

DESENVOLVIMENTO DE UM MÓDULO DE COMUNICAÇÃO DE ALTA VELOCIDADE UTILIZANDO O PROTOCOLO SPI EM FPGA

Afonso Oliveira de Matos ¹; Felipe Cafezeiro Plech ² Valéria Loureiro da Silva ³

¹ Bolsista; Iniciação tecnológica – PIBIT; afonso2000@gmail.com

² Bacharel em Eng. Elétrica; Centro Universitário SENAI CIMATEC; Salvador-BA; felipe.plech@fieb.org.br

³ Dra em Física; Centro Universitário SENAI CIMATEC; Salvador-BA; valeira.dasilva@fieb.org.br

RESUMO

O avanço das pesquisas em diversas áreas têm necessitado de sistemas eletrônicos com alto poder de processamento. O dispositivo FPGA, do inglês, *field-programmable gate array*, vêm se tornando uma opção viável para suprir esta necessidade, no entanto, este elemento trata-se de hardware de baixo nível, assim surge a necessidade do desenvolvimento de métodos para com que este elemento se comunique com outros periféricos. Para viabilizar a troca de informações são utilizados protocolos de comunicação, desta forma, é necessário instruir através da linguagem de programação adequada os elementos a se comunicarem respeitando as diretrizes do protocolo escolhido. Neste artigo será proposto um método para gerenciar o processo de comunicação em uma FPGA utilizando a linguagem de programação SystemVerilog. Para validar o sistema foi realizado o testbench utilizando o software de simulação Modelsim.

PALAVRAS-CHAVE: FPGA; SPI; Protocolo comunicação, Dispositivos, Linguagem de programação, *Testbench*,

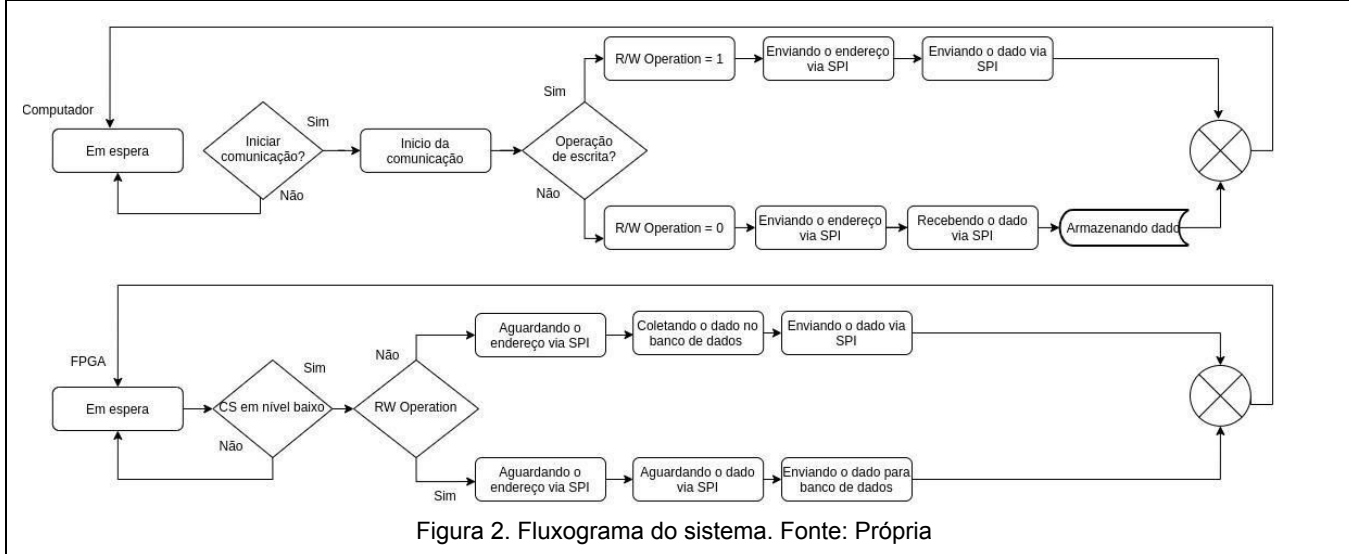
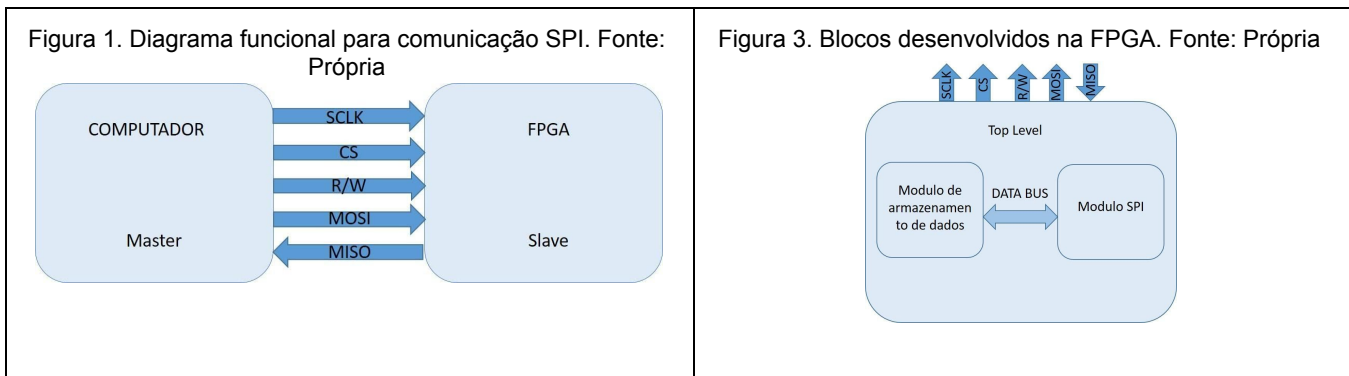
1. INTRODUÇÃO

A evolução das pesquisas em diversas áreas, como por exemplo, militar, aeroespacial, automobilística, inteligência artificial, como também a química e a física têm exigido elementos eletrônicos que possuam capacidade de processamento extremamente acelerado e preciso ¹. Observando essas demandas, o campo da engenharia eletrônica tem feito uso de um dispositivo lógico programável nomeado de FPGA, do inglês, *field-programmable gate array*. Devido às características construtivas, este componente permite a sintetização otimizada dos seus circuitos a partir do modelo escrito utilizando a linguagem de programação adequada. Esta característica permite atender as demandas de alta velocidade e baixa latência, as quais seriam impossíveis em dispositivos microcontrolados e computadores convencionais ². Existem três elementos básicos em uma FPGA, são eles, a *configuration Logical Blocks*, os *Input/Output Block* e às *Switch Matrix*. De forma resumida, através dos *Logical Blocks* é possível criar elementos funcionais lógicos, *Input/Output Block* são os elementos que permitem a interface com o meio externo através dos pinos, enquanto que as *Switch Matrix*, são uma espécie de fios (*wires*) que conectam os elementos anteriores ². Para viabilizar a troca de dados entre os periféricos e o hardware FPGA é necessário a utilização de algum protocolo de comunicação comum aos dois dispositivos, este protocolo deve oferecer uma velocidade de comunicação adequada. Entre os protocolos existentes, a *SPI*³ e o *I2C*³, são modelos consolidados para comunicações de baixa e média velocidade de transmissão de dados, porém o *I2C* possui um limite de até 1Mb/s, enquanto que a *SPI* não possui limite de velocidade, desta forma, optou-se pelo uso do protocolo *SPI*. De forma simples, a comunicação *SPI* pode ser descrita pela existência de um dispositivo mestre, o qual gerencia toda comunicação, e os escravos, os quais recebem ou enviam dados em função do comando dado pelo mestre ³. Como protocolos são apenas uma série de leis que governam como as comunicações vão ocorrer, não é necessário que ambos os dispositivos sejam programados utilizando a mesma linguagem, sendo apenas fundamental respeitar todas as diretrizes do protocolo. Neste artigo será discutido o projeto e implementação do modelo de comunicação entre a FPGA e um periférico, em especial evidenciado o processo de construção do conceito pela perspectiva do FPGA.

2. METODOLOGIA

Inicialmente, foi desenvolvido um diagrama de blocos evidenciando a arquitetura dos elementos a partir do protocolo escolhido, apresentado na figura 1. As portas *MOSI*, *MISO*, *CS* e *SCLK* são necessárias para atender os requisitos do protocolo *SPI*, já a porta "*R/W*" tem a função de informar se a operação que foi iniciada pelo master (computador) será de leitura ou escrita de dados para o escravo (FPGA).

A linguagem de programação escolhida para construção do sistema de comunicação na FPGA foi o *SystemVerilog*, uma evolução do *Verilog* ⁵. A programação em ambas linguagens é baseada em módulos, nos quais são construídas todas as lógicas necessárias para implementação das funcionalidades desejadas. Distintos módulos podem trocar informações entre si, estas interconexões são feitas utilizando os fios (*wires*) mencionados anteriormente. O fluxograma do *design* proposto é apresentado na figura 2 e busca apresentar como ocorre o fluxo de comandos e informações durante o processo de comunicação em ambos dispositivos.



A partir da figura 3, é possível observar que foram criados três módulos na FPGA, sendo o módulo de armazenamento de dados, o responsável por manter em registradores todas as informações enviadas pelo computador, bem como todos os dados enviados por outros módulos existentes no sistema e então, quando solicitado, retornar essas informações. Já o módulo gerenciador SPI, tem o objetivo de codificar e enviar os dados via os pinos pertinentes ou decodificar os dados recebido e logo após enviar para o módulo de armazenamento.

Para realizar a integração entre os módulos citados anteriormente, existe um terceiro módulo nomeado de *top level*. Este módulo é a principal peça de integração existente na programação em *SystemVerilog*, onde são instanciados os demais módulos do sistema e com o auxílio de fios (*wires*), é possível realizar a comunicação entre os módulos conforme representado na figura 3 pelo *Data Bus*. O *top level* também é responsável por ser a interface com o meio externo, ou seja, mesmo que um módulo específico realize o tratamento das informações adquiridas externamente, quem realiza o apanho e redirecionamento dos dados é o *top level*. Vale ressaltar, que no projeto em questão, dentro do gerenciador SPI existe um outro módulo chamado *SPI Slave*, este módulo auxilia o gerenciador SPI a serializar e deserializar os dados a serem transmitidos ou recebidos de acordo com as diretrizes do protocolo. O módulo *SPI Slave* foi obtido através da comunidade de compartilhamento de código *Open Source* do *SystemVerilog* ⁴. Para realizar a validação do sistema proposto foi realizado um *testbench* utilizando o software *ModelSim* ⁵, o qual permite realizar a simulação

interativa dos módulos e com isso é possível encontrar possíveis problemas antes de efetuar a sintetização do algoritmo no chip da FPGA.

3. RESULTADOS E DISCUSSÃO

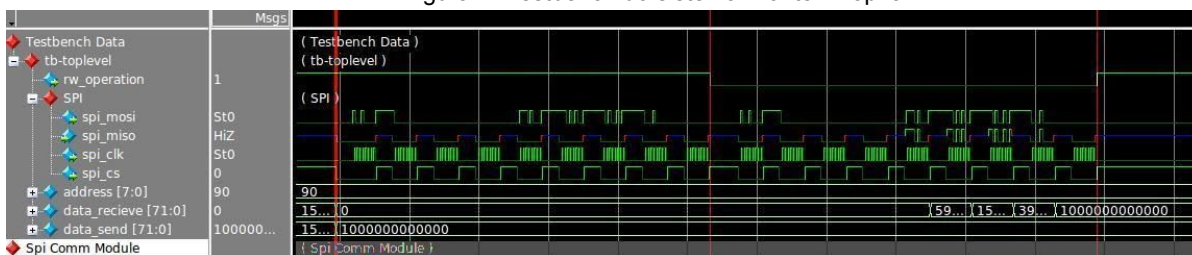
Como o software ModelSim trata-se de um simulador para teste e validação de lógica antes de realizar a sintetização na FPGA, foi necessário a criação de um dispositivo *master* dentro do ambiente de simulação, este módulo foi configurado de modo a possuir as mesmas características encontradas no dispositivo do usuário durante a solicitação e troca de dados. Para realizar a integração entre o *top-level* e o dispositivo *master*, foi criado um módulo chamado *tb_top-level*, dentro deste módulo foi instanciado o módulo *top-level*, bem como, o dispositivo *master*. Os fios (*wires*) que seriam conectados aos terminais de saída da FPGA pelo módulo *top-level*, foram redirecionados para o módulo *master*, desta forma, possibilitando realizar os teste desejados.

Utilizando o fluxo apresentado na figura 2, primeiramente será enviado um endereço e em seguida uma informação para o hardware, logo após será solicitado a FPGA a leitura do dado armazenado no mesmo endereço enviado anteriormente, desta forma, validando o sistema. Na tabela 1 são apresentados o endereço, bem como, o dado em formato hexadecimal que serão enviado e lidos, já na figura 4, é apresentado o fluxo de dados observados através do software ModelSim.

Tabela 1. Endereço e dados utilizados na simulação.

| Endereço (Hexadecimal) | Dado (Decimal) |
|------------------------|----------------|
| 0x90 | 1000000000000 |

Figura 4. Testbench do sistema. Fonte: Própria



4. CONSIDERAÇÕES FINAIS

Neste artigo foi proposto um modelo de comunicação em FPGA utilizando o protocolo SPI, para isso foi desenvolvido um módulo de controle o qual deve gerenciar toda troca de informações entre a FPGA e o meio externo. Os testes utilizando este modelo apresentaram os resultados desejados, dessa maneira, este módulo apresenta-se adequado a ser utilizado como responsável pelo processo de comunicação em qualquer sistema desenvolvido utilizando FPGA e SystemVerilog.

Agradecimentos

Agradecemos a EMBRAPPII, SENAI-CIMATEC, UE-DCC/UFMG, LabNS-UFMG e a CODEMGE pelo apoio financeiro e tecnológico.

5. REFERÊNCIAS

- PROWESS, Disponível em <[What Is an FPGA and Why Is It a Big Deal?](#)>. Acessado em 29/03/2020.
- Moore, Andrew. **FPGAs For Dummies**. New Jersey, John Wiley & Sons, Inc, 2017.
- Leens, Frédéric. **An Introduction to I2C and SPI Protocols**. IEEE Instrumentation, 1094-6969/09/ ©2009IEEE
- OPENCORES, Disponível em <<https://opencores.org/>>. Acessado em 15/10/2019.
- INTEL MODELSIM, Disponível em <[FPGA Simulation - ModelSim*-Intel® FPGA](#)>. Acessado em 29/03/2020.