

O Processador (CPU)

Exceções/Interrupções

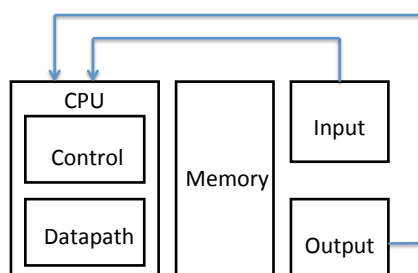
António de Brito Ferrari
ferrari@ua.pt

Exceções/Interrupções

- **Exceção**: evento que modifica o fluxo normal de execução do programa corrente
- Dois tipos de eventos:
 - Internos ao processador, gerados pela execução do programa
 - Externos, gerados por periféricos - **interrupções**
- MIPS: eventos internos – **exceções** (em sentido estrito)
- Outras nomenclaturas:
 - Motorola: eventos internos – **traps**
 - Intel: **interrupção** designa exceções de qualquer tipo

Interrupções

Linhas de pedido de interrupção dos periféricos – se as interrupções estiverem *enabled* CPU termina execução da instrução corrente e “salta” para a execução da rotina de tratamento das interrupções (*interrupt handler*)



Interrupções – tema de AC II em que é tratado o sistema de Entradas/Saídas (I/O)

ABF - AC1_Exceções

3

Exceções (internas)

- Usadas para tratar situações excepcionais que ocorrem na execução de um programa:
 - **Overflow** em operações aritméticas
 - **Illegal Instruction** - tentativa de executar instrução indefinida
 - *system call* - invocação pelo programa dos serviços do sistema de operação
 - *page fault* num sistema de memória virtual, ...

ABF - AC1_Exceções

4

Processamento de Exceções

Após a ocorrência de uma exceção:

1. Processador:
 - a. termina instrução corrente (exceção externa) **ou**
 - b. cancela execução da instrução corrente (exceção interna)reconhece a exceção e inibe novas exceções
2. Processador salvaguarda contexto, i.e. a informação mínima que permita retomar a execução do programa (PC, ...)
3. PC = endereço de subrotina de tratamento da exceção (***exception handler***)

ABF - AC1_Exceções

5

Processamento de Exceções (2)

- Análogo à invocação de uma subrotina - controlo passa do programa que está a ser executado para um outro que vai tratar a exceção – ***exception handler***
- Diferença: a execução da subrotina não é desencadeada por uma instrução de invocação de subrotina do programa em execução mas por um mecanismo externo ao programa

ABF - AC1_Exceções

6

Processamento de Excepções (3)

- Questões:
 1. Onde salvar o contexto?
 2. Qual a informação a salvar?
 3. Como obter o endereço do *exception handler*?
- Arquitecturas diferentes dão respostas diferentes a estas questões

ABF - AC1_Excepções

7

Excepções

- Como obter o endereço da subrotina de tratamento das excepções (*exception handler*)?
- Alternativas:
 1. Excepções vectorizadas – cada excepção tem associado um *vector da excepção* (***interrupt vector***) que é fornecido ao processador quando este reconhece (*acknowledge*) a excepção, e que é usado para indexar uma tabela que contém os endereços de entrada dos *exception handlers*
 2. Excepções não-vectorizadas - *exception handler* único, comum a todas as excepções: identificação de qual a causa da excepção a cargo do *exception handler*

ABF - AC1_Excepções

8

Exceções – informação a preservar

1. Qual a informação a salvar?
 - Mínima para reduzir tempo de resposta à exceção (crítico em *sistemas tempo real*):
MIPS: apenas PC e modo de execução
2. Onde a salvar?
 - **EPC** guarda conteúdo PC
 - **Status Register** guarda modo de execução do programa (modo: Kernel/User; interrupções: enabled/disabled)

ABF - AC1_Exceções

9

Processamento das Exceções: MIPS

- Duas ações básicas:
 1. Guardar o endereço da instrução que causou a exceção no **Exception Program Counter (EPC)**
 2. Transferir o controle para o sistema de operação num endereço especificado (**0x80000080** no MIPS)
- Subrotina de tratamento das exceções (*exception handler*) necessita de identificar a causa da exceção:
 - Cause Register** – permite identificar o tipo de exceção que ocorreu

ABF - AC1_Exceções

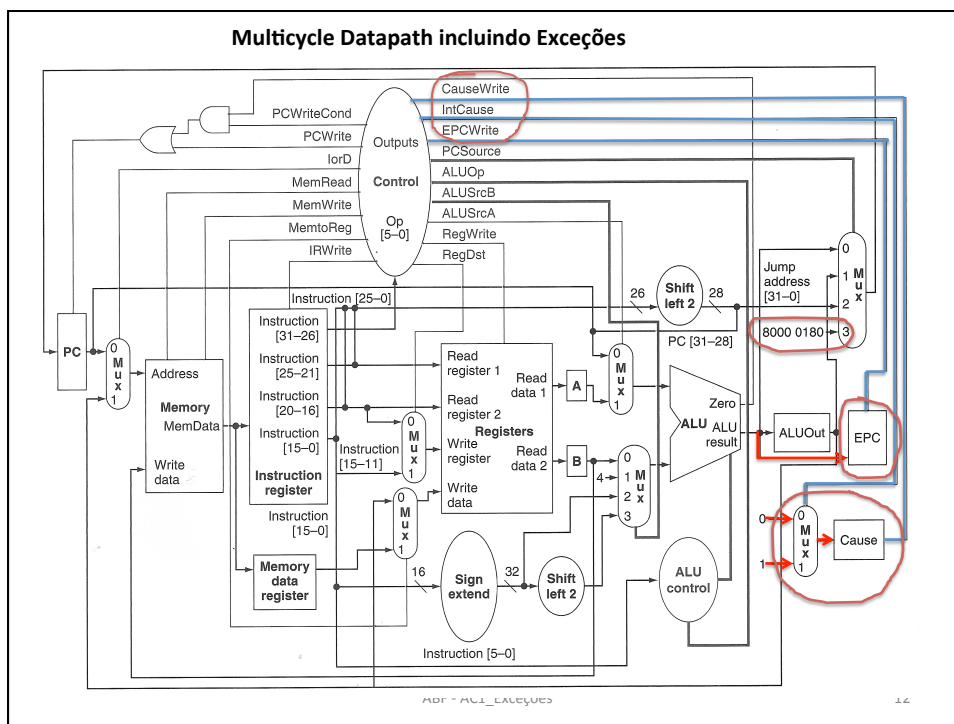
10

Adições ao Datapath

- **EPC** – Exception Program Counter
- **Cause Register** – permite que a subrotina de tratamento de exceções identifique a causa da exceção que ocorreu
 - Registo de 32-bits dos quais só alguns têm significado atribuído (os restantes não são usados)
 - Para nós apenas o bit menos significativo tem significado:
 - **Instrução indefinida = 0**
 - **Overflow = 1**

ABF - AC1_Exceções

11



Illegal Instruction e Overflow

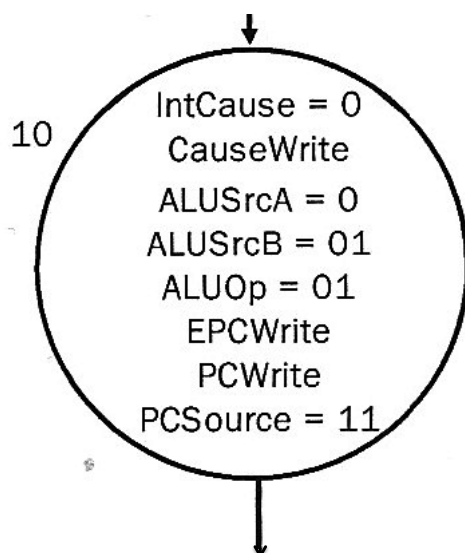
- Guardar o endereço da instrução que causou a exceção no **Exception Program Counter (EPC)**
 - PC aponta para a instrução seguinte, logo:

$$\text{EPC} = \text{PC} - 4$$
- Sinais de Controlo:
 - ALUSrcA = 0 (seleciona PC como 1ª entrada da ALU)
 - AluSrcB = 01 (seleciona 4 como 2ª entrada da ALU)
 - ALUOp = 01 (subtract)
 - EPCWrite
- **PC = 0x80000080** (endereço do *exception handler*)
 - Sinais de Controlo:
 - PCSource = 11
 - PCWrite
- Escrever no **Cause Register** a causa da exceção
 - CauseWrite

ABF - AC1_Exceções

13

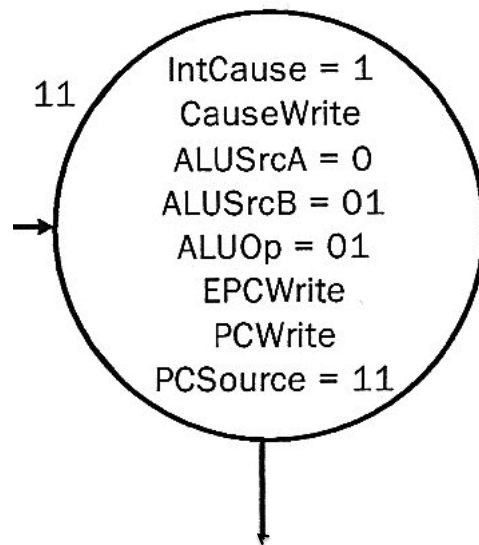
Illegal Instruction: IntCause = 0



ABF - AC1_Exceções

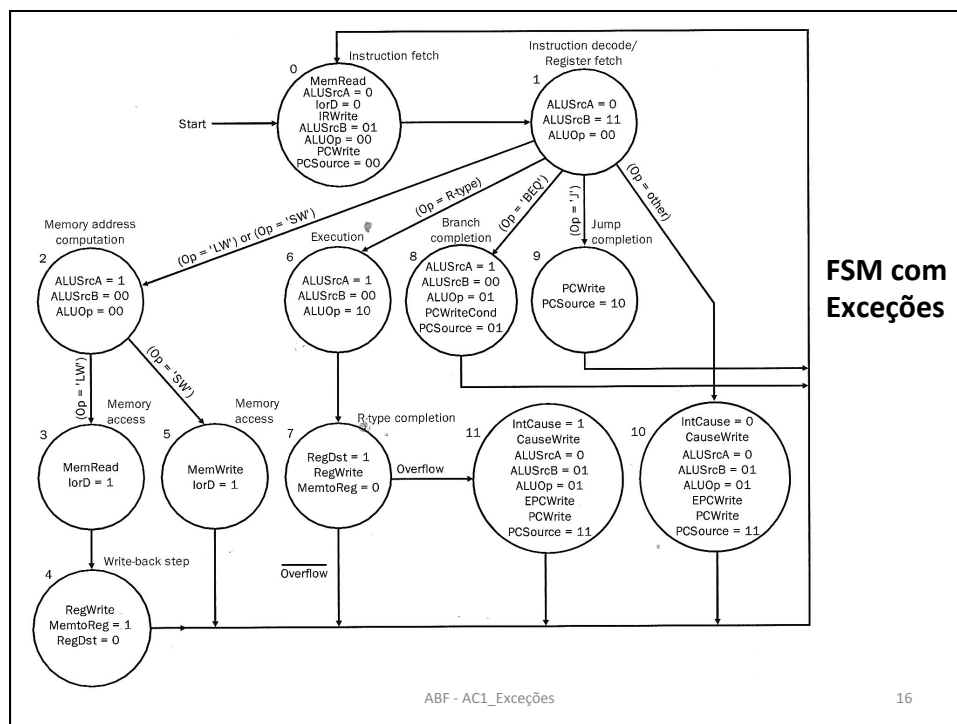
14

Overflow: IntCause = 1



ABF - AC1_Exceções

15



ABF - AC1_Exceções

16

Desenho do CPU Multiycle terminado.
Como construir um processador mais rápido?

PIPELINING