

Computer Structure
&
Introduction to Digital Computers
Lecture Notes

by
Guy Even

Dept. of Electrical Engineering - Systems, Tel-Aviv University.
Spring 2003

Copyright 2003 by Guy Even
Send comments to: *guy@eng.tau.ac.il*

Contents

1	The digital abstraction	1
1.1	Transistors	1
1.2	From analog signals to digital signals	3
1.3	Transfer functions of gates	5
1.4	The bounded-noise model	6
1.5	The digital abstraction in presence of noise	7
1.5.1	Redefining the digital interpretation of analog signals	7
1.6	Stable signals	9
1.7	Summary	10
2	Foundations of combinational circuits	11
2.1	Boolean functions	11
2.2	Gates as implementations of Boolean functions	11
2.3	Building blocks	12
2.4	Combinational circuits	14
2.5	Cost and propagation delay	18
2.6	Syntax and semantics	19
2.7	Summary	20
3	Trees	21
3.1	Trees of associative Boolean gates	21
3.1.1	Associative Boolean functions	21
3.1.2	OR-trees	22
3.1.3	Cost and delay analysis	23
3.2	Optimality of trees	25
3.2.1	Definitions	25
3.2.2	Lower bounds	26
3.3	Summary	29
4	Decoders and Encoders	31
4.1	Notation	31
4.2	Values represented by binary strings	33
4.3	Decoders	34
4.3.1	Brute force design	34
4.3.2	An optimal decoder design	35

4.3.3	Correctness	35
4.3.4	Cost and delay analysis	36
4.4	Encoders	37
4.4.1	Implementation	38
4.4.2	Cost and delay analysis	41
4.4.3	Yet another encoder	42
4.5	Summary	43
5	Combinational modules	47
5.1	Multiplexers	47
5.1.1	Implementation	47
5.2	Cyclic Shifters	49
5.2.1	Implementation	50
5.3	Priority Encoders	53
5.3.1	Implementation of U-PENC(n)	54
5.3.2	Implementation of B-PENC	55
5.4	Half-Decoders	59
5.4.1	Preliminaries	59
5.4.2	Implementation	60
5.4.3	Correctness	62
5.4.4	Cost and delay analysis	62
5.5	Logical Shifters	63
5.6	Summary	63
6	Addition	65
6.1	Definition of a binary adder	65
6.2	Ripple Carry Adder	66
6.2.1	Correctness proof	66
6.2.2	Delay and cost analysis	67
6.3	Carry bits	67
6.4	Conditional Sum Adder	69
6.4.1	Motivation	69
6.4.2	Implementation	69
6.4.3	Delay and cost analysis	69
6.5	Compound Adder	71
6.5.1	Implementation	72
6.5.2	Correctness	72
6.5.3	Delay and cost analysis	73
6.6	Summary	74
7	Fast Addition	75
7.1	Reduction: sum-bits \mapsto carry-bits	75
7.2	Computing the carry-bits	75
7.2.1	Carry-Lookahead Adders	76
7.2.2	Reduction to prefix computation	79

7.3	Parallel prefix computation	81
7.3.1	Implementation	81
7.3.2	Correctness	82
7.3.3	Delay and cost analysis	82
7.4	Putting it all together	83
7.5	Summary	84
8	Signed Addition	85
8.1	Representation of negative integers	85
8.2	Negation in two's complement representation	86
8.3	Properties of two's complement representation	88
8.4	Reduction: two's complement addition to binary addition	89
8.4.1	Detecting overflow	91
8.4.2	Determining the sign of the sum	92
8.5	A two's-complement adder	93
8.6	A two's complement adder/subtractor	94
8.7	Additional questions	96
8.8	Summary	98
9	Flip-Flops	99
9.1	The clock	99
9.2	Edge-triggered Flip-Flop	100
9.3	Arbitration	102
9.4	Arbiters - an impossibility result	103
9.5	Necessity of critical segments	105
9.6	An example	106
9.6.1	Non-empty intersection of C_i and A_i	108
9.7	Other types of memory devices	109
9.7.1	D-Latch	109
9.7.2	Clock enabled flip-flops	109
10	Synchronous Circuits	111
10.1	Syntactic definition	111
10.2	Timing analysis: the canonic form	111
10.2.1	Canonic form of a synchronous circuit	113
10.2.2	Timing constraints	113
10.2.3	Sufficient conditions	114
10.2.4	Satisfying the timing constraints	116
10.2.5	Minimum clock period	116
10.2.6	Initialization	117
10.2.7	Functionality	118
10.3	Timing analysis: the general case	119
10.3.1	Timing constraints	119
10.3.2	Algorithm: minimum clock period	120
10.3.3	Algorithm: correctness	121

[Click here to download full PDF material](#)