

## 3 Tecnologia FPGA

Ao longo do presente capítulo será apresentada uma descrição introdutória da tecnologia FPGA e dos módulos básicos que a constitui.

### 3.1. FPGA: Histórico, linguagens e blocos

Muitos dos circuitos integrados em nosso dia-a-dia vêm programados de fábrica, destinados a aplicações específicas, ou seja, sua programação é realizada no ato da fabricação, fisicamente, sem a possibilidade de alteração posterior. O custo fixo de produção deste tipo de dispositivo é extremamente alto, ou seja, o processo para a criação da primeira unidade é milhares, ou milhões, de vezes mais caro que a produção individual das demais. Portanto, só é justificado produzi-lo se for utilizado para produção de dispositivos em massa. Outra desvantagem de seu uso é que toda a sua programação é definida durante seu projeto, não sendo possível qualquer alteração depois que a matriz é criada. Caso seja encontrado um erro no projeto, será necessário descartar toda a produção.

Em 1985 uma empresa Americana, a Xilinx Inc, apresentou um novo modelo de *chip*, capaz de ser reprogramado de acordo com as aplicações do usuário, chamado de FPGA[6], abreviação de “*Field Programmable Gate Array*”. O FPGA é composto, basicamente, por:

- blocos lógicos: circuitos idênticos, constituídos pelo agrupamento de *flip-flops*, permitem a criação de elementos lógicos funcionais;
- blocos de entrada e saída: permitem o interfaceamento dos sinais entre os blocos lógicos e a saída do circuito;
- matriz de interconexão: são chaves e trilhas usadas para interligar os blocos. Este processo, conhecido como roteamento, é bastante complexo e definido pelo software usado na programação do *chip*. Se for mal realizado, a lógica pode não funcionar corretamente;

Um *chip* FPGA contém uma enorme quantidade de blocos lógicos. Através do roteamento entre eles, é possível a implantação de lógicas bastante complexas. O desenvolvimento de módulos, ou dispositivos, nos *chips* FPGA geralmente é feito através de linguagens de descrição de hardware, do inglês “*Hardware Description Language*” ou, simplesmente, HDL. À primeira vista semelhantes a linguagens de computador tradicionais, as linguagens HDL são na verdade um conjunto de instruções que permitem aos desenvolvedores descrever a lógica desejada para o hardware de forma mais intuitiva e rápida. Uma vez concluído, o código passa por um processo de compilação, ou tradução, sendo então implantando no *chip*. As linguagens mais conhecidas são VHDL[7] e Verilog[8].

Existem inúmeras aplicações para os *chips* FPGA, sendo que algumas são tão comuns que os fabricantes desenvolvem modelos especializados em determinadas funções, como para redes de comunicação ou processamento de sinais. Vários modelos de *chips* FPGA, como o usado neste trabalho, possuem microprocessadores embutidos de forma que é possível executar programas em paralelo com a lógica do FPGA. É curioso ainda salientar que, mesmo em *chips* FPGA de menor custo, onde não há a presença de processadores, é possível implantar um através de programação de lógica física. Vários fabricantes possuem versões já prontas de seus processadores, que podem ser implantadas se necessário. Obviamente, processadores “criados” com lógica no FPGA jamais possuirão eficiência comparável a de processadores fabricados diretamente no *chip*. É, portanto, importante frisar que existe um nicho de mercado ideal para os circuitos FPGA, onde existe demanda por desempenho, mas não é viável o desenvolvimento de *chips* dedicados.

### 3.2.

#### **Componentes lógicos de comunicação em *chips* FPGA**

Quaisquer projetos desenvolvidos em circuitos FPGA, mesmo com lógicas diminutas, devem ser estruturados de forma modular. Esta abordagem contribui com a simplicidade do projeto, além de permitir reuso destes blocos inúmeras vezes dentro do mesmo circuito. A comunicação entre esses blocos lógicos pode

ser realizada diretamente, através da interconexão direta dos mesmos, ou por mecanismos mais inteligentes, com controles de transmissão ou através de barramentos de dados. Abaixo serão descritos, de forma sucinta, dois mecanismos de comunicação de grande importância na realização deste trabalho:

### 3.2.1. Interface *Local-link*

Este padrão foi desenvolvido visando a criação de canais de comunicação entre módulos, idealmente em dispositivos de redes. Admite o uso de largura de dados variável, permitindo ajuste à taxa de transmissão desejada, tamanho variável dos quadros, comunicação bidirecional e síncrona, além de possuir controle de fluxo.

A figura abaixo mostra o diagrama básico de funcionamento da interface *local-link*[9].

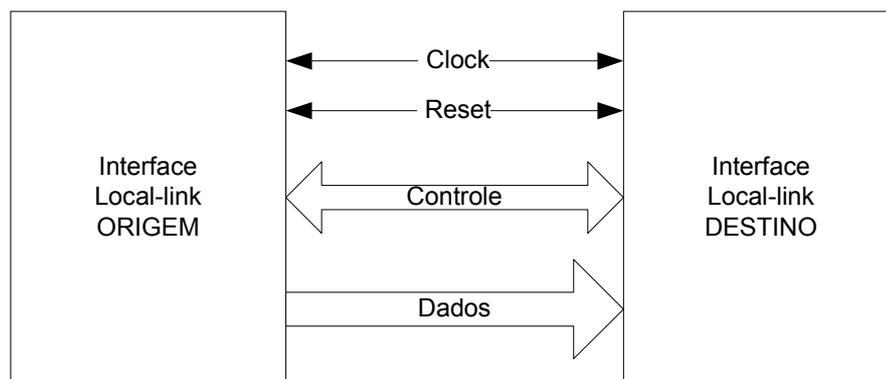


Figura 1 - Diagrama básico da interface *local-link*

O uso comum dos sinais de *clock* e *reset* é uma consequência do modo de operação síncrona. A interface *local-link* pode ser usada entre quaisquer módulos que necessitem transferir o conteúdo de quadros entre si. No projeto, objeto deste trabalho, a *local-link* será usada entre os módulos de transmissão/recepção de quadros e os módulos responsáveis pela camada de enlace do padrão Ethernet.

A figura abaixo ilustra a organização de um dispositivo com várias portas Ethernet. Para cada uma delas, é usada uma interface *local-link* para realizar a comunicação com o módulo central de comutação de quadros ou pacotes.

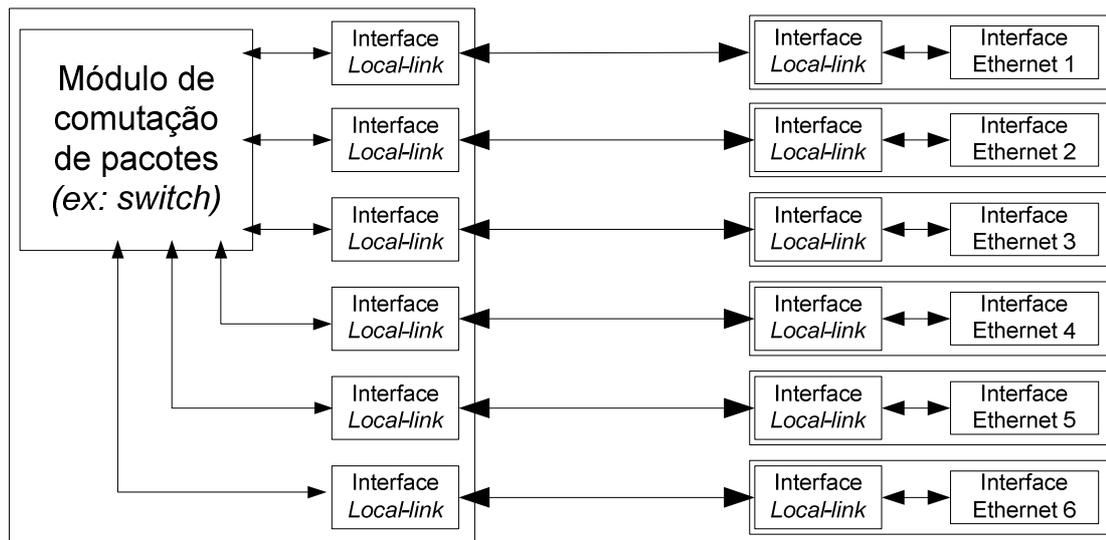


Figura 2 - Diagrama básico da interface *local-link*

### 3.2.2. Barramento OPB

Ao contrário da interface *local-link*, onde as transferências são realizadas ponto-a-ponto e com elevada taxa de transmissão, o OPB[10], sigla do inglês “*On-chip peripheral bus*”, oferece um canal de comunicação em barra conectado a diversos módulos, ou dispositivos, implantados no mesmo *chip*. Dispositivos conectados ao barramento OPB podem transferir dados entre si usando este canal comum. Desenvolvido pela IBM, o OPB tem como ponto forte o seu mecanismo de organização, responsável por evitar problemas em seu funcionamento e, especialmente, controlar as permissões e privilégios de acesso ao meio. Esse controle é realizado por um dispositivo ativo, chamado de árbitro.

Por ter sido desenvolvido para trabalhar com taxas mais reduzidas de transmissão, tipicamente bem inferiores às atingíveis com o *local-link*, o OPB não se mostra adequado para a transferência de quadros, porém é bastante atraente para operações de controle e gerência, devido à sua simplicidade e capacidade de interconectar diversos dispositivos. O OPB também pode permitir a comunicação com outros dispositivos de comunicação de baixa velocidade, como portas seriais, paralelas, modems analógicos, telas, ou até mesmo módulos de memória RAM, ROM ou flash.

### **3.3. Comentários finais e conclusões**

Ao longo do presente capítulo foram introduzidos os tópicos básicos da tecnologia FPGA, destacando-se as linguagens utilizadas (VHDL e Verilog), os blocos lógicos, os blocos de entrada e saída, a matriz de interconexão e as interfaces *local-link* e OPB. Alguns comentários envolvendo a programação e utilização destes blocos em redes de comunicação e no gerenciamento e controle das funcionalidades envolvidas foram adicionados.