

Laboratório 4

Entrada/Saída Paralela

O objetivo deste laboratório é investigar o uso de dispositivos que fornecem formas de entrada e saída para o processador. Examinaremos duas abordagens de processamento: uma baseada em *polling* e outra em interrupção. Para isso começaremos a usar as interfaces com portas paralelas disponíveis no sistema *DE2 Basic Computer*.

Uma interface paralela pode possibilitar a transferência de dados tanto para entrada como saída. As interfaces paralelas no *DE2 Basic Computer* são implementadas na forma de um componente PIO (*Parallel Input/Output*). A transferência é feita em paralelo e pode envolver de 1 a 32 bits. O número de bits, n , e o tipo de transferência variam de dispositivo para dispositivo. A interface PIO pode conter, no máximo, quatro registradores apresentados na Figura 1.

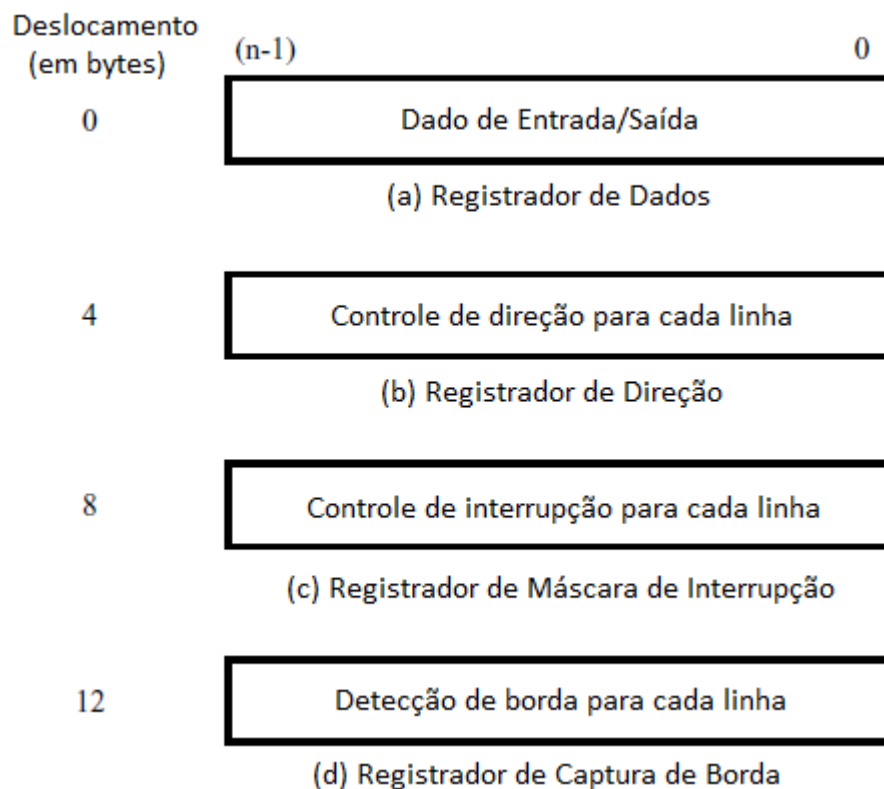


Figura 1. Registradores da interface PIO.

Cada registrador tem n bits. Os registradores possuem os seguintes propósitos:

- *Registrador de Dados* armazena n bits de dados que são transferidos entre a interface PIO e o processador Nios II. Ele pode ser implementado como entrada, saída ou ambos (bidirecional).
- *Registrador de Direção* define a direção da transferência para cada um dos n bits de dados quando uma

interface bidirecional é selecionada.

- *Registrador de Máscara de Interrupção* é usado para habilitar interrupções a partir das linhas de entradas conectadas ao componente PIO.
- *Registrador de Captura de Borda* indica quando uma mudança de valor lógico é detectado nos sinais das linhas de entrada conectadas ao componente PIO.

Nem todos os registradores são gerados em uma dada interface PIO. Por exemplo, o registrador de direção somente é incluído quando uma interface bidirecional é especificada. Os registradores de máscara de interrupção e o de captura de borda são incluídos apenas se o método baseado em interrupção for selecionado.

Os registradores PIO são acessados como se fossem posições de memória (esquema conhecido como *entrada e saída mapeadas em memória*). Qualquer endereço de memória base que tenha os 4 bits menos significativos iguais a 0 pode ser associado a um PIO. Este endereço base torna-se o endereço do registrador de dados. Os endereços dos registradores restantes possuem deslocamentos de 4, 8 ou 12 bytes (1, 2 ou 3 palavras).

O *DE2 Basic Computer* inclui diversos PIOs que estão configurados para usos diversos. Os detalhes desses PIOs estão descritos no tutorial *Basic Computer System for Altera DE-series Board*, disponível na página da disciplina.

A tarefa deste laboratório consiste em adicionar um conjunto de números sinalizados de 8 bits entrados via *toggle switches* da placa DE2. A soma resultante é mostrada nos LEDs e displays de 7 segmentos.

Parte I

Use 8 *toggle switches*, SW_{7-0} , como forma de entrada para os números. Use os LEDs verdes, *LEDG*, para mostrar a soma acumulada. Todos estes componentes estão conectados através de portas paralelas ao *DE2 Basic Computer*.

Implemente a tarefa desejada usando a linguagem de montagem do Nios II como a seguir:

1. Escreva um programa em linguagem de montagem do Nios II que leia o conteúdo das chaves (*switches*), adicione este número à soma que está sendo acumulada, e mostre a soma nos LEDs verdes.
2. Crie um novo diretório chamado *lab4_part1*. Coloque seu programa, *lab4_part1.s*, dentro deste diretório.
3. Use o Altera Monitor para criar um novo projeto, *part1*, neste diretório. Selecione seu programa e faça o download do *DE2 Basic Computer* para a placa FPGA. Escolha SDRAM como a memória que será usada pelo seu programa. Compile e carregue seu programa.
4. Use o modo passo-a-passo (*single-step*) para executar o programa e verificar sua corretude para vários números de entrada. Note que o modo passo-a-passo permite que você mude os números de entrada sem que você leia o mesmo número múltiplas vezes.

Parte II

Nesta parte vamos adicionar a habilidade de executar a aplicação continuamente e controlar a leitura de novos números. Para isso, vamos incluir um botão (*pushbutton*) que é ativado pelo usuário quando o novo número estiver pronto para ser lido. Desta forma, o usuário fornece o novo número ao configurar as chaves (*toggle switches*) apropriadamente e então pressiona o botão KEY_1 para indicar que o número está pronto para leitura.

Para realizar esta tarefa é necessário implementar um mecanismo para monitorar o estado do circuito usado para entrada dos números. Um esquema de entrada e saída muito comum é usar uma *flag de estado* que é inicialmente zerada. Esta flag é então configurada com o valor 1 tão logo a interface do dispositivo de entrada e saída esteja pronta para a transferência do próximo dado. Ao transferir o dado, a flag é novamente zerada. Logo, o

processador pode olhar para o estado da flag e determinar quando a transferência pode ser feita. Esta técnica é conhecida como *polling*.

No nosso caso, o dispositivo de entrada e saída é o próprio usuário que manualmente configura as chaves. A interface de entrada e saída é uma porta paralela presente no *DE2 Basic Computer*. Para a flag de estado vamos usar a porta paralela referente aos botões, na qual a posição do bit b_1 está conectada à KEY_1 .

Prossiga da seguinte forma:

1. Crie um novo projeto em um novo diretório para esta parte.
2. Modifique sua aplicação da Parte I para aceitar um novo número quando o botão KEY_1 for pressionado. Esta ação vai setar o bit da flag de estado no registrador de captura de borda. Após adicionar o número e acumular sua soma, seu programa deve limpar a flag ao escrever o valor 0 no registrador de captura de borda.
3. Carregue e execute seu programa para demonstrar que ele funciona apropriadamente. O programa deve executar continuamente e um novo número deve ser adicionado cada vez que o botão KEY_1 for pressionado.

Parte III

Nas partes anteriores a soma acumulada foi mostrada através dos LEDs verdes. Para esta parte você deve estender seu projeto para que a soma seja mostrada nos displays de 7 segmentos HEX3-HEX0 como um número hexadecimal. Note que os displays de 7 segmentos também estão conectados a uma porta paralela do *DE2 Basic Computer*.

Parte IV

Faça a mesma tarefa da parte anterior, mas usando interrupção ao invés de *polling*. Um pedido de interrupção deve ser gerado toda vez que o botão KEY_1 for pressionado. A rotina de tratamento de interrupção deve computar e mostrar a soma acumulada. Note que o programa principal deve configurar todos os registradores necessários para o processamento de interrupções e esperar em um laço infinito.