OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

This document has been produced without formal United Nations editing. The designations employed and the presentation of the material in this document do not imply the expression of any opinion whatsoever on the part of the Secretariat of the United Nations Industrial Development Organization (UNIDO) concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries, or its economic system or degree of development. Designations such as "developed", "industrialized" and "developing" are intended for statistical convenience and do not necessarily express a judgment about the stage reached by a particular country or area in the development process. Mention of firm names or commercial products does not constitute an endorsement by UNIDO.

FAIR USE POLICY

Any part of this publication may be quoted and referenced for educational and research purposes without additional permission from UNIDO. However, those who make use of quoting and referencing this publication are requested to follow the Fair Use Policy of giving due credit to UNIDO.

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

16714

Integrated Hardware Software Design

H. Kopetz

Technical University of Vienna

ABSTRACT

The recent developments in the design of integrated software/hardware systems open new opportunities for product development in the area of embedded real time control systems. In this short paper the benefits of an integrated software hardware design are presented and an outline of an appropriate design methodology is given. The final section contains some recommendations for immediate actions in the field of computer engineering education.

1. Introduction

The dynamic developments in the fields of computer science and microelectronics have changed the character of these fields in the last few years. Ten years ago, the design and construction of a special purpose computer tailored to the characteristics of a given application was a major and expensive project. Today, with the availability of standard microprocessors, standard peripheral chips and high level VLSI design tools, the implementation of such a project has lost some of its difficulties. Tomorrow, at a time when the VSLI design technology will have matured even further, the construction of an integrated hardware software solution with application specific functionality will be an accepted practice in the area of real time control systems for the volume market.

This short paper is organized in three sections. The first sections examines the need for the integration of hardware software design. In the second section a design methodology is outlined and in the third section some recommendation for immediate actions in the educational sector are given.

2. Why should we integrate the Software/Hardware Design?

In the following paragraphs we will present some arguments which lead us to believe that the integrated software hardware design for real time control systems will be an important design technology of the future leading to products of increased reliability and functionality.

2.1. User needs

The significant decrease in the cost of microelectronic systems has lead to a multitude of new embedded computer applications which interface directly to a user population unfamiliar with computer technology. Examples for these kind of systems are: the control of home appliances, the control of special purpose machinery, automotive electronics etc.. These products are destined for a growing mass market with important economic implications.

In these application the user requires a product with the following characteristics:

- Special purpose simple user interface of focussed functionality.
  Every action required by the enduser to operate such a system must be designed from the point of view of the overall system functionality and not from the "computer" viewpoint. It is therefore necessary to design special easy to operate user interfaces, both in hardware and software. A general purpose terminal interface is not suited for these applications.
- High dependability.
  In many cases the control system and the control object (i.e. the machine which is to be controlled) form an integrated functional unit, i.e. a product. The reliability of the control system must be optimized in order not the compromise the reliability of the total product. Such an optimization of the hardware reliability requires the minimization of the chip count. This is achievable only by an integrated software hardware design.
- Minimal installation effort.
  After the software design has been completed and the software has been tested and released for use, the installation effort has to be minimized. This can be achieved if the software is "frozen" into the hardware (e.g. in ROM). From the users point of view it is not decidable which implementation technology provides the given functionality.

2.2. Technical Developments

As mentioned before, a number of technical developments have taken place in the last few years which make the integrated hardware software design feasible.

- Design Tools

  The achievements in the area of VLSI design techniques in the last ten years have been dramatic. Starting with the seminal work of Mead and Convay /Mea80/ the development of VLSI design tools has reached a state, where it is possible to design application specific VSLI chips of moderate complexity within a period of weeks. At the moment a considerable effort is underway to integrate these design tools with the classical software engineering environments. In the not too far future, it will be possible to consider the design of a VLSI solution as one of a number of alternative implementation strategies for a given system functionality. The integrated software hardware design of the future will start with a computer aided requirement specification /e.g. Senf87/. These requirements will be checked for completeness in relation to established standards and for internal consistency. In the following phase the system functions and the system architecture will be specified. Finally an implementation strategy will be developed. Depending on the expected size of the production series either a classical microcomputer solution on standard off the shelf hardware (for small series), a gate array solution (for medium sized series) or a custom VLSI design solution (for large series) will be chosen.

- Distributed Systems

  The developments in the area of distributed computing are a second driving force for an integrated hardware software design. A distributed system solution has the potential for implementing large application systems with many identical single board computers connected by local area networks. It is evident that the design and implementation of these single board computers is driven by concerns for efficiency, reliability and cost. These leads to a design, where many standard operating systems functions and communication systems functions are integrated in the component hardware.

2.3. Economic Aspects

- Mass Markets

  If an integrated software/hardware design has been pursued, then it is possible to select the appropriate implementation technology in response to market penetration and market success of a given product. Since the different implementation alternatives are supported by an integrated design environment, the switchover from one alternative (e.g. software on standard hardware) to another (e.g. a part of the functions in gate arrays) can be achieved with minimal extra cost.

- Tangible product

  An integrated software hardware solution is a tangible product which can be marketed on its own. Since the integrated software/hardware product is produced by a single source, there are no questions of responsibility in case of problems, which gives the customer an increased confidence in the product. There is no special knowhow required on behalf of the customer to perform the integration of hardware and software at the user site, as is the case with standard software products.

- Know how protection

  The most significant investment during the implementation of a computer application is in the area of software development. If the software is integrated into the hardware then this investment can be protected much better than if the software is sold separately. At the moment, the adequate protection of software products is still an open question.

3. How should we perform the Hardware/Software integration?

In the following it will be assumed, that the design and implementation activities can be broken down into the following phases:

(1) Requirements specification

(2) Architecture design

(3) Analysis of the architecture

(4) Implementation

(5) Testing and Evaluation

(6) Operation

Given a set of requirements--both functional and timing-- it is proposed to realize a stepwise design of a real time control system by an operational approach. The entities of the specification and design language should be closely related to the entities of the object architecture such that the structure clash between the specification, design and implementation is minimized. Since the timing properties of a design are related to the capability of the underlying hardware, an early binding of software functions to hardware units is required. This makes it possible to analyse the performance properties of the design representation early in the development cycle --even before every detail of the design is given-- in order to determine if this still unrefined design will meet the given timing and reliability requirements.

The effectiveness of any design methodology can be significantly enhanced if it is supported by an appropriate set of software tools, i.e. a design environment. We distinguish between architecture design tools, analysis tools, implementation tools and management tools /Kop86/. The architecture design tools support the system analyst in the requirements specification phase and the architecture design phase of a real time application development. The analysis tools can be used for an analysis of a given architecture design, e.g in respect to timing and reliability and a comparison of different designs. The management tools support the project management and documentation.

In present industrial practice, disjoint development teams use different sets of specification, design and analysis tools for alternative implementation technologies (e.g. implementation by a microprocessor with the appropriate application software or implementation by an application specific VLSI chip). However considerable research is in progress to integrate these different design tools and the design databases into a coherent toolset such that the duplication of efforts for the implementation of the same functionality with different implementation technologies is eliminated. However, such an integration can only be successful if the design engineers are experienced in both, software and hardware design.

4. What Actions have to be taken in the educational institutions?

Looking at the speed of change in the field of information technology it is of paramount importance to realize the important longterm trends in their early phases in order to take appropriate actions in the educational system.

Only if engineers of the required background in software and hardware are trained in a given country will the industry be in a position to take advantage of these new trends. The most important longterm action relates to the establishment of courses of study on integrated software hardware design techniques at the trade schools and at the university level for interested students. These courses must be backed up by appropriate laboratory exercises.

Teaching software and hardware technology without the possibility of practical work on the machine is a dangerous undertaking. Since the lectures tend to become too theoretical, the student will not grasp the elementary concepts and might shy away instead of developing a positive attitude towards this new technology.

Therefore any education initiative in this field must be supported by an initiative to provide the necessary computer equipment and software for the practical training and access to a silicon foundry for VLSI manufacturing.

Furthermore, suitable channels for marketing and maintaining the new products must be set up.

5. Conclusion

It is felt that the integrated software/hardware development will lead to a number of new intelligent products, which will have a significant impact on our work environment and our society in general. Only if a country is in a position to take part in the production of these products will it fully participate in the benefits of the revolution caused by the information technology.

6. Literatur

/Alf77/ Alford M.W., A requirement engineering methodology for real time processing requirements, IEEE Trans. Softw. Eng. Vol SE-3, p.60-69, 1977

/Kop86/Kopetz, H., Design Principles for Fault Tolerant Real Time Systems Proc. of the Nineteenth Annual Hawaii International Conference on System Science, Western Periodicals, North Hollywood, Cal, 1986, p:53-62

/MSA80/Mascot Suppliers Association, The Official Handbook of Mascot, Malvern, U.K., 1980

/Mea80/Mead, C., Conway, L., Introduction to VLSI Systems, Addison Wesley Publishing Company, Reading, Mass, 1980

/Par72/Parnas, D., On the Criteria to be used in Decomposing Systems into Modules, Comm. ACM., Vol. 15, Nr. 11, Nov.1972, p. 1053 -1058

/Ros77/Ross, D.T., Structured Analysis, A Language for Communicating Ideas, IEEE Transactions on Software Engineering, Vol. SE-3 p.16-34, 1977

/Senf87/Senft, C., A computer Aided Design Environment for Distributed Real Time Systems, Proc. Eurocomp. April 1988

/Was83/Wassermann, A.I., Freeman, P., ADA Methodologies, Concepts and Requirements, ACM Software Engineering Notes, Vol 8, Nr. 1, Jan. 1983 p.33-98.

TU Wien		title

INTEGRATED SOFTWARE HARDWARE DESIGN

H.Kopetz
TU Vienna, Austria
Dec. 1987

Integrated Hardware Software design is characterised by:

* Requirements specification independent of implementation technology

* Early binding of software to hardware

* Different implementation choices, e.g.
  - Standard Microprocessor + Software
  - Interconnection of Standard Hardware Modules (e.G. VME Bus)
  - Gate Array Technology
  - Customized VLSI chips

Overall Focus on System Functionality.

Typical products where an integrated hardware software design is required:

* home appliances, e.g. washing machine, kitchen appliances, Television etc..

* automotive electronics

* industrial products e.g. instrumentation, robotics,

* Communication, e.g. telephone

These are mass market products of significant economic importance.

But also
* process control systems in industry

Why is this integration of hardware/software design so important?

(1) User needs

(2) Technological Developments

(3) Economic Aspects

User needs:

(1) Special Purpose Man Machine Interfaces
supports the functionality of the
product without any unnecessary
hardware or software facilities.

(2) High dependability
Minimization of chip count.
Optimal maintenance strategy.

(3) Integrated product
no integration knowhow or effort
at the retailer or enduser site.

Technical developments:

* Standard VLSI design tools make the design of Customized VLSI chips relatively easy

* Integrated Software engineering environments are gaining industrial acceptance.

* The integration of these two fields will be achieved in the forseeable future i.e. integrated tools for software and hardware design

* Market penetration of Distributed Systems

Economic Aspects:

* Mass market products
  Appropriate implementation technology can be selected depending on market penetration

* Tangible products of given functionality

* Know how protection.
  Software, which is "frozen" into the hardware design can be protected

Development phases

(1) Requirements specification

(2) Architecture design

(3) Analysis of the architecture

(4) Implementation

(5) Testing and Evaluation

(6) Operation

Actions in the Educational Sector:

* Training of engineers in integrated hardware software design techniques.

* Lectures plus extensive laboratory work obligatory

* Access to VLSI Tools and Silicon Foundry