Interactive High-Performance Processor Understanding Via the Web

→ LEVOVIS

Augustus K. Uht
Sean Langford
Dept. of Electrical and
Computer Engineering
University of Rhode Island

&

David Morano
Dept. of Electrical and
Computer Engineering
Northeastern University

SSGRRw January 23, 2002

Acknowledgements

- Work supported by the Intel Corporation, URI, NSF, Mentor Graphics, Xilinx, VCC.
- Simulator by Dave Morano, Ali Khalafi and Marcos de Alba.
- Other members of the Levo team:
  - Tom Wenisch, Prof. David Kaeli (NEU)
- Constant advice and editing:
  - Laurette Bradley

Outline

1 Motivation

2 CPU Performance Basics

3 Instruction Level Parallelism (ILP)

4 LEVO CPU Overview

5 LEVOVIS Architecture

6 LEVOVIS Examples

7 LEVOVIS Online or Offline Demo

8 Summary

Motivation

- CPU chip *complexity* high and growing
  - 10's of millions of transistors → billions
- → *functional verification* costs growing
- → *time-to-market* excessive
- → *education* difficult
- → *debugging* difficult

Related Work

- Many specialized simulators, e.g., IBM BRAT
  - Not readily adaptable to other machines
- "General Purpose" visualizers, e.g., Stanford Rivet
  - May be adaptable to many types of systems
  - Scalability is an issue: much state
  - Often, adaptability is through custom scripts
    - LevoVis based on std. XML and SVG
  - Not readily accessible; LevoVis is Web-based

CPU Performance Basics

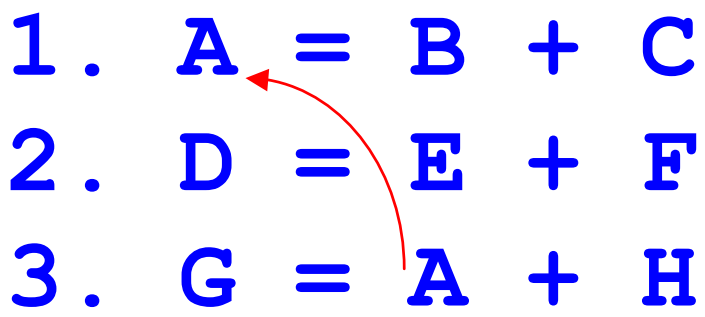
- Two elements to processor performance - *P*:
  - Clock frequency – *f*
    - Technology dependent
  - Instructions executed per cycle – *IPC*
    - Architecture and Implementation dependent
- Fundamental relation:

$$P = f * IPC$$

- LEVO focuses on high IPC via ILP →

Instruction Level Parallelism (ILP)

- Execute more than 1 instruction per cycle
- Example:



instructions 1 and 2 can execute in parallel;

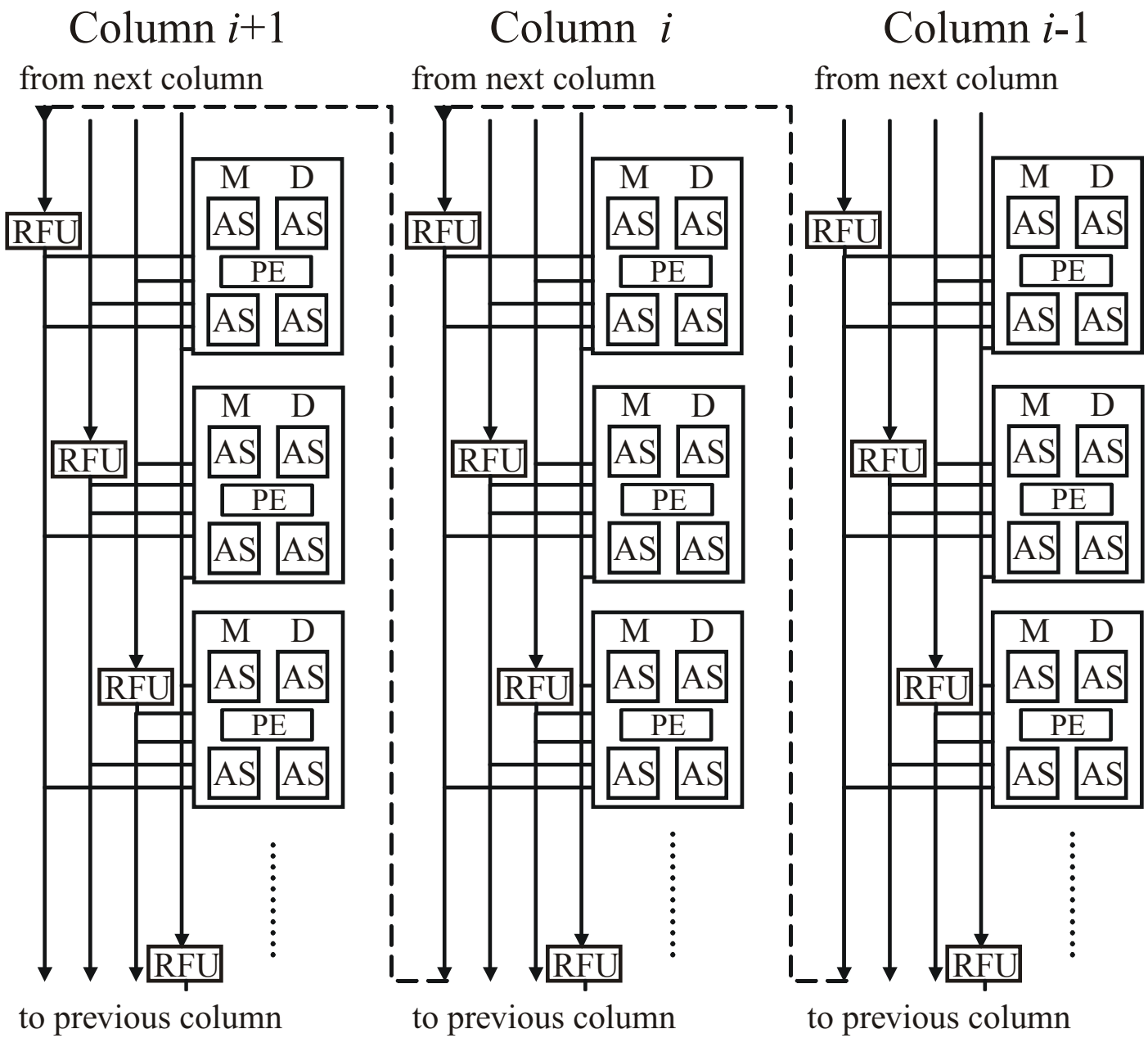
1 and 3 cannot (data dependency)

LEVO CPU Overview

- Uses modification of Tomasulo algorithm
  - (The original algorithm dates to 1964 and is used today in the Intel Pentium Pro, II, III & 4.)
- Instruction *time tags* enforce dependencies
- *Active Stations (AS)* hold instructions & data
- Communication buses segmented
  - →Short delay, high *f*
  - *Register Filter/Forwarding Units (RFUs)* link segments

LEVO

Archi-tecture

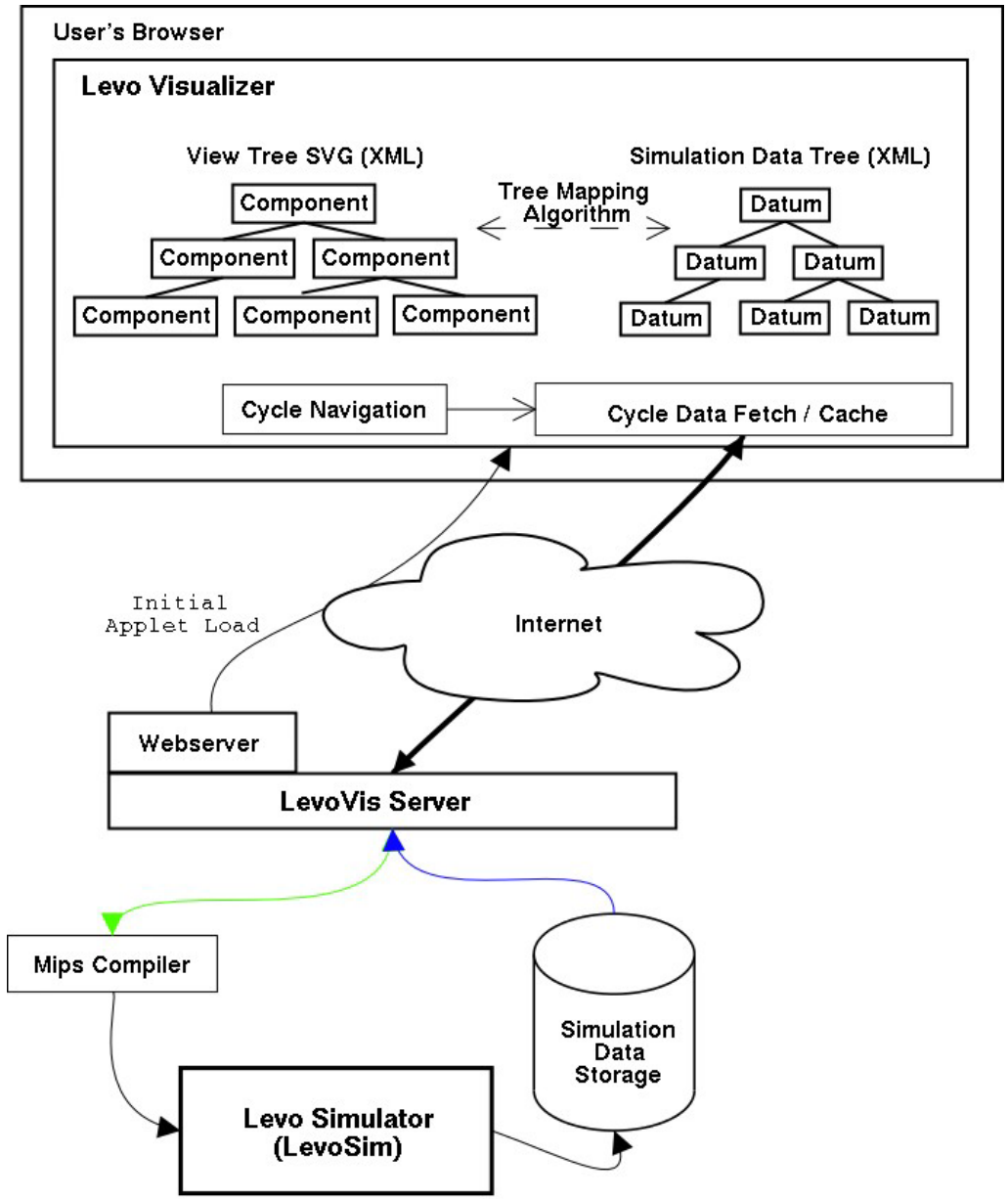


LEVOVIS Overview

- Based on XML and SVG
- SVG used to generate graphics
- XML links graphics with simulation data
- Simulation data files very large
  - → Kept on server
  - Data for individual cycles (10) brought over Web *as needed*
- *User able to navigate to arbitrary cycles*
- *Arbitrary display of machine elements*

LEVOVIS

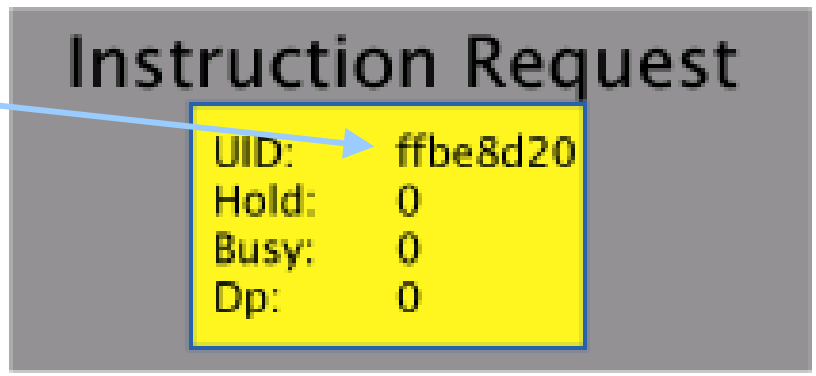
Architecture



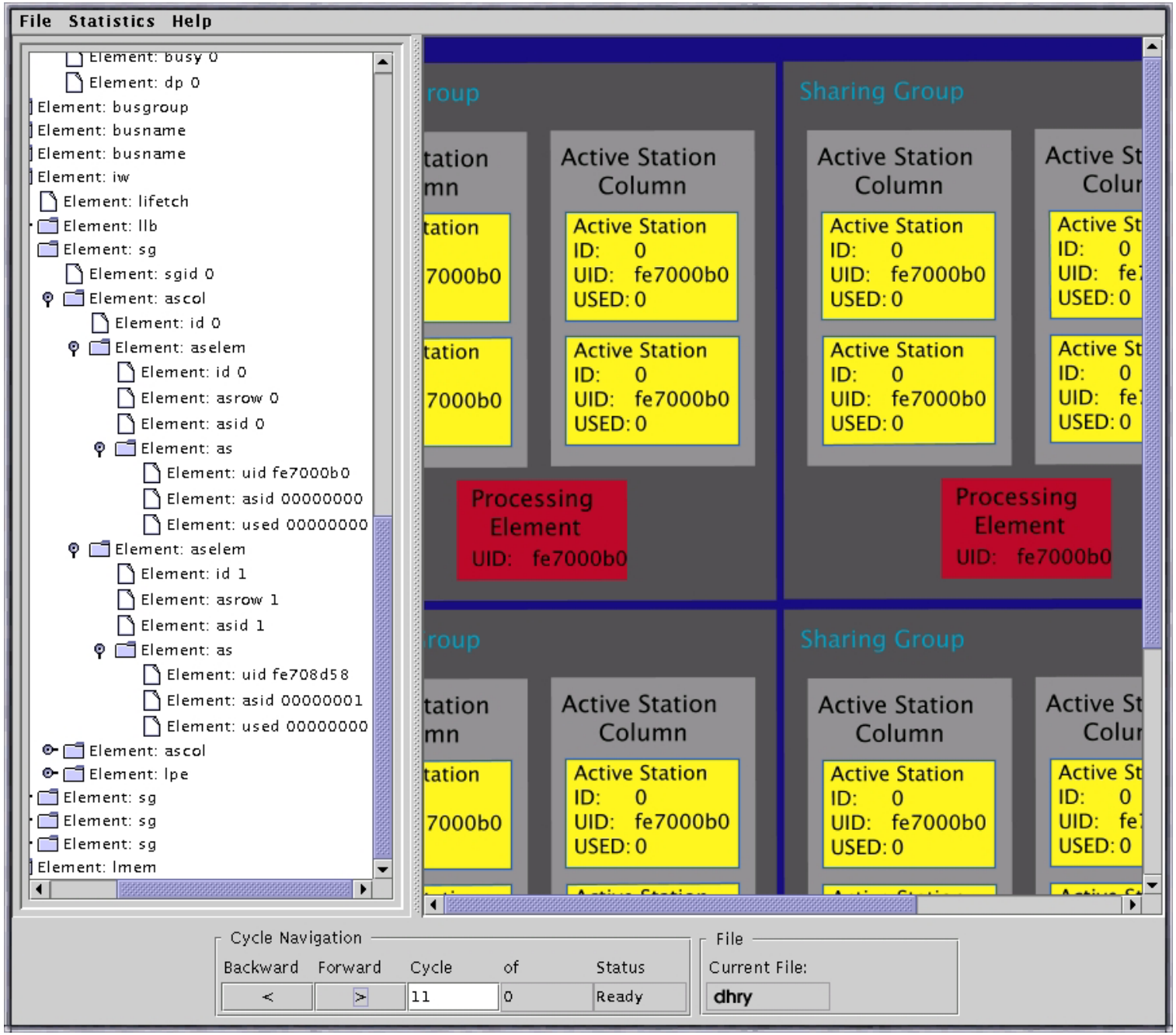
Data-to-Graphic Mapping Example

Simulation Data XML	SVG Graphic Component XML
<uid>	<g id="uid">
ffbe8d20	<tspan x="36" y="0">ffbe8d20</tspan>
</uid>	</g>

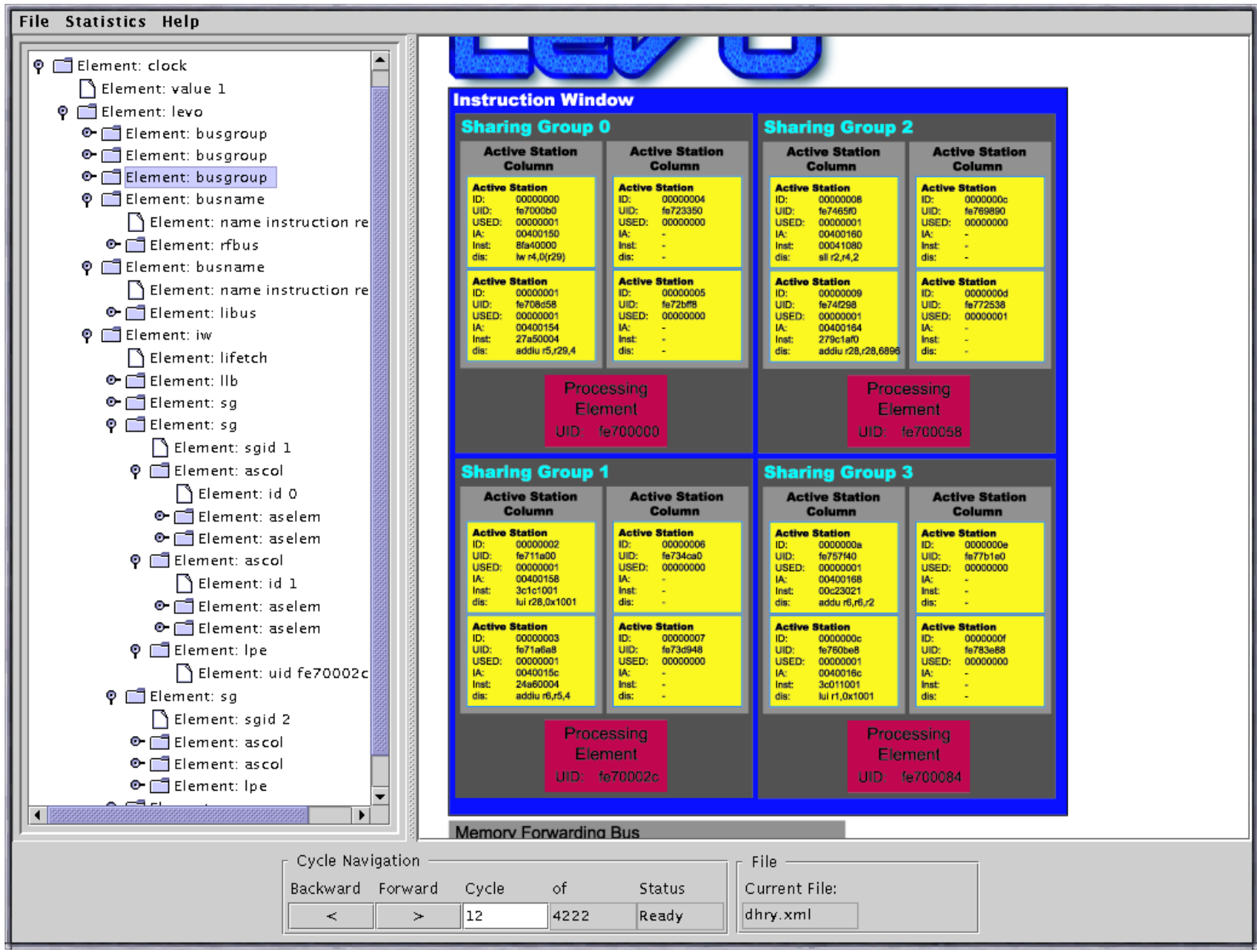
Gives



LEVOVIS Example 1



LEVO Example 2



Online or Offline Demonstration

- ONLINE: Goto → WWW LevoVis

- OFFLINE: Goto → local LevoVis

Summary

- LEVOVIS:
  - Many *flexible state visualization* capabilities
  - Ideal for *complex CPU's*
  - Usable for *any synchronous digital system*
  - Gives *understanding*, *analysis* and *debugging* for researchers, students and engineers
  - Allows *world-wide concurrent access* (Web-based)
  - *Adaptable* to new systems or system requirements

Relevant Web Sites

Levo links:

www.ele.uri.edu/~uht

Or: www.levo.org

LevoVis direct:

ovel.ele.uri.edu:8080