A Multi-Paradigm Approach to Teaching Students Embedded Systems Design using FPGAs and CPLDs

David C. Dyer
School of Engineering, University of Warwick
Coventry, Warwickshire, CV3 4AL
d.c.dyer@warwick.ac.uk

Yan Lin Aung
CHiPES Research Centre, Nanyang Technological University, 50 Nanyang Drive, Singapore 637553
layan@ntu.edu.sg

ABSTRACT

To create optimal embedded electronic systems, it is essential to ensure all implementation options are considered, and students of electronics and computer engineering must be educated in hardware, software and firmware.

We begin by reviewing in an educational context various implementation techniques. These include commercial microcontrollers, custom instruction set architectures (ISA), Field Programmable Gate Arrays (FPGAs) for 'soft-core' processors and dedicated digital 'engines', as well as Complex Programmable Logic Devices (CPLDs) for interface management.

Thereafter, we describe our work to create a platform that incorporates the above but is extended to include software development and tools. Regarding ISAs, we use an FPGA configured with a soft-core ARM Cortex-M1 32-bit processor but also introduce a custom hybrid RISC/CISC 12-bit processor called VIP. This helps students explore and compare multiple soft-core implementation issues. Furthermore, unlike most proprietary platforms, we can provide students with the HDL code of all our peripherals and interfaces. Especially those for the address and data lines used communicate with devices on our associated custom Teaching Auxiliary Board (TAB); which itself uses a CPLD programmed to provide features such as bus handshake, protocol conversions, timers, interrupts and simulation of 'slow memory locations'.

We believe that our holistic approach provides exceptional learning opportunities to show how implementations may be partitioned across FPGAs and CPLDs acting as dedicated programmed logic or programmable soft-core processors.

Categories and Subject Descriptors

K.3.2 Computer science education, Information systems education

General Terms

Design, Experimentation

Keywords

FPGA, soft-core, CPLD, microcontroller, ADC, DAC, interfacing, embedded

FPGAWorld'14, Sept 9-11, 2014, Stockholm and Copenhagen

1. INTRODUCTION

It is with educators of 'novice' students in mind that we have prepared this paper and recognize fully that commercial product development has additional constraints not described herein.

Traditionally, teachers describe sequentially operational principles of electronic devices but how best to combine them concurrently in practical applications can be much more challenging to explain and there has to be a balance between abstraction and detail. For example, a floating-gate field-effect transistor is at the heart of 'flash' memory but knowing the physical mechanisms alone does not mean that one can store encoded pictures.

'Education' in engineering must include tuition on concepts, applications, implementations and deployment – with due regard to the psychology of learning. These are all essential if students are to become creative designers. Many opportunities are afforded in systems-level design by the abundance of programmable devices but it is the wise combination of devices and techniques that underpins profitable commercial exploitation. To segregate topics, and present them to students in isolated classes for recall alone does not develop the skills demonstrated by experienced inter-disciplinary designers.

The diversity of field programmable devices and their rapidly changing parameters means that teachers can present too much material for novice students to assimilate well. Topics presented in a series of lectures have to be applied concurrently and eventually perhaps subconsciously. However, like riding a bicycle, once learned it is nearly impossible to forget. Consequently, experienced designers and educators can find it difficult to remember what it was like not to 'know' the myriad of details that contribute to robust embedded systems design.

Much of the terminology in common use is context sensitive and has mutated over time. For example, the term 'field' was used to differentiate 'user' programmability from unalterable behaviour defined during fabrication. Where 'programming' is now done, it what way, and by whom, and for what purpose must be explained in detail to students.

We believe educators have to make programmable technologies more amenable to understanding so they may be better applied in creative ways. In our work, we have tried to address these challenges and satisfy multiple and diverse educational expectations to improve the overall 'student experience'.

The TAB platform and environment has been used since 2012 by a total of 125 2nd year students of Computer Engineering at NTU. While the VIP ISA has been used each year by approximately 350 1st year students. VIP was introduced in 2013 at the University of Warwick to 140 students and adoption of TAB is being planned.

2. REVIEW OF FIELD PROGRAMMABLE TECHNOLOGIES AS RELATED TO EDUCATIONAL NEEDS

The benefits of various field programmable technologies to commercial product development and manufacture are not necessarily the same as those in education. A multitude of issues of commercial concern do not impact teachers at all, such as supply chain management and warranty provisions. Experienced designers understand very well the capabilities of each device and how to achieve the specified functional requirements subject to constraints like overall cost and development time. Expressions like 'big memory' are understood relative to previously developed products but have little, or no, meaning in isolation to students.

In this section, we review briefly field programmable technologies and highlight key parameters. This serves to identify some of the gaps between the commercial and educational communities.

2.1 Confusion and Misrepresentation

When devices have names that sound similar they are easily confused, for example, thermostat vs. thermocouple and thermistor vs. thyristor. Similarly, for the terms microcomputer vs. microcontroller. Students who refer to web-based definitions of microcontrollers may adopt incorrect qualitative descriptions. Many misrepresentations are perpetuated because undated legacy lecture material is available.

Novice students may fail to recognize when downloaded material is so out of date that it is no longer applicable. For example, one has only to search the web for 'JK flip-flop' to discover a wealth of descriptions stating the behaviour but rarely explaining it is no longer the foundation of sequential digital designs. Hence students may not know what to believe.

2.2 EPROM to Flash and others

Erasable Programmable Read Only Memories (EPROMS) require exposure to ultraviolet light to erase all data. But how many program/erase cycles are permitted? How is the retention time affected by temperature? What is the cost per bit? Such questions are not often addressed by teachers who instead may rather consider only capacity or changes in capacity over time.

In any case, EPROMs are used only to support legacy designs and not new developments where flash memories are commonplace. Even for these, the focus may be too easily placed on capacity and data transfer rate, often leaving aside endurance and retention. Perhaps what should be taught are the modelling and statistical methods used to predict the probability of data failure as a function of temperature or, from a practical point of view, the steps needed to write data in blocked pages.

2.3 From PLAs to CPLDs

Programmable Logic Arrays (PLA) and later, in 1978, Programmable Array Logic (PAL) from Monolithic Memories were field programmable and able to replace 'random logic'. They used one-time fusible links. In textbooks and the classroom the fuses are often assumed 'ideal' when in fact there are problems with long-term reliability.

Over time, the fabrication technologies have changed and the complexity has increased from AND/OR arrays to include registered outputs and macro-cells. However, the flash-based technology is often limited to about 100 re-programming cycles and it is too easily assumed similar to 100,000 cycles allowed for some flash memories. Of more practical significance is the ability to perform in-system re-programming.

As the functionality grew the use of programming languages such as ABLE and PALASM gave way to Verilog and VHDL. Indeed without Hardware Description Languages and synthesis tools, Complex Programmable Logic Devices (CPLDs) are 'inaccessible'. But perhaps what is most needed is an understanding of the applications to which CPLDs rather than FPGAs are best suited and vice versa.

2.4 Microcontrollers – RISC or CISC?

Free-standing microcontrollers with flash memory for code storage represent another field-programmable technology but the capabilities and complexity can be misrepresented in educational environments. This may be due to limited funding and inappropriate reuse of dated teaching materials. Regrettably, it may also be to the dated knowledge of teachers in some circumstances. Qualitative descriptions such as 'small' or 'fast' can be the most misleading and give students false impressions and expectations. For example, what is meant by a 'high-speed' microcontroller, as overall performance is influenced by the instruction set architecture and not merely the clock frequency.

The terms Reduced Instruction Set Computer (RISC) and its retrospective counterpart, Complex Instruction Set Computer (CISC), are confusing to students who have no personal historical perspective. RISC should be taught in the context of load/store architectures and not the number of mnemonics. Also RISC is frequently presented as better than CISC but is untrue. CISC ISAs with memory-to-memory operations can solve some problems related to atomicity [1] which is crucial in some real products.

As industry begins to use highly-integrated heterogeneous multi-core microcontrollers in safety-critical applications, educators need to ensure students have a thorough understanding of all their characteristics.

2.5 Field Programmable Gate Arrays - FPGAs

Since the creation of FPGAs by Xilinx in about 1984, a plethora of devices of this genre have been produced by different manufacturers and it is becoming increasingly convenient for educators to teach digital design at a level of abstraction that is agnostic to the detailed behaviour of 'target' devices.

In taught classes, the challenge is usually to create a design but in commercial applications the time and cost of full characterization and verification can be dominant. For example, the viability of a Reed-Solomon Forward Error Correction (FEC) unit in photonic networks may depend critically on worst-case timing delays. Once proven, the 'floorplan' will be locked to avoid unnecessary regression testing. These aspects are needed for a real product but may not be taught.

Also of commercial importance are the consequences of selecting static RAM-based FPGAs vs. distributed flash vs. immunity to single point failures. While it is fully understood that novice students have much to learn, keeping commercial needs in mind will help guide curricula development. Therefore, the integration of hard-core processors vs. soft-core processors is worthy of more classroom discussion. If 'floorplans' are locked to ensure timing closure, the flexibility afforded by embedded soft-core processors to monitor and manage subsystems can be attractive. Changes in functionality are then provided via programs written in high-level languages like 'C' rather than via re-synthesis.

3. LEARNING OBJECTIVES

Across many different taught classes, high-level objectives are to provide students with skills needed to create innovative and effective solutions to problems broadly associated with embedded and cyber physical systems. This encompasses a) electronic peripheral and interface hardware circuits, b) embedded real-time software running on a processor, c) dedicated firmware and d) possibly custom digital elements defined via a hardware description language.

Depending on the application, robust, fault-tolerant, high-integrity or safely-critical techniques may also be needed but these follow understanding of principles of functional decomposition, physical composition, construction and testing.

Full-custom application-specific integrated circuits (ASICs) may be most appropriate for some high-volume or intrinsically secure applications but field programmable gate arrays (FPGAs) and other programmable logic devices, offer unique possibilities for many solutions. However, commercial success increasingly depends upon the integration of different principles that are traditionally taught independently.

3.1 Multiple Paradigms

Our platform, described in detail in Section 5, incorporates an FPGA and CPLD together with associated components and programming environments. They are intended to support classes from years 1 to 3 (or 4) of BSc degree programs, such as Computer Engineering, Electronics or Information Engineering. The concepts provide opportunities for improved holistic understanding of principles that may be developed further to improve commercial opportunities. The expected classes include.

- Introduction to Instruction Set Architecture (ISA) use and design.
- Simplicity, or complexity, of ISA implementation via Hardware Description Languages (HDLs) with FPGA targets. Leading to Advanced ISA design and implementation with focus on low-power.
- Hardware peripheral interfacing and dedicated peripheral implementations in HDLs for FPGAs
- Real-time programming in assembler and 'C'.
- Use of 'Intellectual Property' and integration with proprietary busses and architectures.
- Schedulers, Kernels and Operating systems. Particularly, uC/OS III and ThreadX
- Compiler design and program language translation.

4. WHICH SOFT-CORE PROCESSOR ?

The efficient use of FPGA resources is a commonly taught but as the number of Logic Elements (LEs) has increased, more and more sophisticated designs can be implemented. This is particularly true when considering the instruction set architecture of a processor that is to be implemented in an FPGA. While some entities in the programmer's model like registers can be mapped efficiently to a group of LEs, the logic needed to decode instructions encourages the creation of RISC-like ISAs. However, this is contrary to the complexity of now commonly found in ISAs such as Thumb-2 [2] or PowerISA [3].

Another counter argument relates to the cost in terms of memory size and the energy needed to fetch many RISC-like instructions over fewer CISC-like instructions.

4.1 Some Commercial Offerings

We recognize that FPGA manufactures offer encrypted soft-core processors with different ISAs. These include Nios-II from Altera and MicroBlaze from Xilinx. Each has features that are of potential educational benefit but they are neither simple to describe to novice students nor support migration to commercial microcontrollers.

However, we regard essential characteristics to be a) simplicity of introduction, b) representation of multiple ISA paradigms and c) ease of upward migration. Also, we wish to enable students to use commercial integrated development environments (IDEs) that are representative of commercial practices.

4.2 Selection of DE0 Host and ARM soft-core Cortex-M1 processor

We surveyed available products and selected an off-the-shelf circuit board called the DE0 from Terasic [4] that contains a Cyclone III FPGA made by Altera. There were four main reasons: a) allows ARM soft-core Cortex-M1 [5] to be deployed, b) has numerous on-board input/output devices, c) has sufficient input/output connections to support a parallel interface bus to our custom Teaching Auxiliary Board. We wanted to provide a migration path to other similar microcontrollers from an established 'ecosystem' and judged that ARM Cortex-M1 was optimal.

An immediate possibility was to use the Cortex-M0 processor that is part of a 'DesignStart' kit from ARM. It is provided as a synthesizable, obfuscated Verilog netlist for academics, starts-up and ad-hoc technology teams [6]. Martos et al. implemented this -M0 in a Xilinx FPGA based Nexys2 board from Digilent and validated the functionality of the processor in hardware by observing the values on the data read bus during the memory fetches with ChipScope Pro on-chip logic analyzer tool from Xilinx with an LED toggling at programmed intervals [7]. However, one major limitation of the Cortex-M0 Design-Start processor is that it lacks the hardware debugger interface for seamless compilation of the software applications, downloading, and debugging (e.g. single stepping, inserting breakpoints, watchpoints) of the hardware.

Consequently, we licensed a soft-core Cortex-M1 from ARM with debug capability and a USB to DE0 interface, which enabled a synthesized M1 to be downloaded to the Cyclone III FPGA (with 49% of LEs used) and the application code to be downloaded to associated RAM, but also a virtual JTAG interface to support single-step and trace using IDEs from Keil.

Figure 3 shows the physical appearance of the DE0 and Figure 4 depicts the functional elements.

4.3 Justification for VIP – A 12-Bit processor

The soft-core Cortex-M1 uses 'Thumb' instructions and programs written in 'C' can easily be re-complied for other more capable ARM processors such as the M3 and M4 that use Thumb-2. However, we believe that working at assembly level with the M1 is still too challenging as an entry point for novices. Consequently, we have designed a 12-bit processor called VIP from Various Instruction Paradigms. Its ISA is designed to aid understanding in a) nature of low-level instructions, b) their uses when combined b) how the ISA may be simulated in 'C', d) compiler techniques, and e) how it may be defined in a HDL and synthesized into an FPGA.

Using VIP avoids important issues related to potential copyright infringement.

Graphical simulators for VIP have been written and used since January 2012 in 1st year classes in the School of Computer Engineering at NTU and since October 2013 in 2nd and 3rd year classes in School of Engineering at the University of Warwick, UK. The core of these is written a very efficient form in 'C' and is available for discussion in class to explore the effects of different coding styles.

Some of VIP's features are listed below but a full description is available [8] and [9].

- The ISA is simple but not trivial.
- Unique feature - Memory Mode Flag in status register selects either Von-Neumann mode (default) or Harvard.
- Addressing modes for destination, d, and source,s, are R0, R1, R2, R3, [R0], [R1], [R2+n], [R3+n], AR, SR, SP, PC, #n, [n], [SP+n], [PC+n].
- Register-based and Accumulator based mnemonics
- RISC-like and CISC-like characteristics, including atomic memory-to-memory operations.
- Flags V N Z C behave exactly as in ARM processors.
- Has instructions that exhibit relative simplicity or complexity in implementation via HDLs.

11	10	9	8	7	6	5	4	3	2	1	0
0-7 Dual operand				d				s			
8 Short Move				d				n			
9-A Unary/Control				s/d				n			
B-F JMP				2's complement -128 to +127 relative							

Figure 1. VIP's op-code format

4.4 Opportunity of other cores

Because the DE0 uses an FPGA from Altera, students can readily work with the Nios-II processor. However, while the RTL code of the Cortex-M1 processor is proprietary, our DE0/TAB platform is well-suited to host a variety of open-source soft-core processors. For example, the Tiger MIPS 32 bit processor with a five-stage pipeline and a RISC instruction set from [10] is designed with an Altera Avalon bus interface. The GNU tool chain and an IDE to compile the application and download the executable to the processor via JTAG interface are also available. Furthermore, the Tiger MIPS processor has been deployed as a host processor for automatic compilation of processor/accelerator systems for a well-known open-source high-level synthesis tool [11]. Although, real-time debugging functionality is not yet incorporated, we believe that Tiger MIPS could be nearly a drop-in replacement for the Cortex-M1 thus opening up a new opportunity for students to study internal workings of a soft-core processor and its interaction with a variety of peripherals hosted on our platform both at hardware and software levels in a holistic fashion. One complement to this approach is that we can release the Verilog code which has been implemented in a CPLD to students to enhance learning experiences further.

4.5 Migration to Cortex-M3 and -M4

Some students may design products using commercial field programmable microcontrollers but others may go on to design application-specific processors that are implemented in FPGAs. Towards either goal, migration from Cortex-M1 that uses a von-Neumann memory model to the -M3 or -M4 that use Harvard architecture is beneficial. The M3 also has hardware divide, while the M4 has DSP features and optional floating point. It is anticipated that newer processor will offer compatible features.

5. TAB - TEACHING AUXILIARY BOARD

We recognized that laboratory exercises are often crafted around the facilities of a selected board and it may not be possible to support some experiments of particular educational worth. Consequently, we decided to define practical activities and then ensure our custom board could support them. For example, we wanted to have an LED on each address and data 'line' and single-step bus transfers. Although this feature is of limited use to experienced designers it is very helpful when introducing the idea of addressable memories. The board became known as the Teaching Auxiliary Board, or TAB, and is shown in Figure 2.

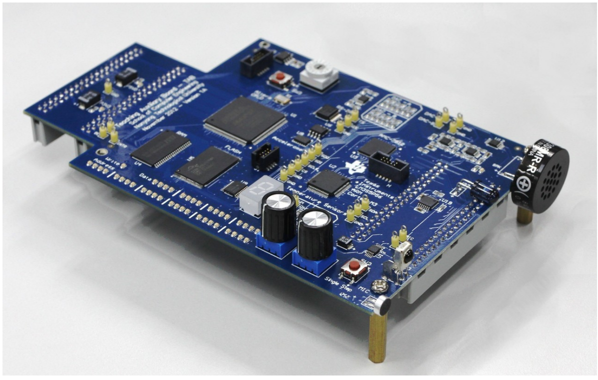

Figure 2. Teaching Auxiliary Board – TAB

5.1 Major Features of TAB

- Plugs into DE0 and both are powered from a USB port.
- Uses Altera MAX II CPLD as address decoder, simulated delayed memory, PWM generator, 38 kHz modulated infra-red transceiver, protocol conversions and various 'glue' logic. All written in Verilog and available to students to use and/or modify.
- LEDs on each address and data line in groups of 4.
- Has hard-core ARM Cortex-M3 mixed-signal micro-controller (type LM3S5P56) to act as an intelligent programmable peripheral, or to provide stand-alone operation. By default, it is programmed to measure the

voltage at two user-adjustable potentiometers which then determines the volume of the loudspeaker and the gain associated with the microphone.

- 3-axis accelerometer, 3-axis compass, 7-segment display, temperature sensor, 16 KByte EEPROM, all accessed via Inter Integrated Circuit (I^2C) serial bus.
- AMC7812 - twelve 12-bit ADCs and twelve 12-bit DACs accessed via Serial Peripheral Interface (SPI).
- Memory - 512 KByte RAM and 1 MByte NOR Flash.
- Parallel access to two 4-bit DACs built using discrete resistors. One hexadecimal switch. RS-232 transceiver, Class-D audio amplifier, loudspeaker, microphone, visible LED and PWM heater. Two analogue temperature sensors. Two digital potentiometers.

5.2 DE0 and TAB Boards in Operation

All the major components of DE0 and TAB are shown in the photograph in Figure 3 which was taken while operating in single-step mode. For the purpose of illustrating the diverse capabilities of our platform and demonstrating multiple paradigms, the soft-core Cortex-M1 processor in the FPGA is running a 'C' program that is behaving as a simulator for the VIP ISA. It is reading 12-bit VIP instructions from RAM on TAB and writing changing data values to both 4-bit DACs on TAB according to the VIP register-based assembly program shown immediately below. Pseudo random levels appear on DACA and a 'ramp' on non-ideal DACB. All peripheral address decoding and hand-shake timing is managed by the CPLD.

```
80000 02C FF0  MOV   R2,#0xFF0
80002 800      MOVS  R0,#0
80003 811      MOVS  R1,#1
80004 060 000  MOV   [R2+0x000],R0
80006 061 001  MOV   [R2+0x001],R1
80008 970      PRSG  R0    ;pseudo random
80009 901      INC   R1    ;ramp
8000A BF9      JMP   -7
```

According to the preferences of the teacher, or the student, VIP's intentionally equivalent accumulator-based syntax may be used to demonstrate another programming paradigm, as shown below.

```
80000 02C FF0  LDX    #0xFF0
80002 800      LDA    #0
80003 811      LDB    #1
80004 060 000  STA    [X+0x000]
80006 061 001  STB    [X+0x001]
80008 970      PRSGA       ;pseudo random
80009 901      INCB        ;ramp
8000A BF9      JMP    -7
```

It is also expected that students will work directly with 'C' programs and/or assembler using native Thumb code for the Cortex-M1.

5.3 CPLD Block Diagram

The CPLD provides functions to support communication with, and/or control of, all the devices and peripherals on TAB. Some are listed below and these should be read in conjunction with the diagrams in Figure 5 and Figure 6.

- 38 kHz infra-red (IR) transmitter and receiver in NEC format as used by TV remote controls. UART over infra-red. Interrupt generation from infra-red received packet. IR signal router and IR LED control.
- Pulse width modulation module (PWM) for resistive heater for closed loop control with temperature sensors.
- SPI format conversion of two 12-bit SPI packets from Cortex-M3 into one 24-bit SPI packet needed by AMC7812.
- Three simulated memory location with configurable timing and access requirements.
- Multiplexing busses within CPLD to bi-directional off-chip busses connected to DE0. Write and read interfaces to registers within CPLD. Memory-mapped registers. Interrupt edge capturing and clearing.
- SRAM address decoding and control signals. Flash address decoding and control signals. Acknowledge generation for SRAM or Flash.
- Meta-stability compensation for 'slow' digital signals.
- Single-stepping mode changer. Single-stepping acknowledge generation.

5.4 CLPD Resource Utilization

The Verilog RTL code is implemented in an Altera Max II CPLD type EPM1270T144C5N which has 1270 logic elements, of which 470 are combinational with no register, 100 are register only, and 442 are mixed; making 1012/1270 or 80% capacity.

5.5 Experiments Supported By TAB

A copious list of experiments is supported by the DE0/TAB combination and many are published [12]. One is summarized below to illustrate well some capabilities of our platform.

TAB has two memory-mapped 4 bit digital-to-analogue converters built with resistor 'ladders'. DACA uses nominal values and has a linear transfer function but DACB uses inappropriate values and has a non-linear response. Using either software breakpoints in a 'C' program or hardware single stepping of a VIP assembler program, the voltage associated with each 4-bit code on each DAC is measured.

In the latter case, pressing the single-step button on TAB for more than 3 seconds causes the mode of interaction between the soft-core processor and the CPLD to toggle. Instead of acknowledging each bus transfer in a sub-microsecond time frame, it is necessary to press this button to complete each bus cycle. In this way students can see and interpret the binary patterns on static address and data lines via individual LEDs arranged in groups of 4. See Figure 3.

The measurements and subsequent analysis may be regarded as exemplars of 'experimental methods' by accreditation agencies. Using a spreadsheet, the best-fit straight can be found and students can determine the worst case deviation. In this way, parameters such as monotonicity, differential and integral non-linearities are explained in a way that is not practical with proprietary DACs.

6. STUDENT REACTIONS

We believe we have created a platform that enables students to explore better multiple paradigms when learning about embedded systems design using field programmable technologies. Specifically, a) FPGAs to host commercial and custom soft-core processors b) FPGAs to provide integrated peripherals and bus interfaces and c) CPLDs to provide peripherals and support functions. Especially when judged with respect to relative performance and complexity.

Since our work is to assist educators, the opinion of students is important. Consequently, we include some verbatim comments from interns at the Centre for High Performance Embedded Systems (CHiPES) who were instrumental in the development of TAB during 2012. Each has now graduated with degrees in Bachelor of Technology (Electronics and Communication Engineering) from Indian Institute of Information Technology, Allahabad, India.

“Working on TAB, helped me get really comfortable with what happens at the intersection of hardware and software. Students can now learn aspects of computer hardware and software in a way that makes them immediately able to make a contribution to any industrial or research project”.

“Everything is available for learning, and that’s what makes it a really special platform for teaching computer engineering.”

“Since it comprises various chips like FPGA, CPLD and memories along with sensors like infrared, temperature and accelerometer, there is a long list of experiments that can be done on TAB. Usually, students don’t get this amount of exposure at the college level. It is {usually} limited to performing some basic activities by simply writing in a few registers and getting LEDs to flash.”

7. FUTURE WORK

There are multiple future opportunities for us as developers and/or teachers and/or students. These include:

- Conduct activities to assess and quantify the benefits of combining VIP, DE0/TAB and FPGA soft-cores and CPLD multi-paradigm approaches.
- Refine Verilog as used in the CPLD and publish.
- Establish a web site to ease distribution of materials.
- Develop synthesizable implementations of VIP in VHDL and Verilog that can be downloaded to the Cyclone III on the DE0 and elsewhere. All models and code will be made available to students.
- Develop a fully-integrated environment where VIP assembler programs are investigated first in software-only environments with graphical interfaces, but later may be downloaded to a synthesized core on DE0 and interact with TAB to explore real-time interfacing and enhanced CPLD functions.
- Express the behaviour of VIP in Language for Instruction Set Architecture (LISA) as used at RWTH Aachen University; with the intention of automating the creation of a ‘C’ compiler. This presents some interesting challenges because of the VIPs ISA.
- Develop student exercises related to extending VIP’s ISA to 16 bits with application specific instructions.

8. ACKNOWLEDGMENTS

The authors express gratitude to Professor T. Srikanthan, the Chair of the School of Computer Engineering at NTU, who in 2011 supported the use of the Cortex-M1 soft-core processor and the development of the DE0/TAB combination for use in taught courses. Our sincere appreciation goes also to Mr. Param Aggarwal and Ms. Ayushi Sinha who, as interns in CHiPES and with our guidance, help to develop the embedded firmware and software. Also thanks are due to Mr. Fateen Mubarak who assisted in the CAD design of the printed circuit board.

9. REFERENCES

[1] IBM “Save your code from meltdown using PowerPC atomic instructions” at http://www.ibm.com/developerworks/library/pa-atom/

[2] ARM – Thumb-2 Reference Card http://infocenter.arm.com/help/topic/com.arm.doc.qrc0001l/QRC0001_UAL.pdf

[3] IBM – Power ISA 2.07 May 3, 2013. See https://www.power.org

[4] Terasic DE0 Board using Altera Cyclone III FPGA. Available http://www.terasic.com.tw/cgi-bin/page/archive.pl?No=364

[5] ARM soft-core Cortex-M1 specifically for implementation in FPGAs. http://infocenter.arm.com/help/topic/com.arm.doc.qrc0001l/QRC0001_UAL.pdf

[6] ARM soft-core Cortex-M0 DesignStart Processor. http://www.arm.com/support/university/ip/index.php

[7] P. I. Martos and F. Baglivo, "Implementing the Cortex-M0 DesignStart Processor in a Low-end FPGA," presented at the Proceedings of the VII Southern Programmable Logic Conference, 2011.

[8] D. C. Dyer, VIP-1T Technical Reference Manual. School of Computer Engineering, Nanyang Technological University, 25/1/2012, Singapore.

[9] D. C. Dyer, “VIP: A novel 12-bit ISA for introducing novice students to instruction set paradigms, their use and implementation”. Proceedings of the 8th Workshop on Embedded and Cyber-Physical Systems Education (WESE). October 3, 2013, Montreal, Canada. (in press)

[10] The Tiger MIPS Processor Available: http://www.cl.cam.ac.uk/teaching/0910//ECAD+Arch/mips.html

[11] A. Canis, J. Choi, M. Aldham, V. Zhang, A. Kammoona, J. H. Anderson, et al., "LegUp: high-level synthesis for FPGA-based processor/accelerator systems," presented at the Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays, Monterey, CA, USA, 2011.

[12] D. C. Dyer and Y. L. Aung, "Microprocessor-based Systems Design Teaching Platform for Undergraduate Students in Computer Engineering" presented at the Proceedings of the 7th Workshop on Embedded and Cyber-Physical Systems Education., Tampere, Finland, 2012.

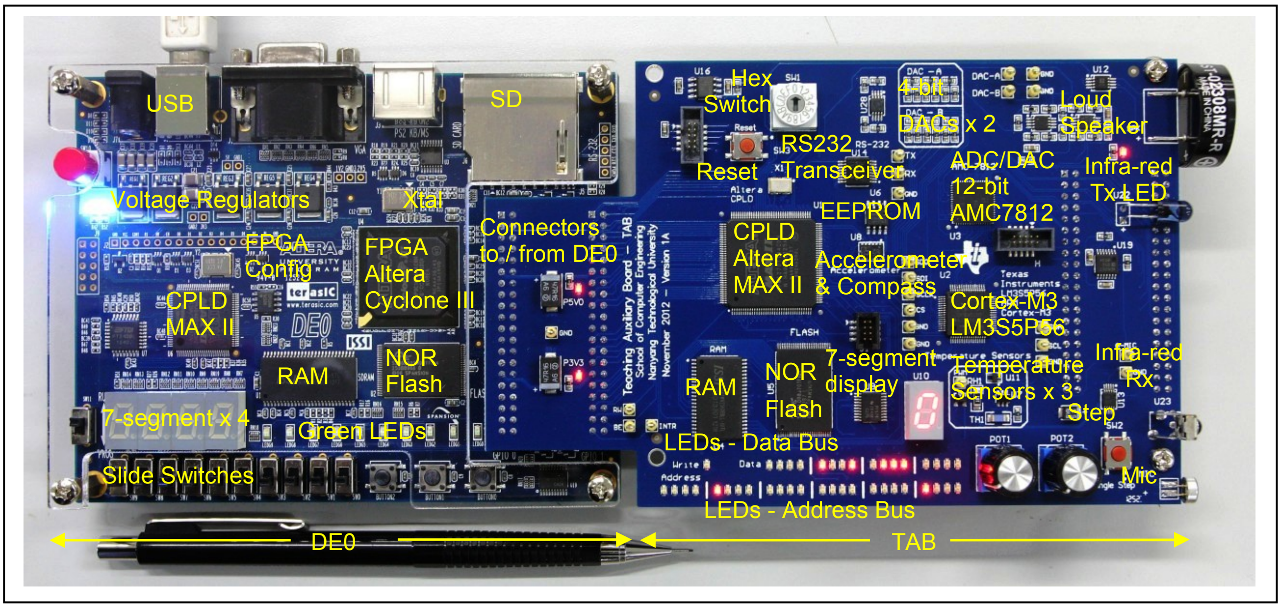

Figure 3. DE0 and TAB in operation with Cortex-M1soft-core in FPGA fetching VIP instruction PRSG R0 (0x970) in single-step mode from address 0x80008

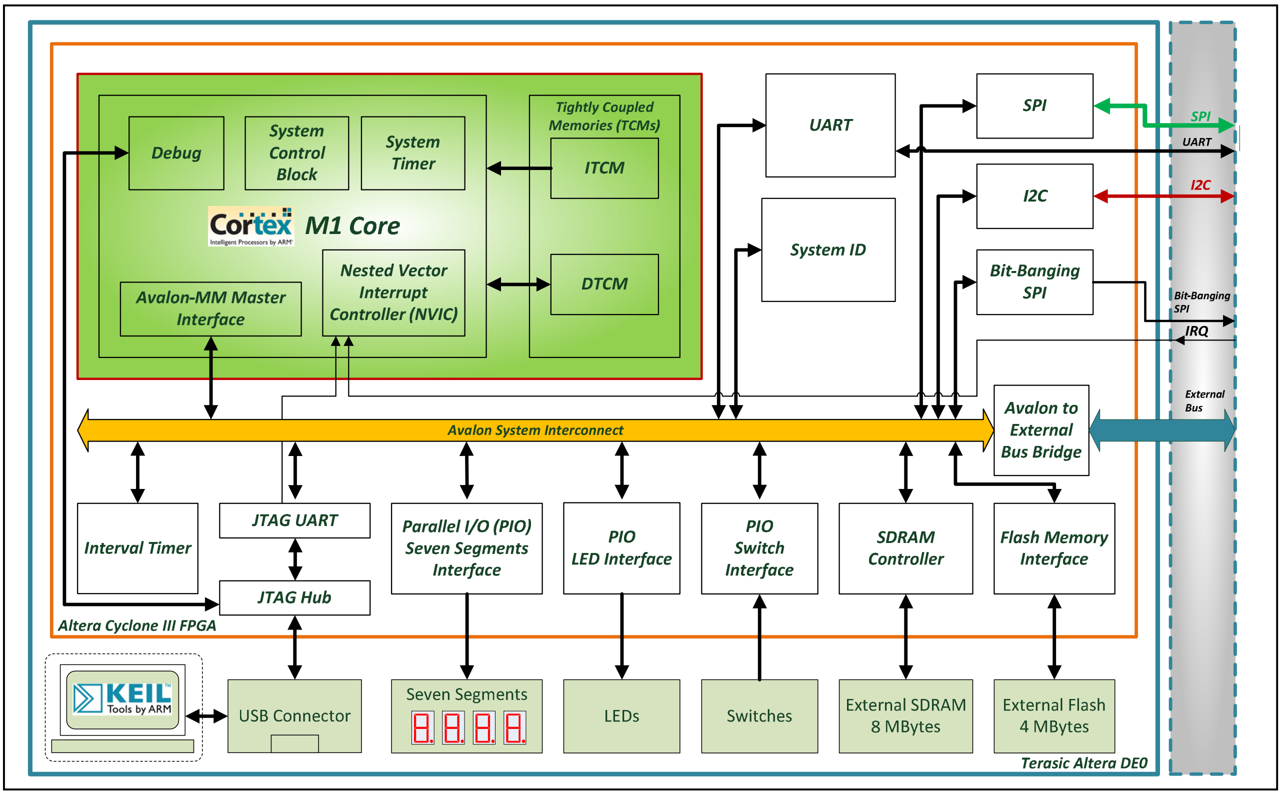

Figure 4. Interconnection of functional components on Terasic's DE0 including those configured within the Cyclone III FPGA

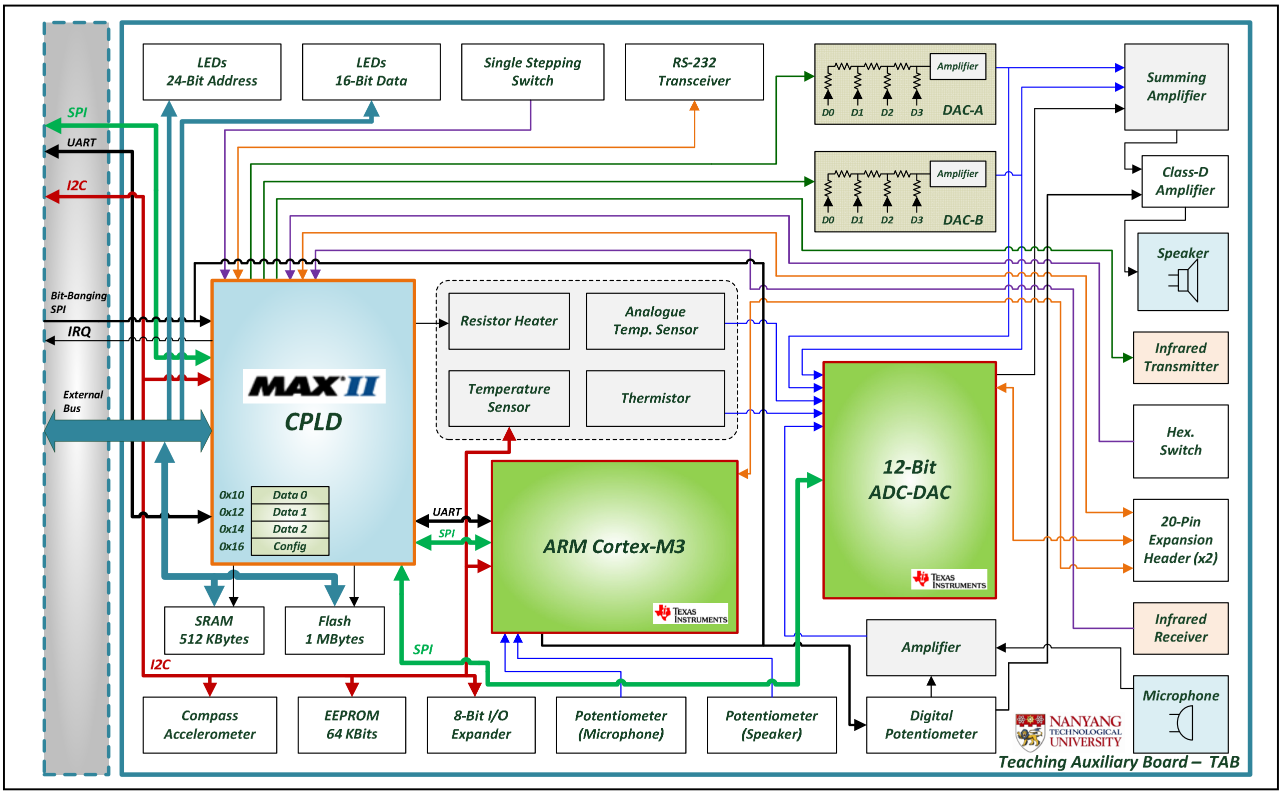

Figure 5. Interconnection of functional components on the Teaching Auxiliary Board (TAB) of our design

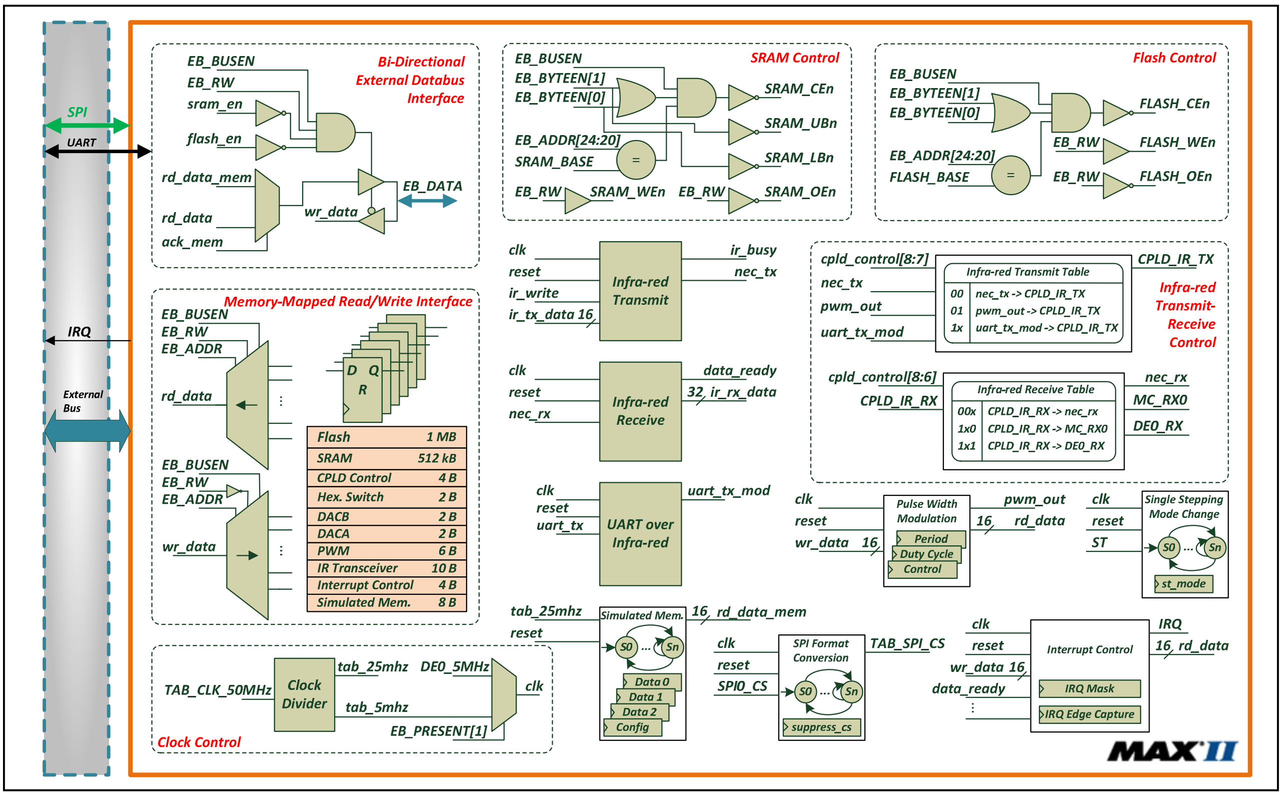

Figure 6. Summary of functional units within the CPLD on TAB