Embedded System Design and Synthesis

Robert Dick

http://robertdick.org/esds-two-week
http://ziyang.eecs.northwestern.edu/~dickrp/esds-two-week
Department of Electrical Engineering and Computer Science
Northwestern University

Office at Tsinghua University: 9–310 East Main Building

Outline

1. Introduction
   - Why work on embedded systems?
   - Class organization and sources of information

2. Embedded system research areas
   - Definitions
   - Topics to cover in class

3. Homework

Embedded system definition

Embedded system: A computer within a host device, when the host device itself is not generally considered to be a computer.

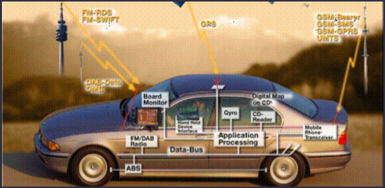
Not a general-purpose desktop computer.

In many applications, well-designed, correctly functioning embedded systems are almost invisible to their users.

Embedded systems examples

Medical devices

Sensor networks



Automobiles

Smartphones

Embedded system market size

Dominates general-purpose computing market in volume

Similar in monetary size to general-purpose computing market

Growing at 15% per year, 10% for general-purpose computing

Embedded system requirements

Hard real-time: Deadlines must not be violated

Wireless: Effects of the communication medium important

Reliable: Better crash desktops than cars

First time correct: Field repairs difficult

Rapidly implemented: IP use, HW–SW co-design

Low price: Fierce competition between many companies

High-performance: Massively parallel, using ASICs

Low power: Battery life and cooling costs

Embedded systems research goals

Develop better embedded system design ideas

Automate embedded system design process

Today's goals

1. Know how to get access to course resources
   - Website
   - References
   - Mailing list
2. Understand work and grading policies
3. Rough understanding of topics we'll cover in course

Administration

- Lectures
  - Monday–Friday 13:30–14:30
  - 10–309 East Main Building
- Quizzes, question and answer sessions
  - Monday–Friday 14:30–15:00
- PDF files posted after lectures
- If something isn't clear and you ask about it in class, I'll sometimes add more detail to the slides before posting

Class prerequisites

No prerequisites. Knowledge of some of the following helps

- Computer architecture
  - Distributed systems
  - Cache effects
  - Power consumption impact of architectural decisions
- Systems programming
  - Project-oriented course with substantial programming component
- Algorithm design and analysis
  - Computational complexity analysis
  - Efficient algorithm design

Course structure

- I'll give research-oriented lectures for a few days
- Then you will select research projects
- Each project will have a research, design, and presentation component
- Similar to short conference papers and sessions
- Main purpose of class: prepare students for independent research on embedded systems design and synthesis
  - Short course so goal will only be partially achieved

Decide office hours

- I can reschedule office hours based on your comments
- Suggest Monday–Friday 15:00-16:00 in 9–310
    - Will stay later if students still present

Grading policies

Literature summaries:	15%
Quizzes:	20%
Project presentation:	25%
Project quality and report:	40%

Active class participation by students is strongly encouraged

Project

- Open to your project suggestions
- Will also provide a few default projects
- Examples:
  - Develop/improve a clean way to specify the behavior and cost constraints for a domain of embedded systems
  - Synthesize and model communication architecture, e.g., bus topology, protocol translators, and schedulers
  - Improve embedded operating system
  - More details on these later

Projects

- Multiple people may work on the same topic and collaborate
- However, each person must present own work

Subscribe to mailing list

- Please subscribe to the ESDS mailing list by sending a to listserv@listserv.it.northwestern.edu with no subject and a body of

  SUBSCRIBE ESDS [Firstname] [Lastname]

- Useful for getting questions rapidly answered

Course goals

After finishing this course you should

- Be prepared for independent research in embedded system design automation
  - Short course so goal will only be partially achieved
- Have a high-level understanding the major research topics in embedded system design automation
  - Know the context of new work

Course goals

After finishing this course you should

- Be better at writing research papers and doing research presentations
- Understand a research topic within embedded system design automation in detail
- Have completed a project that can naturally be developed into substantial and novel research

The detailed topics we cover can be guided by your interests

Reference books

- Wayne H. Wolf. *Computers as Components: Principles of Embedded Computing System Design*. Morgan Kaufmann Publishers, CA, 2001
- Robert P. Dick. *Multiobjective Synthesis of Low-Power Real-Time Distributed Embedded Systems*. PhD thesis, Dept. of Electrical Engineering, Princeton University, July 2002
- Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. McGraw-Hill Book Company, NY, 1990

Web resources

- Deep Chip and e-mail Synopsys user's group. http://www.deepchip.com
- Electrical Engineering Times. http://www.eetimes.com
- Embedded.com. http://www.embedded.com

Journals of note

- ACM Transactions on Embedded Computing Systems
- IEEE Transactions on Computer-Aided Design
- IEEE Transactions on VLSI Systems
- Design Automation for Embedded Systems

Conferences of note

- Embedded Systems Week
  - International Workshop on Hardware/Software Codesign
  - International Conference on Compilers and Architecture for Embedded Systems
  - Conference on Embedded Systems Software
- Design Automation Conference
- Design, Automation, and Test in Europe
- International Conference on Computer-Aided Design
- Asia South Pacific Design Automation Conference

Outline

Review: embedded system definition

An embedded system is a computer within a host device, when the host device, itself, is not generally considered to be a computer.

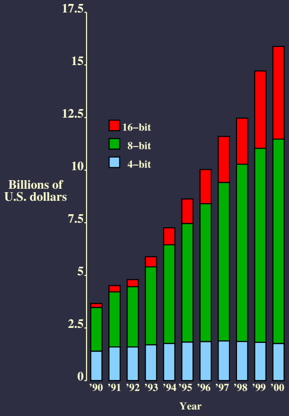
For example, the computers within automobiles, medical devices, and portable communication devices are embedded systems.

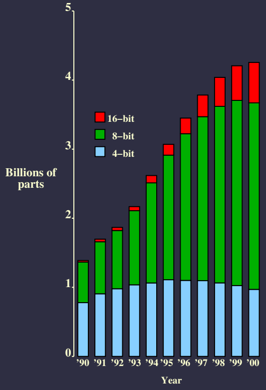
In most applications, well-designed, correctly functioning embedded systems are almost invisible to their users.

Review: embedded system market size

- Dominates general-purpose computing market in volume
- Similar in monetary size to general-purpose computing market
- Growing twice as fast
- Electronics market over $1,000,000,000/year

Global microcontroller sales





Source: Embedded Processor and Microcontroller Primer and FAQ by Russ Hersch

Embedded system requirements

- Hard real-time: Deadlines must not be violated
- Wireless: Effects of the communication medium important
- Reliable: Better crash desktops than cars
- First time correct: Field repairs difficult
- Rapidly implemented: IP use, HW–SW co-design
- Low price: Fierce competition between many companies
- High-performance: Massively parallel, using ASICs
- Low power: Battery life and cooling costs

Low-power motivation

- Embedded systems frequently battery-powered, portable
- High heat dissipation results in
  - Expensive, bulky packaging
  - Limited performance
  - Short battery life
- High-level trade-offs between
  - Power
  - Speed
  - Price
  - Area

Design automation

- These numerous conflicting expectations make manual design difficult and unpredictable
- 2001 CMP Media LLC survey
  - 1,100 embedded system developers
  - Majority of projects were running late
    - Four-month delay normal
  - Majority had lower performance than predicted
    - 50% expected and planned performance normal

Design automation

- These numerous conflicting expectations make manual design difficult and unpredictable
- 2001 CMP Media LLC survey
  - 1,100 embedded system developers
  - Majority of projects were running late
    - Four-month delay normal
  - Majority had lower performance than predicted
    - 50% expected and planned performance normal
- Conjecture: Design process unpredictability due to manual, ad-hoc design

Embedded system design automation

- Anything allowing computers to do a portion of embedded system design
- Broad scope: Try to solve the whole system-level design problem automatically
  - May need to make limiting assumptions or target narrow problem domains to make scope reasonable
  - Too large for course project
  - Can start from existing system, though
- Narrow scope: Thoroughly solve a sub-problem within embedded system design

Embedded system design automation

- System-level design automation is embedded system design automation
- General-purpose system architecture largely already decided
- Improvements can undergo laborious special-case manual analysis due to high volume

Embedded system design automation

- Embedded system architectures more flexible
- Flexibility gives synthesis algorithms freedom to consider numerous solutions
- Small design runs make it difficult to justify assigning many engineers to manual design

Embedded system design

- Design constraints and resources more varied than in general-purpose computing
- This requires different design techniques
- Many ideas highly-successful ideas in embedded systems would never work in general-purpose computing
- Many ideas highly-successful ideas in general-purpose computing would never work in embedded systems
- However, there is also some overlap

Topics to cover in class

- Will cover each topic
- May add and shorten topics as directed by your interests

Overview of embedded system research areas

- Heterogeneous multiprocessor system-on-chip design problem
- Models and languages
- Formal methods for designing reliable embedded systems
- Heterogeneous multiprocessor synthesis
- Reliability optimization

Overview of embedded system research areas

- Real-time systems
- Scheduling
- Compilation techniques for embedded systems
- Embedded operating systems
- Low-power and power-aware design

Overview of embedded system research areas

- Low-power and power-aware design (continued)
- Novel fabrication techniques for compact and low-power embedded systems
- Emerging applications: sensor networks
- Hardware and software data compression for use in embedded systems
- Review and student presentations on short projects

Example projects

- This course, and your projects, are not constrained to these topics
- They are presented as examples
- I can give access to the source code for many of these projects to use as starting points

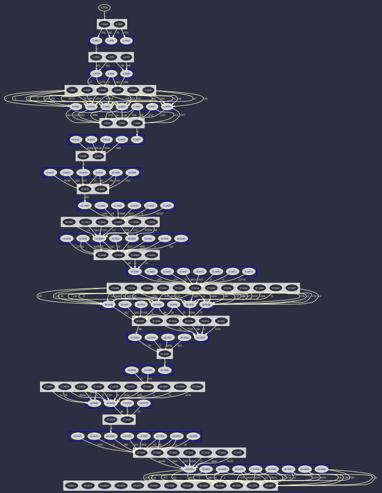
Embedded system specification

- How does one best describe an embedded system
- Must be precise and complete
- Must not constrain solutions unnecessarily
  - Leave as many options open to the designer/synthesis system as possible
- Pipelining vs. parallelism important

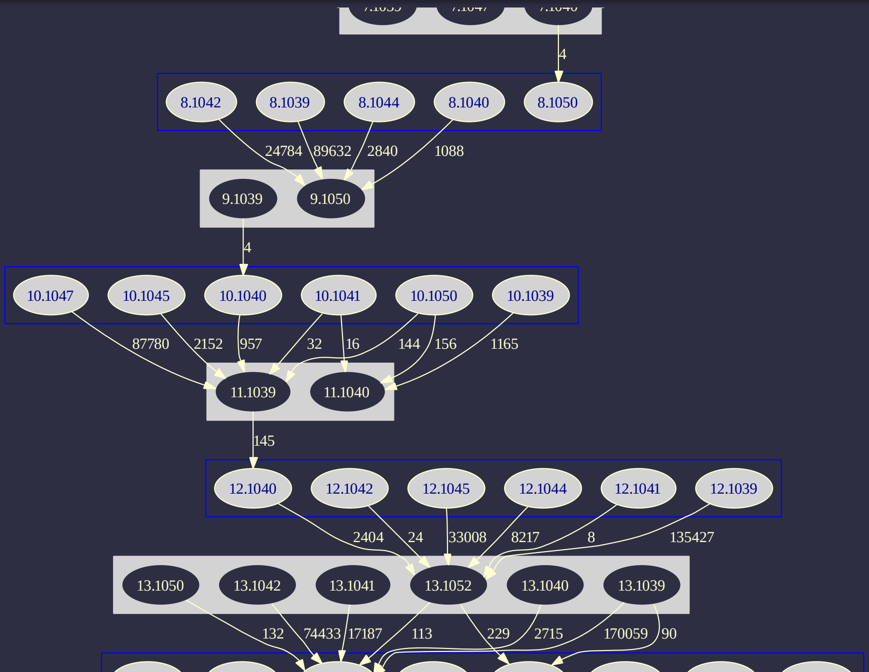
Embedded system specification

- Many use TGFF to generates parametric task graphs and resource databases
  - Robert P. Dick, David L. Rhodes, and Wayne Wolf. TGFF: Task Graphs for Free. In *Proc. Int. Wkshp. Hardware/Software Co-Design*, pages 97–101, March 1998
  - Acceptable for debugging and to demonstrate ability to scale
  - Often inappropriate
- Is there a better way?

Example MPEG encoding task set



Example MPEG encoding task set



Hardware-software co-design

- Simultaneous design of hardware and software components
- Partitioning system-level specification among heterogeneous components
- Partially automated HW/SW compilation

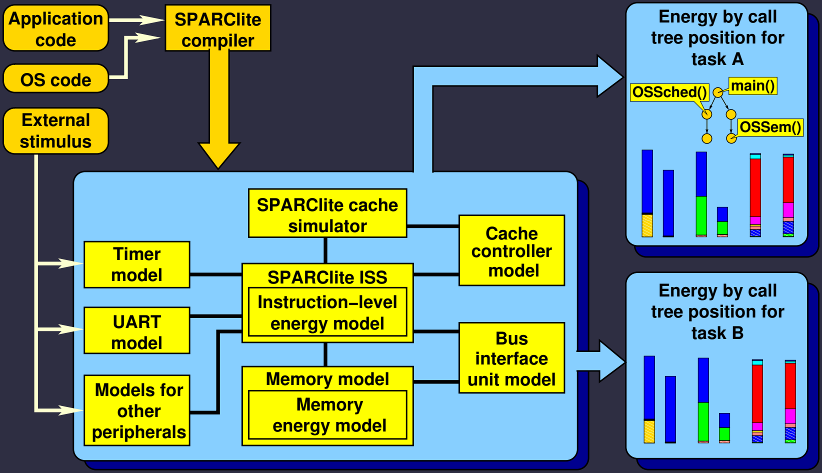
Embedded system synthesis and optimization

- Synthesize embedded systems
  - Heterogeneous processors and communication resources
  - Multi-rate
  - Hard real-time
- Optimize
  - Price
  - Power consumption
  - Response time

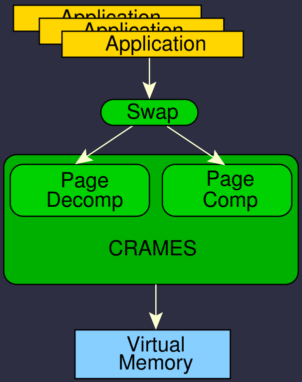
System synthesis

- MOGAC: Multi-chip distributed systems
- CORDS: Dynamically reconfigurable
- COWLS: Multi-chip distributed, wireless, client-server
- MOCSYN: System-on-a-chip composed of hard cores, area optimized
- Temperature-aware reliable MPSoC synthesis

RTOS power analysis



Compressed RAM for embedded systems

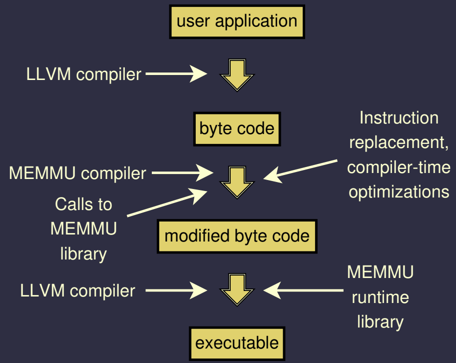


Results

- More than doubles available memory
- No hardware or application changes
- Negligible performance and energy consumption penalties
- Tested over wide range of applications
- CODES-ISSS'05, DAC'06, TECS'07
- Being used in next-generation cellphones from NEC

Lei Yang, Dr. Haris Lekatsas, and Dr. Srimat Chakradhar

Memory expansion for MMU-less embedded systems

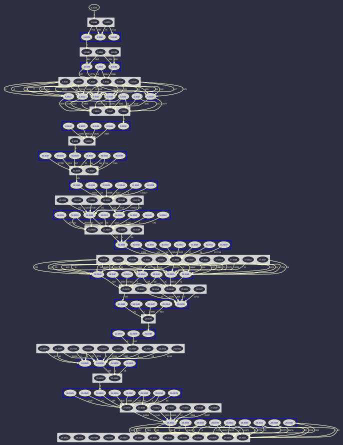


Observations and Results

- Main application: Sensor network nodes
- Implemented in LLVM and tested on TelosB nodes
- Increases usable memory by 50%, no changes to applications
- Performance and energy penalties small after compiler optimizations
- CASES'06

Lan Bai and Lei Yang

Application characterization for system synthesis

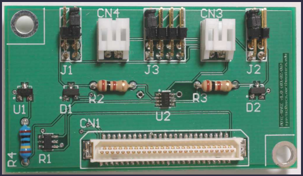


Applications

- Extract communication graphs from arbitrary multithreaded applications
- Non-intrusive
- Use for application-specific multiprocessor synthesis
- CODES-ISSS'06
- Publicly released

Ai-Hsin Liu

Event-driven sensor network architectures



Observations and Results

- Existing architectures assume nothing interesting happens when they nap
- Must always sense, but with extremely low power
- 250× power improvement for structural integrity monitoring
- 16 μW sensor board power consumption
- Crossbow MICAz-compatible hardware fabricated and tested

Sasha Jevtic, Mat Kotowsky, Prof. Peter Dinda, and Prof. Charles Dowding

Design of reliable real-time MPSoC systems

- Modern processors throttle in response to thermal emergencies
- This prevents adherence to real-time constraints
- Instead, plan real-time system design according to temperature predictions
- Status: Optimal phased steady-state real-time assignment and scheduling algorithm

Scheduling

- Many of these projects contain schedulers
- Power-aware list scheduling
- Scheduling for dynamically reconfigurable systems

Outline

Reading assignment

- Robert P. Dick. *Multiobjective Synthesis of Low-Power Real-Time Distributed Embedded Systems*. PhD thesis, Dept. of Electrical Engineering, Princeton University, July 2002
- Read Chapters 1–3 by next class
- Mainly introduction and definitions

Determine topics of interest

- Due next class
- Determine and rank three embedded systems topics you want to learn more about
- Use electronic resources, research papers, and questions posted to the mailing list to answer the following questions for each topic, using three or fewer sentences for each
    1. How useful will this be to designers in the next ten years?
    2. Is this topic of special interest to embedded system designers?
    3. Identify a potential research project that is related to this topic and can be completed within the time-frame of this course
- We will discuss your answers in the next class to select topics to focus on in the lectures
- This is not a commitment to a particular topic
    - That will come on Wednesday
- Used to guide selection of appropriate lecture topics

Next lecture

- Heterogeneous multiprocessor system-on-chip design problem
- Models and languages
- Discussion to decide which topics will be covered in most detail