Universidade Federal de Santa Catarina - Centro Tecnológico - Departamento de Informática e Estatística
INE 5411 - Organização de Computadores
Roteiro do Laboratório 6 - Tratamento de exceções
Pré-requisitos para a compreensão e execução do laboratório: interrupções e exceções.

1. Objetivo

O objetivo desta aula é implementar e testar um tratador de exceções utilizando o suporte do coprocessador 0 da arquitetura do MIPS. O estudo de caso proposto consiste de dois programas:

- O programa aplicativo `calcFibonacci`, que implementa o cálculo de um número da série de Fibonacci;
- Um programa tratador de exceções denominado `exceptionHandler`, o qual é responsável pela leitura de valores digitados no teclado e também por lidar com problemas de *overflow*.

Os códigos em linguagem de montagem do aplicativo `calcFibonacci` e do tratador de exceções/interrupções serão fornecidos de forma parcial e deverão ser completados durante esta aula.

2. Programa aplicativo: calcFibonacci

Os números de Fibonacci são números inteiros pertencentes à seguinte sequência: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55...
Por definição, os dois primeiros números da série, F_0 e F_1, são iguais a zero e um, respectivamente. Os demais números são obtidos de forma recursiva usando a seguinte fórmula: $F_n = F_{n-1} + F_{n-2}$. Assim, observa-se que um número da série de Fibonacci é igual à soma dos dois números que o precedem em tal série.

O aplicativo `calcFibonacci` calcula iterativamente o número de índice *n* da série de Fibonacci, armazenando o resultado na posição de memória indicada pelo label "resultado". O índice *n* é digitado pelo usuário a partir do teclado. A leitura e o processamento dos caracteres digitados são feitos pelo tratador de exceções para que o andamento do programa não seja afetado durante o cálculo de um número. O usuário deve digitar o índice do número da série de Fibonacci que deve ser calculado e depois pressionar ENTER para que o tratador de exceções transfira o valor digitado para uma posição de memória conhecida pelo programa principal (posição de memória essa indicada pelo label "argumento"). **Observação:** o índice pode ter mais de um dígito como, por exemplo, 157.

O programa em linguagem de montagem `calcFibonacci` está disponível no Moodle.

Entrada de dados com o auxílio de interrupções

Para lidar com as operações de entrada e saída, **o MARS simula a existência de um teclado e um display usando um esquema de entrada/saída (E/S) mapeado em memória**. Com isso, a escrita em uma determinada posição de memória (0xFFFF000C) resulta na impressão do caractere ASCII correspondente no display, enquanto a digitação de um caractere no teclado resulta no armazenamento do valor ASCII correspondente em uma determinada posição de memória (0xFFFF0004). Além disso, os valores (*words*) armazenados nas posições de memória 0xFFFF0000 e 0xFFFF0008 são utilizados para controlar as operações de E/S. **A área de memória entre 0xFFFF0000 e 0xFFFF000F é denominada como MMIO no MARS.**

Neste experimento, o esquema de E/S disponível no MARS será utilizado para interação com o usuário do programa `calcFibonacci`. Com respeito à apresentação de informações no display, serão realizadas escritas diretas à posição de memória 0xFFFF000C para impressão de caracteres. Por outro lado, para a leitura de caracteres do teclado, será utilizado o mecanismo de geração de interrupções disponível no esquema de E/S do MARS. **O valor armazenado na posição de memória 0xFFFF0000 é quem controla o funcionamento desse mecanismo.** Conforme ilustrado na Figura 1, apenas os bits 0 e 1 de tal valor são utilizados, sendo o bit 1 o habilitador de interrupções e o bit 0 o indicador que um caractere está disponível na memória para ser lido. Uma vez colocado o bit 1 em nível lógico alto, cada tecla pressionada - com o **Keyboard and Display Simulator** em funcionamento - irá gerar uma chamada ao tratador de exceções e o armazenamento do valor digitado na posição de memória 0xFFFF0004.

Dica: Não confunda os endereços relacionados a MMIO com os registradores Cause e Status.

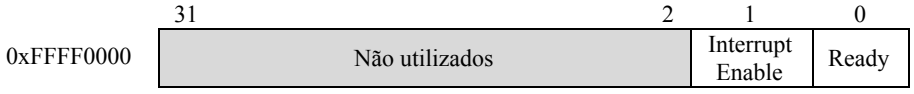

Figura 1. Ilustração da word que controla as operações de entrada pelo teclado no MARS.

3. Programa tratador de exceções: exceptionHandler

O tratador de exceções consiste das seguintes etapas:

- **Etapa 1 - Salvamento de contexto:** Os registradores usados pelo tratador são preservados, o que não pode ser feito usando a pilha.
- **Etapa 2 - Decodificação do registrador de causa:** O objetivo é descobrir a causa da exceção para realizar a ação apropriada (por exemplo, se for uma interrupção, fazer a leitura e o processamento do valor digitado).
- **Etapa 3 – Tratamento de interrupção:** Leitura da posição de memória 0xFFFF0004 e processamento do valor digitado. Isso é feito chamando um procedimento **lechar**.

- **Etapa 4 - Tratamento de *overflow*:** O cálculo de um número da série de Fibonacci pode resultar em *overflow* em função do índice digitado. Nesse caso, é preciso tratar o *overflow*, o que deve ser feito reiniciando o programa na **Etapa 7**, o que acabará imprimindo um "**I**" no display do usuário.
- **Etapa 5 - Preparação do sistema para novas exceções:** Consiste na atualização dos registradores `Cause` e `Status` para habilitar o tratamento de novas exceções que vierem a ser detectadas.
- **Etapa 6 - Restauração de contexto:** Recuperação dos valores de registradores preservados.
- **Etapa 7 - Retorno ao fluxo normal de execução ou ao início do programa:** No caso da exceção gerada pelo uso do teclado (interrupção), voltar à execução da instrução que deveria ter sido executada. No caso de *overflow*, reiniciar o programa.

Um esboço do código do tratador em linguagem de montagem, que deve ser completado, está disponível no Moodle.

Exercício 1: implementação e teste do programa e do tratador

Faça o download dos arquivos calcFibonacci.asm e exceptionHandler.asm. Abra os dois arquivos no MARS e inclua o exceptionHandler.asm como tratador de interrupções (utilize o menu ***Settings***, opção ***Exception Handler...***).

a) No arquivo calcFibonacci.asm, complete a parte indicada no início do programa de tal forma que interrupções sejam geradas a partir de entradas no teclado. **Restrição: Ao habilitar interrupções, você deve preservar todos os bits que nada tenham a ver com essa habilitação.**

b) Responda à Questão 1.1 do relatório de aula.

c) No arquivo exceptionHandler.asm, implemente as **Etapas 2, 3, 4 e 5** do tratador obedecendo às convenções adotadas e às orientações disponíveis no próprio código. No código do tratador, use somente os registradores `$k0`, `$k1`, `$t0`, `$t1` (além do registrador `$ra` para o mecanismo de chamada de procedimento).
Dica: se você tiver a impressão de que precisa de mais registradores, lembre-se de verificar se um dos registradores-fonte pode virar destino; lembre-se também de diminuir a distância entre a definição de um valor como destino e o seu uso como fonte (essa diminuição do intervalo de vida de um valor aumenta as chances de reuso de registradores).

d) Com o programa e o tratador prontos, teste o seu funcionamento. Para tal, faça a montagem do programa calcFibonacci e inicie a sua execução com uma velocidade de 20 instruções por segundo (você pode utilizar uma velocidade menor ou maior se for de sua preferência). Em seguida, abra o **Keyboard and Display Simulator** (acessível a partir do menu **Tools** do MARS), desmarque a opção "DAD" e clique no botão "Connect to MIPS". Você pode, com esse simulador, utilizar as teclas para digitar valores a partir da janela "Keyboard" e observar valores enviados pelo programa a partir da janela "Display". No caso do programa considerado neste experimento, você pode digitar números de 0 a 9 no teclado e também a tecla ENTER. **Atenção: ao digitar um número ou ENTER, aguarde o seu aparecimento na janela "Display" para digitar algo novamente**. Faça os ajustes necessários para o funcionamento adequado do programa.

e) Responda às Questões 2.1 a 2.6 do relatório de aula.