System Integration at ISY, LiU - background and status report

O. Dahl[1]

[1]Electrical Engineering, Linköping University, Linköping, Sweden

Seminar at SaS, IDA, LiU, November 30, 2011

Outline

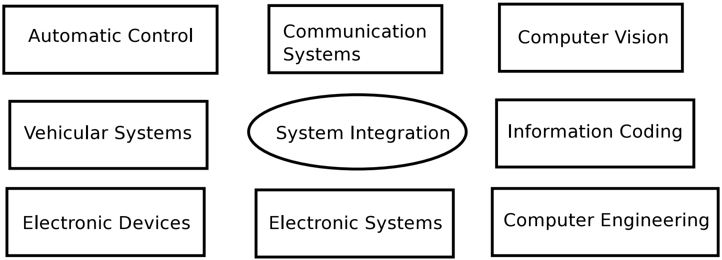
Personal background

- At LiU since 2011-01-01, at ISY (Institutionen för Systemteknik) - Associate Professor in System Integration (a new subject at the department) http://www.da.isy.liu.se/∼olad/
- Moved from ST-Ericsson
- Started at Ericsson November 2006 - worked with applications, software architecture, LTE design, simulation for software development
- Before that: engineer, consultant, manager, associate professor in Computer Science and Automatic Control
- Experience in software development, system engineering, system development, simulation, real-time systems, control
- Ph D Automatic Control, Lund, 1992

Outline

System integration at ISY

A view of its virtual placement, its real placement is at Division of Computer Engineering (where I am employed)



Interactions

- HiPEAC research network - http://www.hipeac.net/ - member of the Design and Simulation cluster
- Industrial contacts - Ericsson, FOI, Coresonic, ST-Ericsson, Saab
- ISY - Communication Systems, Computer Engineering, Parallel programming
- IDA - Real-time systems, Simulation, Parallel programming, Information systems
- MAI - Optimization
- Build-up of external System engineering contacts

From application announcement

- The field of system integration covers system solutions, software integration, and low-level software developlent, for the purpose of achieving efficient embedded systems. The relation to modern technology for processors and signal processors, ASIP, FPGA, ASIC, and GPU is important.
- Pedagogically there is a clear connection to existing courses in computer engineering, where successful project courses exist. New courses in embedded systems, system integration, and low-level programming would be beneficial for the department.

Outline

System integration

What is system integration?

- How to put together parts so that they become a system?
- How to understand the interactions between the parts?
- How to understand the structure of the resulting system?

These questions are studied in

- System engineering
- Software engineering

System integration

Is there a need for more (or new types of) work with a System perspective (System integration, System engineering, System design)?

A new class of engineer?

Is System-Level Design Creating a New Class of Engineer? [Goering, 2011]:

> *Panelists talked about who will be responsible for hardware/software partitioning. One suggested that experts in system target applications will do this task, while others spoke of engineers whose expertise spans both hardware and software.*
>
> *a "new level" of designer is appearing below system architects and is focusing on high-level modeling as a "new area of engineering."*

Hardware and Software

The Tides of EDA [Sangiovanni-Vincentelli, 2003], also Keynote speech at DAC 2003:

> *We need ways to design hardware and software concurrently but not independently. Software developers will have to deal with parameters that characterize the hardware's behavior. Doing so will permit them to predict physical quantities associated with the software's execution on the implementation platform; these quantities include timing, memory occupation, and power consumption. In turn, hardware designers must know what is important for the application software to work correctly.*

DAC 2011

- Freescale [Su, 2011] - hardware/software connection, software driving force for power consumption, power-aware software needed, better cooperation between HW and SW engineers needed
- Freescale [Su, 2011] - efficiency in system simulation, e.g. in trouble-shooting with need to reproduce errors found in field
- Panel discussion [DAC, 2011a] - “there is no time for sequential development” (i.e. hardware and software must be developed in parallel), “much time is spent on debugging things that are due to miscommunication between hardware people and software people”.

DAC 2011

- Intel [Singer, 2011], need to optimize a system over all layers, IP needs to be delivered as hardware combined with the appropriate software
- System-Level Design tutorial [DAC, 2011b] - need to model also the software, e.g. for prediction of power consumption

Virtual platforms

- [Burgio, 2011] - design cycles with increased pressure on schedule, need for concurrent design
- [Corleto, 2009] - shorter marked demands, longer developmend time due to increased product complexity, delays are (very) costly, virtual platforms can decouple SW development from silocon availability

SSF

Electronics and Photonics systems [SSF, 2010]

... while the industry as a whole is growing, especially the electronics manufacturing has left this country and Swedish industry is moving more towards systems development. Research supporting electronic systems development should thus receive priority.

Co-operation with research in signal processing, embedded software and hardware is important.

SSF

Informationintensive systems [SSF, 2011]

The Call focuses on projects that are based on a clear systems perspective

Each proposed project should contain a description of the three parts: (a) demonstrator, (b) system-oriented research and/or expertise and (c) disciplinary research.

Outline

Systems and Models

- A more system-oriented view is needed
- Hardware and Software must be considered together
- Models are important*
- ESL was used for illustration, also interesting to look at e.g. automotive (e.g. AUTOSAR and its usage)

*Models on different *abstraction levels*, with different *purposes*

LiU proposal to SSF

- “Strategiskt centrum för simulatorforsking” [Nielsen, 2011]
- proposed from ISY, in cooperation with partners from Saab, Scania, Ericsson, VTI, Volvo CE, LiU IDA, LiU ISY.

  Simulatorer utgör en allt viktigare del av den komplexa produktutvecklingen i svensk systembyggande industri. Det satsas stora resurser både i form av investeringar och genomförande. Det är strategiskt viktigt att dessa satsningar får största nytta.

LiU proposal to SSF

Linköping har i flera tidigare forskningsprogram visat sig framgångsrikt i att brygga över mellan systemteknik, datalogi och domänkunskap (exempelvis i SSF-satsningarna ECSEL, VisiMod och MOVIII). Några specifika styrkeområden är modellbygge, simuleringsspråk (exempelvis Modelica), samt domänkunskap inom telekommunikation och reglerteknik/mekatronik.

Outline

Simulation of multicore embedded systems

- Wireless baseband (e.g.) Heterogenous (control processors, signal processors, ASIC blocks, radio interface)
- Possible to run production software
- Transaction-level modeling
  - Model functional behavior of hardware
  - Bit-accurate interfaces
  - Timing can be added (timing annotations, not cycle-accurate)
  - SystemC, TLM-2 - http://www.systemc.org/home/

SW design with a HW perspective, HW model design with a SW perspective, multicore on the host, timing and synchronization, simulator design

Outline

Modeling and optimization of Wireless sensor networks

- SSF application
- Modeling nodes and modeling the network
- Model transformations
- Optimization
- Mixed-signal modeling
- ISY, IDA, MAI

Modeling and optimization for increased predictability and better trade-off decisions, node requirements vs network requirements, languages and tools - Modelica, OMOptim, SystemC, SystemC-AMS

Outline

Parallel architectures for nonlinear filters

- ELLIIT application
- Sampling-based filters used in industry today
- Computationally demanding
- Hardware solutions needed
- Example applications - positioning, computer vision, signal processing
- Methodology and architecture for parallel implementation in hardware of sampling based filters, as a means for enabling more applications than what is possible today
- Parallel custom hardware in FPGA or ASIC, general multicore architectures e.g. GPUs, standard multicore processors, and multicore DSP architectures such as ePUMA

Outline

Problem formulation

- Trade-off quality and computing power
- Signal processing scenario
- Voltage scaling and frequency scaling
- Determine how to control the number of iterations, in an iterative algorithm, so that power consumption is optimized, while keeping a stated quality goal (quality-constrained energy minimization)

Related work

- [Nawab et al., 1997] - defines *Approximate Signal Processing* as a general framework for trade-offs between quality and computational effort
- General discussion and problem formulations in [Larsson and Gustafsson, 2011] - discusses implications of per-core DVFS for multicore DSP applications
- [Cirkic et al., 2011] - optimize detection quality under computational complexity constraint, choose adaptively between different detection algorithms
- [Andrei et al., 2011] - time-constrained energy minimization

Related work

- [Amador et al., 2010] - power management strategy for iterative decoding of LDPC codes, can achieve significant energy savings with a relatively low loss in error-correcting performance
- [Alt and Simon, 2010] - applies automatic control methods to video decoding, adjusting voltage and frequency for energy minimization, given a set-point for quality

Model propagation delay and energy consumption

Alpha model for propagation delay [Rabaey, 2009]

$$t_p = \frac{kV_{DD}}{(V_{DD} - V_{TH})^{\alpha}} \tag{1}$$

Energy per operation [Rabaey, 2009]

$$E_{op} = C_{op} V_{DD}^2 \tag{2}$$

where C_{op} is a constant that bundles the capacitances and the number of transistor switches involved in the operation.

Number of operations and Energy consumption

A job with N tasks, with n_i, $i \in 1, 2, \ldots N$ operations per task.
Total number of operations $n = \sum_{i=1}^{N} n_i$.
Assume constant $V_{DD} = V_{DD_i}$ for all operations in one task.
Energy per operation for task i

$$E_{op_i} = C_{op} V_{DD_i}^2$$

gives total energy for task i

$$E_i = n_i E_{op_i} = n_i C_{op} V_{DD_i}^2$$

and total energy for all tasks

$$E_N = \sum_{i=1}^{N} E_i = \sum_{i=1}^{N} n_i E_{op_i} = \sum_{i=1}^{N} n_i C_{op} V_{DD_i}^2$$

Quality constrained Power optimization

- Minimize total energy, subject to a constraint on the overall quality of the job.
- Assume a function $Q_i(n_i)$, relating the number of operations for task i to quality for task i.
- The function $Q_i(n_i)$ also depends on other variables, e.g. SNR.
- Such a function exists for several algorithms, as described in [Nawab et al., 1997].

Total time $T = \sum_{i=1}^{N} T_i$ of all tasks is given as a constraint.

Quality constrained Power optimization

Quality constraint expressed e.g using average quality

$$Q_{avg}(n_1, \ldots, n_N) = \frac{1}{N}\sum_{i=1}^{N} Q_i(n_i)$$

constrained as

$$Q_{avg}(n_1, \ldots, n_N) \geq Q_{min}$$

Optimization problem

Minimize

$$\min_{n_i, T_i, V_{DD_i}, i \in \{1,\ldots,N\}} \sum_{i=1}^{N} n_i C_{op} V_{DD_i}^2$$

subject to

$$\sum_{i=1}^{N} T_i = T, \quad Q_{avg}(n_1, \ldots, n_N) \geq Q_{min}$$

Just-in time scheduling with DVFS

Choose V_{DD_i} in

$$t_{p_i} = f_{t_p}(V_{DD_i}) = \frac{kV_{DD_i}}{(V_{DD_i} - V_{TH})^\alpha}$$

so that

$$t_{p_i} = t_{p_i}^{DVFS} = \frac{T_i}{n_i}$$

Optimization problem with just-in-time DVFS

Introduce the function $f_{V_{DD}}$, defined by

$$V_{DD} = f_{t_p}^{-1}(t_p) = f_{V_{DD}}(t_p)$$

Minimize

$$\min_{n_i, T_i, i \in \{1,\ldots,N\}} \sum_{i=1}^{N} n_i C_{op} f_{V_{DD}} \left(\frac{T_i}{n_i}\right)^2$$

subject to

$$\sum_{i=1}^{N} T_i = T, \quad Q_{avg}(n_1, \ldots, n_N) \geq Q_{min}$$

Why is this interesting?

- the sensitivity of V_{DD} as a function of t_p and hence as a function of the number of operations per task, is strongly varying, and it depends heavily on the value of α
- the sensitivity of Q_i as a function of the number of operations is more regular
- a small change in number of operations may therefore lead to a large change in energy consumption (due to a large change in V_{DD} with a corresponding adjustment of the operating frequency)
- for a given job, a given average quality can be obtained by different selections of the individual n_i values, hence it should be possible to select the set of n_i values giving the lowest energy consumption

Challenges

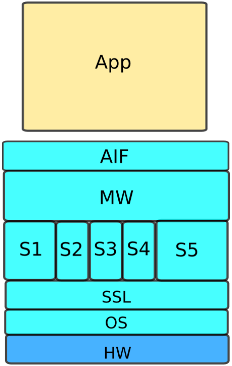
- for a given algorithm, determine quality as a function of the number of operations
- for a given problem, design an algorithm where is is easy to determine the quality as a function of the number of operations, and which gives meaningful results for a continous range of number of operations (i.e. an *anytime algorithm*)
- planning in advance using optimization for a given task, vs. using feedback for on-line adjustments
- choice of model for energy consumption

Outline

Industrial motivation

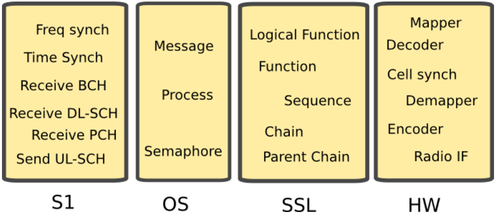
- Increasing system complexity
- Large projects, many engineers, shorter lead-times
- Software often designed using concepts and methods from software engineering, with a demand to make a general enough design (e.g. to secure adaptation to changes in future product generations)
- Software design may become *unneccessary* general - leading to loss of insight regarding hardware properties, and loss of performance
- Layered architectures may lead to abstraction levels with *unneccesary information hiding*

Example system architecture



Layers, Many (million) lines of code, Abstraction levels, APIs, Vocabularies - applications, processes, events, signals, The architecture and its use in different project phases

Using views



Vocabularies, Definitions of terms in a vocabulary, Mappings (relations) between terms in a vocabulary, Mappings between vocabularies, Which view are we in now? Are we in the same view? Does S1 mean Requirements? (or SW?), Do our requirements and design cover all terms in all views?

Where to go from here?

- Choose a domain - 3GPP LTE
- Discuss with industry - Ericsson, Enea
- Start with a realistic example - 3GPP LTE MAC spec - 36.321 - model from a requirements view
- Choose a use case - generation of code for simulation, from a formal model representing the chosen requirements (from 36.321 to start with)
- Investigate formal representations for view modeling
- Plan for other use cases - e.g. consistency checking, traffic model generation, configuration code generation, stimuli and expected outputs generation, system description generation, documentation and training material generation

View modeling - possible approaches

- Ontologies e.g.
  [Assawamekin et al., 2008, Chen et al., 2011]
- Behavior modeling, behavior engineering e.g.
  [Myers et al., 2011, Powell, 2010]
- Model checking inspired methods and tools, e.g. TLA, e.g. used in [Shrotri et al., 2003]

Courses relating to system integration

TSEA81 - http://www.da.isy.liu.se/courses/tsea81/ - Computer Engineering and Real-Time Systems

- shall give an *understanding of hardware/software interactions in computer systems with parallel activities and time constraints, and to develop basic skills for integration of software using a real-time operating system*
- uses an RTOS, running as soft-kernel in Linux and Windows, and also on ARM Cortex-A8 on Beagleboard - http://beagleboard.org/
- given mainly for Y, with additional students from other programs

Courses relating to system integration

TSEA-TBD - System integration - a *possible* course,

- covering system design with concurrent hardware and software design and verification
- giving a deepened knowledge of the hardware/software-interface, especially in concurrent and real-time systems, with heterogenous hardware
- covering methods for *system analysis*, e.g. with respect to perfomance related issues (MIPS, cache misses) and quality related issues (e.g. power consumption)
- covering methodology for system engineering (e.g. as defined by INCOSE - http://www.incose.org/)
- targeting students from TSEA81, D-students, and Master program students

Summarizing notes

System integration

- System aspects more and more important in embedded systems engineering
- Some research areas outlined and activities have been started

Next steps

- Continue with ongoing startup-work
- Continue cooperation and networking
- Continue to apply for funding

Alt, A.-M. and Simon, D. (2010).
Control strategies for h.264 video decoding under resources constraints.
SIGOPS Oper. Syst. Rev., 44:53–58.

Amador, E., Knopp, R., Rezard, V., and Pacalet, R. (2010).
Dynamic power management on ldpc decoders.
In *Proc. IEEE Computer Society Annual Symp. VLSI (ISVLSI)*, pages 416–421.

Andrei, A., Eles, P., Jovanovic, O., Schmitz, M., Ogniewski, J., and Peng, Z. (2011).
Quasi-static voltage scaling for energy minimization with time constraints.
IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 19(1):10–23.

Assawamekin, N., Sunetnanta, T., and Pluempitiwiriyawej, C. (2008).

Resolving multiperspective requirements traceability through ontology integration.
In *Proc. IEEE Int Semantic Computing Conf*, pages 362–369.

Burgio, P. (2011).
Virtual platforms for embedded systems prototyping.
http://www-micrel.deis.unibo.it/MPHS/slidecorso1

Chen, F., Zhou, H., Yang, H., Ward, M., and Chu, W. C.-C. (2011).
Requirements recovery by matching domain ontology and program ontology.
In *Proc. IEEE 35th Annual Computer Software and Applications Conf. (COMPSAC)*, pages 602–607.

Cirkic, M., Persson, D., and Larsson, E. G. (2011).
New results on adaptive computational resource allocation in soft mimo detection.

In *Proc. IEEE Int Acoustics, Speech and Signal Processing (ICASSP) Conf*, pages 2972–2975.

Corleto, J. (2009).
Virtual platforms: Enablement challenges.
http://www.ovpworld.org/view.php?doc=qualcomm_co

DAC (2011a).
Software-hardware verification battle: Prototyping vs. emulation.
http://www.dac.com/conference+program+panels.asp

DAC (2011b).
System-level design and software development for energy efficient platforms: Challenges from models to methods.
http://www.dac.com/conference+program+tutorials.

Goering, R. (2011).
Is system-level design creating a new class of engineer?
Cadence.

http://www.cadence.com/Community/blogs/ii/archiv

Larsson, E. G. and Gustafsson, O. (2011).
The impact of dynamic voltage and frequency scaling on multicore dsp algorithm design [exploratory dsp].
IEEE Signal Processing Magazine, 28(3):127–144.

Myers, T., Dromey, G., and Fritzson, P. (2011).
Comodeling: From requirements to an integrated software/hardware model.
Computer, 44(4):62–70.

Nawab, S. H., Oppenheim, A. V., Chandrakasan, A. P., M.Winograd, J., and T.Ludwig, J. (1997).
Approximate signal processing.
J. VLSI Signal Process. Syst., 15:177–200.

Nielsen, L. (2011).
Strategiskt centrum för simulatorforskning.
http://www.stratresearch.se/sv/Strategiprocessen

Powell, D. (2010).
Behavior engineering - a scalable modeling and analysis method.
In *Software Engineering and Formal Methods (SEFM), 2010 8th IEEE International Conference on*, pages 31 –40.

Rabaey, J. (2009).
Low Power Design Essentials.
Springer.

Sangiovanni-Vincentelli, A. (2003).
The tides of EDA.
IEEE Design & Test of Computers, 20(6):59–75.

Shrotri, U., Bhaduri, P., and Venkatesh, R. (2003).
Model checking visual specification of requirements.
In *Software Engineering and Formal Methods, 2003.Proceedings. First International Conference on*, pages 202 – 209.

Singer, G. (2011).
DAC keynote: The imminent EDA transformation.
http://www.dac.com/conference+program+keynotes.a

SSF (2010).
SSF announces research group grants for research on electronics and photonics systems.
http://www.stratresearch.se/Global/utlysning_pdf

SSF (2011).
SSF announces research group grants for research on informationintensive systems: Making good use of everincreasing data volumes.
http://www.stratresearch.se/Global/utlysning_pdf

Su, L. (2011).
DAC keynote - megatrends driving embedded multicore innovation.
http://www.dac.com/conference+program+keynotes.a