

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

PSI-2553- Projeto de Sistemas Integrados

Experiência 3: Utilizando Interrupção no Plasma (Parte Prática)

M.S. / W.J.C / M.A.R.J (17)

Conteúdo:

1. OBJETIVOS	2
2. PARTE EXPERIMENTAL	2
2.1. Buscando os Arquivos do Plasma para Compilação	2
2.2. Entendendo os arquivos a serem compilados	2
2.3. Aprontando o Arquivo count_2553.c para Compilação	4
2.4. Pós-compilação	4
2.5. Inserção e Montagem do Sistema Plasma com o Módulo Perif_Exp3	5
2.6. Simulação no Quartus	6
2.7. Prototipando e Testando o Plasma	7

1. Objetivos

Esta experiência visa a exploração dos conceitos de comunicação entre processadores e periféricos em sistemas digitais, particularmente por interrupção.

2. Parte Experimental

O Plasma utilizado nesta experiência é inicialmente o mesmo que o utilizado na exp2. No entanto, um novo módulo Perif_Exp3 (veja seção 2.3 da parte teórica) foi inserido no sistema e, para isto, uma série de mudanças no hardware e software foram feitas. As partes novas, hardware ou software são disponibilizadas prontas para o(s) aluno(s), porém, todas as modificações de hardware (no código VHDL) são apresentadas no documento Modificações_Plasma_17.pdf, para conhecimento (é importante entendê-las).

2.1 Buscando os Arquivos do Plasma para Compilação

- 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) Na 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**. Atenção: a compilação e a montagem serão feitas no subdiretório **X:\psi2553\exp3\infra\tools**, porém, como em experiências anteriores, o aluno deverá compilar os programas via linhas de comando (diretamente no DOS).

2.2 Entendendo os arquivos a serem compilados

Em **X:\psi2553\exp3\infra\tools**, existem quatro (4) arquivos a examinar:

boot.asm – contém funções de inicialização, diretamente ligadas ao hardware do Plasma;

plasma.h – definição de variáveis e funções globais; você observará que com a inserção do módulo perif_exp3, várias definições referentes ao bloco ethernet foram substituídas.

no_os_2553.c – contém funções de sistema do Plasma, incluindo a rotina de interrupção do usuário;

count_2553.c – programa do usuário;

Estes arquivos estão interligados através de chamadas e de definição das variáveis globais e na análise de um poderá querer informações de outros.

- 1) 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. Tenha certeza que compreende o código do *main*. Atente para os seguintes aspectos (use o **plasma.h** para auxiliá-lo):
 - a. O valor de MAIN_COUNT;
 - b. Os significados de MemoryRead e MemoryWrite

→ No relatório:

ANÁLISE E DISCUSSÃO:

- I. Qual é o valor de `MAIN_COUNT`?
- II. Qual é o argumento de `MemoryRead` e o que a função retorna?
- III. Quais são os argumentos de `MemoryWrite` e que a função faz?
- IV. Explique como funciona e o que faz o laço infinito de `count_2553.c`? Que valores serão escritos e em que posições absolutas de memória, nas primeiras 3 iterações?

2) Ainda em `count_2553.c`, observar que existe a chamada para a função `OS_Init()`, que é declarada em `no_os_2553.c`. Observe que esta função chama outras duas rotinas, `OS_InterruptMaskSet()`, também em `no_os_2553.c` e `OS_AsmInterruptEnable()` de `boot.asm`; ao mesmo tempo, as funções usam definições de `plasma.h`.

→ No relatório:

EXPERIMENTAL:

Copie os trechos de código onde as funções `OS_Init()`, `OS_InterruptMaskSet()` e `OS_AsmInterruptEnable()` são definidas.

ANÁLISE E DISCUSSÃO:

- I. Baseado no código, explique o que faz `OS_InterruptMaskSet()` ao ser chamado em `OS_Init`.
 - II. Baseado no código, explique o que faz `OS_AsmInterruptEnable()`
- 3) Na mesma pasta, examinar o arquivo `boot.asm` novamente. Procure entender o trecho referente ao label `interrupt_service_routine`. Examinar também no arquivo `no_os_2553.c`, a função `OS_InterruptServiceRoutine`, que é chamada dentro do `interrupt_service_routine`. Use as definições de `plasma.h` para lhe auxiliar.

→ No relatório:

EXPERIMENTAL:

Imprima os trechos onde as funções `interrupt_service_routine` e `OS_InterruptServiceRoutine ()` são definidas.

ANÁLISE E DISCUSSÃO:

- I. Explicar genericamente qual a relação entre `interrupt_service_routine`, e `OS_InterruptServiceRoutine`.
- II. Explicar quais as tarefas realizadas pela `interrupt_service_routine`, justificando-as pelas suas instruções Assembly (associe os trechos com as tarefas).
- III. Liste as variáveis de `OS_InterruptServiceRoutine`, cujos valores são definidos em `plasma.h`. Anote os valores.
- IV. Explícite todos os valores envolvidos em cada escrita e leitura de memória nas duas primeiras iterações de `OS_InterruptServiceRoutine`.

- 4) Como ocorrido no uso do tutorial, o arquivo **makefile** já está ajustado para gerar o código do programa **count_2553.c**. Examine o arquivo **makefile** em **X:\psi2553\exp3\infra\tools** e verifique que está tudo certo.

2.3 Aprontando o Arquivo **count_2553.c** para Compilação

- 1) Vamos realizar uma modificação no código de **count_2553.c** da mesma forma que na experiência 3:
 - a) Calcule $\text{num_aluno} = \text{No_USP} \bmod 21$; caso este número seja igual a 0, faça $\text{num_aluno} = 21$
 - b) No código C, atualize o **num_aluno** e apague algumas linhas com as instruções $x[k] =$ sequência de números primos, de tal forma que a sequência que permanece no código, vá de $x[1]$ até $x[\text{num_aluno}]$.

→ No relatório:

DADOS EXPERIMENTAIS:

Impressão do arquivo `count_2553.c`.

ANÁLISE E DISCUSSÃO:

Identifique as novas inserções de acordo com o número `num_aluno` acima no código impresso.

2.4 Pós-compilação

- 1) Seguir os passos apresentados no tutorial do Plasma para realizar a compilação do código do **count_2553.c** através do arquivo **makefile**.
- 2) Olhar o arquivo **X:\psi2553\exp3\infra\tools\test.map**. Localize os endereços de **main**, **OS_Init**, **OS_AsmInterruptEnable**, **OS_InterruptMaskSet** e **OS_InterruptService Routine**.

→ No relatório:

DADOS EXPERIMENTAIS:

Impressão do arquivo `test.map`.

ANÁLISE E DISCUSSÃO:

Identifique os endereços das cinco rotinas acima.

- 3) Olhar o arquivo **X:\psi2553\exp3\infra\tools\test.lst**. Observe o código *assembly*. Localize os seguintes pontos de interesse:
 - 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**
 - f. Return de **OS_InterruptServiceRoutine**
 - g. Return do **ISR global**
 - h. Um ponto de leitura do registrador **IRQ_status**

→ No relatório:

DADOS EXPERIMENTAIS:

Impressão do arquivo completo test.lst.

ANÁLISE E DISCUSSÃO:

Identificar na impressão os pontos a até h.

2.5 Inserção e Montagem do Sistema Plasma com o Módulo Perif_Exp3

- 1) Copiar para X:\psi2553\exp3\ a pasta **Plasma_com_Perif_Exp3** que está em Rede\NEWSERVERLAB\psi2553\exp3\. Corresponde aos arquivos VHDL do sistema plasma usado na exp3 com as modificações necessárias para a inclusão do periférico.

ATENÇÃO: em Rede\NEWSERVERLAB\psi2553\exp3, o aluno poderá encontrar as pastas **Plasma_Original** e **Perif_Exp3**. Eles correspondem respectivamente aos arquivos VHDL do sistema plasma usado na exp2 e os arquivos do periférico antes das modificações que estão explicadas, passo a passo no arquivo **Modificações_Plasma.pdf** (no Moodle). Estas modificações deverão ser seguidas sempre que um novo periférico for incluído-caso da Exp.4)

- 2) Olhe os códigos VHDL de perif_exp3.vhd e seus submódulos. Utilize a apostila de teoria para ajudá-lo no entendimento dos portos de perif_exp3.vhd.
- 3) Olhe o código VHDL de plasma_tbw.vhd e faça as seguintes modificações nele:

Você deverá gerar dois números decimais a partir de seu número USP

primeiro_num = composto pelos primeiros quatro algarismos do seu número USP

segundo_num = composto pelos últimos quatro algarismos do seu número USP

Exemplo: número USP= 7350422

primeiro_num= 7350

segundo_num= 0422

Substituí-los em XXXX nas designações de data_in_ext, por exemplo:

```
data_in_ext <= std_logic_vector(to_unsigned (XXXX, 32)); /* usar o
primeiro_num
```

→ No relatório:

ANÁLISE E DISCUSSÃO:

- I. Mostre o cálculo de seus números primeiro_num e segundo_num
 - II. Explique qual é o objetivo destes números no testbench e simulação
- 4) Copie os quatro arquivos codex.hex de X:\psi2553\exp3\infra\tools para X:\psi2553\exp3\Plasma_com_Perif_Exp3.

2.6 Simulação no Quartus

- 1) Carregue o programa *Quartus II* da Altera para fazer as simulações através do seguinte caminho: **Iniciar/Programas/Altera/Quartus II 9.1.**
 - 2) Crie um projeto como o nome *plasma*. Para isto, use o **New Project Wizard**. Após atribuir o nome do projeto, incluir todos os arquivos *.vhd, assinalar o dispositivo, na caixa de diálogo seguinte, no campo de simulação, adicione em Tool name: ModelSim-Altera e no Format: VHDL Após finalizar esta etapa poderá observar as características do projeto na janela **Project Navigator**. Clicando duas vezes sobre o *plasma.vhd* (em **Files**) poderá visualizar o código vhd do projeto. Lembre-se que os arquivos *.hex já estão incluídos pela referência feita nas memórias RAM.
 - 3) Usaremos o simulador ModelSim nesta experiência pela facilidade de se acessar os sinais internos do projeto. Para configurar o Quartus para este objetivo, acesse o menu **Assignments > EDAToolSettings > Simulation**.
 - 4) Na opção “Compile test bench”:
 - a. clique em testbench
 - b. clique em NEW
 - c. Completar: “Test bench name” = plasma_tbw. “Top level module in test bench” = plasma_tbw. “Design instance name in test bench” = behavior.
 - d. No campo “Test bench files”, adicionar o arquivo “plasma_tbw.vhd”
 - 5) Na opção “Use script to set up simulation” indique a rota do arquivo “inicio.txt”, que será o script para a simulação no ModelSim.
 - 6) Abra os arquivos plasma_tbw.vhd e inicio.txt e os estude cuidadosamente.
 - 7) Compile o arquivo *plasma.vhd*. Selecione no menu **Processing > Start > Start Analysis & Synthesis**. O resultado da compilação é apresentado numa janela tipo *pop-up*, pressione **OK**. Uma janela com o relatório da compilação é aberta automaticamente ou pode ser acessada pela opção do menu **Processing > Compilation report**.
 - 8) Realize a simulação funcional do arquivo plasma. Para realizar a simulação funcional: **Tools > Run EDA Simulation Tool > EDA RTL Simulation**. O ModelSim deverá abrir já com a janela de simulação pronta.
- ATENÇÃO: lembrar que estão sendo observados os sinais em cada estágio de um pipeline. Portanto existe uma defasagem no tempo entre os sinais que correspondem a cada instrução. Por exemplo, se o resultado do estágio *fetch* ocorrer num ciclo *i* o resultado do estágio *decode/control* correspondente à mesma instrução deverá ocorrer num ciclo posterior *i+n*.**
- 9) Na simulação, você deverá localizar por meio dos endereços de instrução obtidos em **test.lst** os pontos:
 - a. de **c** até **h** do item 3) da **seção 2.4**
 - b. As ativações do sinal ext e dos dados de entrada do periférico (para as duas interrupções)
 - c. Ativações e desativações dos sinais de irq (para as duas interrupções)
 - d. Uma escrita no e uma leitura do registrador do periférico.
 - 10) Imprima o resultado da simulação, nos trechos de interesse.

→ No relatório:

DADOS EXPERIMENTAIS:

Carta de tempos com simulação com todos os trechos de interesse acima.

ANÁLISE E DISCUSSÃO:

- I. Indique/identifique na carta de tempos (cuidado para que tenha uma resolução adequada) todos os pontos de interesse como descrito on item 9.a (identifique-os explicitamente através dos endereços das instruções e os opcodes associados)
- II. Indique/identifique na carta de tempos (cuidado para que tenha uma resolução adequada) todos os pontos de interesse como descrito on item 9.b (para as duas interrupções)
- III. Indique/identifique na carta de tempos (cuidado para que tenha uma resolução adequada) os pontos de interesse como descrito on item 9.c (para apenas uma das interrupções)
- IV. Identifique na carta de tempos uma leitura do periférico (identifique-a explicitamente através do endereço do periférico, do endereço das instruções , dos valores dos dados envolvidos e dos protocolos associados).
- V. Identifique na carta de tempos uma escrita para o periférico (identifique-a explicitamente através do endereço do periférico, do endereço das instruções , dos valores dos dados envolvidos e dos protocolos associados).
- VI. Qual é o método de controle usado para os protocolos de comunicação processador-memória ou processador-periférico? Justifique.

2.7 Prototipando e Testando o Plasma

O objetivo da prototipagem é observar o comportamento do Plasma em uma operação normal (programa count_2553) e quando da ativação da interrupção feita pelo bloco Perif_Exp3. Na interrupção, a execução da tarefa definida no ISR deverá ser observada, assim como o retorno ao programa principal.

- 1) Copiar para X:\psi2553\exp3\ a pasta **Plasma_Interrupt_DE2** que está em Rede\NEWSERVERLAB\psi2553\exp3\.

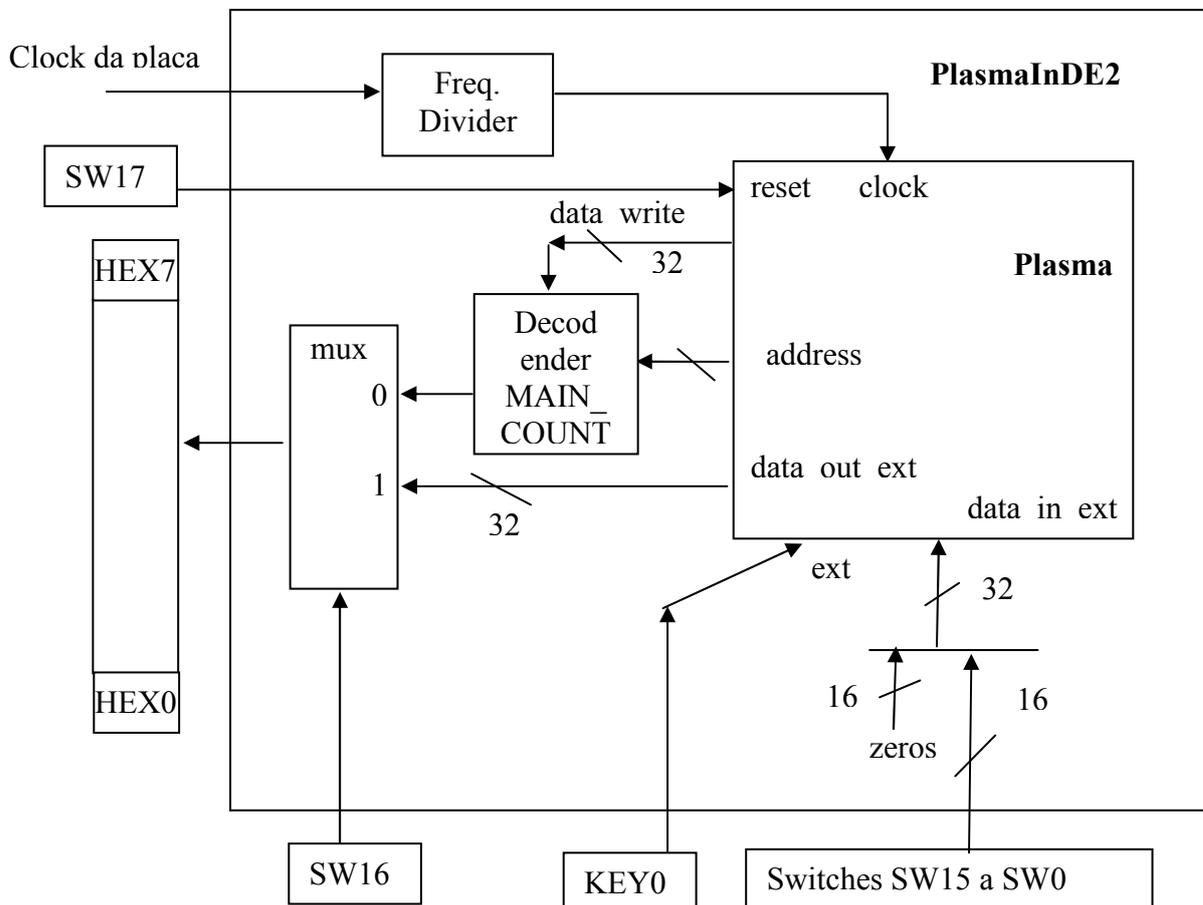
Obs. Uma pequena explicação:

Diferente da simulação, o arquivo topo **plasmainDE2.vhd** é um arquivo topo que inclui o arquivo **plasma.vhd**, além das interfaces necessárias para o seu uso na placa DE2. O arquivo **plasma.vhd** é equivalente ao arquivo montado na seção 2.4. Os arquivos devem ser observados com atenção no que se refere ao acesso externo dos dados.

A estrutura básica do módulo PlasmaInDE2 é dada na figura abaixo para orientar o(a) aluno(a). Consiste dos seguintes elementos:

- A) Divisor de frequência do clock da placa para tornar a velocidade das operações compatíveis com o olhar humano. Deve-se, neste caso fornecer uma frequência de clock do Plasma aproximadamente 5 milhões de vezes menor que o da placa.
- B) A chave SW17 realiza o reset do Plasma.
- C) A chave SW16 seleciona duas possibilidades de visualização de dados nos displays de 7 segmentos: a) os dados escritos na memória na execução de main (sel=0) ; b) dados enviados ao periférico durante a interrupção (sel=1);

- D) O valor de entrada externo para a interrupção é dado pelas chaves SW15 a SW0 (16 bits); como a rotina de interrupção o envia à memória RAM e depois escreve no periférico, este valor é verificável nos displays;
- E) 8 displays de 7-segmentos com valores Hexa;
- F) O pino de acesso externo ext (que indica a disponibilidade de dados de entrada) é acionado pelo botão KEY0;



- 2) Copiar para **X:\psi2553\exp3\Plasma Interrupt DE2** todos os arquivos que você simulou anteriormente e que estão em **X:\psi2553\exp3\ Plasma_com_Perif_Exp3**, incluindo os quatro arquivos codex.hex.
- 3) Carregue o programa *Quartus II* da Altera para fazer síntese do sistema **plasmainDE2.vhd** através do seguinte caminho: **Iniciar/Programas/Altera/Quartus II 9.1**.
- 4) Crie um projeto como o nome *plasmainDE2*. Para isto, use o **New Project Wizard**. Como feito anteriormente.
- 5) Antes de compilar, faça a designação de pinos. Use o *assignment editor*, clicando em menu **Assignment> Import Assignments...** Inclua o arquivo **DE2_pin_assignments.csv**.
- 6) Faça a compilação completa.
- 7) Antes de programar o dispositivo é necessário fazer a configuração do cabo. Para isto execute as seguintes operações:
 - a) Ligue o cabo na porta USB do PC.

- b) Ligue a outra extremidade na placa de prototipagem.
- c) Ligue a fonte na placa e na tomada, energizando o conjunto.
- d) Deixe todas as chaves SW_n em '0'.
- e) Pressione o Botão Vermelho.

Atenção: como o FPGA é baseado na tecnologia **SRAM**, se a fonte for desligada (por qualquer motivo) o dispositivo terá de ser configurado novamente.

- 8) Clique em Quartus II no menu **Tools > Programmer**. No parâmetro **Mode** selecione **JTAG**. Clique na opção **Hardware Setup > Hardware Settings > Add Hardware**; escolha a opção **USB-BLASTER(USB-0)**. Clique em **OK** e **Close**
- 9) Na coluna **File** deve estar o arquivo *plasmainDE2.sof*, resultado da compilação. Caso não estiver, para adicionar este arquivo clique na opção **Add File**. Confirme com **OK**. O dispositivo (*EP2C35F672C6*) foi carregado na coluna dispositivo. Selecione a caixa na coluna **Program/Configure**.

Pronto! Seu “chip” deverá estar funcionando.

- 10) Acione a opção Start para programar o circuito no FPGA.

ATENÇÃO : Neste momento, o aluno deverá testar o seu sistema rodando **count_2552.c** e observar o seu funcionamento correto em caso de interrupção.

- 11) Acione a chave SW17 para efetivar o reset do processador (primeiramente em '1' e depois em '0').
- 12) Aguarde alguns segundos e você verá os displays apontarem alguns valores que variam.

ATENÇÃO: Você perceberá que os resultados de saída observados mudam muito rápido. Diminua a frequência de operação do relógio no plasma alterando o fator de divisão em **div_freq.vhd** (na realidade, tal controle é feito no **GENERIC** da entidade **topo** em **plasmainDE2.vhd**). Faça ali a modificação, escolhendo uma opção, de tal forma que você possa identificar claramente os valores nos displays, em um passo mais lento (fique ciente que a parte de inicialização do programa ficará razoavelmente mais lento...).

- 13) Com o novo valor de frequência, refaça a compilação no Quartus e a programação, além dos itens 11 e 12 acima.
- 14) Agora, interprete o que o você está vendo nos displays (lembrar que a chave SW16 está em '0').
- 15) Programe as chaves SW15 a SW0 com um dos dois números que você utilizou para os experimentos de simulação de interrupção.
- 16) Aperte o botão de habilitação do dado externo (ext) para interrupção (KEY0). Use a chave SW16 para mudar o modo de observação.

A demonstração ao instrutor é obrigatória!!

Tenha certeza que consegue indentificar e explicar todos os números das sequências que aparecem nos displays, em cada uma das duas etapas/situações abaixo:

- 1) quando o main é rodado;
- 2) quando a interrupção é acionada e a ISR é chamada.