

DSP FUNDAMENTALS AND SYSTEM DESIGN

TABLE OF CONTENTS

1. INTRODUCTION

- 1.1 OVERVIEW
- 1.2 USE IN ACCELERATORS

2. DSP EVOLUTION AND CURRENT SCENERY

- 2.1 DSP EVOLUTION
- 2.2 CURRENT MAINSTREAM DSPS

3. DSP CORE ARCHITECTURE

- 3.1 INTRODUCTION
- 3.2 FAST DATA ACCESS
- 3.3 FAST COMPUTATION
- 3.4 NUMERICAL FIDELITY
- 3.5 FAST-EXECUTION CONTROL

4. DSP PERIPHERALS

- 4.1 INTRODUCTION
- 4.2 INTERCONNECT & DATA I/O
- 4.3 SERVICES
- 4.4 C6713 EXAMPLE
- 4.5 MEMORY INTERFACING
- 4.6 DATA CONVERTER INTERFACING
- 4.7 DSP BOOTING

5. RT DESIGN FLOW: INTRODUCTION

6. RT DESIGN FLOW: S/W DEVELOPMENT

- 6.1 DSP PROGRAMMING INTRODUCTION
- 6.2 DEVELOPMENT SETUP AND ENVIRONMENT
- 6.3 LANGUAGES: ASSEMBLY, C, C++, GRAPHICAL
- 6.4 RTOS
- 6.5 CODE BUILDING PROCESS

7. RT DESIGN FLOW: DEBUGGING

- 7.1 BUGS AND DEBUGGING
- 7.2 SIMULATION

- 7.3 EMULATION
- 7.4 TRADITIONAL EMULATION TECHNIQUES
- 7.5 REAL-TIME DEBUGGING

8. RT DESIGN FLOW: ANALYSIS AND OPTIMISATION

- 8.1 INTRODUCTION
- 8.2 OPTIMISER ON
- 8.3 ANALYSIS TOOLS
- 8.4 OPTIMISATION GUIDELINES

9. RT DESIGN FLOW: SYSTEM DESIGN

- 9.1 INTRODUCTION: DSP AND ARCHITECTURE CHOICE
- 9.2 DSP: FIXED VS. FLOATING
- 9.3 DSP: BENCHMARKING
- 9.4 ARCHITECTURE: MULTI-PROCESSING OPTION
- 9.5 ARCHITECTURE: RADIATION EFFECTS
- 9.6 ARCHITECTURE: INTERFACES
- 9.7 CODE DESIGN: INTERRUPT-DRIVEN VS. RTOS
- 9.8 CODE DESIGN: GOOD PRACTICES
- 9.9 GENERAL RECOMMENDATIONS

10. RT DESIGN FLOW: SYSTEM INTEGRATION

- 10.1 INTRODUCTION
- 10.2 GOOD PRACTICES

11. PUTTING IT ALL TOGETHER ... A DIGITAL SYSTEM EXAMPLE: