

Outline

- Introduction
- Architecture of a RISC Processor
- Implementation**



Pirouz Bazargan Sabet

December 2008

Implementation

- RISC vs. CISC concept
- Concept of pipeline
- An implementation of Mips
- Pipeline's problems**



Pirouz Bazargan Sabet

December 2008

Exception / Reset

Implementation of the :

- Reset mechanism
- Interrupt mechanism
- Exception mechanism**



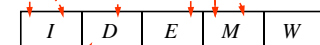
Pirouz Bazargan Sabet

December 2008

Exception

How stop the pipeline ?

- Overflow
- Illegal read address
- Illegal write address
- Coprocessor unusable
- Unknown instruction
- Syscall
- Break
- Trap
- Data bus error
- Instruction bus error
- Machine check

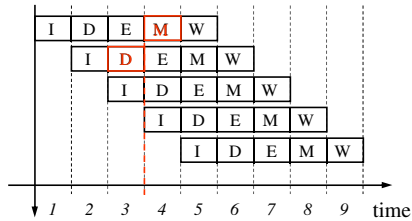


Pirouz Bazargan Sabet

December 2008

Exception

How stop the pipeline ?



Exceptions should be seen in the same order as the instructions (exact exception)

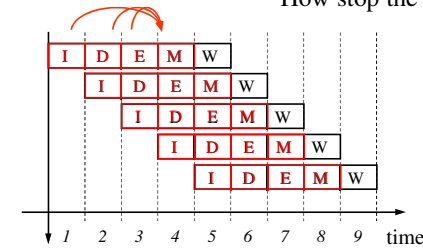


Pirouz Bazargan Sabet

December 2008

Exception

How stop the pipeline ?



Forward all the sources to the same cycle

Postpone the reaction to an exception



Pirouz Bazargan Sabet

December 2008

Exception

When stop the pipeline ?

Exceptions are events that denote a malfunction in the program

An exception is an error
 ⇒ the faulty instruction should NOT be executed

A faulty instruction should not modify the state of the system ... from the software point of view

Protect software visible registers and the memory

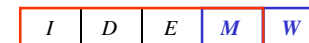


Pirouz Bazargan Sabet

December 2008

Exception

When stop the pipeline ?



Pirouz Bazargan Sabet

December 2008

Exception

How stop the pipeline ?

- Overflow
- Illegal read address
- Illegal write address
- Coprocessor unusable
- Unknown instruction
- Syscall
- Break
- Trap
- Data bus error
- Instruction bus error
- Machine check

The diagram shows a list of exception types on the left. Dashed lines of various colors (blue, red, orange) connect these exceptions to specific stages of a five-stage pipeline (I, D, E, M, W) shown at the bottom. For example, 'Overflow' and 'Illegal read address' point to the Instruction (I) stage, while 'Data bus error' points to the Memory (M) stage.

LIP 6 Pirouz Bazargan Sabet December 2008

Exception

When stop the pipeline ?

The diagram shows a horizontal pipeline with five stages: I, D, E, M, and W. A vertical red line is drawn between the E and M stages, indicating an exception point. The M and W stages are highlighted with a blue background.

LIP 6 Pirouz Bazargan Sabet December 2008

Exception

When stop the pipeline ?

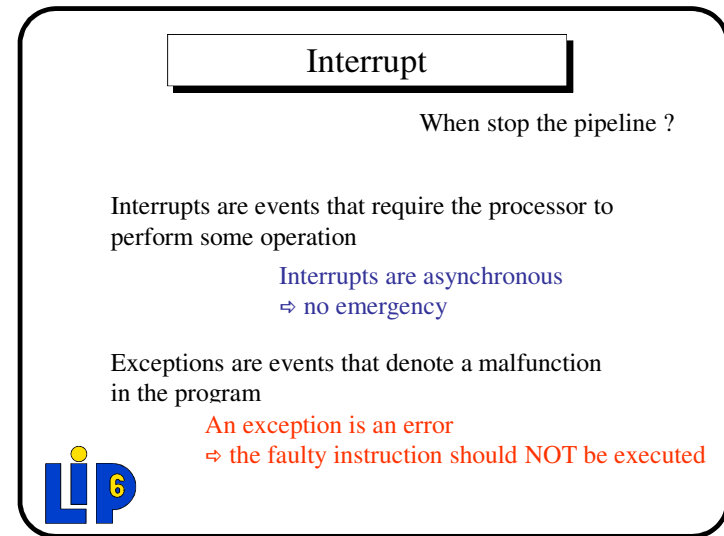
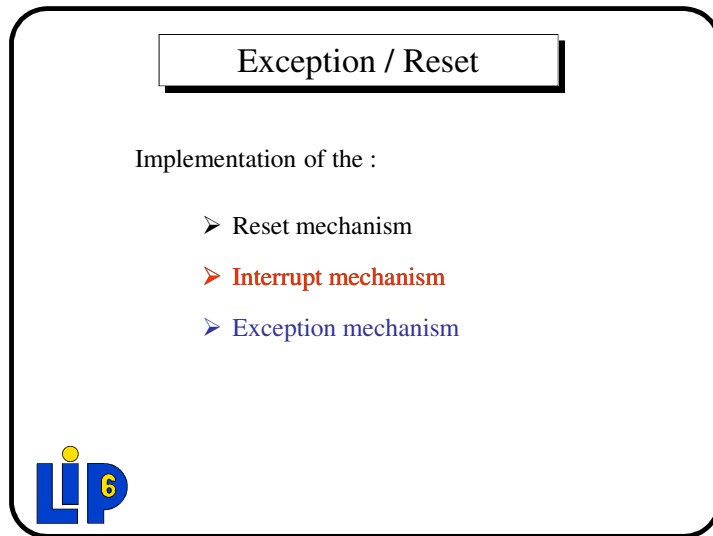
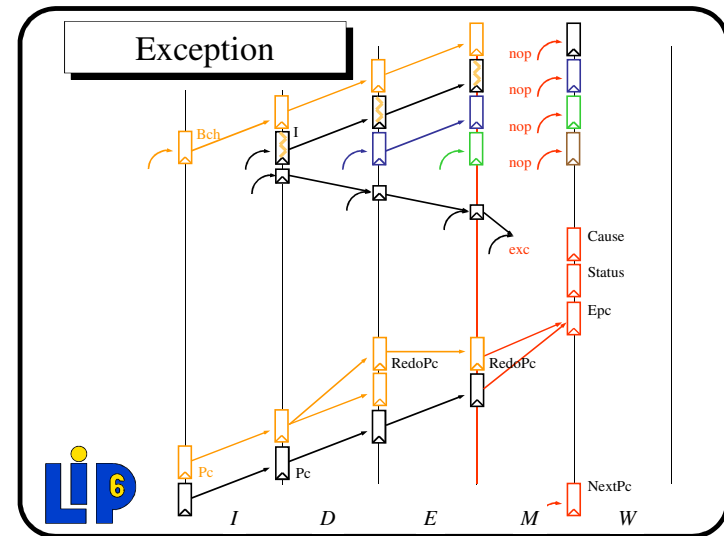
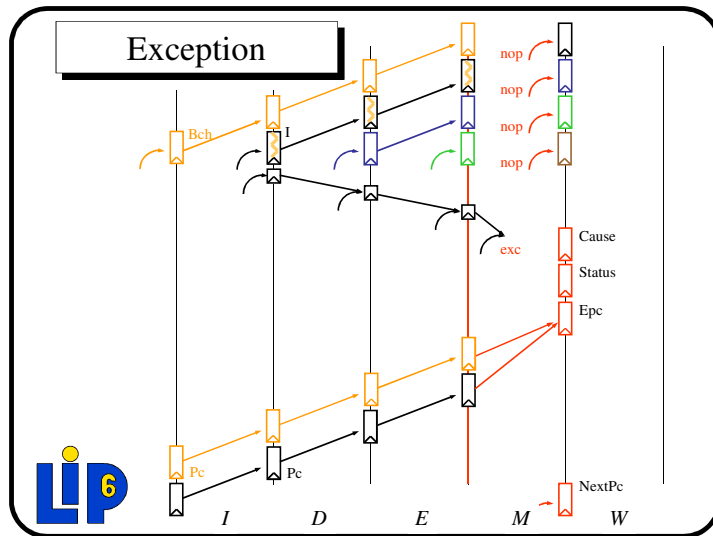
The diagram shows a pipeline with five stages (I, D, E, M, W) over a time axis from 1 to 9. A red vertical line at time 5 indicates an exception. The pipeline state at each time step is shown as a row of boxes. At time 5, the M stage of the previous instruction is still present, and the next instruction has not yet started its I stage.

LIP 6 Pirouz Bazargan Sabet December 2008

Exception

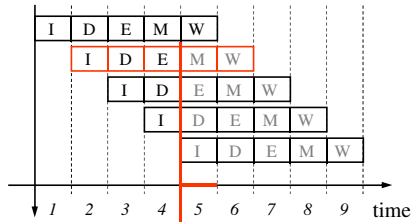
This diagram illustrates the exception handling process. It shows a pipeline with stages I, D, E, M, and W. An exception (exc) is triggered during the M stage. The diagram shows the state of the pipeline at each stage, with arrows indicating the flow of data and the exception signal. The exception signal is shown as a red line that branches off from the M stage and connects to a block labeled 'Cause Status Epc'. The 'NextPc' (Next Program Counter) is also shown, indicating the address of the instruction following the one that caused the exception.

LIP 6 Pirouz Bazargan Sabet December 2008



Interrupt

When stop the pipeline ?



Exception / Reset

Implementation of the :

- Reset mechanism
- Interrupt mechanism
- Exception mechanism



Exception / Reset

Reset Abort the current program
Jump to the Reset Handler

- Initialize the address of the next instruction : **0xBFC0 0000**
- Initialize the Status Register : **0x0040 0004**
- Initialize the Cause Register : **0x0000 0000**
- Save the return address into Eepc
- Initialize the Exception Base Register : **0x8000 0000**



Reset

