C for Rapid Prototyping of Embedded Systems

NIBEC - The Nanotechnology and Integrated Bio-Engineering Centre is a well-established world-class research complex at the University of Ulster's Jordanstown campus. NIBEC represents a consolidation of research groups, associated with advanced material types used in medical devices, electronics, photonics, nanotechnology, sensors, MEMS, optical and environmental devices. The multi-million pound purpose-built facilities house some of the most sophisticated nano-fabrication, biological characterisation equipment and embedded systems in the world. NIBEC is staffed by an internationally recognised and well-experienced team of researchers and academics working predominantly at the interface of bioengineering and nanotechnology.

Nanotechnology and Integrated Bioengineering Centre

Contact Information
Ann Blair
Nanotechnology and Integrated Bio-Engineering Centre (NIBEC)
Faculty of Computing and Engineering
University of Ulster (Jordanstown Campus), Shore Road
Newtownabbey Co. Antrim BT37 0QB, Northern Ireland

Email: a.blair@ulster.ac.uk

Date: 30 June-1 July 2011

Nanotechnology and Integrated Bio-Engineering Centre (NIBEC)
University of Ulster (Jordanstown Campus)
Overview

The course will help understanding the principles and the role of embedded systems in real world applications. The course will provide familiarity and experience with a range of architectural and programming techniques for embedded engineering systems and their evaluation. The attendees will understand the process of efficient implementation and optimisation of algorithms on embedded platforms such as Field Programmable Gate Arrays (FPGAs). Unlike other C to FPGA tools which rely on going via several intermediate stages, Mentor Graphics C Synthesis tools using Handel-C allows hardware to be directly targeted from software, allowing a more efficient implementation to be created. The language is designed around a simple timing model that makes it very accessible to system architects and software engineers. The first part of the course will focus on rapid prototyping and writing code using C.

The course will provide a detailed knowledge of real-time computing for embedded computer systems and will illustrate and develop an understanding of the various engineering, scientific and economic tradeoffs necessary in the design of embedded systems using C-Synthesis tools for FPGAs and optimised C for microcontrollers. A set programming languages and CAD tools such as Handel-C, C, ISE, DK, Integrated Development Environments (IDE) will be used to deliver this course and demonstrate the examples.

Who Should Attend?

- FPGA, DSP, Microcontrollers System Development Engineers
- Researchers working in the areas of embedded systems design
- Post-graduate Students working on Software and Hardware type Projects
- Project managers who wish to enhance their technical understanding of FPGAs, Rapid Prototyping using C, Implementation and Design Methodologies

Attendees should have some previous experience of programming (Assembler, Java, C#, Visual Basic etc.). Although previous experience of C would be useful, it is not essential as the course will include an overview of the C programming language.

Course Details

Dates: 30 June – 1 July
Times: 9 am – 4 pm
Venue: Nanotechnology and Integrated Bio-Engineering Centre (NIBEC), Faculty of Computing and Engineering, University of Ulster (Jordonstown Campus), Shore Road, Newtownabbey Co. Antrim BT37 0QB, Northern Ireland
Meals: Refreshment and lunch will be provided
Accommodations: Participants should arrange their accommodation and there are numerous exceptional locations in Jordanstown area

Day 1 C Programming Language

This session will provide an overview of the fundamentals of the C programming language, with the emphasis on writing code for embedded systems. Topics covered will include:

1. C Program Structure
2. Types and Operators
3. Control Flow
4. Functions
5. Pointers, Arrays, and Strings
6. Bit Manipulation
7. The C Pre-processor
8. Program Design and Maintenance

Day 1-2 C Programming for FPGAs

1. Introduction to Embedded Systems using FPGAs
2. Reconfigurable Computing and FPGAs Structures
3. Hardware Compilation
   - Mentor Graphics C Synthesis tools: The Handel-C approach
   - Interfacing HDL, Core Generator with Handel-C
   - Design Flows: DK, ISE: From Algorithm to Hardware Implementation
4. Introduction to Handel-C
   - Variables and Parallelism: How to achieve high performance?
   - Arrays and Memories
   - Some Advanced Features: Signals, Channel, Interfaces
5. Rapid Prototyping using Handel-C: Workshop and Examples

Day 2-3 C Programming and Use of FPGAs

1. Hardware Design
   - Introduction to Handel-C
   - Variables and Parallelism: How to achieve high performance?
   - Arrays and Memories
   - Some Advanced Features: Signals, Channel, Interfaces
2. Software Design using Handel-C
   - Design Flows: DK, ISE: From Algorithm to Hardware Implementation
3. Rapid Prototyping using Handel-C: Workshop and Examples

Day 3-4 C Programming and Use of FPGAs

1. Hardware Design
   - Introduction to Handel-C
   - Variables and Parallelism: How to achieve high performance?
   - Arrays and Memories
   - Some Advanced Features: Signals, Channel, Interfaces
2. Software Design using Handel-C
   - Design Flows: DK, ISE: From Algorithm to Hardware Implementation
3. Rapid Prototyping using Handel-C: Workshop and Examples

Day 4 C Programming and Use of FPGAs

1. Hardware Design
   - Introduction to Handel-C
   - Variables and Parallelism: How to achieve high performance?
   - Arrays and Memories
   - Some Advanced Features: Signals, Channel, Interfaces
2. Software Design using Handel-C
   - Design Flows: DK, ISE: From Algorithm to Hardware Implementation
3. Rapid Prototyping using Handel-C: Workshop and Examples

Day 5 C Programming and Use of FPGAs

1. Hardware Design
   - Introduction to Handel-C
   - Variables and Parallelism: How to achieve high performance?
   - Arrays and Memories
   - Some Advanced Features: Signals, Channel, Interfaces
2. Software Design using Handel-C
   - Design Flows: DK, ISE: From Algorithm to Hardware Implementation
3. Rapid Prototyping using Handel-C: Workshop and Examples

Who should attend?

- FPGA, DSP, Microcontrollers System Development Engineers
- Researchers working in the areas of embedded systems design
- Post-graduate Students working on Software and Hardware type Projects
- Project managers who wish to enhance their technical understanding of FPGAs, Rapid Prototyping using C, Implementation and Design Methodologies
- Engineers and technicians involved in production and project management

Course Contents

Day 1 C Programming Language

This session will provide an overview of the fundamentals of the C programming language, with the emphasis on writing code for embedded systems. Topics covered will include:

1. C Program Structure
2. Types and Operators
3. Control Flow
4. Functions
5. Pointers, Arrays, and Strings
6. Bit Manipulation
7. The C Pre-processor
8. Program Design and Maintenance

Day 1-2 C Programming for FPGAs

1. Introduction to Embedded Systems using FPGAs
2. Reconfigurable Computing and FPGAs Structures
3. Hardware Compilation
   - Mentor Graphics C Synthesis tools: The Handel-C approach
   - Interfacing HDL, Core Generator with Handel-C
   - Design Flows: DK, ISE: From Algorithm to Hardware Implementation
4. Introduction to Handel-C
   - Variables and Parallelism: How to achieve high performance?
   - Arrays and Memories
   - Some Advanced Features: Signals, Channel, Interfaces
5. Rapid Prototyping using Handel-C: Workshop and Examples

Day 3-4 C Programming and Use of FPGAs

1. Hardware Design
   - Introduction to Handel-C
   - Variables and Parallelism: How to achieve high performance?
   - Arrays and Memories
   - Some Advanced Features: Signals, Channel, Interfaces
2. Software Design using Handel-C
   - Design Flows: DK, ISE: From Algorithm to Hardware Implementation
3. Rapid Prototyping using Handel-C: Workshop and Examples

Day 5 C Programming and Use of FPGAs

1. Hardware Design
   - Introduction to Handel-C
   - Variables and Parallelism: How to achieve high performance?
   - Arrays and Memories
   - Some Advanced Features: Signals, Channel, Interfaces
2. Software Design using Handel-C
   - Design Flows: DK, ISE: From Algorithm to Hardware Implementation
3. Rapid Prototyping using Handel-C: Workshop and Examples

Overview

Day 1 C Programming Language

This session will provide an overview of the fundamentals of the C programming language, with the emphasis on writing code for embedded systems. Topics covered will include:

1. C Program Structure
2. Types and Operators
3. Control Flow
4. Functions
5. Pointers, Arrays, and Strings
6. Bit Manipulation
7. The C Pre-processor
8. Program Design and Maintenance

Day 1-2 C Programming for FPGAs

1. Introduction to Embedded Systems using FPGAs
2. Reconfigurable Computing and FPGAs Structures
3. Hardware Compilation
   - Mentor Graphics C Synthesis tools: The Handel-C approach
   - Interfacing HDL, Core Generator with Handel-C
   - Design Flows: DK, ISE: From Algorithm to Hardware Implementation
4. Introduction to Handel-C
   - Variables and Parallelism: How to achieve high performance?
   - Arrays and Memories
   - Some Advanced Features: Signals, Channel, Interfaces
5. Rapid Prototyping using Handel-C: Workshop and Examples

Overview

Dr. Abbes Amira is a Reader in embedded systems in the Nanotechnology and Integrated Bio-Engineering Centre (NIBEC) at the University of Ulster, United Kingdom. From May 2006-March 2010, he was a senior lecturer at Brunel University, West London within the division of Electronic and Computer Engineering. Before he joined Brunel University, he has held a lectureship in Computer Science at Queen’s University, Belfast (QUB) since November 2001. He received his Ph.D. in Computer Science from Queen’s University Belfast in 2001. He has been awarded a number of grants from government and industry and has published over 150 publications in the area of reconfigurable computing and image processing during his career to date. He has been invited to give talks and tutorials at universities and international conferences and being chair, program committee for a number of conferences. He was one of the tutors presenters at ICIP 2009, Conference Chair of ECVW 2011, Program Chair of ECVW2010, Program Co-Chair of DELTA 2008 and IMVIP 2005. He is also one of the 2008 VARIAN prize recipients. He is a senior member of the IEEE, senior member of ACM, member of IET and Fellow of the Higher Education Academy. His research interests include: embedded systems, high performance reconfigurable computing, image processing, multi-resolution analysis and medical applications.

Dr Peter Graham has over 19 years’ experience across a range of software engineering and research projects, the last 11 of which have been as Software Development Manager at Heartsine Technologies Ltd, a local firm that designs and manufactures portable defibrillators. A set programming languages and CAD tools such as Handel-C, C, ISE, DK, Integrated Development Environments (IDE) will be used to deliver this course and demonstrate the examples.

Prior to joining Heartsine in 1999, Peter was a senior research programmer with Meridian Medical Technologies where he helped develop an 80 -lead ECG body mapping system. Peter has also collaborated on projects involving the development of Web based ECG recording and retrieval systems via cell phone networks, ambulatory cardiac monitoring systems and cardiac diagnostic algorithm development. Apart from medical related software, Peter has been professionally involved in software projects as diverse as PC based video conferencing systems, expert advisory systems, CAD-CAM systems, stock report display systems for the New York Stock Exchange, central heating balancing systems, as well as evolutionary architectural design systems. Before becoming a software engineer, Peter taught in further and higher education as well as running his own design and manufacturing business. Peter holds a BA in 3-Dimensional Design (1984), an MSc in Computer Science (1990) and a DPhil in Software (1996).