

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia de Sistemas Eletrônicos - PSI

PSI-2553- Projeto de Sistemas Integrados

O Processador Plasma - Parte 3
(Interrupção)

Objetivos

Esta experiência visa ilustrar o uso do recurso de interrupção utilizado na comunicação entre o processador MLITE e os componentes periféricos disponíveis no sistema Plasma.

Parte Prática

O modelo do sistema Plasma utilizado nesta experiência é uma ampliação do utilizado na exp2. Um módulo periférico, designado por Perif_Exp3 (veja seção 2.3 da apostila teórica) foi inserido no sistema, substituindo o módulo ETH-DMA cujo registrador está no endereço fixo 0x20000070. Este periférico realiza algumas operações de escrita na memória semelhantes às que ocorrem num periférico real. Para isto foram feitas algumas adaptações no hardware e no software. As modificações de hardware (no código VHDL) são apresentadas no documento **Modificações_Plasma_17.pdf**.

O roteiro da parte prática será uma combinação das experiências 1 e 2:

- Preparando as pastas e arquivos
- Compilação dos códigos C (conforme realizado na exp1) para gerar o código *assembly* (arquivos *test.lst* e *test.map*). Um programa contém a aplicação (algoritmo de Fibonacci) acrescido das instruções necessárias para executar a interrupção do processo e o outro contém a rotina de tratamento da interrupção do periférico (ISR).
- Geração do formato *.hex, conforme realizado na exp2, para gerar os arquivos que serão incluídos na memória RAM do Plasma.
- Captura e compilação do projeto no ambiente Quartus da Altera.
- Simulação do projeto através do simulador ModelSim.

Preparando as pastas e os arquivos

- 1) Criar o diretório **X:\psi2553\exp3\infra**.
- 2) Copiar para **X:\psi2553\exp3\infra** o arquivo **Rede\NEWSERVERLAB\psi2553\exp3\tools_exp3.zip**
- 3) No subdiretório **X:\psi2553\exp3\infra** extraia as pastas comprimidas que estão em **tools_exp3.zip**. Com isso duas pastas deverão aparecer: **X:\psi2553\exp3\infra\tools** e **X:\psi2553\exp3\infra\gccmips_elf**. A compilação e a montagem serão feitas no subdiretório **X:\psi2553\exp3\infra\tools**.

Compilação do programa em C e geração dos arquivos *.hex

1. Na pasta **tools** existem 2 arquivos .C: count_2553.c e no_os_2553.c
 2. Modifique seu arquivo count_2553.c substituindo *seu_numero* por VALOR???????????
 3. Siga as instruções da apostila exp plasma_1 para abrir uma janela de terminal.
 4. Tecle **gmake count_2553** para realizar a compilação e montagem do programa count_2553.c assim como para gerar os arquivos .hex.
 5. Crie uma nova pasta com o nome **projeto_I** em **X:\psi2553\exp3** (ficará ao lado de infra).
 6. Copie para esta pasta
 - a. Os arquivos VHDL que estão no *serverlab* em **plasma_com_perif_exp3**
 - b. Os arquivos .hex que estão na pasta **X:\psi2553\exp3\infra\tools**
 7. Siga as instruções da apostila exp plasma_2 para incluir os arquivos *.hex na memória.
- Observe os códigos VHDL existentes na pasta plasma_com_perif_exp3. Localize o componente periférico *perif_exp3* dentro do código plasma.vhd. Observe os portos deste componente. Utilize a apostila de teoria plasma_I_teorias.pdf para ajudá-lo a entender os portos de perif_exp3.vhd. Observe também o código VHDL de plasma_tbw.vhd.
 - Examine o código arquivo count_2553.c. O código do *main* é um laço infinito, ou seja, de um programa de roda continuamente, que será eventualmente interrompido pelo módulo perif_exp3. Observe e explique o significado de (use o plasma.h para auxiliá-lo):
 - a. MAIN_COUNT
 - b. MemoryRead e MemoryWrite

→ Anote no seu roteiro:

1. O número que você usará na sua experiência
2. Qual é o valor de MAIN_COUNT?
3. Qual é o argumento de MemoryRead e o que a função retorna?
4. Quais são os argumentos de MemoryWrite e que a função faz?
5. Explique como funciona e o que faz o laço infinito de count_2553.c?

Examinar o arquivo count_2553.c. Nele se encontra:

- Chamada para a função OS_Init() (declarada em no_os_2553.c)

Examinar o arquivo no_os_2553.c. Nele se encontram:

- Rotinas OS_InterruptMaskSet() e OS_AsmInterruptEnable()
- Função OS_InterruptServiceRoutine (chamada dentro do *interrupt_service_routine*)

Examinar o arquivo boot.asm.

Qual a finalidade destes recursos?

Use as definições de plasma.h para lhe auxiliar.

Examinar o arquivo test.lst (que contém o seu código *assembly*)

→ **Anote no seu roteiro:**

Os opcodes de:

- a. Início do main
- b. Última instrução do main
- c. Dentro do main, a escrita do vetor $x[\text{num_aluno}]$ na posição MAIN_COUNT+4
- d. Início do ISR global (endereço da interrupção fixa)
- e. Chamada ao OS_InterruptServiceRoutine

Compilação e simulação no Quartus

Siga as instruções da apostila *exp plasma_2* para realizar as seguintes tarefas:

- 1) Disparar o programa *Quartus II* da Altera
- 2) Criar um projeto com o nome *plasma*.
- 3) Compilar o projeto (já incluindo o *testbench*)

→ **Anote no seu roteiro:**

- a. código do componente
 - b. número de elementos lógicos utilizados
 - c. número de registradores utilizados
- 4) Fazer a simulação funcional do arquivo *plasma*. No final da simulação, você deverá:
- Identificar todos os sinais que aparecem na simulação e correlaciona-los com o modelo que aparece na apostila de teoria.
 - Identificar a sequência dos códigos das instruções (*opcodes*) mostrados na simulação e verificar que são os que aparecem no código *assembly* gerado na aula *plasma_1*. Para isto observe o conteúdo do arquivo *test.lst*.
 - Identificar a sequência dos endereços das instruções mostradas na simulação e verificar que são as que aparecem no arquivo *test.lst*.
- 5) Salve o resultado da simulação, nos trechos de interesse (veja a seguir os momentos da simulação que devem ser observados).

→ **Anote no seu roteiro:**

- Quais registradores do *Register File* foram utilizados durante a simulação (anotar os números e nomes destes registradores)
- Os *opcodes*, endereços e instantes de tempo de simulação dos seguintes eventos:
 1. Início do *main*

2. Dentro do *main*, a escrita do vetor $x[\text{num_aluno}]$ na posição MAIN_COUNT+4
3. Início do Fibonacci
4. As ativações do sinal ext e dos dados de entrada do periférico (para a primeira interrupção)
5. Ativações e desativações dos sinais de irq (para a primeira interrupção)
6. Início ISR global (endereço da interrupção fixa)
7. Chamada ao OS_InterruptServiceRoutine
8. Anotar o instante e o valor que estava sendo calculado no momento em que o Fibonacci foi interrompido pela primeira vez
9. Anotar o instante em que o cálculo foi reiniciado pela primeira vez
10. Identifique na carta de tempos uma leitura do periférico através de:
 1. Endereço do periférico
 2. Endereço das instruções
 3. Valores dos dados envolvidos.
11. Identifique na carta de tempos uma escrita no periférico através de:
 1. Endereço do periférico
 2. Endereço das instruções
 3. Valores dos dados envolvidos.