Embedded System Design and Synthesis

Robert Dick

http://robertdick.org/esds/
Office: EECS 2417-E
Department of Electrical Engineering and Computer Science
University of Michigan



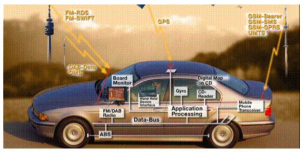

Embedded system definition

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

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
- Wireless
- Reliable
- First time correct
- Rapidly implemented
- Low price
- High-performance
- Low power

More on these later.

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

- Tuesdays and Thursdays 10:40–12:00.
- 3427 EECS.

Office hours

- Thursdays 12:00–1:00 and by appointment.
- It will be necessary to schedule a number of meetings to discuss course projects.
- 2417-E EECS.

Class prerequisites

Knowledge of some of the following topics 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.

Pick project team members carefully.

Course structure I

Lectures

- First few weeks will be background lectures.
- Then discussion of recent research papers.
- Presentations on projects.

Projects: 50% weight

- 8 September: Mini-project selection begins.
- 20 October: Mini-project report due.
- 25 October: Main project selection.
- 13 December: Main project due.

Course structure II

Presentations: 25% weight

- Week of 17 October: Mini-project presentation.
- Week of 5 December: Main project presentation.
- Near end of semester: Presentation of relevant literature.

Reading assignments and literature summaries: 10% weight

- Every week.
- One paragraph summaries are sufficient.

Exams: 15% weight

- Will have infrequent quizes on reading material.
- Final 4:00–6:00, 16 December.

Project

- Open to project suggestions.
- Will also provide a list of possible project topics.
- 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 an embedded operating system.
  - Evaluate the implications of a new device technology on embedded systems, e.g., energy scavenging.
  - More details on these in next lecture.

Projects

- Multiple people may work on the same topic and collaborate.
  - Teams of two unless there is a special reason for three.
- However, each person's contributions must be made clear in report and presentation.

Mailing list

- I will subscribe you to the course mailing list.
- If you are not officially registered but may take the course, email me ASAP to get on the course mailing list.

Course goals I

After finishing this course you should

- Be prepared for independent research in embedded system design automation.
- Understand the major research topics in embedded system design automation.
- 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 ideally novel) research.

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, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. McGraw-Hill Book Company, NY, second edition, 2002.
- Jack Ganssle. *The Art of Designing Embedded Systems*. Newnes/Elsevier, 1989.
- Frank Vahid and Tony D. Givargis. *Embedded System Design: A Unified Hardware/Software Introduction*. John Wiley & Sons, 2001.

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.
- ACM Transactions on Design Automation for Electronic Systems.

Conferences of note

- Embedded Systems Week
  - International Conference on Hardware/Software Codesign and System Synthesis.
  - International Conference on Compilers, Architecture, and Synthesis for Embedded Systems.
  - International Conference on Embedded Systems Software.
- Design Automation Conference.
- International Conference on Information Processing in Sensor Networks.
- Conference on Embedded Networked Sensor Systems.
- Design, Automation, and Test in Europe.
- International Conference on Computer-Aided Design.
- Asia South Pacific Design Automation Conference.

Fair warning

I will probably require major spine surgery some time in the coming half year.

If this happens during the course, there are plans in place to minimize impact.

There may be one or two guest lectures by other faculty and senior Ph.D. students working on topics related to embedded systems.

I may need to cover one or two lectures via video conference.

Summary: Regardless of whether I get surgery, the course will proceed and you will learn about important topics in embedded system research from experts in the field.

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.

Embedded systems research goals

Develop better embedded system design ideas.

Automate embedded system design process.

Two major sources of changing problems

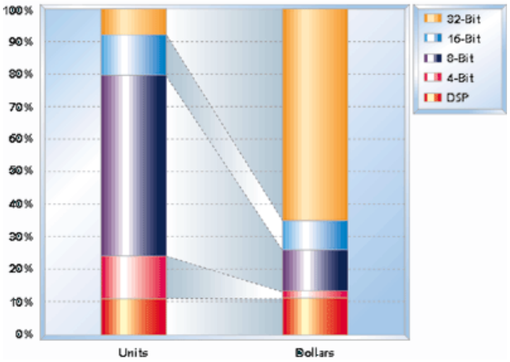
New implementation technologies.

New applications.

Review: embedded system market size

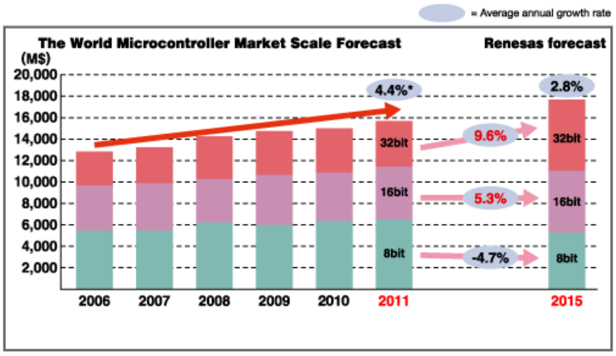
- Dominates general-purpose computing market in volume.
- Similar in monetary size to general-purpose computing market.
- Growing twice as fast.
- Electronics market over \$1,000,000,000/year.

Processor sales by type



From *The Two Percent Solution* by Jim Turley, Embedded Systems Design, 2002.

Worldwide microcontroller market scale forecast



Renesas estimates.

Microcontroller market shares

2007 and 2008 Worldwide Microcontroller Revenue Share by Supplier

Company	2008 Rank	2008 $M	2008 Share	2007 Rank	2007 $M	2007 Share	Y/Y %
Renesas Technology	1	2,770	20.1%	1	2,944	21.2%	-6%
Freescale Semiconductor	2	1,518	11.0%	2	1,743	12.6%	-13%
NEC	3	1,330	9.7%	3	1,296	9.3%	3%
Fujitsu	4	1,065	7.7%	4	1,115	8.0%	-5%
Infineon Technologies	5	983	7.2%	5	1,023	7.4%	-4%
Microchip Technology	6	812	5.9%	6	778	5.6%	4%
STMicroelectronics	7	645	4.7%	7	662	4.8%	-3%
Texas Instruments	8	601	4.4%	8	607	4.4%	-1%
Atmel	9	511	3.7%	9	458	3.3%	12%
NXP Semiconductors	10	286	2.1%	10	303	2.2%	-6%
Other		3,229	23.5%		2,936	21.2%	10%
Total		13,749			13,866		-1%

databeans estimates, Company Reports

From *Renesas Technology Still Dominates the Microcontroller Market*, Electronics Design, Strategy, News, 2009.

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, and
  - Short battery life.
- High-level trade-offs between
  - Power,
  - Speed,
  - Price, and
  - Area.

Design automation

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.

Possible explanations

- Differences between applications require greater degree of re-design than in general-purpose computing.
- More limited resources per design.
- 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.
- Smaller 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 highly successful ideas in embedded system design do not work in general-purpose computing.
- Many highly successful ideas in general-purpose computing do not work in embedded systems.
- However, there is also some overlap.

Overview of topics I

- Heterogeneous multiprocessor system-on-chip design problem.
- Models and languages.
- Introduction to complete and stochastic optimization.
- Heterogeneous multiprocessor synthesis.
- Reliability optimization.
- Real-time systems and scheduling.
- Hardware and software data compression for use in embedded systems.
- Memory hierarchies in embedded systems.
- Low-power and power-aware design.
- Embedded operating systems.
- Emerging applications: machine-to-machine.

Overview of topics II

- Emerging applications: pervasive health care.
- Emerging applications: sensor networks.
- Emerging applications: CPS.
- Emerging applications: mobile social networking.
- Impact of device technology on low-power embedded systems.
- Compilation techniques for embedded systems.
- Human-driven computer design.
- Energy supply in embedded systems.

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.

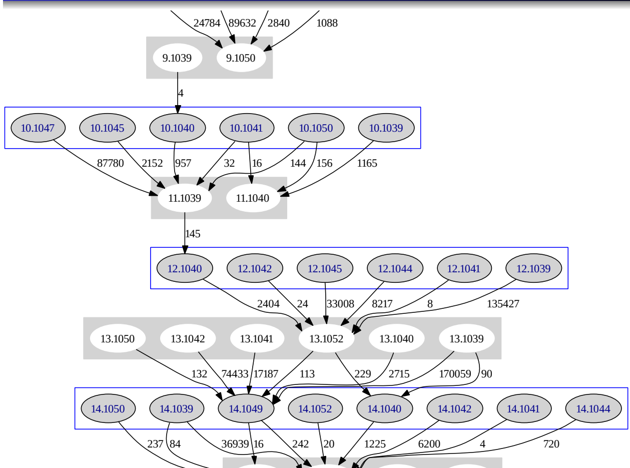
Embedded system specification

- 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.

Embedded system specification example

- 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 very inappropriate.
- Is there a better way?

Example MPEG encoding task set



Automated data flow graph extraction

- This was a project in a similar course.
- Appropriate topic in terms of scope.
- Will explain in more detail later.

Hardware-software co-design

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

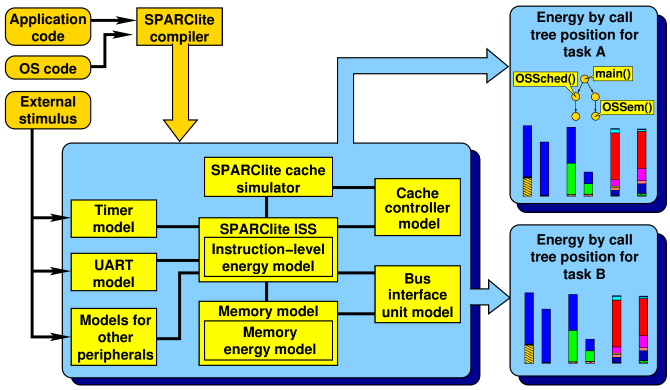
Embedded system synthesis and optimization

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

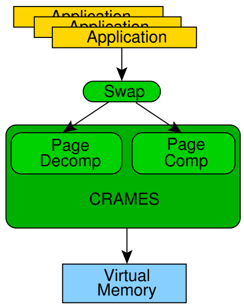
System synthesis

- 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

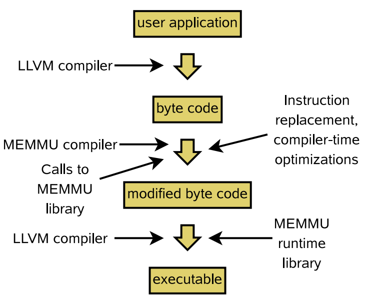


Results

- 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

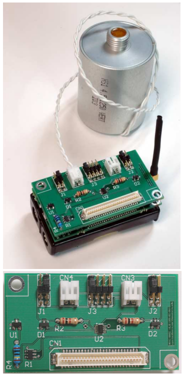
Application characterization for system synthesis

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.

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 13 September.
- Mainly introduction and definitions.

Determine topics of interest

- Due at noon on 7 September.
- List three embedded systems topics you are interested in.
- This will be used in preparing a list of suggested mini-projects.
- This is not a commitment to a particular topic.
  - That will come soon, though.

Begin study of topics of interest

- Due in class on 8 September.
- Use electronic resources, research papers, and questions posted to the mailing list to answer the following questions for each of the three topics of interest, 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.

Next lecture

- More detailed overview of example mini-project topics.
- Heterogeneous multiprocessor system-on-chip design problem.