

Index

- \$readmemb – system function 237
- \$readmemh – system function 237
- \$signed() – system function 178
- \$unsigned() – system function 178
- @(event_expression) statement 220

- Active low output signals 14, 64, 65
- Address activated FSM control 291
- Address activated FSM control state diagram 292
- always** block 199
- Arithmetic right-shift (\gg) 178
- Arrays of **reg**'s 231
- assign** parallel statement 148
- Asynchronous FSM 267
 - important note 280
- Asynchronous FSM to relay circuit 296, 299
- Asynchronous FSM using more than two event cells 305
- Asynchronous Petri net structure 317
- Asynchronous receiver block diagram 90, 94, 331, 359
- Asynchronous receiver complete system simulation 358–365
- Asynchronous receiver Protocol 89, 93
- Asynchronous reset 54, 58, 221
- Asynchronous serial receiver 88
- Asynchronous serial receiver block diagram 90, 97, 359

- Asynchronous serial receiver simulation 96, 365
- Asynchronous serial receiver state diagram 91
- Asynchronous serial receiver with parity 92
- Asynchronous serial receiver with parity state diagram 95
- Asynchronous serial transmitter 95
- Asynchronous serial transmitter simulation 99
- Asynchronous state diagrams without inputs along transitions 290
- Asynchronous transmitter 95
- Asynchronous transmitter state diagram 98

- Base (number) 169
- Basic logic gate symbols 337
- begin..end** block 199
- Behavioural Asynchronous (event) FSM development 379
- Bi-directional port 178
- Binary counters using D type flip flops 355
- Binary counters using D type flip flops - generic parallel inputs equation 353, 354
- Binary counters using D type flip flops - with parallel inputs 353, 354
- Binary counters using T type flip flops 347, 349
- Binary counters using T type flip flops generic parallel inputs equation 352
- Binary counters using T type flip flops with parallel inputs 353
- Binary data serial transmitter 83

- Binary sequence detector (4 bit) 136
- Binary sequence detector (8 bit)
 - programmable 138–143
- Bit-range 151
- Bit-selection 156
- Bit-wise logical operators 178
- Blocking Assignment 206
- Boolean 170
- Boolean algebra laws 337
 - and rules 339
 - associative law 340
 - auxiliary law 341, 346
 - auxiliary law Proof 341
 - communicative law 340
 - consensus theorem 342
 - De Morgans theorem 343
 - De Morgans theorem - converting AND-OR to NAND 345
 - De Morgans theorem - converting AND-OR to NOR 345
 - distributive law 340
 - exclusive NOR 338
 - exclusive OR 338
 - logical adjacency rule 345, 346
 - or rules 339
- Buses 150
- Byte wide binary code detector 139, 140
- Byte wide binary code detector simulation 142, 143
- Byte wide binary code detector state diagram 141
- case..endcase** statement 228
- Case-equality 184, 216
- casex** statement 249
- Class C type FSM 6
- Clock circuit for use with FSM systems 355
- Clocked FSM 2
- Clocked watchdog timer FSM 100
- Clocked watchdog timer FSM simulation 102
- Clocked watchdog timer FSM state diagram 101
- Clothes spinner 304
 - block diagram 304
 - equations 305
 - gate level simulations 309
 - simulation using equations 308
 - state diagram 305
 - test bench module 307
 - verilog module 306
- Combinational Logic (using sequential block) 209
- Comma separated event expression 203
- Comments (Verilog) 150
- Compilation 161
- Compiler directive- timescale 159
- Concatenation 180
- Concatenation operator 184
- Conditional operator (?:) 175
- Continuous assignment 148
- Controlling an Analogue to Digital Converter (ADC) 26, 33, 73, 111
- Controlling a Digital to Analogue Converter (DAC) 76, 117
- Counter design using don't care states 355–357
- D type flip flops 47–65
- D type flip flop equations 47
 - 0 to 1 transitions 49
 - 1 to 0 transitions with leaving terms
 - Rule 1 49, 51
 - 1 to 0 transitions without leaving terms 51
 - 1 to 1 transitions Rule 2 50, 51
- D type flip flop two way branches Rule 3 50–53
- Data acquisition system FSM 110
- Data acquisition system FSM simulation 113
- Data acquisition system FSM state diagram 112
- Dataflow style 148
- Dealing with unused states 69
- Default assignment 210
- default** branch (case) 228
- Define compiler directive 242
- Delayed sequential assignment (#) 201
- Delta delay 149
- Detecting binary sequences without memory 134
- Dice game 79
- Dice game simulation 82, 83
- Dice game state diagram 81
- Divide by 11 counter design 335, 362
- Dynamic Memory Access (DMA) 127
- Dynamic Memory Access (DMA) Block Diagram 128, 129
- Dynamic Memory Access FSM simulation 132
- Dynamic Memory Access FSM state diagram 130
- endmodule** 147
- Equality operators (Verilog) 182
- Event cell 269
- Event cell characteristic equation tests 271

- Event cell derivation 270
- Event driven FSM (see Asynchronous FSM) 267
- Event driven FSM to relay circuit 299
- Event driven single pulse with memory FSM 277
- Event driven single pulse with memory FSM Circuit 278
- Event driven state diagrams without inputs along transitions 290
- Event expression (Verilog) 202
- Exclusive-OR 40, 92, 174, 338
- Explicit association (Verilog) 153
- External Timer Unit 23–26

- for** loop 160, 214
- forever** loop 200
- Four-valued Logic 168

- Gate-level module 172, 306

- Handshaking mechanisms 23, 90, 98, 114, 117, 120, 133, 135, 291, 331
- Hierarchical design (Verilog) 152
- Hover Mower FSM 285
- Hover Mower circuit 287
- Hover Mower simulation 289
- Hover Mower state diagram 286
- Hover Mower Verilog code 287

- if..else** statement 210
- Incomplete assignment 210
- Inertial delay 172
- Inferred latch 210
- Infinite loop 204
- inout** port (read/write memory) 232
- Instantiation (of modules) 152
- integer** 160
- intial** block 199, 220

- Literal values (Verilog) 169
- Local parameters (**localparam**) 186
- Logic Synthesis 146
- Logical-AND 174

- Mealy active low outputs (with examples of use) 65, 73, 74, 113, 116, 118, 120, 125, 126, 127, 131, 303
- Mealy FSM (Verilog) 381
- Mealy outputs effect of clock and other signal delays 17
- Mealy type FSM 4, 15, 241
- Mealy type outputs 16, 295
- Memory chip tester 123
- Memory chip tester Block Diagram 124
- Memory chip tester state diagram 125
- Memory cycle (device) timing 28
- Memory device control 29–34
- Memory device control of chip select and read 28–31
- Memory device control of chip select and write 28–31
- Memory device controlled by an FSM 28
- Memory Device waveforms 29
- Meta-logical values (Verilog) 168
- Microprocessor control for waveform synthesiser 120
- Microprocessor control of DMA system 132
- module** 147
- Module header 147
- Module instantiation statements 153
- Module ports 147
- Monitoring input for changes 35
- Moore FSM (Verilog) 383
- Moore type active low outputs 14
- Moore type FSM 5, 241
- Moore type outputs 5, 16, 314
- Motor controller FSM with fault monitoring 281
 - circuit 282
 - simulation 285
 - diagram 281
 - verilog code 283
- Multi way branches in state diagrams 61–63
- Multi-bit ports 150
- Multiply and Divide operators (Verilog) 176

- Named sequential block 213
- NAND sequential equations 271
- negedge** event qualifier 217
- Non-Blocking Assignment 206
- Null statement (;) 228

- One Hot method 105–143
- One Hot method - dealing with two way branches 108
- One Hot method - schematic circuit arrangement 107
- One Hot method to produce flip flop equations 105–110
- One hot single pulse detector 105, 106
- Operators (Verilog) 172

- or event expression 202, 203
- Override (a parameter default value) 229
- Parallel statements (Verilog) 147
- parameter** 214
- parameter** to set size 226
- Parity detector 92
- Parity for error detection 92
- Petri nets 313
- Petri net arc's 313
- Petri net asynchronous receiver 329–335
- Petri net asynchronous receiver petri net diagram 332
- Petri net asynchronous receiver example Details of sequence 335
- Petri net asynchronous receiver example simulation 334
- Petri net based asynchronous serial receiver 329–336
- Petri net circuits 316, 319
- Petri net comparison with state diagram 314
- Petri net diagram fork 319, 321
- Petri net diagram join 319
- Petri net disabling arcs 325, 326
- Petri net enabling arcs 325
- Petri net equations 314, 315
- Petri net full cycle of design 316
- Petri net outputs 316, 319, 322, 328, 333
- Petri net parallel controllers 319–323
- Petri net placeholder equations 314
- Petri net placeholders 313
- Petri net serial controllers 318
- Petri net shared resource example 327
- Petri net shared resource example simulation 329
- Petri net synchronisation between parallel nets 324–327
- Petri net Tokens 313, 320
- Petri net transition equations 314, 315
- Petri net transition equations with disabling arcs 326
- Petri net transition equations with enabling arcs 325
- Petri net transitions 313
- posedge** event qualifier 217
- Positional association 154
- Primary and secondary signal gate tolerances 301
- Primary inputs 8
- Primary outputs 8
- Primitive Gates 170
- Propagation delays 170, 343
- Race conditions in event FSM's 299
- Race conditions in event FSM's - between primary and secondary variables 299, 300
- Race conditions in event FSM's - between primary inputs 299, 300
- Race conditions in event FSM's - between secondary state variables 299, 300
- Race conditions in event FSM's - gate delay tolerance 301
- Raise-to-the-power operator (**) 217
- Reduction NOR 175
- reg** 147, 151
- Register types (Verilog) 164
- Relational operators (Verilog) 181
- repeat** loop 201
- Replication operator 184
- RTL (Register Transfer Level) 145
- Rules (Module Port connectivity) 154
- Samples per waveform 78
- Sampling frequency 78
- Scalability (using parameters) 226
- Secondary state variables 11
- Secondary state variables non unit distance coding 11
- Secondary state variables unit distance coding 12
- Sequential block 198
- Sequential equation for relay implementation 297
- Sequential equations 271, 272
- Sequential equations dropped terms 276
- Sequential equations for PLD implementation 272, 276
- Sequential equations logical adjacency reduction 274, 346
- Sequential equations NAND form 271, 272
- Sequential equations NOR form 271, 272
- Sequential equations short cut rule 275
- Sequential statements (Verilog) 198
- Serial Asynchronous protocol 89, 93
- Serial transmitter 95
- Serial transmitter simulation 99
- Serial transmitter state diagram 98
- Shared memory 114
- Shared memory FSM Block Diagram 114
- Shared memory FSM state diagram 115
- Shift operators (Verilog) 175
- Shift register(s) 357
- Shift register eleven bit design 360, 361
- Shift register empty detection 96, 362

- Shift register equations 88, 357
- Shift register four bit with parallel inputs 358
- Shift registers with parallel loading input equations 357
- Signal delay in logic gates 343, 344
- signed** qualifier (Verilog) 165
- Simple binary up counter 349
- Simple binary up counter simulation 352
- Simulation cycles 149
- Slings 10
- State assignment (Verilog) 241
- State maps for counter design 348, 355, 356
- supply0**, **supply1** nets 163
- Synchronous FSM 2
- Synchronous counter design 347–352
- Synchronous reset 54, 8
- System task - \$stop 160
- SystemC 146
- SystemVerilog 146

- T type equations 41–46
- T type flip flops 40
- Tank water level control final state diagram solution 295
- Tank water level control first state diagram solution 294
- Tank water level control system 293
- Test-fixture 155
- Test-module 155, 372

- Timescale compiler directive 159, 170
- Timing waveforms 162
- Traditional FSM design 67
- tri** 164
- Twisted ring counter design 356
- Two way branches Caution in using 309
- Two way branches in state diagrams Rule 3 50–53
- Two's complement 175

- Unconnected port 154
- Unsigned 176
- Unused states (Verilog FSM) 244

- Verilog - Extreme Simulator 367, 375, 376
- Verilog - simulators Tutorial 367
- Verilog-2001 - simulators 161
- Verilog-HDL 145, 367
- VHDL 146

- Wait state generator for microprocessor 301
- Waveform generator 76
- Waveform generator state diagram 77
- Waveform synthesiser 116
- Waveform synthesiser state diagram 118
- Waveform synthesiser control via a C program 120
- Wildcard event expression 203
- wire** 147

