

Programação de um menu interativo em LCD para configuração e controle de um sistema de transmissão e recepção de sinais de ultrassom baseado em FPGA

Programming of an interactive menu display for configuration and control of an FPGA-based ultrasound transmission and reception system

RESUMO

Pedro Humberto Augusto Paz Teixeira Nunes
pedronunes@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Joaquim Miguel Maia
joaquim@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Amauri Amorin Assef
amauriassef@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Este trabalho apresenta a utilização dos softwares Quartus II e Nios II para a programação de um menu interativo de controle e configuração em LCD 16x2 de um sistema de transmissão e recepção de ultrassom. O kit de processamento DE2-115 (Terasic Inc.), baseado em dispositivo de lógica reconfigurável FPGA da Intel família Cyclone IV, é usado para conexão com a placa de ULTRASSOM_TX_RX, desenvolvida previamente na UTFPR, que consiste em um canal de transmissão de forma de onda arbitrária, etapa de proteção e um canal de conversão analógico para digital com frequência de amostragem de até 80 MHz. Para a programação de hardware e software, além dos periféricos usados na programação do LCD, foram utilizados outros componentes, como, por exemplo, o UART_JTAG, Interval Time e Parallel I/O, dentre outros. Depois dessa etapa, foi escolhida a pinagem, no Quartus II, correspondente ao kit e em seguida a programação do software em linguagem C, no Nios II. Até o momento foram realizadas as etapas de modelagem do hardware e do software do LCD, a modelagem do hardware do menu de configurações e se deu início à modelagem do software do menu.

PALAVRAS-CHAVE: Sistema de pesquisa do ultrassom. FPGA.

TX/RXRecebido: 19 ago. 2019.

Aprovado: 01 out. 2019.

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



ABSTRACT

This paper presents the use of the Quartus II and Nios II softwares to program an interactive control and configuration menu based on an 16x2 LCD for an ultrasound transmission and reception system. The DE2-115 (Terasic Inc.) development board, based on a reconfigurable logic device Intel Cyclone IV FPGA, is used for connection to the ULTRASSOM_TX_RX board, previously developed in UTFPR, which consists of an arbitrary waveform transmission channel, protection stage and an analog-to-digital conversion channel with a sampling frequency of up to 80 MHz. For hardware and software programming, in addition to the peripherals used in LCD programming, other components were used, such as UART_JTAG, Interval Time and Parallel I/O, among others. After this step, the pinout was chosen, in Quartus II, corresponding to the kit and then the C language programming of the software, in Nios II. Up to date, the hardware and software modeling steps of the LCD have been performed, the hardware modeling of the configuration menu and the software modeling of the menu has started.

KEYWORDS: Ultrasound research system. FPGA.

INTRODUÇÃO

Com a busca constante por tecnologias mais seguras e simples na geração de imagens para o diagnóstico médico, equipamentos que utilizam a técnica de ultrassom para a visualização de imagens vêm se destacando cada vez mais. Isso se deve, principalmente, à facilidade de manuseio, baixo custo e a sua natureza não-ionizante e não-invasiva. Porém, apesar desses equipamentos estarem disponíveis no mercado, o acesso à programação dos módulos de transmissão, recepção e processamento ainda são restritos na maioria dos casos, impossibilitando o acesso e o controle de parâmetros de forma individualizada. Dessa forma, diversos núcleos de pesquisa têm apresentado soluções que atenuem esse problema, dentre os quais se destacam no desenvolvimento de soluções customizadas baseadas em computador pessoal (PC), arquiteturas reconfiguráveis e processadores digitais de sinais DSP (CHANG et al., 2007; QIU et al., 2010; ASSEF, 2013).

Com foco no apoio à pesquisas e técnicas mais simples, seguras e baratas no âmbito da saúde, o grupo de pesquisa do Laboratório de Ultrassom (LUS) da Universidade Tecnológica Federal do Paraná (UTFPR) vem desenvolvendo trabalhos de instrumentação ultrassônica utilizando métodos inovadores para promover o desenvolvimento de tecnologia nacional, capacitação e formação de recursos humanos na área de engenharia biomédica. Nesse sentido, este trabalho apresenta a programação de um menu interativo em display de cristal líquido LCD 16x2 para configuração e controle de um sistema de transmissão e recepção de sinais brutos de ultrassom baseado em dispositivo de lógica reconfigurável *Field Programmable Gate Array* (FPGA).

MATERIAL E MÉTODOS

Neste trabalho, utilizou-se os *softwares* Quartus II e Nios II, versão 15.1, para a programação das mensagens e comandos apresentados na tela de LCD. Tais ações visam o controle da placa ULTRASSOM_TX_RX, desenvolvida previamente na UTFPR. A placa representa um módulo completo de um canal com todas as etapas de transmissão de forma de onda arbitrária até 100 Vpp, através dos circuitos integrados MD2131 ou MD2134 (Microchip Technology Inc.), proteção (T/R *switch*) e recepção ultrassônica, com frequência de amostragem em 12 *bits* até a frequência de 80 MHz.

O desenvolvimento deste menu tem a função de facilitar a avaliação e configuração de parâmetros de transmissão de ultrassom, como, por exemplo, taxa de repetição de pulsos (PRF), amplitude de apodização, formato do sinal de excitação e ajuste de fase. Já na recepção, tem a função de apresentar os ajustes de frequência de amostragem e o controle do ganho controlado por tensão, dentre outros. Para esta aplicação, foi adotado o *kit* DE2-115 (Terasic Inc., Taiwan), possibilitando o controle da placa ULTRASSOM_TX_RX, bem como da leitura de comandos e apresentação das mensagens através de botões, chaves, LEDs e, principalmente, do LCD 16x2 disponíveis no *kit* de desenvolvimento.

Primeiramente, foi realizada a modelagem do *hardware* para a FPGA Intel EP4CE115 da família Cyclone IV utilizando a ferramenta Qsys, que é disponibilizada no *software* Quartus. Essa etapa é essencial para que se possa realizar a conexão

interna com o processador Nios II, implementado através do *software* homônimo. Além da unidade central de processamento (CPU) e das memórias internas, o principal componente acrescentado no projeto foi o **16x2 Character Display**. As Figuras 1 e 2 demonstram as ligações efetuadas dentro do *software* Qsys, ressaltando que esses periféricos (LCD, botões, LEDs, dip-switches, etc.) trabalham de forma paralela utilizando o repasse de instruções do processador Nios II.

Os principais periféricos que compõem o sistema Qsys do projeto são:

- NIOS II:** Processador *softcore* de 32 *bits*, definido em uma linguagem de descrição de *hardware*, que pode ser implementada nos dispositivos FPGA e controla todos os periféricos usados;
- PIO (Parallel input/output):** Fornece uma interface mapeada por memória entre uma porta escrava Mapeada para Memória *Avalon* (*Avalon* - MM) e portas de I/O para fins gerais. As portas de I/O conectam-se à lógica do usuário no *chip* ou a pinos de I/O que se conectam a dispositivos externos ao FPGA;
- SDRAM:** Núcleo responsável pelo controle de acesso a memórias SDRAM *off-chip*. Atende memórias padrões conforme a especificação PC100 e realiza operações de atualização, gerenciamento de linhas abertas e sequência de comandos;
- Display_LCD:** Esse núcleo fornece uma interface escrava no qual envia instruções e caracteres para o processador, a fim de serem transmitidas para a tela de LCD.

Figura 1 – Parte 1 das ligações entre unidades de hardware do menu interativo no *software* Qsys

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		cpu	Nios II Processor					
		clk	Clock Input	Double-click to export	clk			
		reset	Reset Input	Double-click to export	[clk]			
		data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
		irq	Interrupt Receiver	Double-click to export	[clk]			
		debug_reset_request	Reset Output	Double-click to export	[clk]			
		debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	# 0x0004_0800	0x0004_0fff	
		custom_instruction_m...	Custom Instruction Master	Double-click to export				
<input checked="" type="checkbox"/>		onchip_ram	On-Chip Memory (RAM or ROM)					
		clk1	Clock Input	Double-click to export	clk1			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	# 0x0002_0000	0x0003_ffff	
		reset1	Reset Input	Double-click to export	[clk1]			
<input checked="" type="checkbox"/>		sysid	System ID Peripheral					
		clk	Clock Input	Double-click to export	clk			
		reset	Reset Input	Double-click to export	[clk]			
		control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	# 0x0004_10c0	0x0004_10c7	
<input checked="" type="checkbox"/>		clk	Clock Source					
		clk_in	Clock Input	clk_clk_in	exported			
		clk_in_reset	Reset Input	clk_clk_in_reset				
		clk_out	Clock Output	Double-click to export	clk			
		clk_reset	Reset Output	Double-click to export				
<input checked="" type="checkbox"/>		LEDs	PIO (Parallel I/O)					
		clk	Clock Input	Double-click to export	clk			
		reset	Reset Input	Double-click to export	[clk]	# 0x0004_1040	0x0004_104f	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]			
		external_connection	Conduit	leds_external_connection				
<input checked="" type="checkbox"/>		sys_clk_timer	Interval Timer					
		clk	Clock Input	Double-click to export	clk			
		reset	Reset Input	Double-click to export	[clk]	# 0x0004_1020	0x0004_103f	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]			
		irq	Interrupt Sender	Double-click to export	[clk]			
<input checked="" type="checkbox"/>		high_res_timer	Interval Timer					
		clk	Clock Input	Double-click to export	clk			
		reset	Reset Input	Double-click to export	[clk]			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	# 0x0004_1000	0x0004_101f	
		irq	Interrupt Sender	Double-click to export	[clk]			
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART					
		clk	Clock Input	Double-click to export	clk			
		reset	Reset Input	Double-click to export	[clk]			
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	# 0x0004_10c8	0x0004_10cf	
		irq	Interrupt Sender	Double-click to export	[clk]			

Fonte: Autoria própria.

Component	Signal	Direction	Value	Address	Value
button	clk	Input	Double-click to export	clk	
	reset	Input	Double-click to export	[clk]	
	s1	Input	Double-click to export	[clk]	
	external_connection	Conduit	button_external_conne...	# 0x0004_10b0	0x0004_10bf
dip_sw	clk	Input	Double-click to export	clk	
	reset	Input	Double-click to export	[clk]	
	s1	Input	Double-click to export	[clk]	
	external_connection	Conduit	dip_sw_external_conne...	# 0x0004_1050	0x0004_105f
character_kd	clk	Input	Double-click to export	clk	
	reset	Input	Double-click to export	[clk]	
	avalon_lsd_slave	Input	Double-click to export	[clk]	
	external_interface	Conduit	character_kd_0_extern...	# 0x0004_1040	0x0004_104f
address_bus	clk	Input	Double-click to export	clk	
	reset	Input	Double-click to export	[clk]	
	s1	Input	Double-click to export	[clk]	
	external_connection	Conduit	address_bus_external_...	# 0x0004_10a0	0x0004_10af
rd_md2131	clk	Input	Double-click to export	clk	
	reset	Input	Double-click to export	[clk]	
	s1	Input	Double-click to export	[clk]	
	external_connection	Conduit	rd_md2131_external_c...	# 0x0004_1090	0x0004_109f
wr_md2131	clk	Input	Double-click to export	clk	
	reset	Input	Double-click to export	[clk]	
	s1	Input	Double-click to export	[clk]	
	external_connection	Conduit	wr_md2131_external_c...	# 0x0004_1080	0x0004_108f
data_bus	clk	Input	Double-click to export	clk	
	reset	Input	Double-click to export	[clk]	
	s1	Input	Double-click to export	[clk]	
	external_connection	Conduit	data_bus_external_con...	# 0x0004_1070	0x0004_107f
disparo	clk	Input	Double-click to export	clk	
	reset	Input	Double-click to export	[clk]	
	s1	Input	Double-click to export	[clk]	
	external_connection	Conduit	disparo_external_conne...	# 0x0004_1060	0x0004_106f

Após a modelagem do *hardware*, elaborou-se o fluxograma principal do funcionamento do *software* da aplicação do menu interativo (Figura 3) e, posteriormente, iniciou-se a modelagem e a programação do *software* em linguagem C para controle de todos os periféricos implementados.

```

graph TD
    Inicio([Inicio]) --> PROJETO[PROJETO MD2131 V0]
    PROJETO --> Autores[Autores: Pedro N]
    Autores --> Nota1[Nota]
    Nota1 --> MENU1[MENU 1 MOSTRA CONFIG]
    MENU1 --> Botao1_1{BOTÃO 1 RESSIONADO?}
    Botao1_1 -- NÃO --> Botao2_1{BOTÃO 2 RESSIONADO?}
    Botao2_1 -- NÃO --> Botao1_1
    Botao2_1 -- SIM --> DAC_DAC255_DAC255[\"DAC=255 PRF=1 FASE = 360°\"]
    DAC_DAC255_DAC255 --> GRAVA_CONFIG[GRAVA CONFIG]
    GRAVA_CONFIG --> DAC_DAC255_DAC255
    Botao1_1 -- SIM --> MENU2[MENU 2 ALTERA CONFIG]
    MENU2 --> Botao1_2{BOTÃO 1 RESSIONADO?}
    Botao1_2 -- NÃO --> Botao2_2{BOTÃO 2 RESSIONADO?}
    Botao2_2 -- NÃO --> Botao1_2
    Botao2_2 -- SIM --> DAC_DAC255_DAC255
    Botao1_2 -- SIM --> MENU3[MENU 3 INICIA TESTE]
    MENU3 --> Botao1_3{BOTÃO 1 RESSIONADO?}
    Botao1_3 -- NÃO --> Botao2_3{BOTÃO 2 RESSIONADO?}
    Botao2_3 -- NÃO --> Botao1_3
    Botao2_3 -- SIM --> RELAIANDO_TESTE[RELAIANDO TESTE]
    RELAIANDO_TESTE --> Botao1_4{BOTÃO 1 RESSIONADO?}
    Botao1_4 -- NÃO --> Botao2_4{BOTÃO 2 RESSIONADO?}
    Botao2_4 -- NÃO --> Botao1_3
    Botao2_4 -- SIM --> FIM_DE_TESTE[FIM DE TESTE]
    FIM_DE_TESTE --> Inicio
  
```

Flowchart illustrating the configuration and testing process for the MD2131 V0 project.

Initial Configuration:

- Projeto: PROJETO MD2131 V0
- Autores: Autores: Pedro N
- Nota: Configuração padrão é enviada

Menu 1: MOSTRA CONFIG

- Botão 1 RESSIONADO? (Sim/Não)
- Botão 2 RESSIONADO? (Sim/Não)
- Se Botão 2 RESSIONADO? = SIM: DAC=255 PRF=1 FASE = 360°

Menu 2: ALTERA CONFIG

- Botão 1 RESSIONADO? (Sim/Não)
- Botão 2 RESSIONADO? (Sim/Não)
- Se Botão 2 RESSIONADO? = SIM: DAC= _ PRF= _ FASE = _

Menu 3: INICIA TESTE

- Botão 1 RESSIONADO? (Sim/Não)
- Botão 2 RESSIONADO? (Sim/Não)
- Se Botão 2 RESSIONADO? = SIM: RELAIANDO TESTE

Relaiando Teste:

- Botão 1 RESSIONADO? (Sim/Não)
- Botão 2 RESSIONADO? (Sim/Não)
- Se Botão 2 RESSIONADO? = SIM: FIM DE TESTE

Final Configuration:

- CURSOR PISCANDO
- DAC: começa com 15 e depois ++16
- FASE vai de 15 em 15
- PRF

Página | 4

RESULTADOS E DISCUSSÃO

Na Figura 4 é apresentada a conexão entre as placas DE2-115 e ULTRASSOM_TX_RX, na qual é possível verificar a mensagem inicial no LCD 16x2.

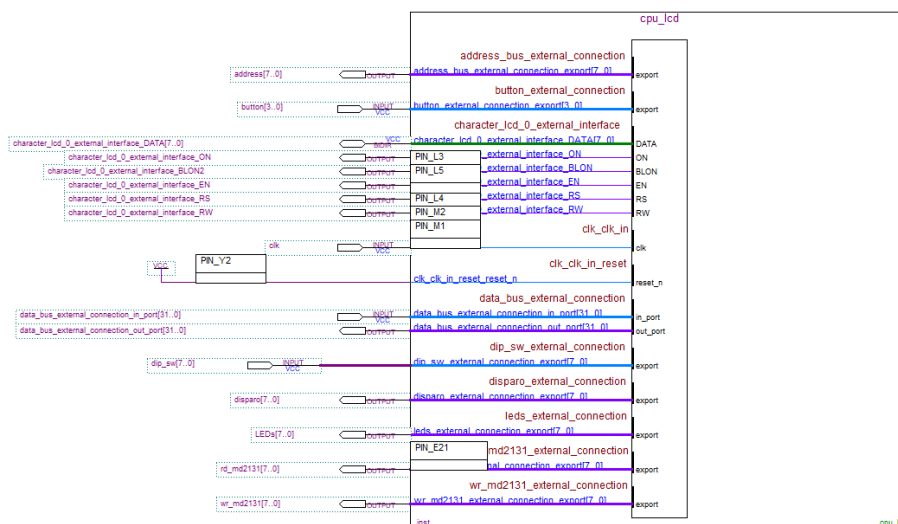
Figura 4 – Foto das placas DE2-115 e ULTRASSOM_TX_RX conectadas



Fonte: Autoria própria.

Para melhor visualização, na Figura 5 é mostrado o resultado do arquivo BDF (*Block Diagram File*), gerado a partir do arquivo em linguagem VHDL (*Very High-Level Design Language*), que foi criado através da compilação do projeto do *hardware* no Qsys. Esse processo foi seguido pela seleção da pinagem da FPGA, correspondente entre os periféricos do *hardware* gerado e os periféricos do kit utilizado, através do recurso *Pin Planner*.

Figura 5 – Inserção do bloco *top level* do tipo BDF.



Fonte: Autoria própria.

Apesar do periférico **16x2 Character Display** ser controlado por um *hardware* exclusivo para a visualização de caracteres no seu display, existe a necessidade de repassar um conjunto de informações para essa unidade com o intuito de estabelecer comandos para apresentação de informações que o usuário necessite.

Assim, está em desenvolvimento o código para que o *display* apresente as informações das etapas de transmissão, recepção e processamento dos dados para o usuário e para que o sistema permita realizar as alterações nas configurações escolhidas.

CONCLUSÃO

Considerando a linha de pesquisa, tendo realizado o estudo do *software* Quartus II para projetos de FPGA e feito o modelamento do *hardware* e do *software* para um display do tipo 16x2, concluiu-se que a modelagem do *hardware* para o menu de configurações e o código que está em desenvolvimento para o menu interativo está bem encaminhado para as próximas pesquisas. Finalizando o código e, assim, o modelamento do *software* do menu, será possível visualizar e alterar o dispositivo portátil de ultrassom tendo como opções de controle de ganho, fase e o PRF, dentre outros. A principal dificuldade encontrada foi o estudo dos *softwares* e da linguagem VHDL. Nas próximas etapas, será continuado o desenvolvimento do código para, assim, terminar o menu de configurações.

AGRADECIMENTOS

Ao CNPq, FINEP, Fundação Araucária, CAPES, UTFPR e Ministério da Saúde pelo apoio financeiro ao desenvolvimento da pesquisa. Agradecemos também à empresa Intel Altera pela doação de *kits* de desenvolvimento.

REFERÊNCIAS

ASSEF, A. A. **Arquitetura de hardware multicanal reconfigurável com excitação multinível para desenvolvimento e testes de novos métodos de geração de imagens por ultrassom**. 2013. Tese (Doutorado em Engenharia Biomédica) – Universidade Tecnológica Federal do Paraná, Curitiba, 2013.

CHANG, J. H.; YEN, J. T.; SHUNG, K. K. A novel envelope detector for high-frame rate, high-frequency ultrasound imaging. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 54, n. 9, p. 1792-1801, 2007.

EMBARCADOS. **Dispositivos SoC da Altera - Alto desempenho para projetos embarcados**. Disponível em: <https://www.embarcados.com.br/dispositivos-soc-da-altera/>. Acesso em: 05 ago. 2019.

QIU, W.; YU, Y.; SUN, L. A programmable, cost-effective, real-time high frequency ultrasound imaging board based on high-speed FPGA. In: **2010 IEEE International Ultrasonics Symposium**. IEEE, 2010. p. 1976-1979.