Web-Based Simulators of

Embedded Software for Programmable Digital Signal Processors

Prof. Brian L. Evans

Department of Electrical and Computer Engineering
The University of Texas at Austin, Austin, Texas

bevans@ece.utexas.edu

Research conducted in collaboration with

Raza Ahmed, *Luis Gutierrez*, and *Dr. José Luis Pino* at UC Berkeley
Dogu Arifler and *Chris Moy* at UT Austin
Keith Larson at Texas Instruments

Presented at the Ptolemy Miniconference
University of California at Berkeley, March 14, 1997.

Outline

Web-Based System-Level Design Tools
Synthesis of Efficient Implementations
Embedded Systems
Programmable Digital Signal Processors (DSPs)
Goals for Standalone DSP Simulators
Impact on DSP Simulators
The C30 Simulator
Impact on Ptolemy
Web-Based Design Tools
Web-Based Configurable Simulators
Conclusion

Web-Based System-Level Design Tools

System-Level Design

- Management of complexity and heterogeneity

- Simulation
- Synthesis

The Web as an Electronic Design Automation Tool

- Distributed design tools
- Object-oriented design
- Desktop access to the latest tools [WELD]

Synthesis of Efficient Implementations

Implementation Cost Metrics

- Software costs such as data size, program size, throughput, and power consumption
- Hardware costs such as area, weight, throughput, and power consumption

Use of Cost Metrics in Design Space Exploration

- Multiprocessor scheduling [Ptolemy Project Papers] [Architecture Trade Tool]
- Hardware/software codesign [Kalavade and Lee] [POLIS]
- Low-power design [Survey] [Hyper] [Concept] [Pleiades]
- Benchmarking [Berkeley Design Technology]

Embedded Systems

Hidden from the User

Products with Embedded Systems

- Audio CD players and music synthesizers
- Disk drives
- Digital cellular phones
- Sonar and radar
- Video disk players and video telephones

Perform Signal Processing, Communications, and Control

Embedded Software Systems [Graduate Course]

Programmable Digital Signal Processors (DSPs)

Common Properties [Talk on DSP Hardware]

- Fall between microcontrollers and the fastest general-purpose processors in terms of processing power and power consumption
- One cycle execution for most instructions
- Zero-overhead looping
- Hardware multiplier(s)
- Extended precision accumulator
- Separate data and program memory on separate busses
- Special addressing modes
- Various data formats:
  - fixed-point such as the TMS320C50 and Motorola 56000 families (comparison)
  - non-IEEE floating-point such as the TMS320C30 family
  - IEEE floating-point such as the ADSP-21020
- Pipelines with hardware interlocking protection
- Compiler tools are generally not very efficient

Goals for Standalone DSP Simulators

Support System-Level Design Tools

- Leverage existing simulators or disassemblers
- Validate them to be cycle-accurate and bit-true
- Extend them to support different boards and hypothetical configurations
- Return implementation cost estimates
- Make portable across computer platforms
- Report pipeline hazards

Support for Control by a Parent Process

- Separate the user interface from the kernel
- Add hooks to configure the parent process(es) that controls it
- Support pipes by flushing all output

Impact on DSP Simulators

Plans for DSP Simulator Development

Tools Available by FTP	*Simulator Available?*	*Current Work at UT*	*Future Work at UT*
C30 DSK Tools (1) [Texas Instruments]	No	C30 Simulator (2)	C30 Simulator (3)
C20 DSK Tools [Texas Instruments]	No	-	C50 Simulator (2)(3)
56000 Tools [Motorola]	Yes	-	56000 Simulator (2)(3)

1. DSK stands for Digital Signal Processing Starter Kit
2. Will return program, data, and execution time
3. Will return estimates of power consumption

The C30 Simulator

Validation

- Based on the validated disassembler for the C30 DSK board
- Written in C++ that has been run through Purify
- Run simulator in lock step with the C30 DSK board to verify state

Interaction

- Can be controlled by a parent process
- User can use textual commands

Portability

- Runs under MS-DOS, Windows, and Unix

Features:

- Instruction set architecture simulation
- Supports the C30 non-IEEE floating-point format
- Models state of the C30 processor including the pipeline

Speed

- Simulates about 80,000 C30 instructions/sec on a 167 MHz UltraSparc
- Reasonable for interpreted simulation

Impact

- First freely distributable C30 simulator
- Changes to the underlying C30 DSK tools have been rolled into C30 DSK tool releases

Impact on Ptolemy

Plans for Incorporating Feedback from Simulators

Ptolemy pre-0.7	*New in Ptolemy 0.7*	*Planned for Ptolemy 0.7+*	*Teamed with*
Code Generation (CG) Targets	CG Target Tracks Costs	-	RA, JLP
CG 56000 (CG56) Domain	CG56 Target Tracks Cost C50 Domain	C50 Target Tracks Cost	RA, JLP LG
CG in C (CGC) Domain	-	CGC/C30 Target CGC/C30 Target Tracks Cost	?

New CG Target Parameters

- Show memory usage?
- Show run time?

Benchmarking of Dual-Tone Modulated-Frequency Decoders [Paper]

Web-Based Design Tools

Web-Based Electronic Design [WELD]

- Finite State Machine Editor
- Schematic Editor for the Synopsys Design Compiler
- Real-Time Package for Java

Tech On-Line DSP Debuggers

- Sluggish telnet access to two C30 boards and two C50 boards
- For demonstration purposes
- No clean way to transfer files

Web-Based Configurable Simulators

Client-Server Architecture

- Java Applets implement the graphical user interface
- Server is written in Java
- Applets communicate with the server over a socket
- Server manages the command-line simulators
- The Applets configure the menus and display based on information returned by the simulator

Web-Enhanced Texas Instruments C30 Simulator (WETICS)

- Tested on Unix platforms and Windows '95
- Should run on all platforms that support Java
- Release expected in April of 1997
- Preliminary proof-of-concept demonstration installed at UT Austin

Conclusion

Simulators

- Provide measures of implementation costs
- Cooperate with system-level design tools and graphical user interfaces

Ptolemy Targets Track Implementation Costs

- Ptolemy 0.7 will track software implementation costs for the 56000
- Profiling performance of blocks of C code in systems [Pino]
- Hooks exists for tracking hardware implementation costs

Web-Enhanced, Configurable Users Interfaces

- Simulators
- Debuggers

Last Updated 03/23/97.