

## Contents

1.1	The Halting Problem . . . . .	1
1.1.1	Universal Computing Machines* . . . . .	1
1.1.2	The Halting Problem* . . . . .	1
1.2	The 2's Complement Representation . . . . .	3
1.2.1	Review of Bits and the Unsigned Representation . . . . .	3
1.2.2	Picking a Good Representation . . . . .	3
1.2.3	The Unsigned Add Unit . . . . .	3
1.2.4	Deriving 2's Complement . . . . .	4
1.2.5	An Algebraic Approach . . . . .	5
1.2.6	Negating 2's Complement Numbers . . . . .	6
1.3	Overflow Conditions . . . . .	7
1.3.1	Implication and Mathematical Notation . . . . .	7
1.3.2	Overflow for Unsigned Addition . . . . .	7
1.3.3	Overflow for 2's Complement Addition . . . . .	8
1.4	Logic Operations . . . . .	11
1.4.1	Truth Tables . . . . .	11
1.4.2	Boolean Logic Operations . . . . .	11
1.4.3	Overflow as Logic Expressions . . . . .	13
1.4.4	Logical Completeness . . . . .	14
1.4.5	Implications of Logical Completeness . . . . .	16
1.4.6	Examples and a Generalization . . . . .	16
1.5	Programming Concepts and the C Language . . . . .	18
1.5.1	The C Programming Language . . . . .	18
1.5.2	Data Types . . . . .	19
1.5.3	Variable Declarations . . . . .	21
1.5.4	Expressions and Operators . . . . .	21
1.5.5	Basic I/O . . . . .	22
1.5.6	Types of Statements in C . . . . .	24
1.5.7	Program Execution . . . . .	25
1.5.8	Compilation and Interpretation* . . . . .	27
1.5.9	The C Preprocessor* . . . . .	28
1.5.10	Changing Types in C* . . . . .	28
1.6	Summary of Part 1 of the Course . . . . .	30
2.1	Optimizing Logic Expressions . . . . .	33
2.1.1	Defining Optimality . . . . .	33

2.1.2	Terminology . . . . .	35
2.1.3	Veitch Charts and Karnaugh Maps . . . . .	36
2.1.4	Canonical Forms . . . . .	40
2.1.5	Two-Level Logic . . . . .	40
2.1.6	Multi-Metric Optimization . . . . .	41
2.2	Boolean Properties and Don't Care Simplification . . . . .	44
2.2.1	Logic Properties . . . . .	44
2.2.2	Choosing the Best Function . . . . .	44
2.2.3	Caring about Don't Cares . . . . .	45
2.2.4	Generalizations and Applications* . . . . .	48
2.3	Example: Bit-Sliced Addition . . . . .	49
2.3.1	One Bit at a Time . . . . .	49
2.3.2	Abstracting the Human Process . . . . .	49
2.3.3	Designing the Logic . . . . .	50
2.3.4	Adders and Word Size . . . . .	51
2.4	Example: Bit-Sliced Comparison . . . . .	53
2.4.1	Comparing Two Numbers . . . . .	53
2.4.2	An Abstract Model . . . . .	53
2.4.3	A Representation and the First Bit . . . . .	54
2.4.4	Optimizing Our Design . . . . .	56
2.4.5	Extending to 2's Complement . . . . .	57
2.4.6	Further Optimization . . . . .	58
2.5	Using Abstraction to Simplify Problems . . . . .	59
2.5.1	Subtraction . . . . .	59
2.5.2	Checking ASCII for Upper-case Letters . . . . .	60
2.5.3	Checking ASCII for Lower-case Letters . . . . .	62
2.5.4	The Multiplexer . . . . .	63
2.6	Sequential Logic . . . . .	65
2.6.1	Storing One Bit . . . . .	65
2.6.2	The Clock Abstraction . . . . .	67
2.6.3	Static Hazards: Causes and Cures* . . . . .	69
2.6.4	Dynamic Hazards* . . . . .	70
2.6.5	Essential Hazards* . . . . .	71
2.6.6	Proof Outline for Clocked Synchronous Design* . . . . .	72
2.7	Registers . . . . .	73
2.7.1	Registers . . . . .	73

2.7.2	Shift Registers . . . . .	74
2.8	Summary of Part 2 of the Course . . . . .	76
3.1	Serialization and Finite State Machines . . . . .	79
3.1.1	Serialization: General Strategy . . . . .	79
3.1.2	Serialization: Comparator Example . . . . .	80
3.1.3	Finite State Machines . . . . .	82
3.1.4	Synchronous Counters . . . . .	84
3.1.5	Ripple Counters . . . . .	86
3.1.6	Timing Issues* . . . . .	87
3.1.7	Machine Models . . . . .	87
3.2	Finite State Machine Design Examples, Part I . . . . .	89
3.2.1	Steps in the Design Process . . . . .	89
3.2.2	Example: A Two-Bit Gray Code Counter . . . . .	90
3.2.3	Example: A Three-Bit Gray Code Counter . . . . .	90
3.2.4	Example: A Color Sequencer . . . . .	91
3.2.5	Identifying an Initial State . . . . .	92
3.2.6	Developing an Abstract Model . . . . .	93
3.2.7	Specifying I/O Behavior . . . . .	94
3.2.8	Completing the Specification . . . . .	94
3.2.9	Choosing a State Representation . . . . .	95
3.2.10	Abstracting Design Symmetries . . . . .	97
3.2.11	Impact of the State Representation . . . . .	99
3.3	Design of the Finite State Machine for the Lab . . . . .	100
3.3.1	Physical Design, Sensors, and Timing . . . . .	100
3.3.2	An Abstract Model . . . . .	101
3.3.3	Picking the Representation . . . . .	102
3.3.4	Testing the Design . . . . .	103
3.4	Extending Keyless Entry with a Timeout . . . . .	105
3.5	Finite State Machine Design Examples, Part II . . . . .	107
3.5.1	Design of a Vending Machine . . . . .	107
3.5.2	Encoders and Decoders . . . . .	109
3.5.3	Vending Machine Implementation . . . . .	111
3.5.4	Design of a Game Controller . . . . .	113
3.5.5	Analysis of a Stoplight Controller . . . . .	115
3.6	Random Access Memories . . . . .	118
3.6.1	Memory . . . . .	118

3.6.2	Static Random Access Memory . . . . .	119
3.6.3	Tri-State Buffers and Combining Chips . . . . .	122
3.6.4	Dynamic Random Access Memory* . . . . .	123
3.7	From FSM to Computer . . . . .	126
3.7.1	Specifying the Problem . . . . .	126
3.7.2	Choosing Components and Identifying States . . . . .	127
3.7.3	Laying Out Components . . . . .	129
3.7.4	Control and Data . . . . .	130
3.7.5	State Representation and Logic Expressions . . . . .	131
3.8	Summary of Part 3 of the Course . . . . .	132
4.1	Control Unit Design . . . . .	135
4.1.1	LC-3 Datapath Control Signals . . . . .	135
4.1.2	Example Control Word: ADD . . . . .	137
4.1.3	Example Control Word: LDR . . . . .	138
4.1.4	Hardwired Control . . . . .	139
4.1.5	Using a Memory for Logic Functions . . . . .	140
4.1.6	Microprogrammed Control . . . . .	142
4.2	Redundancy and Coding . . . . .	144
4.2.1	Sparse Representations . . . . .	144
4.2.2	Error Detection . . . . .	144
4.2.3	Parity . . . . .	145
4.2.4	Hamming Distance . . . . .	146
4.2.5	Error Correction . . . . .	146
4.2.6	Hamming Codes . . . . .	147
4.2.7	SEC-DED Codes . . . . .	148
4.3	Instruction Set Architecture* . . . . .	149
4.3.1	Formats and Fields* . . . . .	149
4.3.2	Addressing Architectures* . . . . .	151
4.3.3	Common Special-Purpose Registers* . . . . .	153
4.3.4	Reduced Instruction Set Computers* . . . . .	153
4.3.5	Procedure and System Calls* . . . . .	154
4.3.6	Interrupts and Exceptions* . . . . .	156
4.3.7	Control Flow Conditions* . . . . .	157
4.3.8	Stack Operations* . . . . .	157
4.3.9	I/O* . . . . .	158
4.4	Summary of Part 4 of the Course . . . . .	159