



Mimis, K., & Kocak, T. (2011). Area efficient System-on-Programmable-Chip design for a wireless touch-triggered machining probe.

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms.html>

Take down policy

Explore Bristol Research is a digital archive and the intention is that deposited content should not be removed. However, if you believe that this version of the work breaches copyright law please contact open-access@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline of the nature of the complaint

On receipt of your message the Open Access Team will immediately investigate your claim, make an initial judgement of the validity of the claim and, where appropriate, withdraw the item in question from public view.

Area efficient System-on-Programmable-Chip Design for a Wireless Touch-Triggered Machining Probe

Konstantinos Mimis

Dept. of Electrical & Electronic Engineering
University of Bristol
Bristol BS8 1UH, UK
Email: k.mimis@bristol.ac.uk

Taskin Kocak

Dept. of Computer Engineering
Bahcesehir University
Istanbul, Turkey 34353
Email: taskin.kocak@bahcesehir.edu.tr

Abstract—In this paper, we present the design and implementation details of a System-on-Programmable-Chip (SoPC), which replaces the circuitry that controls the operation of a touch-triggered machining probe with radio transmission capability. The probe is used to get precise measurements of three dimensional geometric parts. The goal is to achieve size reduction of an existing circuitry inside the probe, that consists of two microcontrollers and a reconfigurable hardware device such as a Field Programmable Gate Array (FPGA), that implements some custom functions. In order to reduce the printed circuit board (PCB) area occupied by these three chips and the associated routing area, we combine them on a single SoPC. An overall 6 fold reduction in PCB area occupied, corresponding to nearly 15cm² is achieved.

I. INTRODUCTION

Precise measurements are required in various industrial applications such as cutting and drilling. A touch probe can provide precise measurements by producing a signal according to the contacts it makes with the part being measured. The generated signal is then fed to a computer numerical control (CNC) machine to get the precise geometrical coordinates as shown in Fig. 1. The transmission of the signal between the probe and the CNC machine can be either wireless or hardwired. The wireless probe has the benefit of increasing the access area on the machine table; however, this comes with the expense of more circuitry to process the wireless connectivity. The printed circuit board (PCB) area on a probe is a prime real estate as the probe needs to be small enough for the machine head to access the required locations on the part being manufactured.

As the available capacity in FGPA devices grows, so does the interest in System-on-Programmable-Chip (SoPC) design for a variety of applications. The use of SoPC designs varies from digital drive controllers [1, 2] and general SoPC implementations for supporting student projects [3, 4] to programmable protocol processors [5] and controllers for robotic applications [6]. Moreover because of FPGAs being reconfigurable in nature, functions can be implemented directly in hardware, reducing computational time [7, 8].

A SoPC design has been demonstrated in [9, 10], in which

the NIOS processor by Altera [11] was used. Similar to that where designs that used the provided by Altera processor in their SoPC designs [2, 12, 13] demonstrating ease of integration and short turn-around-times. Although the design flow is similar, our work differs in the sense that hardware/software co-design could not be implemented, as the software was already developed.

Hence, the goal of this work is to explore a design alternative based on a SoPC integration method. The motivation behind the design of the SoPC is to achieve reduction in the count of assembled components, which will decrease the production cost, radically reduce the area occupied and enhance the quality of the product. More precisely, we aim to reduce the area occupied by the circuitry, while maintaining high compatibility with the rest of the design, which remains intact, as well as with the existing software that is executed on the microcontrollers.

The rest of the paper is organized as follows: Section 2 describes the touch-triggered machining probe used in this work. SoPC components are covered in Section 3. Section 4 discusses the implementation details and section 5 provides experimental results. Finally, section 6 concludes the paper.

II. TOUCH-TRIGGERED MACHINING PROBE

In this work, we consider a touch triggered probe with radio signal transmission for use with machining centers such as the ones from Heidenhain, Marposs, Renishaw and Tesa. The probe is mounted on the machining centre's spindle and comes in direct contact with the part under inspection. It is able to measure complex 3D geometries with high accuracy. A radio interface is mounted within the machine's working envelope and is connected to the CNC machine as shown in Fig. 1. The radio interface connects to the touch probe through a wireless radio link and acts as an interface between the machine and the probe.

The measurement principle of a touch-triggered probe is based on a well established strain gauge technology. The touch probes are typically equipped with multi-axis sensing capability and utilize accelerometers in order to determine if

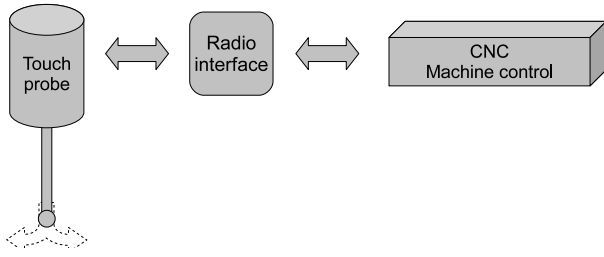


Fig. 1. Touch probe system architecture

forces applied to stylus are valid triggering, or just caused by high speed acceleration of the probe.

A. Probe circuitry

The considered tough-triggered probe electronics amongst other, include two microprocessors, one FPGA and a radio modem. The main processor encodes and decodes the radio messages that are sent or received through the FPGA to or from the radio modem. The main and secondary processors are PIC microcontrollers from MicroChip[14]. The main PIC coordinates the interaction among the components as illustrated in Fig. 2. The second processor, interfaces through the FPGA and handles inputs from the strain gauges and a 3-axis accelerometer. The main PIC monitors the battery and a single axis accelerometer. The FPGA manipulates data sent to and from the radio modem, and it also recognizes messages addressed to the particular probe through the wireless interface. All of them make use of a single crystal oscillator, except for the system PIC which utilizes a second one. The second crystal is of lower frequency and is used to put the system PIC into a standby mode when inactive, in order to reduce power consumption.

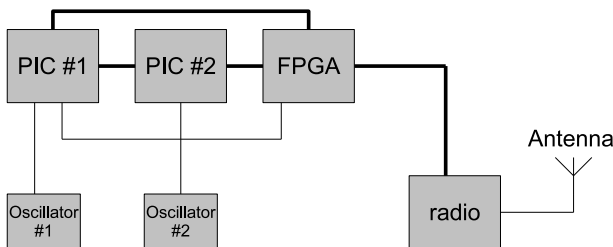


Fig. 2. Circuitry of the touch probe

III. SYSTEM-ON-PROGRAMMABLE-CHIP COMPONENTS

There are several components that make up the SoPC. This section covers the components such as processor and peripheral Intellectual Property (IP) cores as well as the interconnect bus used. When choosing the processor IP core for this work, some preference was given in PIC compatible cores, as the software already exists and is written for the PIC18F family of microcontrollers. By selecting a compatible core we can limit the changes needed in the software.

A. Processor IP

Design reuse is common practice in SoPC design methodology. For that reason we chose to use a processor core which supports a particular bus specification, for which we could find the needed peripherals (counters, timers, etc). The main source for our cores is OpenCores project, an on-line community developing and distributing designs, under Lesser General Public License (LGPL). The supported bus is Wishbone, which is promoted by OpenCores and is supported by the majority of their cores.

A core compatible with the PIC18C family is ae18. It has no peripherals implemented but it is Wishbone compliant, which means that peripherals can be interfaced quite easily. It utilizes a separate instruction and data bus and provides high and low level interrupt sources that can be used to attach an interrupt controller. Hence, ae18 was appropriate choice for this work.

B. Peripheral IPs

Because most of the available microprocessor cores do not include any peripherals, we selected some peripherals from OpenCores that can be integrated in the SoPC, to add functionality to the microprocessors. The cores include an interrupt controller, I/O (Input/Output) controller, timers, counters and an SPI(Serial Peripheral Interface) interface. The GPIO IP core is a user-programmable general-purpose I/O controller and can be used in designs requiring simple input and/or output software controlled signals. The SPI (Serial Peripheral Interface) module supports the well known SPI protocol. The simple programmable interrupt controller supports up to 8 interrupt sources and polarity and sensitivity (either edge or level) are programmable per interrupt source. Finally the PWM/Timer/Counter (PTC) IP core is a user-programmable PWM, Timer and Counter controller used to implement functions like Pulse Width Modulation (PWM), timer and counter facilities.

C. Custom functions IP

The functions previously implemented in a separate FPGA device, will now be integrated in the SoPC. Some of the functions in this block include handling data between the modem device and the PIC microprocessors, sending debugging signals to the exterior of the FPGA, controlling the modem device and more. This block was treated as an individual IP block, and was interfaced with the rest of the components in the SoPC as dictated by the connections in the original implementation.

D. SoPC Interconnect Bus

A bus provides the means of interconnecting functional units. Integrating a number of IPs can be challenging in terms of the interconnection bus used, and may consume a significant amount of time even though it does not add any value. Work has been conducted in this respect in order to automate the procedure [15] and assess the various interconnection buses available [16]. The choice of the bus though remains a decision tightly coupled with the particular application.

In this work, the Wishbone [17] bus is chosen, which is a flexible bus compatible with the IP cores from OpenCores. The choice of Wishbone will ease the integration process and minimize time spend in interconnection issues. It supports multiple types of interconnections as well as multi-master configurations leaving the arbitration scheme up to the designer. More than one Wishbone buses can be used in one design to allow for hierarchical interconnections.

IV. SoPC IMPLEMENTATION

After all the cores were selected, the functionality added through attaching peripherals to each processor core had to be decided and realized. The power of a SoPC implementation is that highly customizable processors can be used in the design, by selecting what kind of peripherals to attach to them. An important consideration was to ease migration to the new design together with reducing the PCB area occupied. To achieve these, special attention was given to the architecture.

A. Architecture

The microcontrollers in the original design come from the same family of microcontrollers (PIC18F) and because the microprocessor cores selected, do not have any peripherals, the first consideration is to interface the needed peripherals to the microprocessor cores. The peripherals are interfaced to the processor through the Wishbone bus. By the use of decoders they can be accessed in a similar way as in the original PIC18F implementation. The SoPC architecture is depicted in Fig. 3.

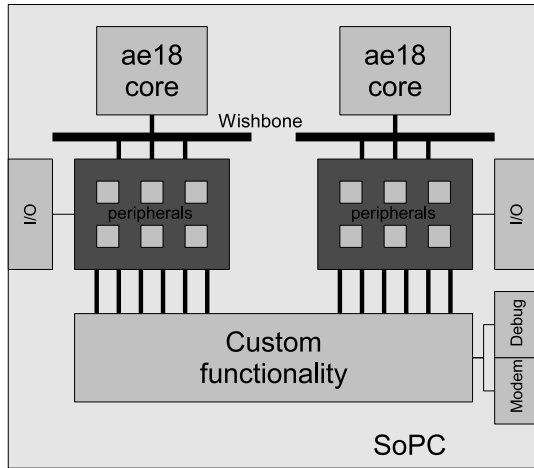


Fig. 3. SoPC architecture

The main processor was interfaced with 5 I/O modules, an SPI module and a PWM/counter/timer module. The secondary processor was interfaced with 3 I/O modules, an interrupt controller and a PWM/counter/timer module. The respective modules' registers were mapped in the processors' memory, so that they can be accessed similarly to the original PIC microcontrollers. This is accomplished by the used of decoders in the data bus, and the mapping can be easily altered to fit different needs.

B. Overall SoPC

The overall architecture of the SoPC can be seen in Fig. 3. The three blocks are instantiated in a top module, in a hierarchical manner. The cores communicate through dedicated signals. These signals are mainly used to synchronize the function of the modules and to configure the operational mode of the custom IP core. The main PIC sends and receives data to and from the custom IP core through the SPI interface. Data is buffered inside the custom IP core and burst to the modem, or send to the main PIC if they have been received by it.

For the I/O interconnections tri-state buffers have been used, and all the pins that are not used for on-chip connections have been assigned to an external port pin, to allow external use. Two more ports exist in the top module. One of them is used to connect the external modem and the other one to debug the operation of the custom IP core. Also if no instruction memory is present on the chip, the memory ports of the two PIC compatible designs can be connected to external memory chips.

V. EXPERIMENTAL RESULTS

The SoPC was compiled and synthesised for a variety of FPGA devices from different manufacturers. Each vendor's device comes with different tools namely ISE from Xilinx, Quartus from Altera and Libero from Actel. The design flow is similar for all of them with steps such as compilation, synthesis, placement and routing. The FPGA families considered were Virtex 4/5 and Spartan 3 from Xilinx, Cyclone, Cyclone II/III from Altera and the Fusion family from Actel. Table I shows the devices able to accommodate the design and each one's packaging.

The area occupied by the microcontrollers and the FPGA in the current design is 1800 mm^2 . That is roughly equivalent to a chip of dimensions $43 \text{ mm} \times 43 \text{ mm}$. Thus, any FPGA that occupies less or even this much area leads to a reduction on the PCB.

A. Results

Our aim is to reduce the area occupied by the components as well as their count. We were able to fit the design in the FPGAs given in Table I. Fig. 4 summarizes the results achieved. By using the 1800 mm^2 area that is occupied by the original design as a reference, we demonstrate the difference between the area occupied by the SoPC implementations and the original area.

As can be seen from the figure, we achieved to reduce the PCB area occupied by as much as 6 times, equal to 15 cm^2 in PCB area. Best results were accomplished by using the Virtex 4 family from Xilinx, but at the same time we have to take into account the resources that Fusion family has to offer (analog functionality, flash memory), that can be useful in the future, even though the dimensions do not make it an appealing choice.

Our second goal of achieving high compatibility with the code currently available has been achieved to a great extent.

Family	Model	Package (available I/O)	Area (mm x mm)
Xilinx Virtex 4	4vlx15	sf363(240)	17x17
Xilinx Virtex 4	4vlx25	sf363(240)	17x17
Altera Cyclone	ep1c12	f324(249)	19x19
Actel Fusion	AFS1500	fg484(223-40)	23x23

TABLE I
FPGA DEVICES

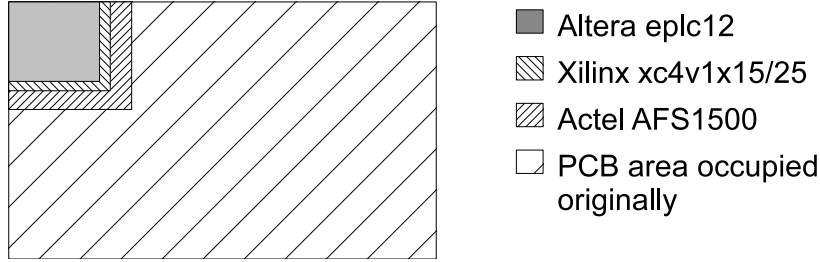


Fig. 4. PCB area reduction of the SoPC design on various FPGA devices compared to the original design

The code has to be modified slightly. That is due to the peripherals attached to the processors. They are similar but do not have exactly the same functionality as those in the original PIC microcontrollers. Moreover the peripherals' registers are mapped into different memory addresses, so the code has to be updated to meet the new memory mapping.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an area-efficient System-on-Programmable-Chip (SoPC) design for a touch-triggered machining probe. The probe, is used to get precise measurements in various industrial applications including cutting and drilling. Such probes need to be small enough for the machine head to access the required locations on the part being manufactured. This work explored a single chip solution to reduce the area required to implement the probe circuitry.

Our results showed that combining two microprocessors and a custom functional block on an SoPC type design and implementing them on an FPGA reduces the overall area by 6 times. This is equivalent to nearly 15 cm^2 area reduction. Moreover the SoPC approach not only provides less component count and area, but at the same time adds flexibility. The product's lifetime can be increased as we have the opportunity to update the design to meet new needs, and the hardware/software partitioning of the SoPC can be reconsidered at a later time. Future directions for this work include merging the existing software to make it run on a single multi-threaded processor and also explore an Application-Specific Integrated Circuit (ASIC) implementation to reduce the area further.

REFERENCES

- [1] M. Lin, "Implementation of a SOPC based switched reluctance drive," in *IEEE Annual Conf. on Industrial Electronics*, pp. 776–780, 2006.
- [2] Y. Rung, P. Huang, and C. Chen, "Development of a SOPC for PMSM drives," in *Midwest Symposium on Circuits and Systems*, vol. 2, pp. 329–332 vol.2, 2004.
- [3] J. Hamblen, "Using second generation SOPC boards for student design projects," in *IEEE Int. Conf. on Microelec. Systems Education*, pp. 69–70, 2005.
- [4] T. Hall and J. Hamblen, "System-on-a-programmable-chip development platforms in the classroom," *IEEE Trans. on Education*, vol. 47, no. 4, pp. 502–507, 2004.
- [5] C. Toal and S. Sezer, "A 32-Bit SoPC implementation of a p5," in *IEEE Computer Society*, p. 504, 2003.
- [6] S. Sanchez-Solano, A. Cabrera, I. Baturone, F. Moreno-Velo, and M. Brox, "FPGA implementation of embedded fuzzy controllers for robotic applications," *IEEE Trans. on Industrial Electronics*, vol. 54, no. 4, pp. 1937–1945, 2007.
- [7] B. Dalay, "Accelerating system performance using SOPC builder," in *Int. Symposium on System-on-Chip*, pp. 3–5, 2003.
- [8] A. Farmahini-Farahani, S. Fakhraie, and S. Safari, "SOPC-Based architecture for discrete particle swarm optimization," in *IEEE Int. Conf. on Electronics, Circuits and Systems*, pp. 1003–1006, 2007.
- [9] S. Poussier, H. Rabah, and S. Weber, "SOPC-based embedded smart strain gage sensor," in *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream*, pp. 1131–1134, Springer, 2002.
- [10] S. Poussier, H. Rabah, and S. Weber, "Smart adaptable strain gage conditioner: hardware/software implementation," *IEEE Sensors Journal*, vol. 4, no. 2, pp. 262–267, 2004.
- [11] Altera, "Nios II processor reference handbook," 2009.
- [12] Y. Kung and G. Shu, "Design and implementation of

- a control IC for vertical articulated robot arm using SOPC technology,” in *IEEE Inter. Conf. on Mechatronics*, pp. 532–536, 2005.
- [13] J. Wang, Y. Chen, J. Xie, B. Chen, and H. Lin, “System structure for FPGA-based SOPC design using hard tasks,” in *IEEE Int. Conf. on Industrial Informatics*, pp. 1154–1159, 2008.
 - [14] Microchip, “PIC18F2420/2520/4420/4520 datasheet,” 2007.
 - [15] S. Zammattio, “SOPC builder, a novel design methodology for IP integration,” in *Int. Symposium on System-on-Chip*, p. 37, 2005.
 - [16] E. Salminen, V. Lahtinen, K. Kuusilinna, and T. Hamalainen, “Overview of bus-based system-on-chip interconnections,” in *IEEE Int. Symposium on Circuits and Systems*, vol. 2, pp. 372–375 vol.2, 2002.
 - [17] OpenCores.org, “WISHBONE System-on-Chip (SoC) interconnection architecture for portable IP cores,” 2002.