



## MIGRATING MICROCONTROLLER BASED CONTROL SYSTEMS INTO THE NANO FABRICS OF FPGAS

Santosh A. Shinde, Vinod G. Shelake and R. K. Kamat\*

VLSI Laboratory, Department of Electronics, Shivaji University, KOLHAPUR 416 004

### ABSTRACT:

*The advent of cheap and immensely capable Field Programmable Gate Arrays (FPGA) facilitates a programmable nano-space that allows the process control designers to move their design to a “soft” re-programmable hardware environment. Although this virtually-instrumented nano-level system bread boarding is cumbersome due to the complexity of the state of the art hardware descriptor languages, there exist completely tested soft processor cores for rapid prototyping. With these soft processor cores, it is possible to migrate the entire embedded systems functionality into the nano-fabrics of the FPGAs. The present paper communicates rapid prototyping of microcontroller based control systems by integration of the Soft IP core on FPGA. A case study is taken up with Picoblaze, the Soft IP core of Xilinx implemented on Spartan III FPGA. The case study reports implementation of single setpoint temperature controller. The customization achieved is evident from the synthesis view of the implemented processor. Usage of resources and redundant hardware presented in the design summary justifies the selection of an appropriate FPGA for the intended application. The results serve as a guiding source for implementation of full custom Nano-ASIC implementation with an added advantage of thorough testing in an accelerated time to market design paradigm.*

### INTRODUCTION

The last decade has witnessed a convergence in molecular-scale architectures towards reconfigurable FPGA platforms. Today's FPGA, eliminates much of the overhead needed to support reconfiguration in traditional CMOS circuits, by using the molecular based switch that holds its own state and can be programmed without extra wires. Since their inception, microcontroller has become the soul of the instrumentation systems owing to the intelligence and flexibility achieved with its inclusion by portioning the part of the design to soft medium using its instruction set. Embedded microcontroller modules offer many advantages over the standard PC such as low cost, small size, low power consumption, direct access to hardware, and if available, access to an efficient preemptive real-time multitasking kernel. However, typical difficulties associated with an embedded solution include long development times, limited memory resources, and restricted memory management capabilities [1]. In order to make the microcontroller with the intended instrumentation,

there is a growing trend of sporadically incorporation of system instrumentation features into processor architectures [2]. However, the design theme of microcontroller based mainly software centric implementations do not guarantee the improvement in performance and power efficiency in all the circumstances due to the general purpose architecture of the target microcontroller itself. The development of Field Programmable Gate Arrays has provided yet another opportunity to the instrumentation professionals to fine tune the controller architecture as per the process requirements. The FPGA based development platform enables the developer to test and add features in parallel without the need for repeating the complete testing of the designed instrument in an iterative fashion [3]. Thus the FPGA based compilation of the instrumentation and control application results in a highly optimized silicon implementation that provides true parallel processing with the performance and reliability benefits of dedicated hardware circuitry.

\*rkk\_eln@unishivaji.ac.in

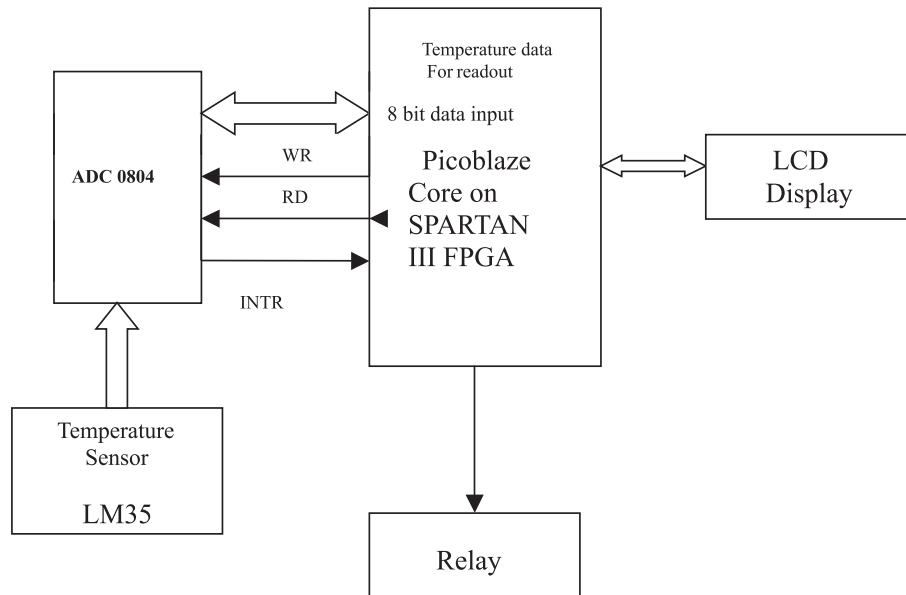
The advantages of the FPGA based design paradigm can be further enhanced combining with the soft processor cores developed by third party. These soft IP cores not only allow the integration of system design into a single FPGA device but also enables optimized division of the system functionality into hardware and software. Thus the FPGA platforms with soft IP cores are emerging for process control applications enabling the designers customization of the processor core in an rapid system development schedules leading to less time to market. This design race is also fuelled by the development of low-cost/high capacity FPGAs built on 90nm/300mm technology, which creates an opportunity very analogous to the 8-bit microprocessor revolution. The present communication reports a case study of picoblaze based soft IP core integration on Xilinx Spartan III FPGA for single setpoint temperature control application.

#### **OVERVIEW OF SOFT IP PROCESSOR CORES BY DIFFERENT VENDORS**

It is worthwhile here to present a brief review of the soft IP cores available from different vendors. Owing to the advantages of the soft IP processors cores towards reconfigurability, customization and emulation, many manufacturers have developed these cores either for a specific FPGA family or as a third party solution. Altera has developed ARM as a hard process core while NIOS and NIOS II as soft IP cores [4]. Similarly ATMEL and Quick Logic [5] have come out with AVR and MIPS as hard processor cores respectively. However, the most popular amongst the design community is Xilinx's PicoBlaze a fully embedded 8-bit RISC microcontroller core. The main features of Pico Blaze are its compactness, and cost-effectiveness (as it is provided as a free), and well documented VHDL source file with royalty-free re-use within Xilinx FPGA's. The VHDL listing of the PicoBlaze frees it from obsolescence as the same can be updated as per the design requirement. The PicoBlaze core can be embedded within the target FPGA, however with certain restriction to the device family selection.

The PicoBlaze [6] design was originally named KCPSM which stands for "Constant(K) Coded Programmable State Machine" (formerly "Ken Chapman's PSM"). Ken Chapman was the Xilinx systems designer who devised and implemented the microcontroller. The KCPSM3 core is designed for Spartan 3 family, whereas the KCPSM2 and KCPSM are meant for vertex 2 or vertex 2 pro Spartan II and Spartan 2(e) or vertex (E) respectively. In general the PicoBlaze soft processor provides 49 different instructions, sixteen 8-bit registers, 256 directly addressable ports, and a maskable interrupt. The program length is 256 instructions, and all address values are specified as 8 bits contained within the instruction coding. The design is based on the RISC "Harvard architecture" model with separate data and instruction ports. Its basic functionality is easily extended by connecting additional logic to the microcontroller's input and output ports. PicoBlaze delivers 50 million instructions per second (MIPS) much faster than commercially available  $\mu$ controller devices, yet occupies a tiny footprint of just 35 Configurable Logic Blocks (CLBs). This processor has an 8-bit bus, which means all registers and arithmetic operations are only 8 bits. There are 16 general purpose registers to quickly access data (i.e S0 to SF), and programs can be up to 256 assembly language instructions long. The PicoBlaze has an internal memory for storing data (called the scratchpad) with 64 locations. This is to store data which isn't used as frequently as other data. Operations can only be performed on data in registers, but there are only 16 registers. However, data can be stored and retrieved from the scratchpad to the registers. There are two flaga ZERO and Carry Flag. The ALU operation results affect the ZERO and CARRY flags. The PicoBlaze module has 256 input ports and 256 output ports. An 8-bit address value provided on the PORT\_ID bus together with READ\_STROBE or WRITE\_STROBE signals indicates the accessed port. The port address can be either supplied in the program as an absolute value, or specified indirectly as the contents of any of the 16 registers.

## Design Problem



**Figure 1 : Block diagram of the temperature control system application with integration of Picoblaze.**

In this paper a temperature control application is developed by integration of the soft IP core of Picoblaze on Spartan III FPGA. The block schematic of the system is shown in figure 1. The ADC 0804 is interfaced in an handshake manner with the Picoblaze. Current temperature and the set point is displayed on an 16 x 2 LCD in an alternate manner. ON-OFF strategy of control is implemented with the assembly language instructions of the Picoblaze.

### DESIGN METHODOLOGY

Design methodology followed for the implementation is given below in step by step manner.

1. PicoBlaze core was downloaded from the Xilinx website after registration.
2. The KCPSM3 version of the Picoblaze was downloaded as the target FPGA is SPARTAN III. The core comprises of VHDL files (named as KCPSM3), Program ROM, and an Assembler (KCPSM3.exe) and manual.
3. The top level VHDL program for the temperature controller was developed in VHDL. Behavioral model is developed the VHDL listing of which is given in the following point of the paper.
4. Appropriate assembly language program was developed for single setpoint temperature controller. The LCD display driver and the handshaking of ADC are the main parts of the program.
5. The assembly language program was executed using KCPSM3.exe. The output is in the form of ROM file which is nothing but the program developed for the Picoblaze.
6. The KCPSM3 VHDL file ,generated ROM file are added to the project environment in Xilinx webpack.
7. The resulting file is then simulated, synthesized and the bit and MCS files are then dumped in the SPARTAN III FPGA board using JTAG port. This forms the customized core of the Picoblaze for the temperature controller application.

## RESULTS AND DISCUSSIONS

The customized version of the Picoblaze for temperature controller application was successfully implemented on a Spartan III FPGA board. The same worked successfully under the umbrella of the assembly language program developed for this purpose. Moreover the design cycle in the course of simulation, verification presents valuable information given in the following paragraphs.

The design information as appeared in the Xilinx webpack is as follows:

```

Design Information
-----
Command Line : C:/Xilinx/bin/nt/map.exe -intstyle
ise -p xc3s400-pq208-4 -cm
area -pr b -k 4 -c 100 -tx off -o temp_controller_
map.ncd temp_controller.ngd
temp_controller.pcf
Target Device : x3s400
Target Package : pq208
Target Speed : -4
Mapper Version : spartan3 — $Revision: 1.16 $
Mapped Date : Fri Feb 1 16:14:39 2008

```

Design summary gives the entire technical specifications of the customized Picoblaze implementation on the target Spartan III FPGA. The logical utilization is wound 1% with details as follows:

### Logic Utilization

- Number of Slice Flip Flops:  
70 out of 7,168 1%
- Number of 4 input LUTs:  
112 out of 7,168 1%

The logic distribution gives an idea of the extent to which the FPGA resources have been used.

### Logic Distribution

- Number of occupied Slices:  
98 out of 3,584 i.e. 2%

- Number of Slices containing only related logic:  
98 out of 98 i.e. 100%
- Number of Slices containing unrelated logic:  
0 out of 98 i.e. 0%

The other finer details of the implementation are as follows:

- Total Number 4 input LUTs:  
184 out of 7,168 2%
- Number used as logic: 112
- Number used as a route-thru: 4
- Number used for Dual Port RAMs: 16  
(Two LUTs used per Dual Port RAM)
- Number used for 32x1 RAMs: 52  
(Two LUTs used per 32x1 RAM)
- Number of bonded IOBs:  
29 out of 141 20%
- IOB Flip Flops: 28
- Number of Block RAMs:  
1 out of 16 6%
- Number of GCLKs :  
1 out of 8 12%

The total equivalent gate count used for the temperature controller design is 75,029. The analysis gives valuable results which guides the user as regards to the effective selection of the target FPGA. It also presents a true picture of the used resources and redundant hardware which could have been avoided in the following revision of the project to optimize the power and speed metrics of the system. The RTL schematic shown in figures 2 and 3 are useful for the full custom ASIC implementation for the dedicated temperature controller applications. Thus the Soft IP core integrated with the FPGA offers a rapid prototyping environment to realize the semicustom ASIC design in the nano-space of the FPGA and progress eventually towards the dedicated full custom ASICs.

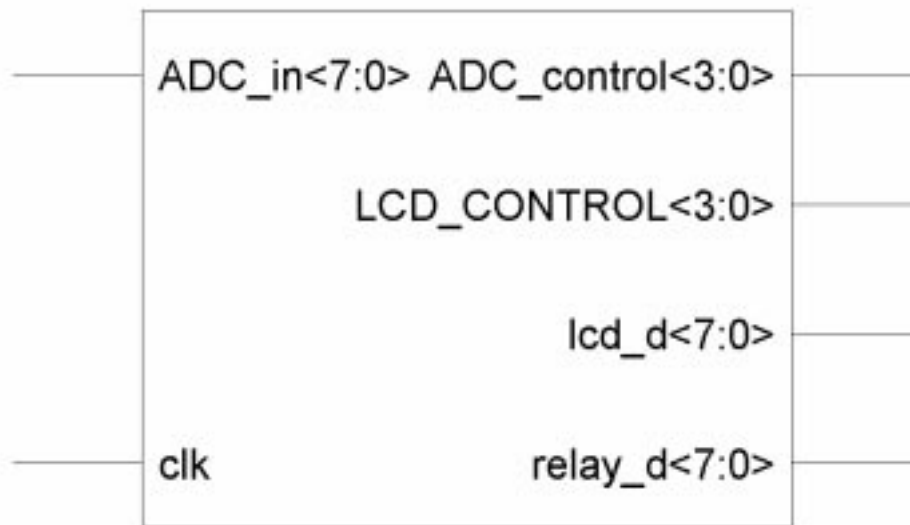


Figure 2 : Higher Level Schematic for the temperature controller

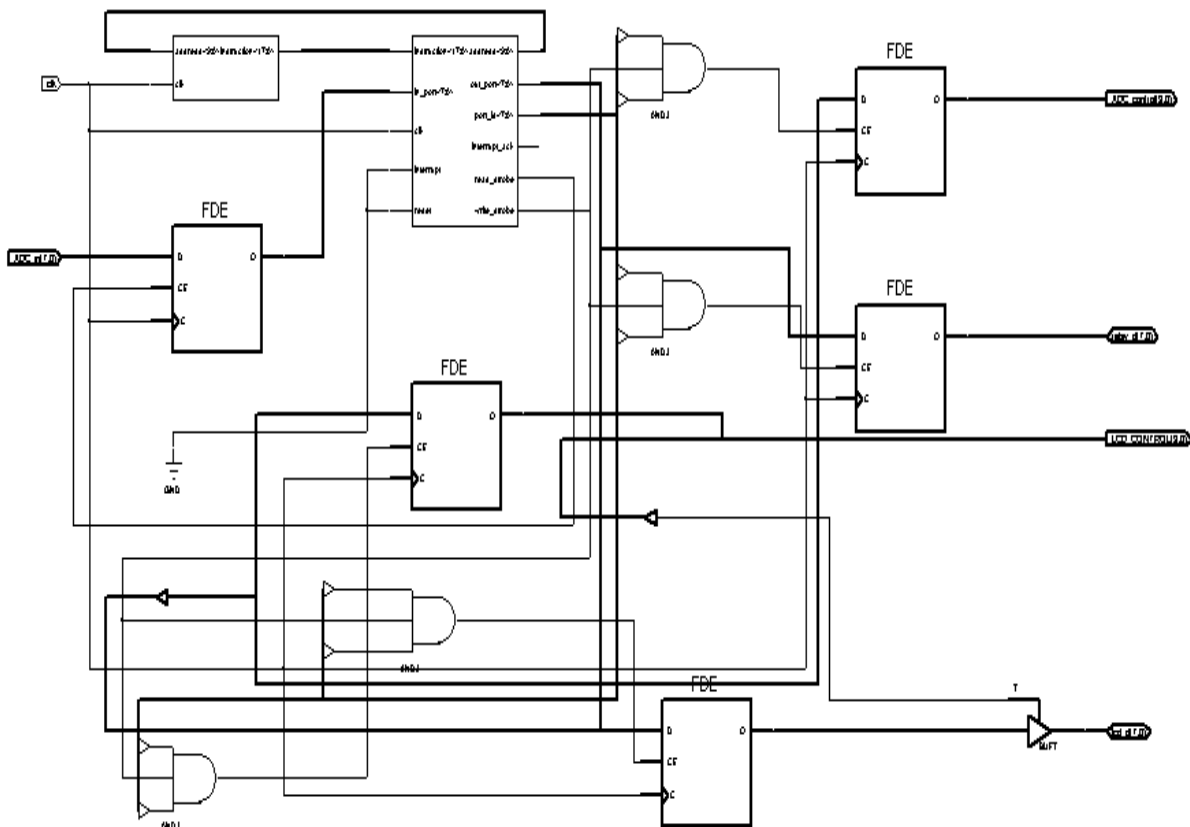


Figure 3: LUT based implementation of the temperature controller

### ACKNOWLEDGEMENTS

This work is supported in part by the DST-SERC Fast Track Project for Young Scientist Grant SR/FTP/ETA-14/2006 entitled "Development of FPGA based open source soft IP cores for parameterized microcontroller design" to Dr. R.K. Kamat.

### REFERENCES

1. Schofield, Ian S.; Naylor, David A., Advanced Software, *Instrumentation control using the Rabbit 2000 embedded microcontroller, Control, and Communication Systems for Astronomy*. Proceedings of the SPIE, Edited by Lewis, Hilton; Raffi, Gianni. Vol. 5496, 2004, PP 392-401.
2. *FPGA based CPU instrumentation for hard real-time embedded system testing*, Richard Fryer, ACM SIGBED Review archive, Vol. 2 , Issue 2 , April 2005, Special issue: IEEE RTAS 2005, ISSN:1551-3688, PP 39 – 42.
3. Parab, J. S. ; Shelake, V. G. ; Kamat R. K.; et...al *Exploring C for Microcontrollers: A Hands on Approach* , Springer Netherlands, 2007.
4. Rambo Susan; *Altera tweaks soft-core processor* <http://www.embedded.com/story/OEG20030225S0022> retrieved on October 20, 2007.
5. *QuickLogic QuickMIPS ESP Is Industry's First — Combines a High-speed Processor With Hardwired Functions and Field Programmability, Design and Reuse*.
6. <http://www.us.design-reuse.com/news/news349.html> Newsletter, retrieved on October 21, 2007.
7. Blaze, Pico; Xilinx Inc. website [http://www.xilinx.com/ipcenter/processor\\_central/picoblaze/picoblaze\\_user\\_resources.htm](http://www.xilinx.com/ipcenter/processor_central/picoblaze/picoblaze_user_resources.htm) retrieved on October 21, 2007.

