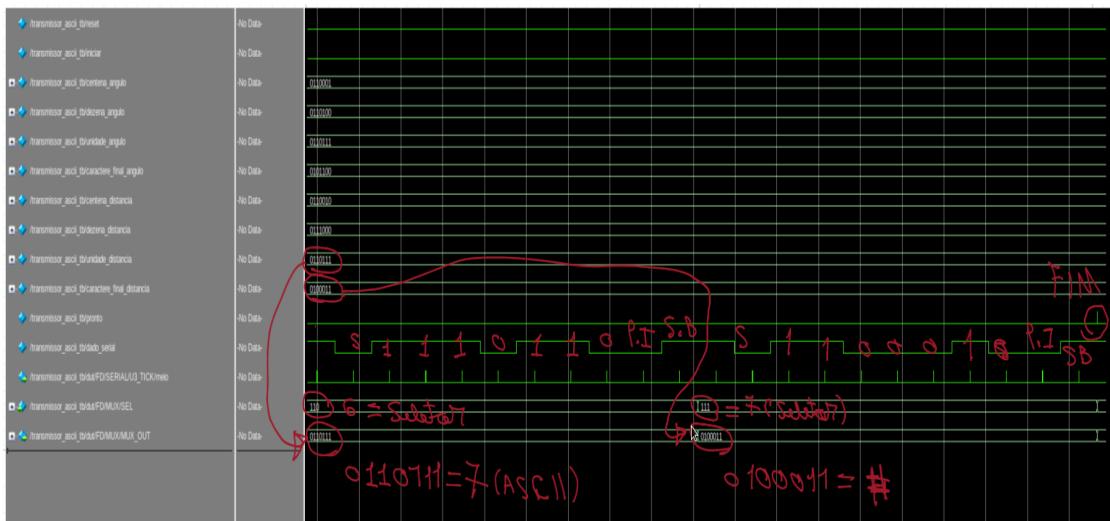


Figura 9: Forma de onda final do testbench do transmissor ascii

Com esses resultados, foi possível validar o funcionamento correto do circuito de transmissão de 8 caracteres, possibilitando seu uso no fluxo de dados principal.

## 2.1 Projeto do Circuito - Unidade de Controle

Tendo o novo Componente bem definido e testado é possível entrar a fundo em como foram realizadas as transições de Estado do Sonar. Para isso, vamos partir do diagrama de Estados, gerado pelo State Machine Viewer e

redesenhado, permitindo maior legibilidade dos estados.

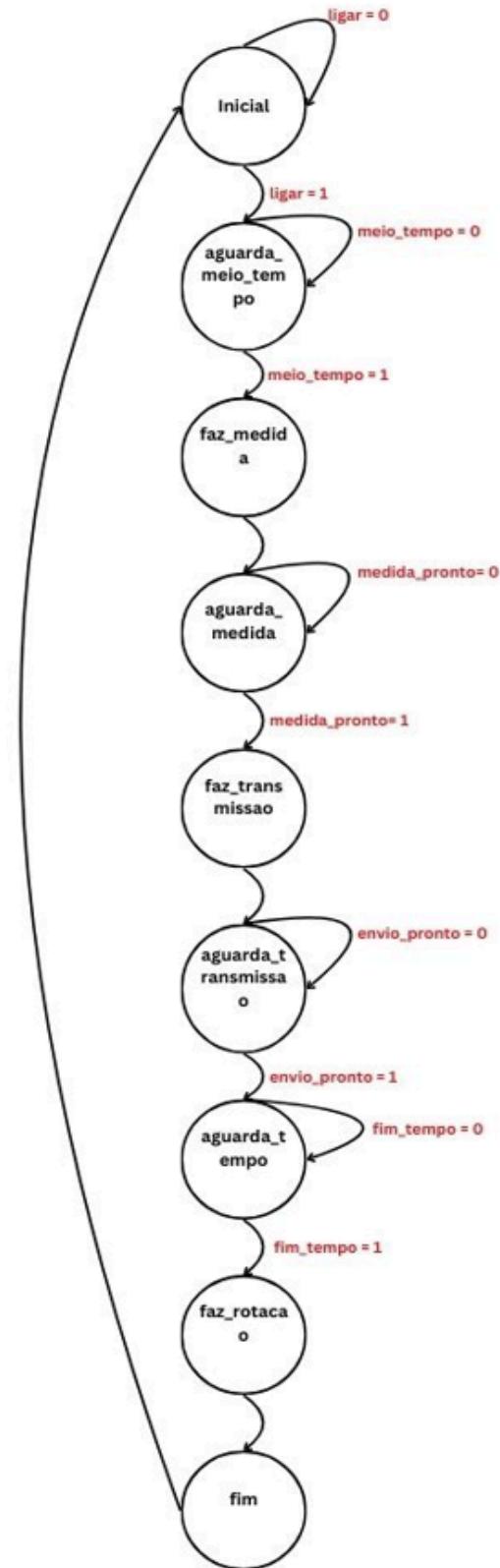


figura 10: Diagrama de estados do Sonar

Pelo diagrama de Estados, é possível entender a lógica que foi desenvolvida. Cabe ressaltar que em todos os estados, caso a chave ligar seja desativada, o circuito retorna ao estado **inicial**. Isso não foi colocado para não poluir o desenho, mas era um requisito do circuito.

**inicial**: estado default do circuito. Sempre que a chave *ligar* estiver inativa, retorna-se para esse estado.

**aguarda\_meio\_tempo**: estado que aguarda 1 segundo (metade do tempo de espera solicitado na experiência). Como o servomotor não possui uma forma de indicar que terminou de realizar sua rotação fisicamente, foi necessário encontrar uma forma de garantir que a medição não fosse realizada no meio da rotação do servo. Por isso criou-se esse estado, que permite que o servo realize sua rotação sem que haja conflito com a medição. Esse estado passa quando a FD envia o sinal de *meio\_tempo*, indicando metade da contagem de 2s.

**faz\_medida**: estado que dá início à medição da distância do objeto. Envia para a FD o sinal em ativo *medir*, que ativa o componente do sensor.

**aguarda\_medida**: estado intermediário que aguarda até que a medição seja realizada e que garante que não haja conflito entre a transmissão e a medição. Passa-se desse estado quando o FD envia o sinal *medida\_pronto* em ativo.

**faz\_transmissao**: estado que dá início à transmissão da mensagem. Envia para a FD o sinal em ativo *transmistar*, que ativa o componente do *transmissao\_ascii*.

**aguarda\_transmissao**: pode-se dizer que este estado possui diversos subestados, melhor especificados na *transmissao\_ascii*. Durante este estado, toda a lógica de transmissão é realizada dentro do componente criado, que - ao terminar de transmitir os dados - envia o sinal *envio\_pronto* em ativo para a UC.

**aguarda\_tempo**: sinal que aguarda até que o contador de 2s termine sua contagem, indicando que é possível iniciar um novo ciclo. Para isso, envia o sinal *fim\_pronto* em ativo para a UC, passando para o estado seguinte.

**faz\_rotacao:** na lógica que foi implementada, a rotação é a última etapa a ser realizada, de forma que não haja conflitos com as outras operações, principalmente, a medição. Assim, nesse estado, é enviado em ativo o sinal *girar*, indicando ao servomotor que pode ser realizada a mudança de posição.

## 2.2 Projeto do Circuito - Fluxo de Dados

Com a UC definida, parte-se para o Fluxo de Dados. Da mesma forma que para a UC, vamos partir do diagrama de projeto, gerado pelo RTL Viewer.

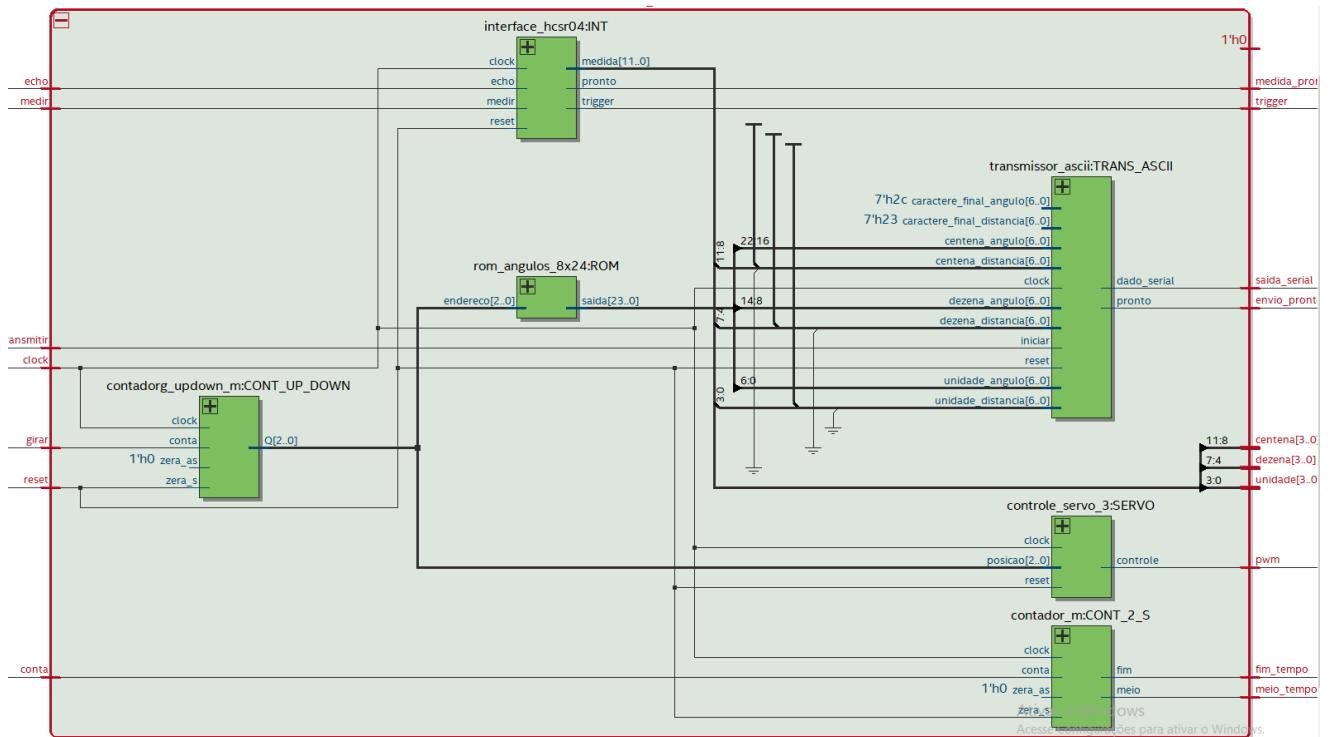


figura II: RTL Viewer do sonar

Como se pode ver, o Fluxo de Dados reúne todos os componentes independentes e realiza sua conversação, quando necessário. Não será abordado o funcionamento de cada componente em específico, uma vez que já foram detalhados nos experimentos anteriores. No entanto, há dois componentes novos que foram passados para essa experiência e que serão detalhados.

*Contador\_m\_updown*: esse componente foi fornecido para esse experimento. Explorando possibilidades de seu uso, chegou-se a conclusão que

ele será usado como entrada do Servomotor, uma vez que para o funcionamento de um radar, é esperado que ele fique girando em meia-lua, em uma sequência de posição  $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 6 \rightarrow 7 \rightarrow 6 \rightarrow \dots \rightarrow 1 \rightarrow 0$ . Para isso, o contador realiza o incremento e decremento contínuo, permitindo a criação desse movimento por parte do radar.

*rom\_angulo\_8x24*: esse outro componente fornecido tem função de auxiliar na transmissão serial dos dados. Dado que os dados devem ser transmitidos em ASCII e o formato pedido de dado é **distância,ângulo#**, é necessário converter os ângulos medidos, ou seja, as posições para caracteres que serão interpretados. Dessa forma, a ROM recebe como entrada a posição do servo e tem como saída o ângulo, já convertido em BCD para ser utilizado na Transmissão serial, pelo componente criado *transmissor\_ascii*.

Por último, o *contador\_m* de 2 segundos foi adicionado para contabilizar o tempo necessário entre as contagens, como pedido no enunciado da experiência.

## 2.3 Teste do Sistema Sonar

Para validar o funcionamento correto de todo o circuito desenvolvido, foi elaborado um testbench. Nele, foi utilizado 100cm como distância simulada. Além disso, como o tempo de espera entre duas rotações é de 2 segundos, seria necessário aguardar 100 milhões de ciclos de clock - com este à 50MHz - o que letificava muito a simulação, cuja escala é de nanossegundos (note que 2 segundos equivalem a 2 bilhões de nanossegundos). Por esse motivo, apenas para facilitar o testbench, o tempo entre 2 rotações foi reduzido de 2 segundos para 40 milissegundos - correspondente a 2 ciclos do servomotor (20ms). Inicialmente, foi resetado o circuito e, portanto, o servomotor deve estar posicionado à 20 graus conforme a ROM de ângulos. Para isso, o sinal *pwm* deve durar 700.000 ns. A primeira etapa a cada ciclo de transição de estados da UC é aguardar metade do tempo entre cada rotação ( $\frac{40ms}{2} = 20ms$ ). Em

seguida, deve ser feita a medida da distância, e - portanto - o sinal *trigger* deve estar ativo por 10 us. Observe a forma de onda inicial.

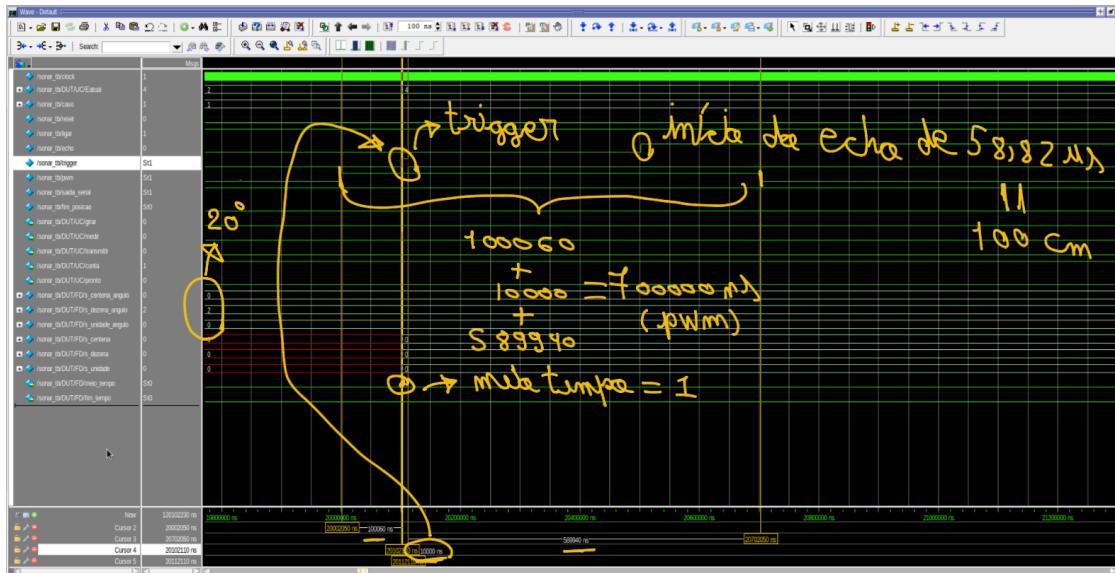


Figura 12: Forma de onda inicial do testbench do sonar

Após o sinal de *echo* com duração de 58,82us (configurado no testbench), foi calculado uma distância de 100 cm, tal como esperado. Em seguida, iniciou-se a transmissão dos 8 dígitos, começando com a centena de ângulos. Observe a forma de onda abaixo.

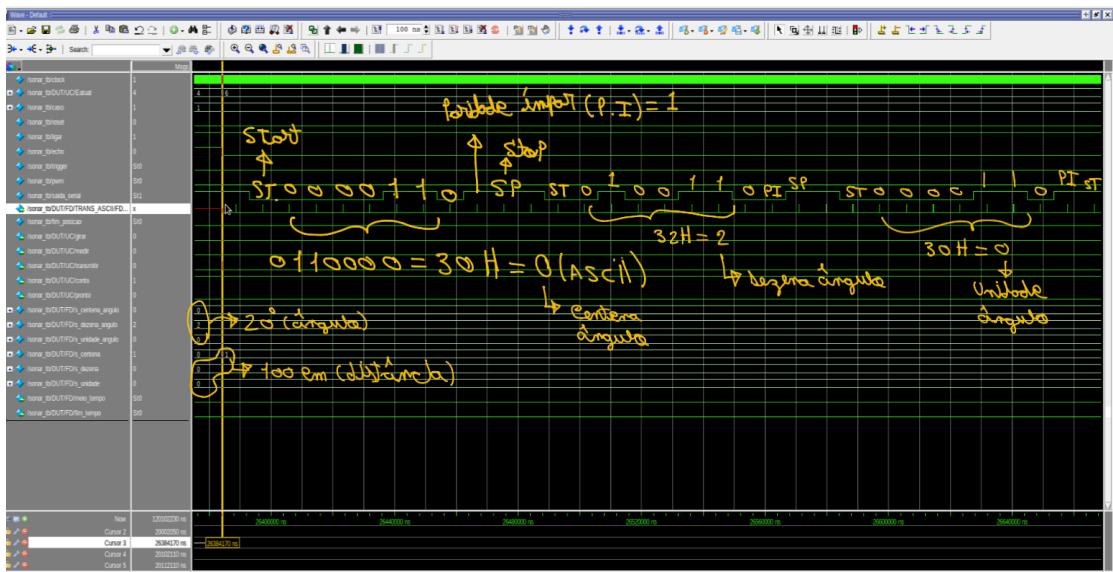


Figura 13: Forma de onda intermediária do sonar (transmissão serial)

Aqui, é possível observar que a transmissão serial ocorre da forma esperada, como visto no estudo do módulo *transmissor\_ascii*, por ser ele o responsável por essa etapa. Por fim, ao final do período que o servo se mantém num mesmo ângulo (40 ms), obteve-se a forma de onda da figura abaixo.

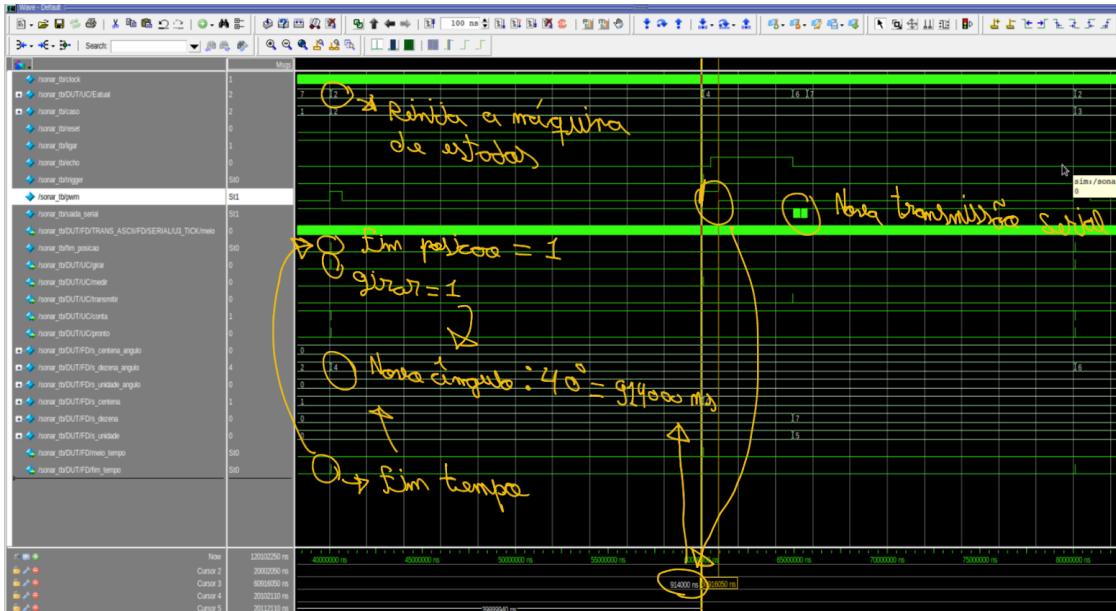


Figura 14: Forma de onda inicial do testbench do sonar

Nessa etapa final, observe que quando o tempo de 40ms chega ao fim (*fim\_tempo=1*), o sinal *fim\_posicao* vale 1 e a máquina de estados reinicia o ciclo visto nas etapas anteriores. Note também que, quando o sinal *girar=1*, o ângulo muda de 20 para 40, como esperado. Nesse novo ângulo, o período do sinal *pwm* deve durar 914000 ns, que é exatamente o que ocorre, como indicado na figura. De forma distante, é possível observar também uma nova transmissão serial ocorrendo quando o sinal de *echo* se encerra, com a distância medida de 75 cm. Assim, valida-se que o sistema de arredondamento também funciona como esperado.

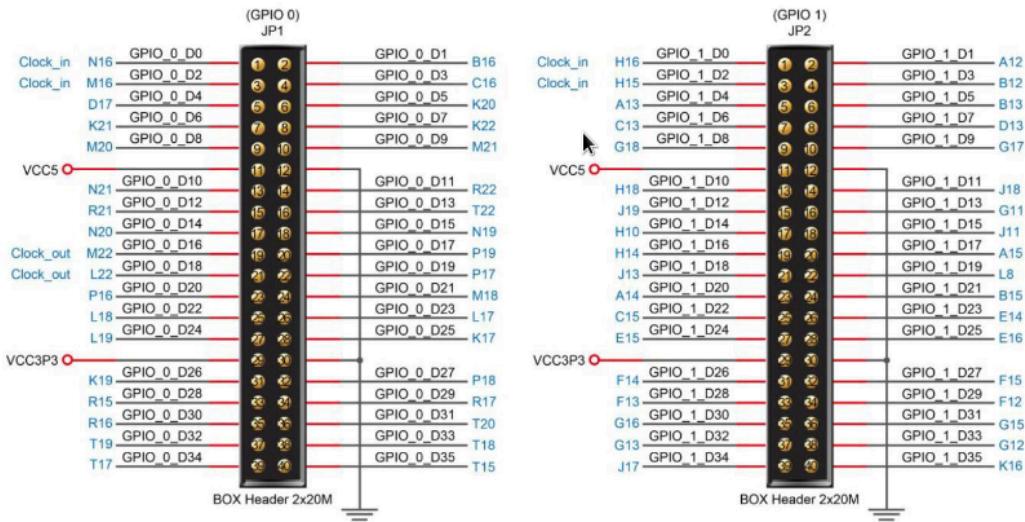
Com esse testbench, foi possível obter resultados satisfatórios para implementá-lo no laboratório digital na próxima experiência, visto que não só o funcionamento do servomotor, da transmissão serial e do sensor ultrassônico estão corretos, mas a integração entre eles - resultando no sonar - também está.

### 3. Plano de execução

Com o testbench elaborado e testado, podemos partir para o Plano de Execução que será realizado presencialmente no laboratório. Antes de elaborar o plano, é importante deixar algumas fotos que sempre auxiliam na montagem física do circuito.

**Table 3-6 Pin Assignment of Clock Inputs**

Signal Name	FPGA Pin No.	Description
<b>CLOCK_50</b>	<b>PIN_M9</b>	<b>50 MHz clock input(Bank 3B)</b>
<b>CLOCK2_50</b>	<b>PIN_H13</b>	<b>50 MHz clock input(Bank 7A)</b>
<b>CLOCK3_50</b>	<b>PIN_E10</b>	<b>50 MHz clock input(Bank 8A)</b>
<b>CLOCK4_50</b>	<b>PIN_V15</b>	<b>50 MHz clock input(Bank 4A)</b>



**Figure 3-12 I/O distribution of the expansion headers**

*Figura 15: Relação entre GPIO e pinos da FPGA*

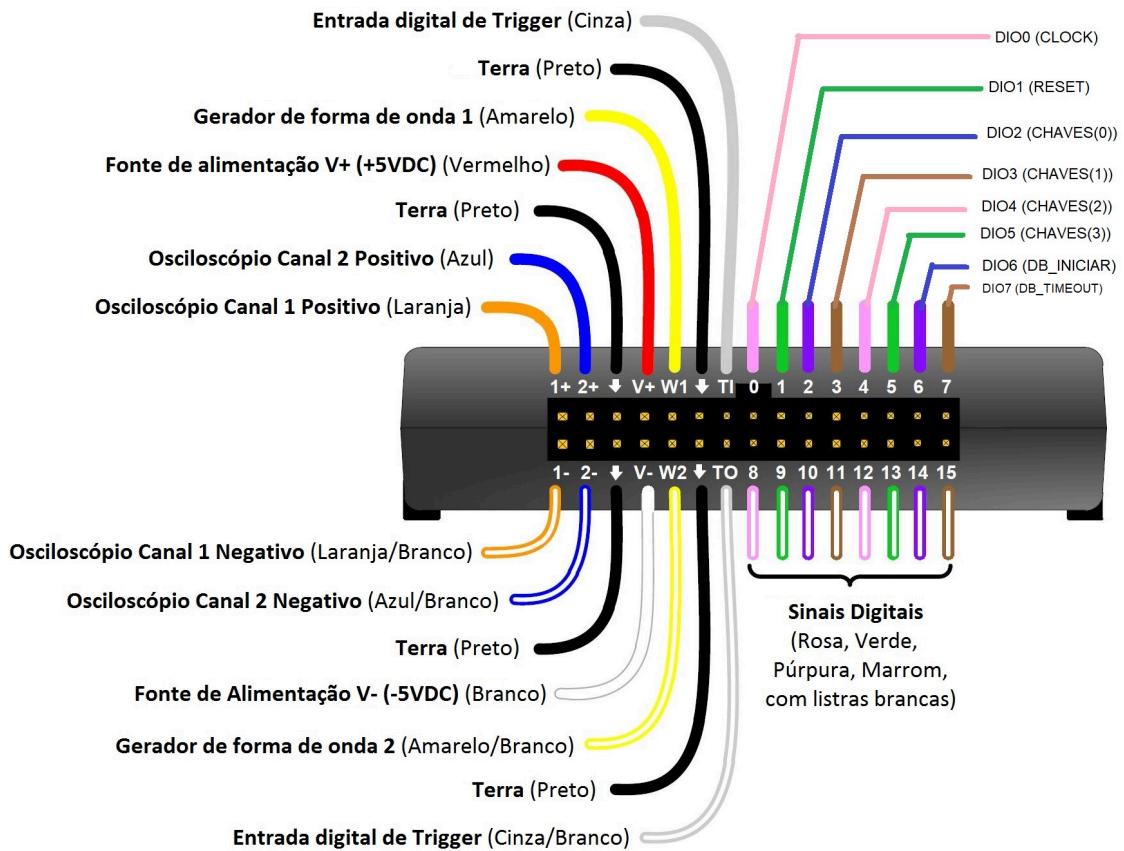


Figura 16: entradas e saídas da GPIO

Para esse circuito, foi realizada a seguinte pinagem:

Sinal	Pino na DE0-CV	Pino na FPGA	Analog Discovery
clock	CLOCK_50	M9	
reset	chave SW0	U13	
ligar	chave SW1	V13	
trigger	GPIO_1_D3	B12	
echo	GPIO_1_D5	B13	
pwm	GPIO_1_D1	A12	
saida_serial	GPIO_0_D1	B15	DIO7
fim_posicao	led LEDR0	AA2	
db_trigger	GPIO_1_D32	G13	CH1+
db_pwm	GPIO_1_D33	G12	CH2+
db_echo	GPIO_1_D34	J17	
db_saida_serial	GPIO_1_D35	K16	

Tabela 2: Tabela com pinagem da FPGA e Analog Discovery

Node Name	Direction	Location	I/O Bank
in clock	Input	PIN_M9	3B
out db_echo	Output	PIN_J17	7A
out db_pwm	Output	PIN_G12	7A
out db_saida_serial	Output	PIN_K16	7A
out db_trigger	Output	PIN_G13	7A
in echo	Input	PIN_B13	7A
out fim_posicao	Output	PIN_AA2	2A
in ligar	Input	PIN_V13	4A
out pwm	Output	PIN_A12	7A
in reset	Input	PIN_U13	4A
out saida_serial	Output	PIN_B15	7A
out trigger	Output	PIN_B12	7A

Figura 17: Tabela com pinagem removida do Pin Planner

Assim como realizado em experiências anteriores, para a montagem do circuito do sonar, o grupo irá realizar a montagem em partes, a fim de verificar o funcionamento gradual dos componentes facilitando a depuração em caso de erro. Para isso será utilizado o roteiro sugerido pelo professor onde inicialmente será testado a integração do servomotor com o sensor HC, em seguida, o sensor HC com transmissão serial, e por fim o sistema completo. A seguir, a figura 12 ilustra a montagem de todos os componentes e suas conexões. Esse esquema será seguido durante toda a experiência.

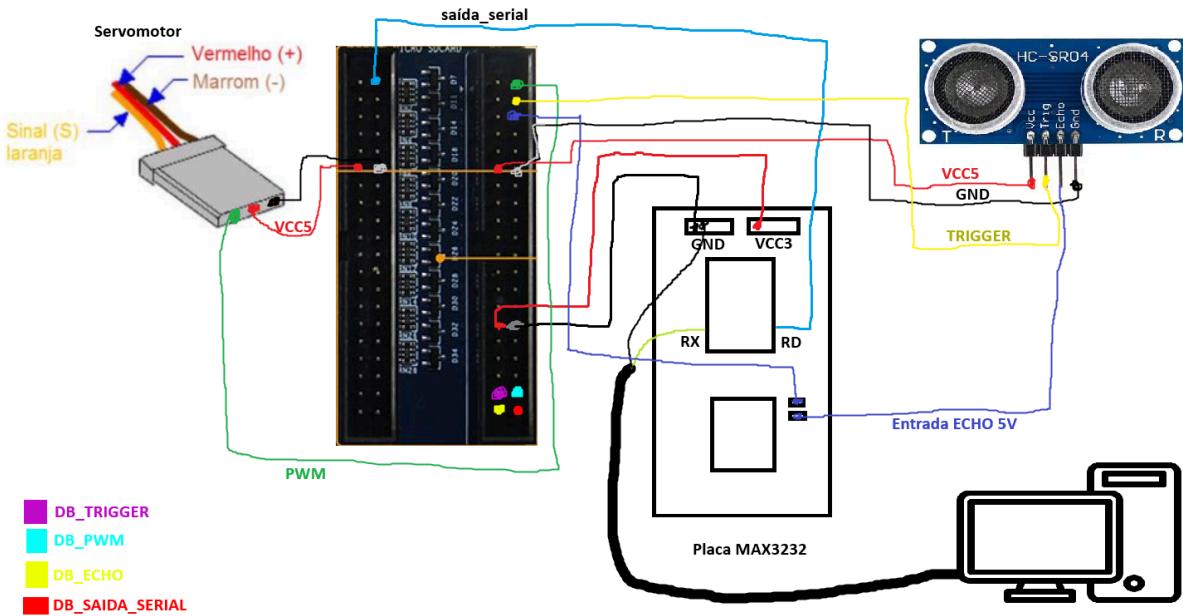


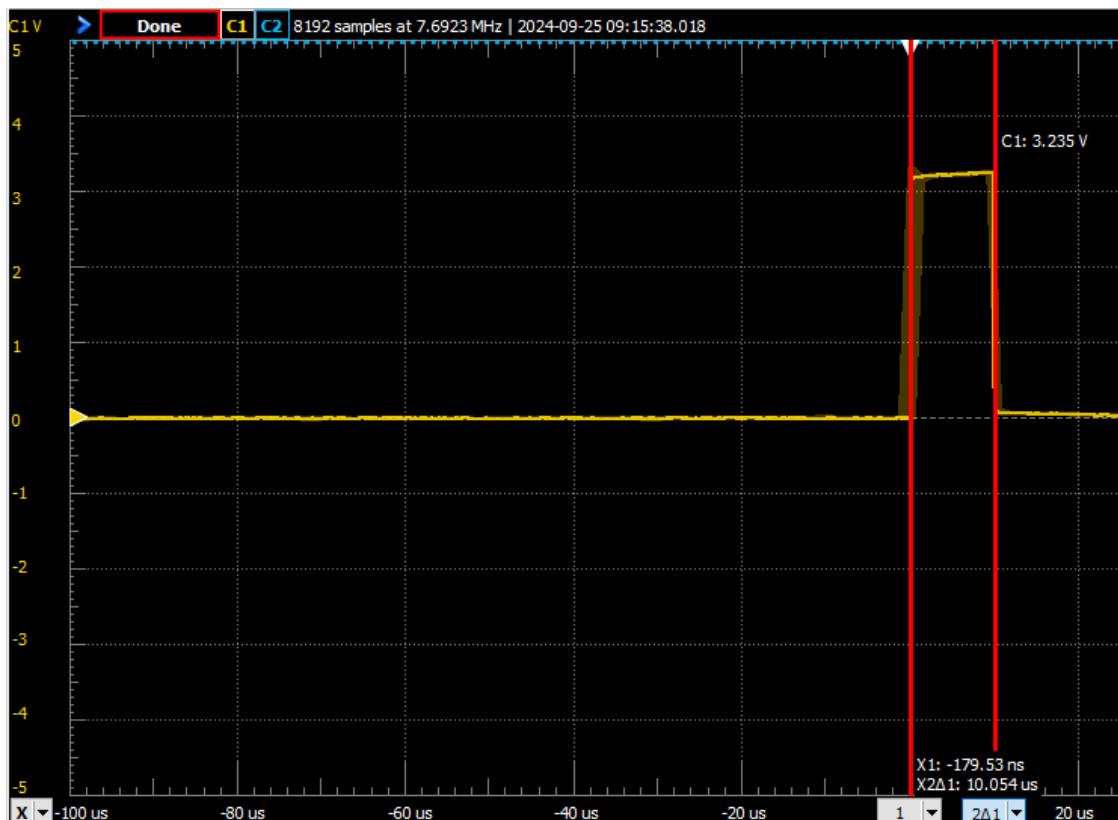
Figura 18: Esquema de montagem

### 3.1 - 1<sup>a</sup> Montagem: Servomotor e Sensor HC-SR04

O início da montagem começa com a montagem desses dois componentes como indicado na figura, porém antes de conectar os componentes é necessário avaliar os sinais de TRIGGER e PWM para verificar o funcionamento e prevenir danos aos aparelhos. A montagem segue o roteiro abaixo:

1. Montar os componentes de acordo com a figura 12
2. Desconectar o sensor e o servo motor da FPGA
3. Conectar o Canal 1 do osciloscópio na Saída TRIGGER da FPGA
4. Conectar o Canal 2 do osciloscópio na saída PWM da FPGA
5. Abrir o Aplicativo WaveForms na funcionalidade Osciloscópio
6. Comparar as ondas com as Figuras 13 e 14
7. Em caso positivo, conectar os componentes ao sistema
8. Conectar o canal 2 do osciloscópio na saída DB\_ECHO

9. Com a utilização de uma régua, posicionar um objeto plano à 10 cm do sensor e realizar a medida
10. Avaliar a largura da onda do ECHO e verificar se corresponde à 582 us (equivalente à 10 cm), com a onda lida no osciloscópio sendo semelhante à figura 15
  - Em caso de erro em alguma etapa ir para a seção 2.3 deste relatório



*Figura 19: Pulso de Trigger esperado*

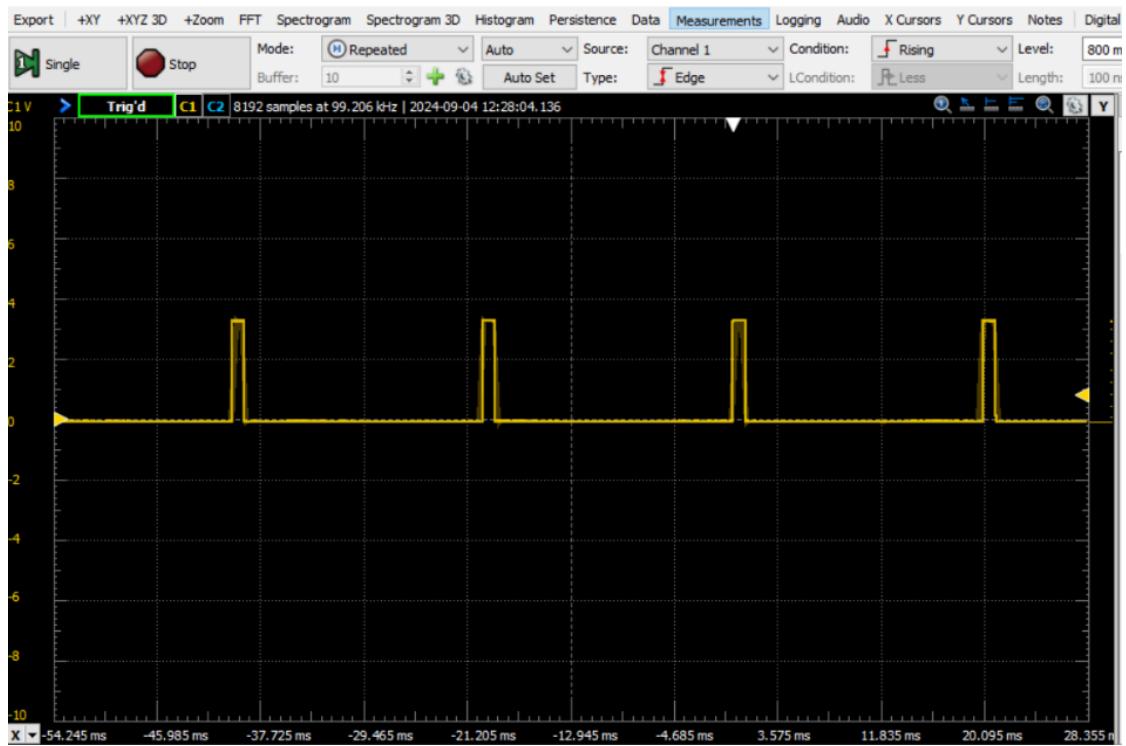


Figura 20: Exemplo de onda PWM

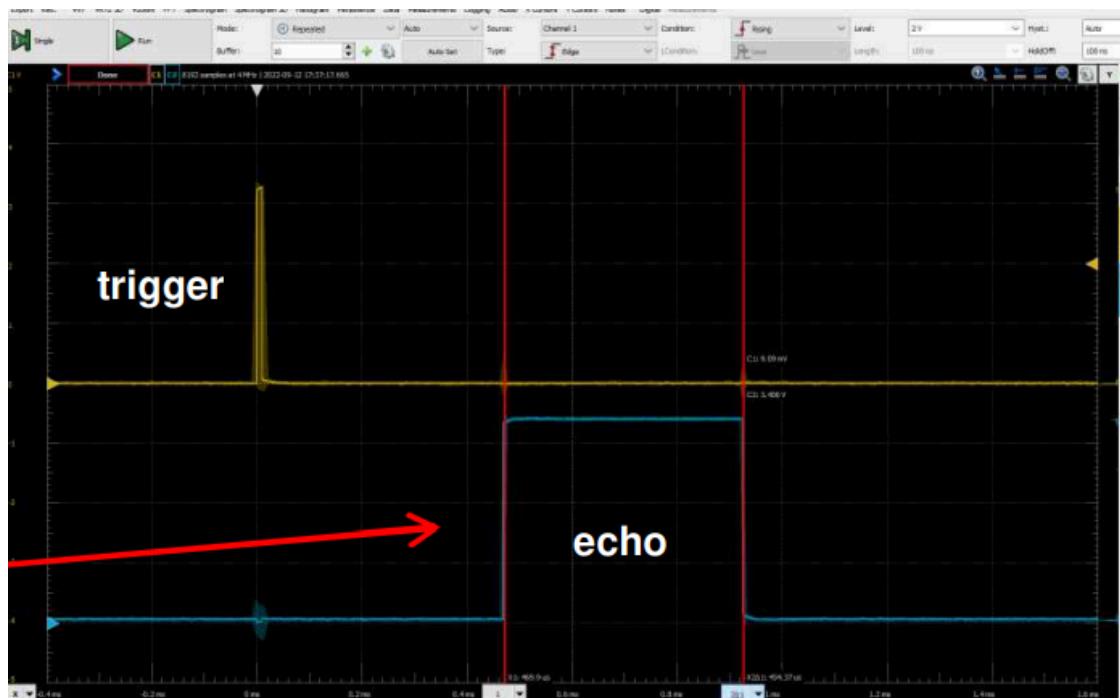


Figura 21: Exemplo de medida do sensor HC-SR04

### **3.2 - 2<sup>a</sup> Montagem: Sensor HC-SR04 e transmissão Serial**

Essa segunda montagem terá como foco a transmissão serial, uma vez que o sensor de medida já foi testado. Então essa montagem deve seguir o seguinte roteiro:

1. Abra o aplicativo Tera Term no computador
2. Realize um curto entre os cabos RX e TX do cabo serial e o conecte ao computador
3. Verifique se aparecem dígitos no terminal quando o teclado é utilizado
4. Com a mesma montagem de 10 cm anterior, conectar o canal 1 do osciloscópio na saída db\_echo
5. Conectar o canal 2 do osciloscópio na saída db\_saida\_serial.
6. Iniciar o sistema para uma única medida e verificar o formato de onda no osciloscópio.
7. Em caso positivo, seguir para a montagem final
  - Em caso de erro em alguma etapa ir para a seção 2.3 deste relatório

### **3.3 Protocolo para a correção de erros**

Esta seção é destinada para a correção de erros na montagem dos sistemas. Para isso deve-se seguir o mesmo procedimento em todos os casos a fim de se corrigir o problema de maneira padronizada. Quando um erro for detectado seguir o seguinte roteiro:

1. Verificar no último display de 7 segmentos o estado atual do sistema
2. Pelo projeto verificar o sinal responsável pela transição deste estado
3. Verificar a montagem e as conexões de acordo com a figura 12
4. Conectar o osciloscópio na saída desse sinal na placa e reiniciar o sistema
5. Analisar o formato da onda: largura, amplitude. Caso o sinal esteja correto, ir para o tópico 6.

- Aferir com o osciloscópio a onda diretamente no componente (conectar a ponta do osciloscópio diretamente no final do fio com o sinal desejado). Caso a onda esteja errada ao final do jumper, trocar o fio e reiniciar o sistema

### 3.4 - Montagem do sistema completo

Com todas as etapas anteriores realizadas, montar o circuito completo de acordo com a figura 12. Posicionar objetos a diferentes distâncias nos ângulos determinados, com uma montagem física semelhante à figura 16. Preparar o computador para a exibição das medidas e dos ângulos. Para isso basta abrir o aplicativo sonar\_processing e configurar com a porta serial correspondente.

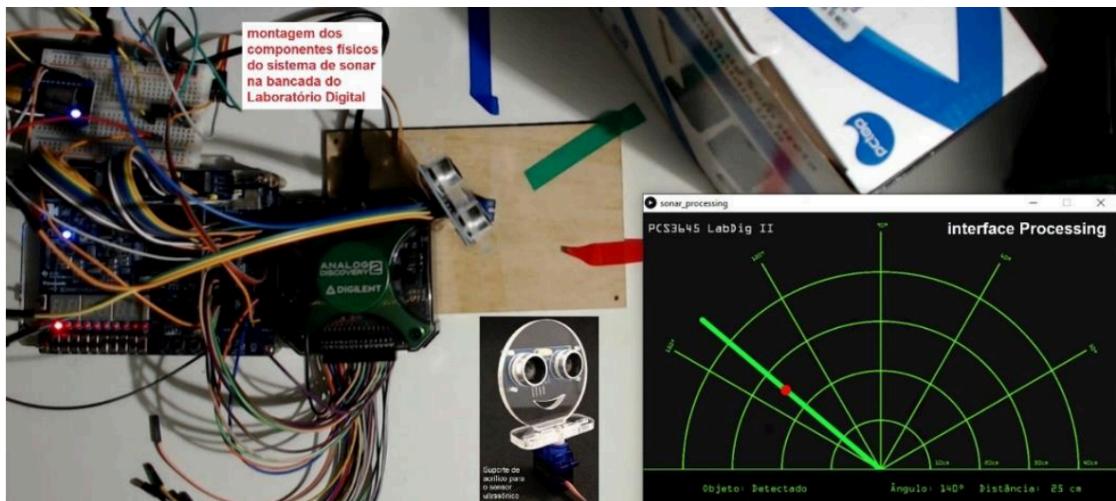


Figura 22: Exemplo de medida do sensor HC-SR04