

INDEX

- Absorption law, 31, 38
- Acyclic graph, 35
 - tree, 36
- Addition operators, in VHDL (VHSIC hardware description language), 192
- Algebraic division, 105
- AND gate, 48–49
- Antisymmetric, 34
- Applicable input sequence, 246
- Architecture, description of
 - behavioral model, 185
 - structural model, 185
 - 2-to-1 multiplier, 184–185
 - VHDL (VHSIC hardware description language), 194–196
- Arcs, 35
- Arithmetic circuits, 125–141
 - BCD adders, 132–133
 - BCD subtractors, 137–138
 - carry save addition, 130–132
 - carry select, 130
 - carry-lookahead adders, 129
 - carry-save, 130
 - comparator, 140–141
 - full adders, 126–129
 - full subtractors, 135
 - half subtractors, 133–135
 - half-adders, 125
 - multiplication, 138–140
 - two's complement subtractors, 135–137
- Arithmetic operations, 5–8
- Associative law, 31
- Asynchronous counter design, 291–295
 - ripple, 291–294
 - up-down, 294–295
- Asynchronous operation, 158–159
- Asynchronous preset, 320, 322
- Asynchronous sequential circuits, 373–397
 - excitation functions, 387–389
 - flow table, 374–376
 - hazards, 390–397
 - output functions, 387–389
 - state assignment, 379–387
- Barrel shifter, 327–328
- Base, 1
- BCD adders, 132–133
- BCD code, 20
- BCD subtractors, 137–138
- Behavioral description, functional simulation, of VHDL (VHSIC hardware description language), 199
- VHDL (VHSIC hardware description language), 199
- Behavioral VHDL, 181
- Binary arithmetic operations, 5–8
- Binary coded decimal. *See* BCD
- Binary codes, 1–24
 - binary encoding, 1, 20–25
- Binary encoding, 1
 - non-weighted codes, 22–25
- Binary numbers, 2–8
 - arithmetic operations, 5–8
 - borrow, 6
 - hexadecimal, 8–11
 - minuend, 6
 - octal numbers, 8–11
 - signed and unsigned, 14
 - subtrahend, 6
- Binary
 - hexadecimal to, conversion of, 12
 - octal from, conversion of, 10
- Bistable element, 162
- Block diagram, 2-to-1 multiplexer, 217
- Blocks, 34

- Boolean algebra, 37–40
 - Huntington's postulates, 37
 - theorems for, 37–40
 - Absorption Laws, 38
 - consensus, 39
 - DeMorgan's theorem, 38–39
 - Idempotent Laws, 37
 - Involution Theorem, 38
 - two valued, 40
- Boolean difference, 112–113
- Boolean division, 105–106
- Boolean expressions, minimization, 60–63
- Boolean functions, 41–43
 - canonical forms, 45
 - classification, 43–45
 - complementation, 41
 - cubical representation, 79–85
 - DeMorgan's Theorem, 43
 - derivation, 43–45
 - product, 41
 - shared product determination, 95
 - sum of products, 43
 - sum, 41
 - symbols, use of, 41
 - truth table, 42–43
 - variables, 41
- Boolean substitution, 104
- Boolean variables, 41
- Branches, 36
- Buffer ports, operation of, 183
- Buffer, definition of, 183
- Byte, 13
 - nibble, 13
- Canonical forms, 45
 - canonical sum of products form, 45–48
 - maxterm, 45
 - minterm, 45
- Canonical sum of product forms, 45–48
- Cardinality, 80
- Carry lookahead adders, 129–130
- Carry save adder, 130
- Carry save addition, 130–132
- Carry select adder, 130
- Cartesian products, 32
 - antisymmetric, 34
 - equivalence relation, 34
 - symmetric, 33
 - transitive, 34
- Case statements, 220–223
- Characters, as lexical elements, 186
- Circulating shift register, 307
- Classification, 43–45
- Clock, 158
- Clocked CMOS circuits, 407–408
- Clocked sequential circuits, 158
- Closure conditions, 248
- Closure property, 37
- CMOS logic, 403–409
 - clocked circuits, 407–408
 - domino, 408–409
 - transmission gates, 405–406
- Code assignments, 268–270
- Cofactors, 82
- Coincidence gate, 53
- Cokernel cube matrix, 107
 - rectangle, 108
 - rectangular cover, 108
- Collapsing, inverse operation of substitution
 - and, 103, 104
- Combinational circuit design, programmable
 - logic devices, 141–150
- Combinational logic design, 50–150, 205–233
 - 2-out-of-4 decoder, 210
 - 4-to-1 multiplexer, 208
 - implied memory, 209
 - 4-to-2 priority encoder, 211
 - arithmetic circuits, 125–141
 - Boolean expressions, minimization of, 60–63
 - Boolean functions, cubical representation, 79–85
 - circuit function example, 205
 - concurrent assignment statements, 206–214
 - conditional assignments, 207–211
 - direct signal assignment, 206–207
 - for loop, 225–229
 - for-generate* statement, 230–233
 - implementation of, 114–117
 - Karnaugh maps, 63–73
 - logic circuit design, 117–125
 - heuristic minimization of, 85–95
 - loops, 225–230
 - multilevel, 102–109
 - multiple output functions, 95–98
 - NAND–NAND logic, 98–101
 - NOR–NOR logic, 101–102
 - Quine–McCluskey method, 73–79
 - selected conditional signal assignment, 211–214
 - sequential assignment statements, 214–224
 - truth table, 60
 - while loops, 229–230

- Combinational logic implementation, EX-OR
 - AND AND gate, 114–117
- Comments, as lexical elements, 186
- Commutative law, 31
- Comparator, 140–141
- Compatibility class, 247
- Complement form, 14
- Complement, 37
- Complementary approach, 70–73
- Complementation, 41, 84–85
- Complex PLDs, 278
- Component instantiation statement,
 - definition of, 198, 206
- Concurrent assignment statements, in
 - combinational logic design, 206–214
- Concurrent statements, in VHDL (VHSIC hardware description language),
 - 192–194
- Conditional assignments, combinational
 - logic design, 207–211
- Connection matrix, 35
- Consensus, 39
- Control equations, 175–176
- Control inputs, multiplexers and, 122
- Counter design, 291–312
 - asynchronous, 291–295
 - gray code, 300–302
 - Johnson counters, 310–313
 - ring, 307–309
 - shift register, 302–307
 - synchronous, 291, 295
- Counters, 332–338
 - decade, 334–335
 - gray code, 335–336
 - Johnson, 337–338
 - ring, 336–337
- Cover, 80
 - cardinality, 80
 - irredundant, 80
 - minimal, 80
 - size, 80
- Covering conditions, 248
- Critical race free state assignment, 381–386
- Critical races, 380
- Crosspoints, fuses, 141
- Cube, 79
 - cover, 80
 - implicant, 80
 - intersection, 82
 - minterm, 80
 - positional cube notation, 81
 - supercube, 82
- Cubical representation, 79–85
 - literal, 79
 - tautology, 82–84
- Cyclic code, 23
 - reflected, 23–25
- D* flip flops, 163–164, 316–318
- D* latch, 315–316
 - level sensitive device, 315
- Data flow description, 181. *See also* RTL
 - description
- Data objects, as lexical elements,
 - 186–187
- Data types, VHDL (VHSIC hardware description language), 187–189
 - bit, 187
 - Boolean, 187
- Decade counters, 334–335
- Decimal numbers, 1–2
- Decoders, 123–125
- Decomposition process, 103
- Decomposition, 261–265
 - reduced dependency, 262
 - substitution property, 262
- DeMorgan's law, 32
- DeMorgan's Theorem, 38–39
- Demultiplexers, 123–125
- Derivation, 43–45
- Digital logic
 - Boolean
 - algebra, 37–40
 - functions, 41–43
 - concepts of, 29–53
 - graphs, 35–37
 - logic gates, 48–53
 - partitions, 34–35
 - relations, 32–34
 - sets, 29–32
- Digraph, 35
- Diminished radix complement, 14–16
 - 1's complement, 14
 - end-around carry, 15
- Direct signal assignments, combinational
 - logic design, 206
- Directed graph, 35
 - digraph, 35
 - acyclic, 35
 - in-degree, 36
 - out-degree, 36
 - path, 35
 - path, cycle, 35
- Disjoint, 31

- Distributive law, 31
- Division operators, in VHDL (VHSIC hardware description language), 192
- Domino CMOS logic, 408–409
- Don't care conditions, 63, 78–79
 - incompletely specified, 68
- Don't cares
 - multilevel circuit minimization and, 109–114
 - observability, 110, 112–114
 - satisfiability, 110–112
- Double rail inputs, 99
- Duality, 37
- Dynamic logic hazards, 395–396

- Edges, 35
- EHDL abstractions, examples of, 182
- Empty sets, 30
- Encoding binary numbers, 20–26
 - weighted codes, 20–22
- End around carry, 15
- Entity-architecture pair, in VHDL, 182
- Enumerated type data, VHDL codes and, 342–345
- EPLDs, 278–285
- Equivalence classes, 35
- Equivalence gate, 53
- Equivalence partition, 241
- Equivalence relation, 34
- ESPRESSO, 91, 92–95
 - Karnaugh map, 92–95
- Essential hazards, 396–397
- Essential prime implicant, 74
- Excess-e code, 22
- Excitation functions, 387–389
- Excitation variables, 158
- Exclusive NOR, EX–OR, 51–53
- Exclusive OR, 51
- EX–NOR gate, 51–53
 - coincidence, 53
 - equivalence, 53
- EX–OR AND AND gate, 114–117
 - parity bit, 115
 - programmable inverter, 116
 - Reed–Muller canonical form, 116–117
 - rules for operation, 115
- EX–OR gate, 51–53
- EXPAND, 85–88
- Extraction process, 103–104

- Factoring process, 103, 105
- Fall delay, 168

- Fan out oriented algorithm, 265–267
- Fan-in oriented algorithm, 265, 267–268
- Finite state machine. *See* Synchronous sequential circuits
- Flattening, inverse operation of substitution and, 104
- Flip flops, 162–168, 316–324
 - asynchronous preset, 320, 322
 - bistable element, 162
 - D*, 316–318
 - hold time, 162
 - JK*, 318–320
 - metastable state, 162
 - next state expression, 249–257
 - transition table, 250
 - setup time, 162
 - synchronous present, 320, 322
 - T*, 318–319
 - types
 - D*, 163–164
 - JK*, 165–167
 - T*, 167–168
- Floating point numbers
 - mantissa, 19
 - normalization, 19
- Flow table, primitive, 376, 377–378
- For loop, in combinational logic design, 225–229
- For-generate* statement, in combinational logic design, 230–233
- Full adders, 126–129
 - ripple, 128
 - truth table, 127
- Full subtractors, 135
- Function hazards, 391–392
- Fuses, 141

- Gated latches, 160
- Graphs, 35–37
 - arcs, 35
 - connection matrix, 35
 - directed, 35
 - edges, 35
 - nodes, 35
 - nondirected, 35
 - vertices, 35
- Gray code, 23–25
 - counters, 300–302, 335–336

- Half adders, 125–126
- Half subtractors, 133–135
- Hardware description language (HDL), 181

- Hazards, 390–397
 - essential, 396–397
 - function, 391–392
 - logic, 393–396
- HDL (hardware description language), 181
- Heuristic minimization, logic circuits and, 85–95
- Hexadecimal, 8–13
 - binary from, conversion of, 12
 - byte, 13
 - nibble, 13
- Hold time, 162
- Huntington’s postulates, 37
 - closure property, 37
 - complement, 37
 - duality, 37
- Idempotent Laws, 32, 37
- If versus case statements*, 223–224
- If-then statements*, 216–220
- Implicant, 80
 - prime, 80
- Implication table, 242–244
 - incompatibles, 242
- Incompatibles, 242
- Incompletely specified, 68
- Incompletely specified sequential circuits
 - applicable input sequence, 246
 - closure conditions, 248
 - compatibility class, 247
 - covering conditions, 248
 - minimization of, 244–249
- In-degree, 36
- Intersection, 31, 82
- Inverse operation of substitution, 104
- Involution Theorem, 38
- Irredundant, 80, 90–92
 - partially redundant prime implicants, 90–92
 - relatively essential prime implicants, 90
 - totally redundant prime implicants, 90–92
- JK flip flop*, 165–167, 318–320
- Johnson counter, 310–313, 337–338
- Karnaugh map, 63–73, 92–95
 - complementary approach, 70–73
 - don’t care conditions, 63
- Kernels, 106–109
 - cokernel cube matrix, 107
 - rectangular covering problem, 107
- Latches, 159–162
 - gated, 160
 - reset input, 159
 - set input, 159
 - SR latch, 159
 - transparent, 160
- Leaves, 36
- Level sensitive device, 315
- Lexical elements, in VHDL descriptions, 185–187
- LFSR. *See* Linear feedback shift registers
- Linear feedback shift registers (LFSR), 329–332
 - maximal length sequence, 329
- Literal, 79
- Literal, cube, 79
- Logic circuit design, 117–125
 - multiplexers, 117–122
- Logic circuits, heuristic minimization of, 85–95
 - ESPRESSO, 91, 92–95
 - EXPAND, 85–88
 - IRREDUNDANT, 90–92
 - REDUCE, 88–90
- Logic design circuit
 - decoders, 123–125
 - demultiplexers, 123–125
- Logic design
 - combinational, 59–150
 - sequential, 59
- Logic gates, 48–53
 - AND, 48
 - exclusive-NOR, 51
 - exclusive-OR, 51
 - NAND, 51
 - NOR, 51
 - NOT, 50–51
 - OR, 48–49
 - truth tables, 48–53
- Logic hazards, 393–396
 - dynamic, 395–396
 - static, 393–394
 - three-valued, 394
- Logic operations, in VHDL (VHSIC hardware description language), 189
- Loop statement, 337
- Loops, in combinational logic design, 225–230
- Majority voter circuit, 196
 - alternate description of, 200
 - VHDL description of, 197
- Mantissa, 19

- Maximal length sequence, 329
- Mealy machine, VHDL and, 345–351
- Mealy models, 172–175
- Mealy type state machines, 341–342
- Metastable state, 162
- Minimal, 80
- Minimization, Boolean expressions and, 60–63
- Minimized two-level representation, 103
 - decomposition, 103
 - extraction, 103–104
 - factoring, 103, 105
 - substitution, 103, 104
- Minterm, 45, 80
- Minuend, 6
- Moore models, 172–175
- Moore type state machines, 338–341
- M -out-of- n code, 271–273
- Multilevel circuits, minimization of, don't cares, 109–114
- Multilevel logic design, 102–109
 - algebraic division, 105
 - Boolean division, 105–106
 - kernels, 106–109
 - minimized two-level representation, 103
- Multiple architectural description, of VHDL (VHSIC hardware description language), 194–196
- Multiple output functions, minimization of, 95–98
- Multiplexers, 117–122
 - control inputs, 122
- Multiplication, 138–140
- Multiplying operators, in VHDL (VHSIC hardware description language), 191–192

- NAND gate, 51
 - entity and, 183
- NAND–NAND logic, 98–101
 - double rail inputs, 99
 - single rail inputs, 100
- Nodes, 35
- Nonbinary counter, 302
- Noncritical races, 380
- Nondirected graph, 35
- Nonweighted codes
 - cyclic code, 23
 - excess-3, 22
- NOR gate, 51
- Normalization, 19
- NOR–NOR logic, 101–102

- NOT gate, 50–51
- Null partitions, 35
- Number systems, 1–24
 - base, 1
 - decimal numbers, 1–2
 - floating point, 19
 - radix, 1
 - signed, 13–19
- Numbers, as lexical elements, 186

- Observability don't cares, 110, 112–114
 - Boolean difference, 112–113
- Octal numbers, 8–11
- Octal, binary to, conversion, 10
- 1 hot encoding, 355–356
- 1's complement, 14
- Operators, in VHDL (VHSIC hardware description language), 189–192
- OR gate, 48–49
- Out degree, 36
- Output functions, 387–389
- Overflow, 14, 18

- PAL, 142, 146–150
 - devices, sequential, 273–286
- PAL22V10 device, 275–277
- Parity bit, 115
- Partially redundant prime implicants, 90–92
- Partitioning approach, 239–242
 - equivalence partition, 241
- Partitions, 34–35
 - blocks, 34
 - equivalence classes, 35
 - null, 35
 - unity, 35
- Path, cycle, 35
- PLA (programmable logic array), 142, 144–146
- PLD (programmable logic devices),
 - crosspoints, 141
- PLD, PAL, 142, 146–150
- PLD, PLA, 142, 144–146
- PLD, PROM, 142–143
- Port, definition of, 183
- Positional association, definition of, 198
- Positional cube notation, 81
- Power sets, 30
- Preset, 322
- Primary signal, synchronous sequential circuits and, 157

- Prime implicants, 74
 - chart, 76–77
 - essential, 74
 - partially redundant, 90–92
 - relatively essential, 90–92
 - totally redundant, 90–92
- Prime, 80
- Primitive flow table, 376, 377–378
 - reduction of, 377–379
- Process statement, 315
- Product, 41
- Programmable inverter, 116
- Programmable logic devices (PLD), 141–150
- PROM (programmable read only memory), 142–143
- Propagation delay, 168
 - fall, 168
 - rise, 168
- Quine–McCluskey method, 73–79
 - don't care conditions, 78–79
 - prime implicant, 74
- Races, 379–381
 - critical, 380
 - noncritical, 380
- Radix, 1
 - complement, 16–19
 - 2's complement, 17
 - overflow, 18
 - sign-extended, 19
- Rectangle, 108
- Rectangular cover, 108
 - problem, 107
- REDUCE, 88–90
- Reduced dependency, 262
- Reed–Muller canonical form, 116–117
- Reflected code (Gray code), 23–25
- Register transfer level (RTL) description, 181
- Registers, 322–324
 - barrel shifter, 327–328
 - linear feedback, 329–332
 - shift, 324–332
 - universal shift, 327
- Relational operators, in VHDL (VHSIC hardware description language), 189–190
- Relations, 32–34
 - Cartesian products, 32
- Relatively essential prime implicants, 90
- Reserved words, in VHDL (VHSIC hardware description language), 192
- Reset input, 159
- Resets, 320–321
- Ring counters, 307–309, 336–337
 - circulating shift register, 307
 - loop statement, 337
- Ripple adder, 128
- Ripple asynchronous counter design, 291–294
- Rise delay, 168
- RTL (register transfer level) description, 181, 200–202
 - concept of, 200
- Satisfiability don't cares (SDCs), 110–112
 - SDCs. *See* Satisfiability don't cares
- Secondary signal
 - excitation variables, 158
 - present state and, 157
 - synchronous sequential circuits and, 157
- Selected conditional signal assignment, in combinational logic design, 211–214
- Self complementing codes, 21
- Self-starting counter, 299
- Sequential assignment statements
 - combinational logic design, 214–224
 - case statement, 220–223
 - if versus case statements, 223–224
 - if-then statements, 216–220
 - process, 214–216
- Sequential circuit design, VHDL, 315–368
- Sequential logic design, 59
- Sequential machine. *See* synchronous sequential circuits
- Sequential PAL devices, 273–286
 - complex PLDs, 278
 - EPLDs, 278–285
 - PAL22V10, 275–277
- Sequential statements, in VHDL (VHSIC hardware description language), 192–194
- Set input, 159
- Set-reset latch. *See* SR latch
- Sets, 30
 - definition of, 30
 - disjoint, 31
 - empty, 30
 - intersection, 31
 - power, 30
 - properties of, 31
 - absorption law, 31
 - associative law, 31
 - commutative law, 31

- Sets (*Continued*)
 - DeMorgan's law, 32
 - distributive law, 31
 - idempotent law, 32
 - singleton, 30
 - union, 30
- Setup time, 162
- Shannon's expansion, 83, 84–85
 - complementation, 84–85
- Shared product, determination of, 95
- Shift and rotate operator
 - functions, 190
- Shift operators, in VHDL (VHSIC hardware description language), 190–191
- Shift register counters, 302–307
 - nonbinary, 302
 - state sequence tree, 302
- Shift registers, 324–332
 - bidirectional, 326
- Sign magnitude representation, 13–14
 - complement, 14
 - overflow, 14
- Signals as wires, example of, 207
- Signed binary numbers, 14
- Signed numbers, 13–19
 - diminished radix complement, 14–16
 - radix complement, 16–19
 - sign-magnitude representation, 13–14
- Sign-extended, 19
- Simulation results, 8-to-1 multiplexer, 212
- Single rail inputs, 100
- Singleton, 30
- Size, 80
- SR latch (set-reset latch), 159
- State assignment, 235, 249, 257–273, 379–387
 - code, 268–270
 - critical race free state assignment, 381–386
 - decomposition, 261–265
 - fan out oriented algorithm, 265–267
 - fan-in oriented algorithm, 265, 267–271
 - m*-out-of-*n* code, 271–273
 - number of, 260
 - races and cycles, 379–381
- State diagram, 170–172
 - state transition graph, 171–172
- State machines, 338–356
 - enumerated types and, VHDL codes, 342–345
 - Mealy-type, 341–342
 - Moore-type, 338–341
 - 1-hot encoding, 355
 - user defined state encoding, 351–355
- State minimization, 235, 239–244
 - implication table, 242
 - incompletely specified sequential circuits, 244–249
 - partitioning approach, 239–242
- State sequence tree, 302
- State tables, 170–172
- State transition graph, 171–172
- Static logic hazards, 393–395
- Strings, as lexical elements, 186
- Structural description, definition of, 196
- Substitution process, 103, 104
 - Boolean, 104
 - inverse operation of, 104
 - collapsing, 103, 104
 - flattening, 104
- Substitution property, 262
- Subtrahend, 6
- Sum of products, 43
- Sum, 41
- Supercube, 82
- Switching algebra, 40
- Symbols, Boolean functions and, 41
- Symmetric, 33
- Synchronizing pulse, clock, 158
- Synchronous counter design, 295–300
 - self-starting counter, 299
- Synchronous logic circuits, 158–159
 - asynchronous operation, 158–159
 - flip-flops, 162–168
 - latches, 159–162
 - synchronizing pulse, 158
- Synchronous preset, 322
- Synchronous reset, 320
- Synchronous sequential circuit design, 235–290
 - flip-flop next state expressions, 249–257
 - intended behavior, 235
 - PAL devices, 273–286
 - problem specification, 236–239
 - specifications of, 235
 - state assignment, 235, 249, 257–273
 - state minimization of, 235, 239
- Synchronous sequential circuits, 157–177
 - analysis, 175–177
 - control equations, 175–176
 - transition table, 176–177
 - clocked sequential circuits, 158
 - finite state machine, 157
 - Mealy models, 172–175

- Moore models, 172–175
- primary signal, 157
- secondary, 157
- sequential machine, 157
- state
 - diagram, 170–172
 - tables, 170–172
- synchronous logic circuits, 158–159
- timing, 168–170
- propagation delay, 168

- T* flip flops, 167–168, 318–319
- Tautology, 82–84
 - cofactors, 82
 - Shannon’s expansion, 83
- Three valued logic hazards, 394
- Timing, synchronous sequential circuits
 - and, 168–170
- Transaction table, 250
- Transition table, 176–177
- Transitive, 34
- Transmission gates, 405–406
- Transparent latches
- Tree, 36
 - branches, 36
 - leaves, 36
- Truth table, 48–53, 60
 - full adders and, 127
- Two level representation, minimized, 103
- Two valued Boolean algebra, 40
 - switching, 40
- Two’s complement, 17
 - subtractors, 135–137
- Two-input NAND gate and entity, 183

- Union, 30
- Unity partitions, 35
- Universal shift registers, 327
- Unsigned binary numbers, 14
- Up-down asynchronous counter design, 294–295
- User defined state encoding, 351–355

- Variables, 41
- Vertices, 35
- Very high speed integrated circuit (VHSIC), 181
- VHDL (VHSIC hardware description
 - language), 181–204
 - addition operators, 192
 - architecture, 184
 - description, 194–196
 - behavioral, 181
 - description, 199
 - functional simulation, 199
 - combinational logic design
 - See* Combinational logic design
 - concurrent and sequential statements, 192–194
 - data types, 187–189
 - bit, 187
 - boolean, 187
 - enumerated types, 188
 - definition of, 181
 - development of, 181
 - division operators, 192
 - entity, 182–183
 - lexical elements, 185–187
 - characters, 186
 - comments, 186
 - data objects, 186–187
 - numbers, 186
 - strings, 186
 - logic operators, 189
 - miscellaneous operators, 192
 - multiple architecture description, 194–196
 - multiplying operators, 191–192
 - operators, 189–192
 - relational operators, 189–190
 - reserved words, 192
 - RTL description, 200–202
 - shift operators, 190–191
 - structural description, 196
- VHDL circuit models, 315
 - case studies, 356–368
 - counters, 332–338
 - D* latch, 315–316
 - flip flops, 316–324
 - process statement, 315
 - registers, 322–324
 - shift registers, 324–332
 - state machines, 338–356
 - Mealy machine and, 345–351
- VHSIC (very high speed integrated circuit), 181
- VHSIC hardware description language (VHDL). *See* VHDL

- Weighted codes, 20–22
 - BCD code, 20
 - self-complementing, 21
- While loops, 229–230

