

Introduction to Embedded System Design Using Field Programmable Gate Arrays

Rahul Dubey

Introduction to Embedded System Design Using Field Programmable Gate Arrays

 Springer

Rahul Dubey, PhD
Dhirubhai Ambani Institute of Information
and Communication Technology (DA-IICT)
Gandhinagar 382007
Gujarat
India

ISBN 978-1-84882-015-9

e-ISBN 978-1-84882-016-6

DOI 10.1007/978-1-84882-016-6

A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2008939445

© 2009 Springer-Verlag London Limited

ChipScope™, MicroBlaze™, PicoBlaze™, ISE™, Spartan™ and the Xilinx logo, are trademarks or registered trademarks of Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124-3400, USA. <http://www.xilinx.com>

Cyclone®, Nios®, Quartus® and SignalTap® are registered trademarks of Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, USA. <http://www.altera.com>

Modbus® is a registered trademark of Schneider Electric SA, 43-45, boulevard Franklin-Roosevelt, 92505 Rueil-Malmaison Cedex, France. <http://www.schneider-electric.com>

Fusion® is a registered trademark of Actel Corporation, 2061 Stierlin Ct., Mountain View, CA 94043, USA. <http://www.actel.com>

Excel® is a registered trademark of Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399, USA. <http://www.microsoft.com>

MATLAB® and Simulink® are registered trademarks of The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA. <http://www.mathworks.com>

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Cover design: eStudio Calamar S.L., Girona, Spain

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

In memory of my father and grandmother

Preface

Overview

The realm of embedded systems is quite large and is predominantly carried out around the general purpose processor and microcontrollers. The use of field programmable gate array (FPGA) in microprocessor-based embedded systems is often for glue logic or for off-loading the processor from tasks that require fast updates. The motivation for writing this text is to present a single source of information that can be used to understand how a FPGA and the Hardware Description Language (HDL) can be used in the design of embedded digital systems.

Digital design methodology has undergone several changes over the past three decades. The use of FPGA and HDL for implementing digital logic has become widespread in the last decade. The use of FPGA in embedded systems is still in its nascent stage. The majority of the embedded applications are divided between an 8-bit microcontroller implementation and a 32-bit processor-based real time operating system (RTOS) implementation. This text provides a starting point for the design of embedded system using FPGA and HDL. To give the text a common thread of thought from the application point of view, a design example of a hypothetical industrial robot controller is taken up. Different chapters of the text provide the necessary background on FPGA and HDL along with its use in designing an industrial robot controller.

Coverage

The first FPGA, introduced in 1985, consisted of 2000 gates. Since then, gate density has grown to tens of millions of gates. With increasing density of FPGAs, varied hardware resources have become a standard feature of contemporary FPGA-based devices. The text includes simulation of digital logic using Verilog HDL, synthesis of HDL code for a given FPGA device and processor-based FPGA devices. The focus of the HDL chapter is to emphasise the synthesizable area of Verilog constructs and to provide a basis to understand the application examples that follow in subsequent chapters. A chapter is devoted to the understanding of hardware–software partitioning in a FPGA device. Proprietary 8-bit and a 32-bit soft processors are discussed along with interfacing methodology using system-on-

chip interconnections. Basic technique for serial data communication, signal conditioning, motor control and hardware prototyping is covered using FPGA and HDL.

How to Use This Book

Moore's law has kept the semiconductor business in a constant state of flux. It is very difficult to write a book that uses FPGA and continues to be relevant despite ongoing technological changes. The author has presented basic concepts and techniques for using FPGA and hence should not change quickly. Since this book covers vast areas of HDL and FPGAs, some sections are brief and sketchy. For this the author recommends that the reader supplement the contents of each chapter with additional available literature. The chapter on HDL coding and simulation should be supplemented by standard textbooks on HDL coding and simulation. The FPGA resources and synthesis topic should be supplemented by EDA tools provided by different FPGA vendors and FPGA device datasheets. The contents on FPGA embedded processors can be supplemented by application notes on interfacing processors to custom codes and datasheets of soft processors.

FPGA Device and Tools Used

For purposes of illustration and consistency, Xilinx ISE™ software and SPARTAN™3E FPGA have been used throughout the book. Though the exemplars are specific to this device, the concepts can be applied to FPGA devices available from other FPGA vendors.

Gandhinagar
October 2008

Rahul Dubey

Acknowledgements

Many people have contributed in the process of writing this text. First and foremost, I wish to thank my PhD supervisors Professor Pramod Agarwal and Professor M.K. Vasantha at IIT Roorkee, for the training they provided during my research. Lots of impetus for being steadfast in my resolve to finish this book came from Professor S.C. Sahasrabudhe, Director DA-IICT. I wish to thank the students and faculty at DA-IICT for their help and support.

The synthesis reports attached with design examples were generated using Xilinx ISE™ software. I would like to thank Xilinx for letting me use their software tool and FPGA to demonstrate various aspects of HDL and FPGA design. I am also thankful to Doctor Parimal Patel, Xilinx, for providing valuable feedback on the text. Certain equipment, used for hardware testing of examples in the text, came through a research grant from the Department of Science and Technology of the Indian Government.

This text would not have seen the light of day without the patience and support of my family. No words can express the thoughtfulness of my wife Sulekha and daughters Aditi and Avni for enduring the extra work hours. Toward the end of the writing process, Sulekha helped out by proofreading large sections of the text. I would also wish to thank my mother and younger brother Rohit for their encouragement during the process of writing this book. My uncle Mr. Rama Shankar Dubey has been a source of strength for all these years.

I thank Mr. Oliver Jackson and Springer for giving me the opportunity to write this book. Also, I wish to thank Sorina Moosdorf and the team at le-tex for meticulously checking the final draft of the manuscript.

Contents

Abbreviations	xv
1 Introduction	1
1.1 Embedded System Overview.....	1
1.2 Hypothetical Robot Control System.....	2
1.3 Digital Design Platforms.....	4
1.3.1 Microprocessor-based Design.....	5
1.3.2 Single-chip Computer/Microcontroller-based Design.....	7
1.3.3 Application Specific Standard Products (ASSPs).....	8
1.3.4 Design Using FPGA.....	10
1.4 Organization of the Book.....	12
Problems.....	14
References.....	15
Further Reading.....	16
2 Hardware Description Language: Verilog	17
2.1 Software and Hardware Description Languages.....	17
2.2 Let's Use Verilog as Our HDL!.....	19
2.3 Design Examples Using Verilog.....	19
2.3.1 Gate Level Model.....	20
2.3.2 Combinational Circuits Using Data Flow Modelling.....	21
2.3.3 Behavioural Logic.....	24
2.3.4 Finite State Machine (FSM).....	27
2.3.5 Arithmetic Using HDL.....	35
2.4 Pipelining.....	40
2.5 Module Instantiation and Port Mapping.....	40
2.6 Use of Pre-designed HDL Codes.....	45
2.7 Simulating Digital Logic Using Verilog.....	47
2.7.1 EDA Tool Flow for Simulation.....	47
2.7.2 Creating a Test Bench for HDL-based Digital Logic.....	49
2.7.3 Post Place and Route Simulation.....	49
2.7.4 Simulation of Algorithm Using Pre-designed Codes.....	51

Problems	51
Further Reading	51
3 FPGA Devices.....	53
3.1 FPGA and CPLD	53
3.2 Architecture of a FPGA	54
3.2.1 FPGA Interconnect Technology	54
3.2.2 Logic Cell	56
3.2.3 FPGA Memory	61
3.2.4 Clock Distribution and Scaling	67
3.2.5 I/O Standards	70
3.2.6 Multipliers	71
3.3 Floor Plan and Routing	72
3.4 Timing Model for a FPGA	74
3.5 FPGA Power Usage	75
Problems	79
Further Reading.....	80
4 FPGA-based Embedded Processor.....	81
4.1 Hardware–Software Task Partitioning	81
4.2 FPGA Fabric Immersed Processors	82
4.2.1 Soft Processors	82
4.2.2 Hard Processors	84
4.2.3 Tool Flow for Hardware–Software Co-design	84
4.3 Interfacing Memory to the Processor	85
4.4 Interfacing Processor with Peripherals	86
4.4.1 Types of On-chip Interfaces	88
4.4.2 Wishbone Interface	89
4.4.3 Avalon Switch Matrix	90
4.4.4 OPB Bus Interface	90
4.5 Design Re-use Using On-chip Bus Interface	92
4.6 Creating a Customized Microcontroller	94
4.7 Robot Axis Position Control	98
Problems	100
References	101
Further Reading.....	101
5 FPGA-based Signal Interfacing and Conditioning	103
5.1 Serial Data Communication	103
5.2 Physical Layer for Serial Communication	106
5.2.1 RS-232-based Point-to-Point Communication	106
5.2.2 RS-485-based Multi-point Communication	106
5.3 Serial Peripheral Interface (SPI)	109
5.4 Signal Conditioning with FPGAs	111
Problems	113
References	114

6 Motor Control Using FPGA	115
6.1 Introduction to Motor Drives	115
6.2 Digital Block Diagram for Robot Axis Control.....	115
6.2.1 Position Loop.....	116
6.2.2 Speed Loop.....	117
6.2.3 Power Module	118
6.3 Case Studies for Motor Control	119
6.3.1 Stepper Motor Controller.....	119
6.3.2 Permanent Magnet DC Motor	122
6.3.3 Brushless DC Motor	125
6.3.4 Permanent Magnet Rotor (PMR) Synchronous Motor	126
6.3.5 Permanent Magnet Synchronous Motor (PMSM).....	131
Problems	135
Further Reading.....	136
7 Prototyping Using FPGA	139
7.1 Prototyping Using FPGAs	139
7.2 Test Environment for the Robot Controller	142
7.3 FPGA Design Test Methodology.....	143
7.3.1 UART for Software Testing	143
7.3.2 FPGA Hardware Testing Methodology	144
Problems	151
References.....	152
Index	153

Abbreviations

ABEL	Advanced Boolean expression language
ADC	Analogue-to-digital converter
ANSI	American National Standards Institute
ASIC	Application specific integrated circuit
ASSP	Application specific standard product
BUFG	Global clock buffer
CAD	Computer aided design
CAN	Controller area network
CE	Clock enable
CLB	Configurable logic block
CLK	Clock signal
CMOS	Complementary metal oxide Semiconductor
CPLD	Complex programmable logic device
DAC	Digital-to- analogue converter
DCI	Digitally controlled impedance
DCM	Digital clock manager
DRAM	Dynamic random access memory
DSP	Digital signal processor
EDA	Electronic design automation
EDIF	Electronic digital interchange format
EMI	Electromagnetic interference
EPROM	Erasable programmable read only memory
FF	Flip flop
FIFO	First in first out
FIR	Finite impulse response (filter)
fMax	Frequency maximum

FPGA	Field programmable gate array
FSM	Finite state machine
GPP	General purpose processor
GPS	Global Positioning System
GPIO	General purpose I/O
GTL	Gunning transceiver logic
GTL ⁺	Gunning transceiver logic plus
GUI	Graphical user interface
HDL	Hardware description language
HEX	Hexadecimal
HSTL	High-speed transceiver logic
I/O	Inputs and outputs
ISR	Interrupt service routine
IEEE	Institute of Electrical and Electronics engineers
ILA	Integrated logic analyzer
IOB	Input output block
IP	Intellectual property
ISA	Instruction set architecture
ISP	In system programming
JEDEC	Joint Electron Device Engineering Council
JTAG	Joint Test Advisory Group
LAN	Local area network
LC	Logic cell
LCD	Liquid crystal display
LSB	Least significant bit
LUT	Look-up table
LVC ⁺ MOS	Low-voltage complementary metal oxide semiconductor
LVDS	Low-voltage differential signaling
LVPECL	Low-voltage positive emitter-coupled logic
LVTTL	Low Voltage transistor to transistor logic
MAC	Multiply and accumulate
MOSFET	Metal oxide semiconductor field effect transistors
MSB	Most significant bit
MUX	Multiplexer
NAND	Not and
NRE	Non-recurring engineering (cost)
NRZ	Non-return to zero

OE	Output enable
OTP	One time programmable
PACE	Pinout and area constraints editor
PCB	Printed circuit board
PCI	Peripheral component interconnect
PCMCIA	Personal Computer Memory Card International Association
PLC	Programmable logic controller
PI	Proportional integral
PLD	Programmable logic device
PWM	Pulse width modulation
RAM	Random access memory
RMS	Root mean square
ROM	Read only memory
SCR	Silicon controlled rectifier
SDF	Standard delay format
SOP	Sum of product
SPI	Serial peripheral interface
SRAM	Static random access memory
SRL16	Shift register LUT
SSTL	Stub series terminated transceiver logic
TTL	Transistor-transistor logic
Tpd	Time of propagation delay (through the device)
UART	Universal asynchronous receiver transmitter
UCF	User constraints file
VHDL	VHSIC high level description language
VHSIC	Very high speed integrated circuit
VREF	Voltage reference
XOR	Exclusive OR
XST	Xilinx synthesis technology

Introduction

Digital systems and their design have evolved greatly over the last four decades. Rising densities and speed have provided designers a huge canvas to create complex digital systems. Present-day embedded systems use single-chip microcontrollers. Contemporary microcontrollers are available with 8-, 16- and 32-bit processing capability along with a peripheral set containing ADC, timer/counter and networks (I²C, CAN, SPI, and UART). For most applications the microcontroller-based board is adequate. For applications where there is a need to integrate custom logic for faster control and additional peripherals, the microcontroller or microprocessor board is augmented by a FPGA or an application specific standard product (ASSP) device. The focus of this chapter is to understand different digital design methodologies before embarking on a full fledged description of the use of a custom digital design based on a FPGA.

1.1 Embedded System Overview

Embedded systems are usually single function applications. Various functional constraints associated with embedded systems are low cost, single-to-fewer components, low power, provide real-time response and support of hardware-software co-existence. A general methodology used in designing an embedded system is shown in Table 1.1.

The decision on the kind of digital platform to be used takes place during the system architecture phase as each embedded application is linked with its unique operational constraints. Some of the constraints of a digital controller of embedded system hardware include (in no particular order) the following:

- Real-time update rate
- Power
- Cost
- Single chip solution
- Ease of programming
- Portability of code

- Libraries of re-usable code
- Programming tools.

Table 1.1. Embedded system design flow [1]

Design phase	Design phase details
Requirements	Functional requirements and non-functional requirements (size, weight, power consumption and cost)
User specifications	User interface details along with operations needed to satisfy user request
Architecture	Hardware components (processor, peripherals, programmable logic and ASSPs), software components (major programs and their operations)
Component design	Pre-designed components, modified components and new components
System integration (hardware and software)	Verification scheme to uncover bugs quickly

1.2 Hypothetical Robot Control System

For understanding different digital design platforms, this text uses the design of a digital controller for a robot as a case study. The robot is a hypothetical, vertically articulated robot system for an automated assembly line. The process of designing this controller will help in understanding various digital design concepts. Figure 1.1 shows the various components of an assembly line robot. Each robot consists of five electric motors that work as actuators for different joints of the robot. A programming pendant or workstation is used to program the movements of the robot along with a communications network to link this robot to other robots on the assembly line. Various sensors are interfaced to the robot control system.

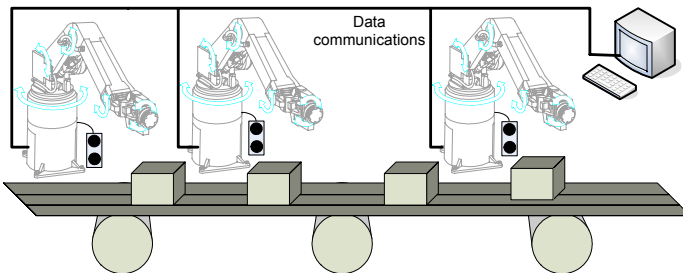


Fig. 1.1. Vertically articulated robot system used in an assembly line environment

The typical requirements of an Industrial robot controller include

- Control method for point-to-point control using servomotors
- Position detection using incremental or absolute encoder system
- Return to origin using limit switches and encoder
- Trajectory control
- Programming using a personal computer.

Table 1.2. Tasks for robot digital controller

Task	Subtask	Update time
Control of joint motors	Gate Driver, protection and current sensing	Fraction of a microsecond
	Dead time	Microseconds
	Closed-loop torque control	Tens of microseconds
	Closed-loop speed control	Hundreds of microseconds
	Position coordinate interpolation	Milliseconds
	Host communications	Tens of milliseconds
Sensor signal processing	ADC, DAC	Tens of milliseconds
Networking applications	Low-speed network	Milliseconds

Control Strategy for the Robot Controller

For implementing the robot controller on a digital system, a list of controller tasks is created in Table 1.2 along with the update time requirements. The major tasks for the robot controller for an articulated factory robot are

- Simultaneous control of five motors with details shown in Table 1.3.
- Signal processing of sensor inputs coming from robot environment — encoders, limit switches, proximity sensors, vision sensor
- Communication of robot co-ordinates to other robots in the vicinity, using CAN bus or Modbus[®]
- Communicating with host controller over serial port
- Computation of trajectory for robot movement.

Table 1.3. Specifications of a micro articulated robot Mitsubishi Movemaster RV-M1 [2]

Axis	Description	Encoder pulses per revolution (PPR) ¹	Gear ratio	Working range in degrees
J1	Waist	200	1:100	300°
J2	Shoulder	200	1:170	130°
J3	Elbow	200	1:110	110°
J4	Wrist pitch	96	1:180	90°
J5	Wrist roll	96	1:110	± 180°

The tasks and their update times are shown graphically in Fig. 1.2.

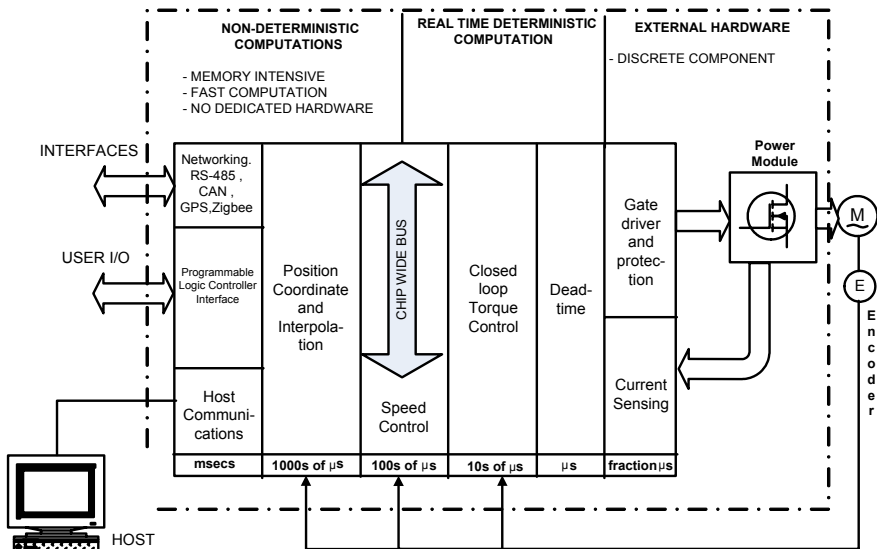


Fig. 1.2. Update times needed for various control functions of a robot control system [3]

1.3 Digital Design Platforms

Till the 1970s, electronic system designs were based on discrete analogue components such as transistors, operational amplifiers, resistors, capacitors and inductors. These circuits offered concurrent processing but had problems of parameter drift with temperature and ageing. The coming of TTL-based

¹ The encoder is used to find the position and speed of the robot joint. The working of the encoder is explained in Chap. 2.