

Laboratório sobre Implementação de Sistemas Digitais com VHDL Implementação de Comunicação Assíncrona entre Dispositivos

Prática: Implementação de um projeto VHDL simulável que realiza a comunicação assíncrona entre dois módulos de hardware

Recursos: Ambiente de Desenvolvimento Active-HDL da Aldec, Inc, CAD Foundation

Parte I – Introdução e Objetivos

Este Laboratório tem por objetivo exercitar o processo de comunicação assíncrona entre dispositivos. O princípio de comunicação assíncrona é importantíssimo sempre que não se pode ter um relógio global que comanda todas as ações de um sistema computacional. Isto ocorre, por exemplo, quando os dispositivos que compõem o sistema possuem velocidade de operação muito diferentes, como ocorre entre CPU e disco e entre disco e impressora.

Ao final do laboratório, os alunos deverão ter compreendido o princípio básico de operação da comunicação assíncrona, bem como a descrição de sistemas assíncronos em VHDL. Para realizar o laboratório o aluno irá utilizar a ferramenta Active-HDL. Todos os resultados serão avaliados apenas através de simulação.

Parte II – Comunicação Assíncrona

Na comunicação assíncrona, os dispositivos comunicantes operam sem a existência de um sinal comum de sincronização dos eventos em suas respectivas interfaces. Ou seja, não existe um sinal que opere como relógio global do sistema. Como os dispositivos neste caso estão sendo executados de forma completamente independente um do outro, não é possível que um dispositivo saiba o estado do outro, para eventualmente trocar informações com este no momento correto. Desta forma devem ser inseridos mecanismos que permitam a comunicação entre os dispositivos, sem que haja falha na informação que será trocada.

Em geral, os mecanismos de comunicação assíncrona são compostos por duas classes de sinais de controle, os que qualificam a informação (**sinais qualificadores**. Exemplo: leitura ou escrita) e sinais que sincronizam a troca de informação (**sinais sincronizadores**. Exemplo: inicie ou pare a comunicação, reconhecimento de dado transmitido, etc.).

Os sistemas de comunicação assíncrona, muitas vezes operam parcialmente com sinais qualificadores e sincronizadores e parcialmente com temporização. A chamada **operação temporizada** se caracteriza por um dispositivo transmissor que disponibiliza uma informação em um determinado sinal e aguarda um instante de tempo pré-determinado antes de prosseguir nas suas tarefas. Independentemente do dispositivo receptor ter reconhecido a informação, o dispositivo transmissor estará liberado para processar novas informações. Na prática, o tempo de espera pré-determinado deve ser o especificado pelo dispositivo receptor. Exemplo típico de operação temporizada ocorre quando um microprocessador lento conecta-se a uma memória rápida.

A quantidade de sinais qualificadores em uma determinada comunicação está intimamente ligada à confiabilidade e ao desempenho do sistema. Em outras palavras, quanto maior o número de qualificadores, maior o número de informações que permitem dar confiabilidade à informação trocada. Em contrapartida, maior será o tempo gasto para garantir que uma informação foi transmitida com sucesso.

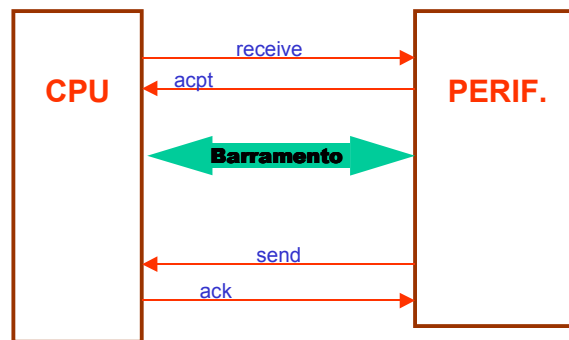


Figura 1 - Comunicação assíncrona.

A Figura 1 ilustra dois dispositivos (CPU e periférico) que se comunicam assincronamente. O controle da comunicação é efetuado através de quatro sinais (receive, acpt, send e ack) que têm a função de qualificadores e temporizadores da informação. A Figura 1 mostra, também, um barramento para a troca da informação propriamente dita. As Figuras 2 e 3 apresentam a troca de informação entre a CPU e o periférico, com um diagrama de tempos ilustrativo.

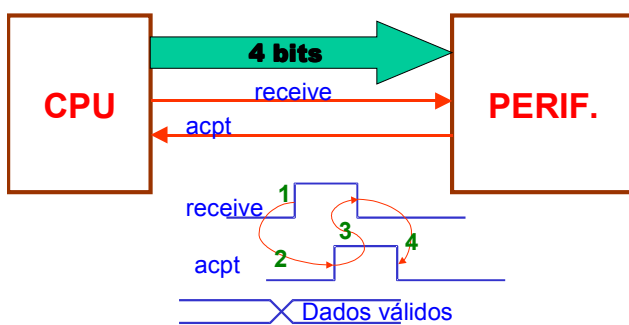


Figura 2 – Envio de dados CPU → periférico.

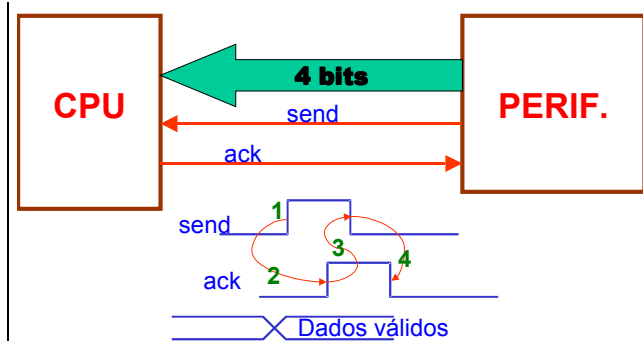


Figura 3 – Envio de dados periférico → CPU.

Pesquisa, Pense e Responda: O método de comunicação assíncrona mostrado nas Figuras 2 e 3 acima caracteriza-se por uma comunicação em quatro partes:

- 1- aviso de dado válido (transmissor, borda de subida do sinal **receive**);
- 2- aviso de dado recebido (receptor, borda de subida do sinal **acpt**);
- 3- aviso de fim de dado válido (transmissor, borda de descida do sinal **receive**);
- 4- aviso de fim da comunicação (receptor, borda de descida do sinal **acpt**).

Esta iteração parece desnecessariamente longa. Imagine uma interação mais sucinta entre transmissor e receptor, e faça um diagrama de tempos ilustrando sua idéia, comparando-a com a versão das Figuras 2 e 3.

Parte III – Especificação do Projeto

Neste Laboratório, deve ser descrito em VHDL um sistema de comunicação conforme ilustrado na Figura 4, que controla um fluxo de dados unidirecional, denominado **Sistema B**. Este sistema pode receber um vetor de 12 bits (*InVector*) do **Sistema A**, e transmiti-lo para o **Sistema C** sempre que a condição descrita abaixo for satisfeita. Caso a condição não seja satisfeita, o sistema deve filtrar a informação fornecida (ou seja, descartá-la sem transmitir ao **Sistema C**) e requisitar um novo vetor.

Sempre que o **Sistema A** tiver um vetor para transmitir para o **sistema C**, ele deve colocar o sinal *InReady* em 1 e permanecer com este valor enquanto o **Sistema B** não colocar o sinal de *Plus* em 1. O sinal de *Plus* deve ir para 0 tão logo o **Sistema B** detecte o sinal de *InReady* em 1, para evitar que o **Sistema A** envie nova informação durante o processamento de informação anterior.

Sempre que o vetor de entrada deva ser repassado para o **Sistema C**, o seu conteúdo deve ser colocado no vetor de saída *OutVector* (vetor de 12 bits) e o sinal *OutReady* (1 bit) deve permanecer em 1 **por exatamente dois ciclos do relógio interno do sistema B** (supõe-se que isto permita que o **sistema C** tenha tempo suficiente para adquirir o conteúdo do *OutVector*).

Pesquise, Pense e Responda: A interface entre o **Sistema A** e o **Sistema B** caracteriza-se por ter uma operação temporizada ou não? E a interface entre o **Sistema B** e o **Sistema C**?

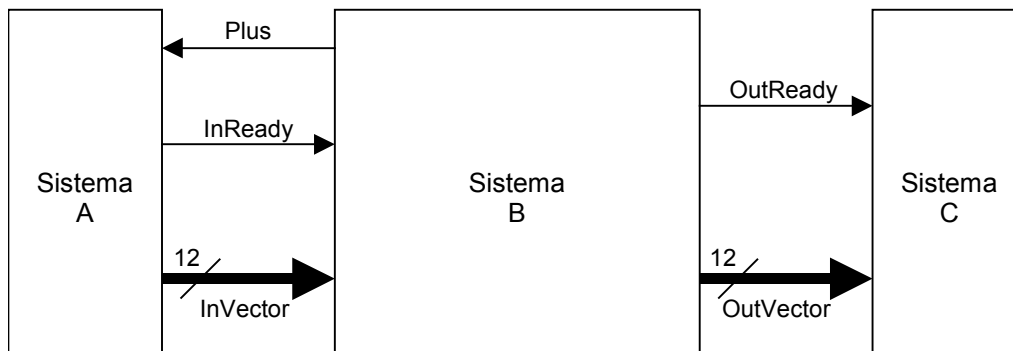


Figura 4 – Diagrama de blocos de um sistemas de comunicação entre dois sistemas.

Abaixo está descrita a condição para filtrar a seqüência de bits:

$$InVector[0, 3] \text{ and } InVector[4, 7] \text{ and } InVector[8, 11] = \text{“0000”}$$

Exemplos:

- 1 - *InVector* = 0100 0011 0111, o vetor de 12 bits será filtrado, pois o resultado é “0000”
- 2 - *InVector* = 0011 0011 0011, o vetor de 12 bits será transmitido, pois o resultado é “0011”

Abaixo está exemplificada a entrada de 2 vetores (representados em hexadecimal, 437H e 333H). Como pode ser observado, o vetor 437 foi filtrado, já que satisfaz a equação.

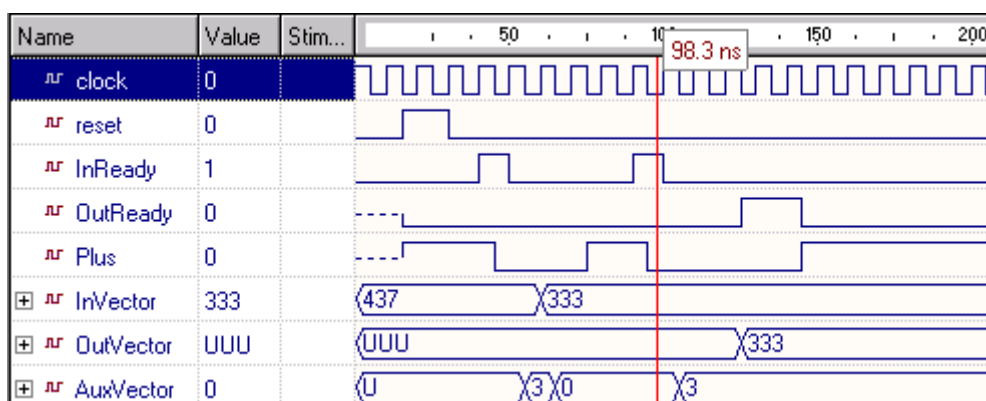


Figura 5 – Simulação do funcionamento do **sistema B**.

Na Figura 5 aparecem, além dos sinais apresentados na Figura 4, um sinal de **reset**, que indica o início de funcionamento dos sistemas e um registrador auxiliar **AuxVector** (de 4 bits), que faz parte do Sistema B e serve para realizar a avaliação do vetor de entrada.

Parte IV – Captura e simulação do projeto

Faça a descrição VHDL de todo o **Sistema B** e o test bench (o que equivale a uma descrição parcial da operação dos **Sistemas A e C**) que deve simular a entrada de pelo menos 10 valores para o vetor InVector, dos quais pelo menos 3 devem ser filtrados pelo Sistema B, e pelo menos a mesma quantidade deve ser transmitida ao **Sistema C**.

O grupo pode optar por escrever o código VHDL ou gerar um diagrama de transição de estados para representar o comportamento do Sistema B, usando o editor de máquinas de estados finitas do Active-HDL (Use opção de Menu **File** → **New** → **State Diagram**). Neste último caso, gere o código VHDL e observe a estrutura do arquivo gerado. Interaja com o editor de máquina de estados até que sua máquina de estados não tenha erros de sintaxe e reflita a funcionalidade desejada.

Pesquise, Pense e Responda: quantos ciclos de relógio é necessário, na sua implementação, para o **Sistema B** realizar uma operação completa de filtragem e/ou transmissão?

Parte V - A Fazer e a Entregar

- O projeto completo na sua versão modificada compactado pelo Foundation (opção de menu Design → Archive Design... do Ambiente Active-HDL), com formas de onda salvas em arquivo dentro do projeto;
- Relatório do projeto, contendo o diagrama de transição de estados da **Parte II** do VHDL e as respostas às questões com menção **Pesquise, Pense e Responda**;
- O diagrama de transição de estados do **Sistema B** (feito manualmente ou com o Active-HDL);

Percentuais de nota atribuídos às Atividades:

Item Avaliado	Percentual
Projeto completo e simulações	50%
Relatório	30%
Diagrama de transição de estados do sistema B	20%