Hardware Compilation, Configurable Platforms and ASICs for Self-validating Sensors

Ian Page

Oxford University Computing Laboratory, Parks Road, Oxford OX1 3QD, U.K.

Abstract. We describe the use of hardware compilation techniques to provide a flexible interfacing and computing platform for building self-validating (SEVA) sensors. SEVA technology has been developed at Oxford as a generic method of supplying a real-time guarantee on the quality of a physical measurement as well as the measurement itself. We describe the reconfigurable hardware platforms that have been constructed to support this work and the development tools and techniques used to generate new sensor implementations quickly and reliably. We further describe the compilation of SEVA systems into RISC-based ASIC technology for mass market applications.

1 Introduction

Self-validating (SEVA) sensor technology has been developed at Oxford over the last nine years as a generic method of supplying a real-time guarantee on the quality of a physical measurement as well as the measurement itself. A major benefit of the approach is that it gives a system-level framework for building fault-tolerant control systems. Such systems require an intimate knowledge of the state of all sensors and effectors both during normal operation and especially in abnormal circumstances so that the appropriate system-level response can be made. The sort of application areas which are intended include chemical process plants, oil rigs and aircraft as well as much smaller scale systems.

These application areas are already being transformed by the introduction of new technology, and 'smart sensors' in particular. For instance, a traditional temperature sensor might employ a thermocouple element and analogue circuitry to turn the varying resistance into a 4-20mA current (representing an arbitrary device-independent scale of 0-100) for onward transmission to the controller at some remote location over dedicated analogue wires. A corresponding 'smart sensor' implementation might employ a local microcontroller to perform self-calibration, maintain an operational history, digitize the thermocouple value locally, process it to obtain a true temperature reading and pass this reading on to the controlling computer over a digital network.

A SEVA sensor will in addition perform detailed checks on the physical device, for example by periodically driving current into a thermocouple and using it as an output device. By examining the response, it can be determined if the thermocouple is open or short circuit, or has lost thermal contact with the substrate being monitored. A SEVA system contains a comprehensive model of the sensor and its environment. For example it might be capable in real-time of detecting a fault in one physical sensor from an ensemble, inverting the model equations with respect to that sensor and continuing to give a 'best efforts' reading with a new, but somewhat poorer, guarantee of quality.

Over a similar timescale the Hardware Compilation Research Group working at Oxford University Computing Laboratory had been developing techniques for compiling programs into hardware implementations. The implementations are realised using FPGA (Field Programmable Gate Array) chips on various hardware platforms that we have constructed. Hardware compilation is the term we use to describe the fully automatic implementation of hardware from a program. It differs from high-level hardware description languages such as VHDL in that the intended users of the language are programmers, not hardware designers. The hardware compilation techniques are described in [8, 9] and other publications.

In 1994 the two groups began to collaborate to see if hardware compilation techniques could be used to produce prototypes of industrial sensor and control systems both quickly and reliably. The results of the collaboration have been wholly positive for both groups and have led to very much faster prototyping and increased re-use of design effort between applications.

2 Hardware Compilation and the Handel-C Language

The Handel-C language has been developed at Oxford for describing programs which are (usually) compiled into hardware. Handel-C is a small subset of C, extended with the parallelism and communication constructs of Hoare's CSP and expressions of arbitrary bitwidth. The language is necessarily different from C itself because of the different nature of hardware and software implementations.

C is not adequate for our task because it is a sequential language. This means that algorithmic parallelism can't be expressed in C programs. However it is in the very nature of hardware implementations that they achieve their speed through exploiting parallelism. It is beyond the state of the art automatically to introduce the appropriate parallelism into a sequential program. For this reason we expect the programmer explicitly to denote the parallelism that is appropriate for the desired hardware implementation.

Programs, written in Handel-C, are mapped into hardware at the level of netlists by a series of transformations [9]. The implementations fully exploit the explicit parallelism of the programs themselves and serial execution is only introduced where explicitly required. We have successfully used the system to compile and execute applications in real-time video, machine vision, data compression and other applications, including a real-time video object detection and tracking system [10].

3 Reconfigurable Hardware Platforms

The majority of industrial sensors are powered over the communication medium itself. These devices tend to be constrained by cost and power consumption. The most sophisticated sensors enjoy separate power supplies; in this smaller market segment, functionality is more important than power consumption or cost.

Partly to address this range of issues, two separate architectures were developed. The first architecture is a low-cost ISA bus FPGA-only card aimed at simple validation applications. Each card can interface to a daughter-card holding application-specific circuitry and components. The second architecture is a transputer-based card for the more sophisticated applications.

It is clearly desirable to minimise the specialist knowledge required to develop applications. We have been tackling this issue on a number of fronts. First and most obvious is the use of a programming language rather than a hardware description language to design the hardware processing units. Secondly we have developed a library of interfaces for commonly used input/output chips. Thirdly we have built higher level Handel library routines (including RMS-DC conversion, high accuracy frequency measurement and noise filtering). We also use National Instrument's Labview for front-end prototyping and our long term goal is to provide all host functionality within Labview, so that any user will be provided with a well-documented and supported front-end tool.

To date the two groups have collaborated on the development of three successful SEVA applications using this technology: a self-validating thermocouple [1], a dissolved oxygen [3], and a Coriolis mass flow [6] metering system. Although the mass flow meter is the most complex and impressive system, it has been particularly gratifying to be able to demonstrate that an entire sensor validation application (the thermocouple) can be coded entirely within Handel and implemented within FPGAs.

In the following sections we give details of the various hardware platforms that we have developed to support this work. Some of these platforms are now available from commercial suppliers which is a good indication that the research work we have been doing has been well focussed and is industrially relevant. We also detail ongoing work which we hope will give us a route into single chip implementations of even quite complex self-validating sensors.

3.1 The 'Valcard' Reconfigurable Platform

The Valcard system was our first-generation reconfigurable platform. It is an ISA bus card which hosts one or two 3000 series Xilinx FPGAs, usually XC3195A parts. It has an internal 16 bit bus which is used for communication between the FPGAs and with the daughter-card, which holds application-specific circuitry. It provides hardware support for interfacing with the PC bus, and Handel routines are available for creating software ports in the PC I/O address space, allowing the transfer of data between PC host software and Handel applications.

The board carries a pair of daughterboard connectors. Typically, a small and very simple daughterboard would be constructed for each new sensor or control interface. This board would carry the application-specific components which could not be part of the general-purpose motherboard, such as AD converters, perhaps a clock, level converters, and external connectors. The FPGA chips on the motherboard usually hold the kernel algorithms for handling the sensors, processing the resulting data, and communicating results back to the host PC. The Valcard system is designed to be cheap, easy to manufacture, and reusable. Further details on this card and all our other hardware platforms can be seen at the world-wide web site for the Oxford Hardware Compilation Group.

Valcard was used in two SEVA applications. The thermocouple project [1] demonstrated that a complete, if simplified, SEVA sensor (including measurement and uncertainty calculation, and fault detection and compensation) can be defined in Handel code and implemented within FPGAs. This implementation proved to be a powerful demonstration to industrial audiences that SEVA is implementable within commercial technology.

The other application is a dissolved oxygen sensor [3]. Here, Valcard provides a simple interface via the PC ports to data from a dissolved oxygen probe and the larger validation program was implemented primarily in software. The interface requirements included controlling a sigma-delta A-D converter, providing a drive current to stimulate the transducer, and operating switching circuitry to control an active diagnostic test.

The availability of the Valcard FPGA board has also stimulated other groups to use it, for instance in a high-speed decoding project in the Engineering Science Department. To show its versatility, we have also demonstrated a number of Handel-C applications which directly generate video signals from the FPGA.

3.2 The RC1000-II Reconfigurable Platform

A second-generation version of Valcard has recently been designed and is being offered for sale commercially by Embedded Solutions Ltd. as the RC1000-II. It differs from the prototype card by being based on the Xilinx 4K part and by carrying a daughterboard interface to the international Industry Pack standard. The intended application areas of the board are high performance signal and control processing, integer/logical co-processing for the PC, legacy hard logic system replacement, and algorithm modelling and architecture design.

The RC1000-II is a standard half size ISA bus card equipped with a Xilinx XC4000E series FPGA in PLCC-84 format. This is socketed, and may be replaced by any pin-compatible FPGA from the Xilinx XC4003E to the XC4010E device range. It is programmable with the Handel and Xilinx XACTstep tools, and a PC provides server microprocessor support. An on-board serial PROM can be used to configure the FPGA for dedicated run time applications. Otherwise the FPGA can be configured over the ISA bus using the PC-based interface programs provided.

The board is equipped with an Industry Pack mezzanine daughter-board slot carrier directly interfaced to the FPGA. The Industry Pack interface is essentially a 16 bit, memory-mapped interface working at either 8 or 32 MHz. A choice of over 400 I/0 modules are commercially available for Industry Pack, covering the needs of most prototyping and low volume bespoke application needs. This mezzanine format is ideal for prototyping custom I/O or additional hardware assist circuitry for the FPGA, such as SRAM or graphics accelerators. In addition general-purpose input/output capability is provided by a 50-pin SCSI-2 style panel connector connected to the FPGA.

The board has a programmable clock which operates over the range 400KHz to 100MHz and can be configured from the host PC. In addition the 33MHz ISA clock, and the Industry Pack 8MHz and 32MHz clocks are available for clocking the FPGA.

In a typical sensor application, this board would be equipped with a suitable Industry Pack module or the general-purpose digital input/output connector would be used to connect to the physical sensors. A control program would be written in Handel-C and compiled to a netlist by the Handel-C compiler. The Xilinx tools would then place and route this netlist and produce a bitmap file for loading into the FPGA. For in-PC operation, the user would probably want to write a PC program which loaded the bitmap into the FPGA over the ISA bus and then went into the main application code. This code would interface with the Handel-C program running in the FPGA to obtain the sensor data after processing. Data exchange between the PC and the FPGA is mediated by a set of software routines and can be run as either an interrupt or a polling interface.

3.3 The 'ValTram' Reconfigurable Module

The 'ValTram' system [7] is based on the Tram module standard originated by Inmos Ltd. It was designed to support the rapid implementation of smart, self-validating sensor systems and conventional control systems through the use of FPGAs and hardware compilation. The ValTram system consists of two daughter boards each of which simply plug into any industry-standard Tram motherboard, or into each other, or into a custom-designed motherboard. These modules are also now available in commercial form from ESL Ltd.

The smaller ComTram (Communication Tram) module is a size 1 Tram which converts a standard transputer 20Mbit/sec serial link into two parallel buses. These buses are linked to as many FPGA modules as necessary by ribbon cable. One bus is used to configure the FPGAs and the other is used to communicate between the host transputer(s) and the FPGA modules once configured. The configuration bus can also be used as an inter-FPGA communication bus.

The larger (size 2) FPGA-Tram module has sockets for up to two Xilinx X3000-series FPGAs. As well as linking to a ComTram module via the ribbon cable buses, it also has a daughterboard connector on which the sensor or control interface can be mounted. Examples of daughterboards which have been constructed range from simple sigma-delta analogue-digital converters up to moderately large interfaces which have their own reprogrammable FPGAs on board.

The Coriolis mass flow meter is an example of one of our more complex sensor systems. The base is a standard, commercial, 10-slot, PC Tram motherboard. The rest of the system is stacked on this mother board and consists of i) three standard, size-1, transputer-based, 4Mbyte compute Trams, ii) one ComTram module, iii) three FPGA-Tram modules driven from the one ComTram, and iv) three application-specific daughterboards on the FPGA-Trams, two of which happen to have their own FPGAs on board. The application is quite a complex one as can be judged from the fact that the implementation has three separate clocks simultaneously active, two active buses, and a total of 5,000 lines of Handel code mapped into hardware implemented in the eight FPGAs.

When running an application, the transputer link is used initially to download FPGA configuration data, and then subsequently to provide data transfer between the Valcard modules and the transputer. Data transfer is controlled by the transputer, using a simple protocol. Bus control is itself almost entirely implemented via Handel, and so can be redefined to suit the requirements of each application. The implementation has been very successful and a commercial partner is currently funding extended industrial trials.

3.4 Aspire Project and ASIC implementations

Our reconfigurable hardware platforms have proved suitable for rapid prototyping of complex sensor systems. However they would usually be too expensive for deployment for industrial control purposes, except perhaps for high end, self-powered systems such as the mass flow meter. For low end, high volume, cost sensitive applications it is necessary to find alternative implementation techniques.

One easily available and attractive route is simply to take the configuration constructed for the FPGA implementation and to transfer it to a commercial gate array technology. In our case the move to Xilinx HardWire technology is particularly simple as functional equivalence is guaranteed by the vendors. The natural implementation of the core of a SEVA sensor would then be a microprocessor system connected to suitable memory components and to the gate array chip(s). Depending on system complexity the processor system might be anything from a single microcontroller to an array of DSP processors. However, in the low cost, simple implementations which are likely to make the earliest impact on the mass market, it is likely that a high-performance microcontroller would be an adequate basis.

It is obviously attractive to consider single chip implementations for high volume systems. This necessitates having a microcontroller core, memory resources and the application-specific circuitry on the same silicon. We have already begun a collaboration with ARM Ltd. and Atmel-ES2 under the European Union OMI (Open Microprocessor Initiative) to develop a route from our language directly into ARM-based ASICs. This ‘Aspire’ Project has the ultimate target of the compilation process being an ASIC which has an ARM processor core, memory resources, network communications, and the designer’s application-specific hardware on the same chip. Given the high MIPS/area and MIPS/watts ratings of the ARM devices, the resulting ASICs should have exactly the right characteristics for the control industry.

The goal of the Aspire project is rather wider than the compilation of RISC-based ASICs however. It is to build a 'shrink-wrap' product for hardware-software co-designers which will allow them easily to design ASICs for embedded computing systems based on ARM processor technology. The Aspire package will consist of a reconfigurable processor system on a PCI-based board, support software including a hardware compiler, and user documentation to enable a programming-literate designer to design such embedded ASICs.

A block diagram of the reconfigurable processor system is shown in Fig. 1. It will be capable of being installed in any standard PCI-based host system. It will be based around the AT91M40200 chip [2] from Atmel-ES2 which is their new microcontroller version of the ARM700TDMI. The box on the left represents the chip boundary of this part. The 'Peripherals' box contains standard timer and interrupt controller modules. The processor, peripherals and small on-chip memory communicate using ARM's on-chip AMBA bus.

Memory resources are expanded on the board-level bus with a bank of DRAM and flash ram. The flash ram will be particularly important for stand-alone uses of the board and it is being designed so that it can operate quite sensibly independently of any PCI host system.

The reconfigurable hardware will consist of three or four large, fast Xilinx 4k series parts one of which supports the PCI interface chip. The FPGAs are bussed together as shown and expansion is provided by daughter-board and back-panel connectors. It is envisaged that an early daughterboard will be one with further FPGA chips.

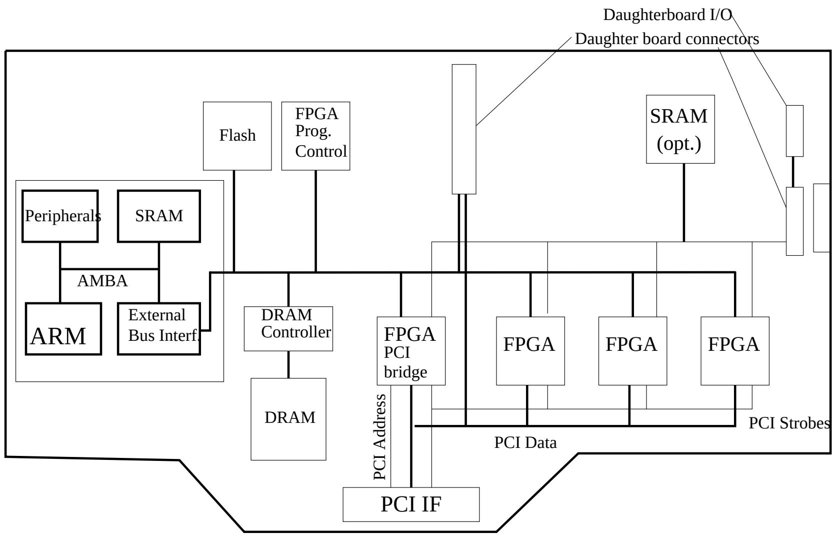

Fig. 1. The Aspire ASIC Prototyping Board

Our support software will compile Handel-C programs into FPGA configurations and will be able to merge these configurations with C or C++ programs for the microprocessor. We have built a co-simulation system which runs C code on the Armulator against a software simulation of the hardware generated by our compiler. This enables us to prototype complex ARM-based systems on a standard workstation with no special hardware. This is clearly important to support SMEs who want to build, or evaluate the building of, ARM-based systems.

After software-based co-simulation, the next step might usually be to implement the same design on the PCI board. We have already taken a number of small hardware/software codesigns through our Handel-C compiler and have them working on the ARM PID7T and PIE boards which we are using as prototypes until the PCI board is completed (in 4Q97). The PCI board is now being designed by ARM to an architectural plan produced jointly by all Aspire partners.

When the user has a co-design operating satisfactorily on the PCI card or similar reconfigurable platform, we produce design files for shipment to Atmel-ES2 from which they can manufacture a single custom ASIC which includes the ARM core together with the application-specific hardware generated from the C-like program. The overall architecture of the chip produced by this process is shown in Fig. 2.

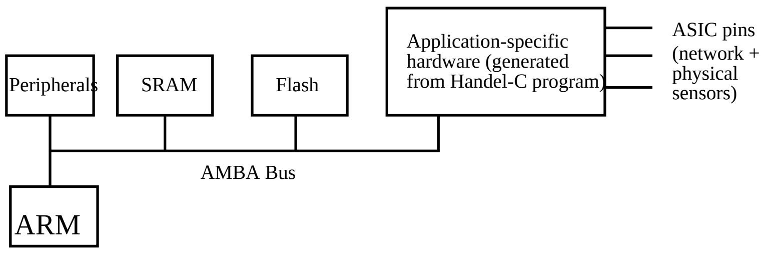
Peripherals
SRAM
Flash
Application-specific hardware (generated from Handel-C program)
ASIC pins (network + physical sensors)
AMBA Bus
ARM

Fig. 2. The Aspire ASIC : General Architecture

We have recently demonstrated a complete flow from our hardware compilation system into ES2's ASIC flow. The result is a hardware design which incorporates our application-specific hardware equivalent to the original Handel-C program, an ARM700TDMI processor core and the interface hardware between the two. This has resulted in satisfactory hardware simulations of the entire system within the ES2-Atmel EDA environment. The only stages we have not yet tested out are final placement and routing and actual manufacture and test. We expect these to be straightforward and so the next major step will be the manufacture and demonstration of an ARM-based ASIC from our Handel-C flow. We are currently planning this design but we have already settled on making it a video-based application.

4 A Large Scale Application

Here we present an example of our Handel-C software and SEVA platforms in use independently of the Oxford groups. Dr. Timo Korhonen at KEK (High Energy Physics Accelerator Research Organization, Tsukuba, Japan) is building a large scale position control system for their Accelerator Test Facility.

This is a positron-electron collider being built to evaluate some of the design considerations for a proposed large linear collider. It consists of 1.54 GeV S-band Injector feeding a damping ring which has a racetrack shape with a circumference of 138.6 m.

The damping ring uses up to 40 ValTram systems to implement dynamic stabilisation of the beam path via the movable alignment tables which carry the beam steering magnets. The positioning of the tables is done with an accuracy better than 20 microns and it is important to remove the effects of traffic-induced vibration and other earth tremors on the apparatus.

It was very pleasing that within four weeks of receiving the system, Timo had interfaced our reconfigurable hardware to the controllers for moving the large magnet-carrying tables, had written Handel-C programs to implement the control algorithm and had the whole system working. His comment that "after two or three days, I completely forgot that it was hardware that I was developing" was especially good to hear.

Even using rather slow Xilinx place and route software, he was still able to test a number of new hardware configurations within a working day. This flexibility and rapid implementation is especially important in an experimental set-up such as this, when the best control algorithm cannot be known in advance of building the system. He is now planning to use this flexibility to enhance the sensing and control systems beyond original brief simply because it is easy to do so.

5 Conclusions and Future Work

Over the next two years we will be designing and building new hardware platforms to support the next generation of self-validating sensor work. For this next phase of work we have identified Fieldbus [5, 4] and PCI bus as important interfaces that need to be supported by our new platforms. We are already confident that these research developments will also result in commercial exploitation.

We are extending the Handel language to raise its level of abstraction while retaining its properties of accurate specification of the temporal properties of programs. We are recoding the compiler to take advantage of the knowledge accumulated by the group through the use of the previous compilation system over the past five years.

Our experience with developing sensor applications has suggested that additional hardware components could be also described within the Handel framework. These include the new generation of 'analogue' FPGAs (e.g. the EPAC chips from IMP Inc.), which are programmed to provide analogue signal conditioning, and programmable inter-connect chips (e.g. Aptix). We hope to extend our compilation tools to target these and other types of programmable hardware components.

We hope to develop some much higher bandwidth smart sensors (not necessarily self-validating sensors) which use video from one or more cameras and execute quite complex algorithms in hardware and software to produce the sensor outputs. One reason for choosing applications such as video-based detection, tracking, and identification is to improve our technology through meeting these challenges. Another is the observed power of video-based demonstrations to effectively carry the message of what our technology actually is and what it can do.

Acknowledgements

I would like to thank all the members of the SEVA group in the Department of Engineering Science at Oxford. I would also like to thank Dr. Timo Korhonen of KEK for his permission to quote details of his work in this paper. Part of the SEVA work has been undertaken as part of a previous grant (GR/34436) and part is being undertaken under a current grant (GR/L41547) for which I wish to acknowledge the support of the UK EPSRC. The Aspire project is supported by the EU OMI office and by ARM Ltd. and Atmel-ES2 which I also wish to acknowledge.

6 References

References

1. M. Atia, J. Bowles, D.W. Clarke, M.P. Henry, I. Page, J. Randall, and J.C. Yang. A self-validating temperature sensor implemented in FPGAs. In W. Luk and W. Moore, editors, *FPL95*, Lecture Notes in Computer Science. Springer Verlag, 1995.
2. Atmel ES2, Zone Industrielle, 13106 Rousset Cedex, France. *AT91 Series Microcontrollers - Product Overview.*
3. D.W. Clarke and P.M Fraher. Model-based validation of a DOx sensor. *Control Engineering Practice*, 4(9):1313–1320, 1995.
4. M. Henry. Fieldbus and sensor validation. *IEE Computing and Control Journal*, 6(6):263–272, 1995.
5. M. Henry. Smart field devices and sensor validation. *IEE Computing and Control Journal*, 6(6):263–272, 1995.
6. M. Henry and D.W. Clarke. The self-validating sensor: Rationale, definitions and examples. *Control Eng. Practice*, 1(4):585–610, 1993.
7. M.P. Henry, N. Archer, M. Atia, J. Bowles, D.W. Clarke, P. Fraher, I. Page, J. Randall, and J.C. Yang. programmable hardware architectures for sensor validation. *Control Eng. Practice*, 4(10):1339–1354, 1996.

8. C.A.R. Hoare and Ian Page. Hardware and software : The closing gap. *Transputer Communications*, 2(2):69–90, June 1994.
9. I. Page. Constructing hardware-software systems from a single description. *Journal of VLSI Signal Processing*, 12(1):87–107, 1996.
10. Ian Page. Video motion tracking using Harp and Handel: A case study. In *Proc. 7th Annual Advanced PLD and FPGA Conference*. Miller Freeman PLC, London, SE18 6QH, May 1997.