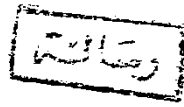
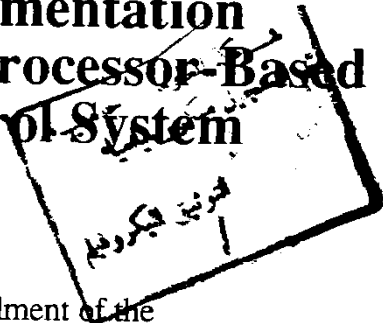


Ain Shams University
Faculty of Engineering
Computer and Systems Engineering Department

Design and Implementation of a Real-Time Multiprocessor-Based Distributed Control System

A Thesis
Submitted in Partial Fulfilment of the
Requirements of the Degree of
Master of Science in Electrical Engineering
(Computer and Systems Engineering)



621.38195
T. F

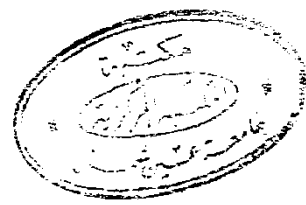
By
Tarek Farouk Ahmed Abdelzaher

B.Sc., Electrical Engineering
(Computer and Systems Engineering)
Ain Shams University, 1990

Supervised by

49176

M.A. Sheirah
Professor of Computer Control Systems
Computer and Systems Engineering Department
Faculty of Engineering
Ain Shams University



Cairo, 1993

Examiners Committee

The thesis on : **Design and Implementation of a Real-Time Multiprocessor-Based Distributed Control System**

by : **Tarek Farouk Ahmed AbdelZaher**

Approved by

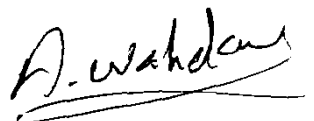
Name

Signature

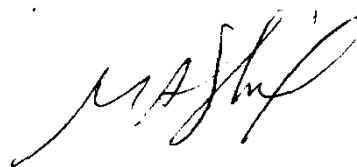
1- **Prof. Abdel Moniem Y. Bilal**
Cairo University, Cairo.



2- **Prof. A. M. Wahdan**
Ain Shams University, Cairo.



3 - **Prof. M. A. Sheirah**
Ain Shams University, Cairo.



Date : 27 / 12 / 1993



Statement

This dissertation is submitted to Ain Shams University for the degree of Master of Science in Electrical Engineering (Computer and Systems Engineering).

The work included in this thesis was carried out by the author at the Computer and Systems Engineering Department, Ain Shams University.

No part of this thesis has been submitted for a degree or qualification at other university or institution.

Date : 10 / 12 / 1993

Signature : *Tarek F.A. AbdelZaher*

Name : Tarek Farouk Ahmed AbdelZaher

To my parents

Acknowledgements

I am most grateful to Professor M.A. Sheirah, my supervisor, for his great support, constant follow up, beneficial comments and strong encouragement.

I would also like to thank all those who helped me getting this work done by their useful advice, constructive criticism and simple words of support.

In particular, I would like to express my deepest gratitude to my parents whose patience and understanding paved the way to the completion of this thesis.

Abstract

A comprehensive survey is carried out on the different existing methods of implementing distributed control systems and the underlying theory is studied

A microcomputer based control system is designed incorporating the following features:

- 1- The system is highly reconfigurable. The particular control algorithm of each individual controller can be changed by connecting the controller to a portable PC, downloading the new algorithm, then disconnecting the PC.
- 2- The control system can be easily connected, if so desired, to a larger system or host computer that may be used for supervisory control or DDC purposes.
- 3- The controllers can be tuned locally or remotely, by a human operator or a self-tuning algorithm.
- 4- The controllers are designed in such a way as to facilitate system maintenance.
- 5- The hardware involved is available at the local market at economic prices.
- 6- The system performance does not degrade abruptly should any individual connection be broken.

A case study is advanced in which an industrial process (complex enough to warrant the use of a distributed control system) is controlled by the above described scheme.

Table of Contents

Chapter 1 : Introduction	1
PART ONE : REAL-TIME DISTRIBUTED COMPUTING	5
General Introduction	5
Chapter 2 : Introduction to Real-Time Systems	6
Introduction.....	6
2.1 Real-Time Computing	6
2.1.1 Time Constraints.....	7
2.1.2 Major Real-Time Requirements	9
2.1.2.1 The Predictability Requirement	10
2.1.2.2 The Adaptability Requirement.....	11
2.1.2.3 The Reliability Requirement.....	11
2.2 Real-Time Software Design.....	12
2.2.1 Polled Loop Systems.....	13
2.2.2 Clocked Polled Loop Systems.....	14
2.2.3 Interrupt Driven Systems.....	15
2.2.4 Foreground/Background Systems.....	18
2.3 A General Classification of Real-Time Tasks	19
2.3.1 Classification by Result Accuracy Function.....	19
2.3.2 Classification by Strictness of Deadline	20
2.4 A Case Study	22
2.4.1 The Boeing 777 Information Management System	22
2.4.2 Conclusions	26
Chapter 3 : Introduction to Distributed Control Systems	27
Introduction.....	27
3.1 An Overview of DCS Design.....	28
3.1.1 The Loop Level.....	29
3.1.2 Operator Stations	30
3.1.3 The Host	31
3.2 Application Engineering of DCS Systems.....	32
Chapter 4 : Hard Real-Time Scheduling.....	34
Introduction.....	34
4.1 Deterministic Scheduling.....	37
4.2 Worst Case Response Time Analysis.....	42
4.3 Dynamic Scheduling.....	52

4.4 Conclusions.....	58
Chapter 5 : Real-Time Programming Languages	61
Introduction.....	61
5.1 The Ada Language.....	62
5.1.1 Ada Programming Features.....	63
5.1.2 Ada Task Communication and Synchronization	65
5.1.3 Task Scheduling in Ada	67
5.1.4 Ada Disadvantages for Real-Time Programming.....	67
5.2 Real-Time Euclid	68
5.2.1 Schedulability Motivated Real-Time Euclid Features.....	69
5.2.2 Tasks in Real-Time Euclid.....	70
5.2.2.1 Task Declaration	70
5.2.2.2 Task Synchronization.....	71
5.2.2.3 Task Termination and Exception Handling.....	72
5.3 Synchronous Languages	73
5.3.1 Multiple-Clocked Recurrent Systems.....	76
5.3.2 State-Based Formalisms.....	77
5.4 Summary.....	79
PART TWO : A SMALL SCALE REAL-TIME SYSTEM.....	82
General Introduction	82
Chapter 6 : Hardware Design and Implementation	83
Introduction.....	83
6.1 High Level System Description	84
6.1.1 The Control System	84
6.1.2 The Computation Processor Card.....	87
6.1.2.1 The 8088 Microprocessor.....	89
6.1.2.2 The Card Processor Bus Interface.....	90
6.1.2.3 The Host Bus Interface.....	91
6.1.2.4 Bus Arbitration and Mode Control Logic	91
6.1.2.5 Memory Subsystem.....	92
6.1.2.6 I/O Control Circuit.....	93
6.1.2.7 The Reconfigurable I/O's.....	93
6.2 Low Level System Description.....	95
6.2.1 The 8088 Processor.....	95
6.2.2 The Basic Control Circuit	96
6.2.2.1 The Bus Arbitration and Mode Control Block	97
6.2.2.2 The Card Processor Bus Interface.....	102
6.2.2.3 The Host Processor Bus Interface.....	103
6.2.3 The Memory Subsystem.....	105

6.2.4 The I/O Subsystem.....	107
6.2.4.1 The I/O Control Circuit.....	108
6.2.4.2 The Reconfigurable I/O's.....	110
Chapter 7 : The Operating System Kernel.....	118
Introduction.....	118
7.1 The Proposed Kernel.....	119
7.1.1 The Tasking Model.....	121
7.1.2 Kernel Architecture.....	122
7.2 The Scheduling Mechanism.....	123
7.2.1 The Scheduler from an Operating System Viewpoint.....	124
7.2.2 The Scheduler Software Anatomy.....	129
7.2.2.1 The Real-Time Clock.....	131
7.2.2.2 The Self-Block Module.....	131
7.2.2.3 The Dispatcher.....	132
7.2.2.4 Invokation Modules.....	132
7.2.2.5 The Scheduler Interface.....	135
7.2.2.6 The Policy Module.....	135
7.2.3 Scheduler Implementation Issues.....	138
7.2.3.1 Scheduling Algorithm Replaceability.....	138
7.2.3.2 Scheduler Reentrancy.....	139
7.2.3.3 Context Switching.....	140
7.2.3.4 The Auxiliary Interrupts.....	143
7.3 Intertask Communication.....	144
7.3.1 <i>Communication Ports</i>	147
7.3.2 The Link Primitive.....	149
7.3.3 <i>Communication Primitives</i>	150
7.3.3.1 The SEND Primitive.....	151
7.3.3.2 The SendToTarget Primitive.....	152
7.3.3.3 The FastSend Primitive.....	156
7.3.3.4 The RECEIVE Primitive.....	157
7.3.3.5 The FastReceive Primitive.....	159
7.3.4 The Complete Picture.....	160
7.4 The Interface Processor System Software.....	162
7.4.1 Load Balancing.....	162
7.4.1.1 The Task Fetch Routine.....	162
7.4.1.2 The Download Routine.....	164
7.4.2 Interprocessor Communication.....	164
Chapter 8 : The Scheduling Policy.....	166
Introduction.....	166
8.1 The Guarantee Module.....	168

8.1.1 Problems with Computing the Available Time	169
8.1.2 A New Algorithm for Computing the Available Time	171
8.1.2.1 The Preemption Case	174
8.1.2.2 The General Case	177
8.2 The Priority Assignment Module	180
8.3 Analysis of the Presented Algorithm	184
PART THREE : AN APPLICATION	190
General Introduction	190
Chapter 9 : Utilities for a Control Application	191
Introduction	191
9.1 SDL : A System Definition Language	193
9.1.1 The Program Header	194
9.1.2 The Declaration Part	194
9.1.3 The Main Body	196
9.2 Elementary Data Acquisition and Control Tasks	200
9.2.1 The Input Module	200
9.2.2 The Output Module	201
9.2.3 The Logger Module	201
9.2.4 The PID Module	202
9.3 The Operating System User Interface	203
9.3.1 Hardware Testing	203
9.3.2 System Configuration	204
9.4 The High Level Monitoring and Control Package	208
9.5 The Hardware Process Emulator	210
Chapter 10 : Real-Time Testing and Results	211
Introduction	211
10.1 Defining the Control System in SDL	212
10.2 Compiling System Definition	214
10.3 Downloading and Running Application Software	215
Chapter 11 : Conclusions	221
Chapter 12 : References	224
Appendix A : The Component Price List	231
Appendix B : Kernel Listing	232
Appendix C : Kernel Data Structures	252
Appendix D : The SDL Grammar	255

List of Figures

Fig. 2.1, Inoptimality of Fixed Priority Scheduling.....	17
Fig. 2.2, Hardware Architecture.....	24
Fig. 3.1, Distributed Control System Architecture.....	29
Fig. 4.1, A Scheduling Example.....	39
Fig. 4.2, Deterministic Scheduling.....	41
Fig. 4.3, Algorithm for Calculating the Maximum Response Time.....	46
Fig. 4.4, Example of Calculation of Maximum Response Time.....	47
Fig. 4.5, Incompatible Blockages.....	48
Fig. 4.6, Incompatibility Reduction.....	49
Fig. 4.7, The Second Largest Reduction.....	50
Fig. 4.8, The Subset Optimal Reduction.....	51
Fig. 4.9, The Latest Start Time.....	53
Fig. 4.10, The Guarantee Routine.....	54
Fig. 4.11, Expressions for DRIF Components.....	56
Fig. 4.12, Illustrations of DRIF Components.....	57
Fig. 6.1, The Control System.....	85
Fig. 6.2, A Single Node Structure.....	86
Fig. 6.3, The Computation Processor Card.....	87
Fig. 6.4, The Card Hardware Overview.....	89
Fig. 6.5, The Card Processor Bus Interface.....	90
Fig. 6.6, Bus Arbitration and Mode Control Logic.....	92
Fig. 6.7, The On-board I/O Component.....	94
Fig. 6.8, The 8088 Pin-Out.....	96
Fig. 6.9, The Bus Arbitration and Mode Control Wiring Diagram.....	97
Fig. 6.10, The Mode Control Register.....	98
Fig. 6.11, Dip Switch Settings.....	100
Fig. 6.12, The Card Processor Bus Interface.....	102
Fig. 6.13, The Host Processor Bus Interface.....	105
Fig. 6.14, Decoder Truth Table.....	106
Fig. 6.15, The Decoding Part.....	108
Fig. 6.16, I/O Address Map.....	109
Fig. 6.17, The I/O Clock Generator.....	109
Fig. 6.18, The Fixed I/O Component.....	112
Fig. 6.19, The I/O Address Map.....	113
Fig. 6.20, The Extension Component.....	116
Fig. 7.1, Task Memory Map.....	122
Fig. 7.2, Task States and Transitions.....	126
Fig. 7.3, Scheduler Software Structure.....	130
Fig. 7.4, The Real-Time Clock.....	131

Fig. 7.5. The Self-Block Module	131
Fig. 7.6. The Dispatcher	132
Fig. 7.7. The Periodic Invocation Module	133
Fig. 7.8. The Internal Sporadic Invocation Module	134
Fig. 7.9. The External Sporadic Invocation Module	134
Fig. 7.10. The Scheduler Interface Module	135
Fig. 7.11. The Policy Module	136
Fig. 7.12. The Stack Manipulation	143
Fig. 7.13. The Link Primitive	150
Fig. 7.14. The Send/SendToTarget Primitives	156
Fig. 7.15. The FastSend Primitive	157
Fig. 7.16. The Receive Primitive	158
Fig. 7.17. The FastReceive Primitive	160
Fig. 7.18. The Card Memory Layout	163
Fig. 7.19. The Task Fetch Routine	163
Fig. 7.20. Interprocessor Communication Handler	164
Fig. 8.1. Task Arrival	175
Fig. 8.2. Task Termination	176
Fig. 8.3. General Task Arrival Algorithm	178
Fig. 8.4. Contents of List	182
Fig. 8.5. Range of Guarantee Ratio Under the Dynamic Algorithm	186
Fig. 8.6. Range of Guarantee Ratio Under the Static Algorithm	186
Fig. 8.7. Comparison of Average Guarantee Ratio	187
Fig. 8.8. A Four-Task Example	187
Fig. 8.9. The Generated Task Schedules	188
Fig. 8.10. Actual Processor Utilization	189
Fig. 9.1. A Simple Control Configuration	195
Fig. 9.2. The Declaration Part	196
Fig. 9.3. The Main Body	197
Fig. 9.4. The Chanel Assignment	198
Fig. 9.5. The Main Menu	203
Fig. 9.6. The Tester Utility	204
Fig. 9.7. Task Editing	205
Fig. 9.8. The Configuration Utilities	206
Fig. 9.9. The Loader Options	207
Fig. 9.10. The High Level Monitoring and Control Package	208
Fig. 10.1. The Control Loop Block Diagram	211
Fig. 10.2. The Test System	212
Fig. 10.3. System Description in SDL	213
Fig. 10.4. Compiler Output Listing	214
Fig. 10.5. Steady State on Display 1	216
Fig. 10.6. Response to Flow Set Point Change	216

Fig. 10.7, Response to Temperature Set Point Change.....	217
Fig. 10.8, Response to Disturbance on Display 2	217
Fig. 10.9, Alarm Condition.....	218
Fig. 10.10, The Alarm Log.....	219

Chapter 1

Introduction

During the last several decades automatic control applications have spread to affect virtually every detail of modern life. Microprocessors have been largely used in such applications. As the applications grew more complex, so did the systems which control them. Today, distributed microprocessor-based control systems have become very common. Such systems are used to control diverse environments. Examples range from process control and factory automation on one side to space shuttle control and complicated military defense systems on the other. As diverse as such systems may be they share one important attribute, time criticalness. The system is typically embedded in an environment which it must monitor, control, or interact with in some other form. The nature of the environment imposes time constraints on the behavior of the system thus making it a real-time one. Distributed Control Systems, DCS, are often organized in a hierarchical manner.

The first and lowest level in the hierarchy is directly connected to the controlled system (environment). It is composed of a set of data acquisition and output modules, each of which is concerned with a single point in the environment.

Each group of data acquisition and output modules belongs to a single processor module, a collection of which comprises the second level of the system hierarchy. Processor modules have computational power and are responsible for data logging as well as different control functions.

A collection of processor modules may be interfaced to a local host computer thus making a single DCS node. The system may be composed of several such nodes together forming the third level of its hierarchy. Each host may have a local user interface as well as mass storage facility. Nodes are typically connected to a central computing facility by data hiways.

For commercial DCS products, the internal design of hardware and software is classified and is a vendor confidential information. Therefore the mere implementation of such a system is a contribution, not to mention an implementation which uses components that are both economical and available at the local market. This consideration has been the motivation for the work presented in this thesis.