

Universidade Federal de Minas Gerais
Escola de Engenharia – Departamento de Engenharia Eletrônica
Laboratório de Arquitetura e Organização de Computadores
Assunto: Projeto de um módulo de GPIO (<i>General Purpose Input Output</i>) programável
Autoria: Prof. Ricardo de Oliveira Duarte

Objetivos:

Apresentar as funcionalidades de um módulo de GPIO (*General Purpose Input Output*) programável.
 Apresentar os passos necessários ao projeto de um módulo de GPIO (*General Purpose Input Output*) programável.

Seção 1: Funcionalidades de um módulo de GPIO (*General Purpose Input Output*) programável:

- 1) Configurar independentemente os pinos de um dispositivo como entradas ou saídas.
- 2) Configurar os pinos de entrada de um dispositivo para serem sensíveis a mudança de nível lógico e gerarem com isso um tipo de interrupção para a controladora de interrupções quando essa mudança ocorrer.
- 3) Configurar independentemente os pinos de saída de um dispositivo com níveis lógicos 0 ou 1.
- 4) Permitir a leitura independente de um estado lógico de um pino de entrada.

Seção 2: Projeto do módulo de GPIO (*General Purpose Input Output*) programável:

Passo 1: Identifique quantos pinos GPIO deverá ter o seu dispositivo (não inclua nessa conta os pinos de *clock*, nem o de *reset* e nem o de *enter*). Agrupe o número de pinos encontrados em 1, 2 ou 3 grupos de quantos pinos desejar. Um grupo de pinos chamaremos de PORTO. Dê um nome para cada PORTO. (Ex.: PORTO A).

Passo 2: A cada PORTO definido, deveremos associar um registrador, cujos *bits* serão capazes de armazenar a configuração individual de cada pino de GPIO do PORTO como entrada (*bit* configurado como 1) ou como saída (*bit* configurado como 0). A esse registrador daremos o nome de DIR. Precisaremos um registrador DIR para cada PORTO definido no passo 1.

Passo 3: A cada PORTO definido, deveremos associar um registrador para armazenar os níveis lógicos dos pinos de GPIOs configurados como entradas. A esse registrador chamaremos DATAIN. Precisaremos um registrador DATAIN para cada PORTO definido no passo 1.

Passo 4: A cada PORTO definido, deveremos associar um registrador para armazenar os níveis lógicos dos pinos de GPIOs configurados como saídas. A esse registrador chamaremos DATAOUT. Precisaremos um registrador DATAOUT para cada PORTO definido no passo 1.

Passo 5: A cada PORTO definido, deveremos associar um bit de habilitação de interrupção no registrador IER definido no módulo controlador de interrupções. Idem no registrador IFR. Devemos também reservar um endereço na memória vetorada para o armazenamento do endereço da ISR referente a interrupção por mudança de nível lógico em qualquer GPIO de um mesmo PORTO. Precisaremos repetir esse passo para cada PORTO definido no passo 1.

Passo 6: Adapte o módulo controlador de interrupções para atender a demanda de interrupções necessária para esse novo módulo.

Passo 7: Escreva um programa em linguagem C que configure os pinos de GPIO conforme o seu desejo e que habilite a interrupção por mudança de estado associada a pelo menos um pino de GPIO. A ISR deve

Universidade Federal de Minas Gerais
Escola de Engenharia – Departamento de Engenharia Eletrônica
Laboratório de Arquitetura e Organização de Computadores
Assunto: Projeto de um módulo de GPIO (<i>General Purpose Input Output</i>) programável
Autoria: Prof. Ricardo de Oliveira Duarte

fazer uma certa ação se ocorrer uma mudança de estado de 0 para 1 e deverá fazer outra ação caso contrário.

Passo 8: Converta o programa que você escreveu no passo anterior para a linguagem *assembly*.

Passo 9: Converta o programa que você escreveu no passo anterior para a linguagem de máquina.

Passo 10: Crie um *testbench* para simular a sua CPU com a configuração necessária da interrupção e a rotina de tratamento de interrupção em *software*.

Passo 11: Compile a sua CPU com o novo módulo de GPIO programável e as modificações realizadas controladora de interrupções. Corrigir eventuais erros de compilação.

Passo 12: Simule o seu sistema com o *testbench* criado no passo.

Passo 13: Teste o seu sistema no kit DE10-Lite.