TABLE OF CONTENTS

List of Figures vi
Code Listings vii
Abstract ix
PART I: THE VHDL LANGUAGE
Chapter 1: TOP LEVEL VHDL OVERVIEW 1
1.0 VHDL Design units 1
1.1 Entity 1
1.2 Architecture 3
1.2.1 Behavioral Models 3
1.2.2 Structural Models 4
1.3 Configuration 5
1.4 Packages and Package Bodies 6
1.4.1 Overloading 7
Chapter 2: VHDL OBJECTS 9
2.0 VHDL Objects 9
2.1 Signals 9
2.2 Signal Operators 9
2.2.1 Signal Type Attributes 9
2.2.2 Array Attributes 11
2.2.3 Signal Attributes that Return a Value 12
2.2.4 Signal Attributes that Return a Signal 13
2.3 Signal Drivers 14
2.4 Signal Resolution Functions 14
2.5 Variables 16
2.6 Constants 16

2.7 Generics 16
Chapter 3: VHDL DATA STRUCTURES 17
3.0 VHDL Types 17
3.1 Signals, Variables and Constants 17
3.1.1 Signals 17
3.1.2 Variables 17
3.1.3 Constants 17
3.2 Scalar Types 18
3.2.1 Integer 18
3.2.2 Real 18
3.2.3 Enumerated 18
3.2.4 Physical 19
3.3 Composite Types 19
3.3.1 Arrays 19
3.3.2 Records 20
3.4 Subtypes 20
3.5 Alias 21
3.6 Unconstrained Types 21
3.7 Overloading 22
3.8 User Defined Attributes 22
3.9 Generate 22
3.10 Access Types 22
3.11 Incomplete Types 23
3.12 File Types 23
Chapter 4: VHDL EXECUTION 24
4.0 Concurrent Processing 24
4.1 Transport and Inertial Delays 25

4.2 Concurrent Statements 25
4.3 Sequential Processing 26
4.4 Processes 26
4.4.1 Sensitive Processes 27
4.4.2 Non-Sensitive Processes 28
4.4.3 Important Notes Regarding Synthesis 28
4.5 Sequential Statements 29
4.5.1 IF-THEN-ELSE 29
4.5.2 Case Statement 30
4.5.3 Looping 32
4.5.3.1 The For Loop 32
4.5.3.2 The While Loop 33
4.5.3.3 Exit 33
4.6 Assert 34
4.6.1 Timing Checks 34
4.7 Wait 35
4.7.1 Wait Examples 35
4.8 Subprograms 36
4.8.1 Functions 36
4.8.2 Procedures 37
PART II: VHDL EXAMPLES
Chapter 5: MUX EXAMPLE 38
5.0 A Complete VHDL Example 38
5.1 Behavioral 4-1 MUX 38
5.1.1 MUX Entity 38
5.1.2 Architecture 39
5.1.3 Configuration 39

5.2 Behavioral 8-1 MUX 40
5.2.1 Entity 40
5.2.2 Architecture 41
5.2.3 Configuration 42
5.3 VHDL Testbenches 42
5.3.1 MUX Testbench 43
Chapter 6: AM2900 VHDL Example 46
6.0 The AM2900 Bit- Sliced Pipelined Processor 46
6.1 I/O 46
6.2 Execution 47
6.2.1 Memory Load 48
6.2.2 Execution 48
6.2.2.1 Source Select 49
6.2.2.2 ALU 50
6.2.2.3 Destination Control Word 52
6.2.2.4 The Next Instruction Control 55
Chapter 7: TESTING THE AM2900 58
7.0 The AM2900 Testbenches 58
7.1 Load the AM2900 58
7.2 Branch and Continue 61
Chapter 8: AM2900 SYNTHESIS 65
8.0 Coding the AM2900 for synthesis 65
8.1 Multiple Processes 65
8.2 Reduced Duplicity 65
8.3 Variable Management 65
8.4 The Result 66
Chapter 9: CONCLUSIONS 67

References 68
Appendix A: AM2900 SOURCE CODE
Appendix B: AM2900 Package
Appendix C: AM2900 Test Bench #1
Appendix D: AM2900 Test Bench #2
Appendix E: AM2900 Synthesis Code
Appendix F: AM2900 TestBench #1 Result
Appendix G: AM2900 TestBench #2 Result

List of Figures

1. Simple NAND Gate 2

2. RS-FF 4

3. 4-1 MUX 38

Code Listings

1. FF ENTITY statement 2
2. Nand Architecture 3
3. RS Flip Flop Example 4
4. RS Flip Flop Configuration 5
5. Parity Package 7
6. Overloading 7
7. Signal Contention 14
8. Sample Resolution Function 15
9. Enumerated Types 18
10. Physical Types 19
11. Array Example 19
12. Record Example 20
13. Subtype Example 20
14. Alias Example 21
15. Unconstarined Types 21
16. Access Types 23
17. Concurrent Processing 24
18. Transport Delay Example 25
19. Process Equivalence 27
20. Sensitive Process (NAND gate) 27
21. Process Example (D Flip Flop) 28
22. VHDL IF-THEN-ELSE Clause 29
23. VHDL CASE Statement 30
24. Case Statement Example 31
25. Finite State Machine Example 31
26. FOR Loop Example 32

27. While Loop Example 33
28. Exit Example 34
29. ASSERT Examples 35
30. Function Example 37
31. Procedure Example 37
32. 4-1 MUX Entity 39
33. 4-1 MUX Architecture 39
34. 4-1 MUX Configuration 40
35. 8-1 MUX Entity 40
36. 8-1 MUX Architecture 42
37. 8-1 MUX Configuration 42
38. The Complete 8-1 MUX VHDL 45
39. AM2900 Execution 47
40. AM2900 Memory Load 48
41. AM2900 Source Select 49
42. AM2900 ALU 50
43. AM2900 Add/Subtract Routine 51
44. AM2900 Destination Control Word 55
45. AM2900 Next Instruction Control 57
46. AM2900 Testbench #1 61
47. AM2900 Branch and Continue 64

VHDL: An Introduction

Abstract

The VHSIC (Very High Speed Integrated Circuit) Hardware Description Language (VHDL) has recently become the industry standard for switch-level hardware modeling. What this means to students of electrical engineering, is that hardware engineering is itself becoming a matter of software. The reason for VHDL's success can be attributed to its ability to model the *behavior* of a circuit, not just the functionality of it. This allows VHDL designs to be hardware independent. A VHDL model can be constructed with the behavior remaining constant, the only change being the use of different technology specific resolution functions. The big impact of this is the ability to re-procure parts at a later date. For Example, if a particular ASIC fabrication process becomes outdated, it is possibly to "synthesize" a new gate level design from the same behavioral models. Both designs functionally identical, but compiled into different ASIC lines. Below are listed some of the advantages and disadvantages of VHDL:

Advantages-

- The ability to code the behavior of a system and use hardware synthesis tools to "synthesize" an actual circuit
- Technology independent hardware descriptions
- Fast switch level simulations
- Industry standard

Disadvantages-

- Only a switch level language. Results can be inaccurate (esp. at high speeds)
- Synthesis tools are not very robust or complete at this time

The main issues discussed in this paper are:

(i) A brief introduction to VHDL

(ii) An introduction to VHDL testbenches

(iii) A VHDL implementation of the AM2900 bit sliced pipe-lined processor

(iv) Sample testbenches for the AM2900

(v) An example synthesis of the AM2900

My experience with VHDL has been coding portions of the F-22 Electronic Warfare System. Shortly I will be starting full VHDL design using synthesis to develop ASICs. I have also been studying VHDL, including development of all the EE126 projects and the EE126 Basic computer. All of these designs which I have been working on have been written in Synthesizeable VHDL.