Index

# character in IS, 132
# character in VASM, 84, 89, 147
# Verilog delay operator, 22
$display task, 21, 67, 68
$finish task, 67, 68
$readmemh statement, 65, 88, 91
$stop task, 68
$time function, 10
$write task, 68
– Verilog operator, 14
! = Verilog operator, 15
!= Verilog operator, 15
!= = Verilog operator, 15
% Verilog operator, 14
& Verilog operator, 14
& & Verilog operator, 15
* Verilog operator, 14
* Verilog operator, 15, 42
[] Verilog operator, 15–16
[] ] Verilog operator, 15
= Verilog operator, 15
~ Verilog operator, 15
+ Verilog operator, 14
< Verilog operator, 15
<< Verilog operator, 15, 60
<= Verilog operator, 15, 23
== Verilog operator, 15
=== = Verilog operator, 15
> Verilog operator, 15
>= Verilog operator, 15
>> Verilog operator, 15
/ Verilog operator, 14
. + verilog operator, 89, 148
. byte directive, 89, 148
. equ directive, 89–90, 148
 . org directive, 86, 148
 . word directive, 89, 148
 // VASM chars, 89, 147
 ; character in VASM, 148
 ; character in ISA, 132
 @ Verilog keyword, 22
' character in Verilog, 63
' define Verilog statement, 18, 62
' else Verilog statement, 81
' ifdef ' - ' endif ' Verilog construct, 80
' undef Verilog directive, 63
' Verilog symbol, 18, 63
0x characters in VASM, 147
absolute addressing, 40
abstraction gap, 1, 2
accumulator, 36
ADD instruction in ISA, 36, 43, 50, 62,
134–135
add operator, Verilog, 14
addressing modes, 35–36, 40, 46, 51, 52
see also byte-addressable memory
and indexed addressing and intermediate addressing
and PC-relative addressing and
register addressing
algorithmic behavioral model
definition of, 58
for ISA testing, 119
for medieval knight, 19–20
ALU, 41, 105
dependencies, 107
always block / statement, 20, 22
AND instruction in ISA, 39, 50, 62, 135–136
anti-dependencies, 99
architected state, 78
architectural design level, 2
arithmetic carry out, 38–39
arithmetic instructions in ISA, 34, 35–39, 51–52
decoding, 111
execution, 74–75
arithmetic overflow, 37–38
arithmetic Verilog operators, 14
arrays, Verilog, 12
assemblers, 83–84
two-pass, 85–88
see also assembly language and VASM
assembly language
definition, 82
format, 84–85
uses, 82–83
see also assemblers and VASM
assign keyword, 26
assignment statements, 22–24
backquote char, 63
basic block, 98
BCC instruction in ISA, 43, 44, 50, 63, 136–137
BCS instruction in ISA, 43, 44, 50, 63, 136–137
behavioral description, medieval knight
FSM, 19–20
behavioral design level, 3, 4, 16
behavioral model see algorithmic behavioral model
BEQ instruction in ISA, 43, 44, 50, 63, 136–137
BGE instruction in ISA, 44, 50, 63, 136–137
BGT instruction in ISA, 44, 50, 63, 136–137
big-endian machine, 69–70
binary numbers in Verilog, 13–14, 63
bitwise operators in Verilog, 14–15
BLE instruction in ISA, 44, 50, 63, 136–137
blocking assignments, 22–23, 24
BLT instruction in ISA, 44, 50, 63, 136–137
BMI instruction in ISA, 43, 44, 50, 64, 136–137
BNE instruction in ISA, 44, 50, 63, 136–137
BIV instruction in ISA, 44, 50, 63, 136–137
Boolean operators in Verilog, 14–15
BPL instruction in ISA, 44, 50, 63, 136–137
Branch instructions, 34, 40, 43–44
decoding, 112
dependencies, 108
branch penalty, 100
bubbles, pipeline, 100, 103
BVC instruction in ISA, 43, 44, 50, 63, 136–137
BVS instruction in ISA, 43, 44, 50, 63, 136–137
Bxx instructions, 75, 136–137
byte-addressable memory, 48–49, 69
C bit, 41, 61, 62, 112, 113
C flag see C bit
caches, 116
carry out arithmetic, 38–39
case statement, 19–20, 73–74
casex statement, 20–21
casez statement, 20–21
CC register, 113
CC4 bit, 109
echeek function, 77–79
checker module, 119, 122, 123
clock frequency, 115
clock period, 115–116
CMP instruction in ISA, 42–43, 50, 52, 62, 75, 137–138
code generation, in VASM, 152
codegen.c, e file, 152
comments, in VASM, 85, 89, 147
compare-and-branch instructions, 40 see also CMP instruction
compilation, software vs. hardware, 5
cmpiler directives (Verilog), 63
compilers, high-level language, 33, 36, 82–83
completeness, logical, 39
concatenation operator, in Verilog, 15–16, 70
conditional assignments and transfers, 23
condition codes, 41–43, 77–79, 133
see also C bit and N bit and V bit and Z bit
conditional branching in ISA, 34, 43–44
conditional operator, in Verilog, 16
constants, in Verilog, 12–13
constrained pseudorandom test vectors, 126
corner cases, 120
critical path, in pipeline, 115
d-caches, 103, 116
data caches, 103, 116
data dependencies, 99
data generator module, 120–121
data transfer instructions, 34, 45–47, 54–56, 112
dataflow level description, 26
De Morgan’s Theorem, 39
deassign statement, 26
decimal numbers, in Verilog, 13–14, 63
decoding see instruction decode stage (ID)
derpazar statement, 12
delay slot, 101
delayed branches, 100
delays critical path, 115
modelling, 22, 28
dependencies, 98
control, 98–99
data, 99, 106–107
see also hazards
design levels, 1–3
direct addressing, 46
directed testing, 118
directives, in Verilog, 63
divide Verilog operator, 14
division in ISA, 37
dnilian-ness, 69–70
endmodule, 9
escape chars, in Verilog, 68
execute task, 71–77
executing ISA instructions, 71–77
tracing, 79–81
execution pipeline stage (EX), 95, 96, 108–109, 113–114
external references, 91
fall delays, 28
fetch-execute cycle, 64, 127
fetch task, 68–70
field programmable gate array (FPGA), 9
finite state machines initialization, 127
medieval knight model, 17–26
pipeline model, 105
flip-flop module for, 30
flow dependencies, 99
flushing, 106
tor loops, in Verilog, 25, 81
forever loops, in Verilog, 24
forward verification, 130
formal specifications, in Verilog, 67
forward quote char, 63
forward referencing problem, 86
forwarding, 103, 104, 106
four-bit ripple-carry adder model, 8–9
full adder model, 7–8
functions, in Verilog, 28, 30, 69
functional design level, 2
gate types, 28
ground nets, 10
hardware compilation, 5
hardware description languages, vii, 6
hardware description level see structural design level
hazard detection unit, 106–108
hazards, 99–100
branch, 100–101, 103
data, 101–102
hazards (cont.)
structural, 102–103
see also dependencies
HDL see hardware description languages
hexadecimal numbers
in VASM, 147
in Verilog, 13–14, 63
hierarchical design flow, 1–3
HLT instruction, 47, 50, 56, 62, 76, 138–139
i-caches, 103, 116
identifiers, 86, 89, 148
immediate addressing, 51, 84
immediate operands, in VASM, 89, 147
immedX notation, 132
indexed addressing, 46, 112
indirect addressing, 46
initial statement, 20, 23, 65
input port type, 10, 11
instruction caches, 103, 116
instruction decode stage (ID), 95, 108, 109,
111–113
instruction execution, tracing, 79–81
instruction fetch stage (IF), 95, 108,
109–111
instruction formats, 48–50
instruction set architecture (ISA), 1–2,
32–56
arithmetic instructions, 34, 35–39, 51–52
decoding, 111
execution, 74–75
control instructions in ISA, 34, 39–44,
52–54
decoding, 112
execution, 75–76
data transfer instructions, 34, 45–47,
54–56, 112
design criteria, 32–33
full specification, 132–146
instruction formats, 48–50
logical instructions in ISA, 34, 39, 51–52
decoding, 111
execution, 74–75
see also memory and pipelines and
registers
integer data type, Verilog, 10, 12, 67
intermediate registers, 108–109
IR register
and condition codes, 77, 78, 79, 113
in pipeline, 109, 110, 113
ISA notation, 133
Verilog modelling, 61
IR2 register, 108, 109, 110, 111
IR3 register, 108, 109, 111, 113
IR4 register, 109, 113
IR5 register, 109
ISA see instruction set architecture
Java Virtual machine, 36
JMP instruction, 40, 54, 75–76, 139
JMPI instruction, 40, 54, 139
jump-with-offset operation, 54
LABEL notation, 132
labels, assembler, 84, 87, 89
LC see location counter
LD instruction, 46, 50, 55, 62, 76, 139–140
LDI instruction, 46, 50, 54–55, 62, 76, 140
LDX instruction, 46–47, 50, 55, 62, 76,
140–141
level-sensitive control, 22
levels of design, 1–3
lex tool, 90, 149
lexical analysis, 90, 150
link register, 40
linking programs, 92
literals, 89
see also immediate operands
little-endian machine, 69–70
load-store architecture, 45–46
load-store instructions, 46–47
loading programs, 92
location counter, 85, 89
logic level implementation, 26
logical instructions in ISA, 34, 39, 51–52
decoding, 111
execution, 74–75
logical operators, in Verilog, 14–15, 74–75
loops, in Verilog, 24–25
machine code, 82, 91
machine language, 82
mapping to ports, 10–11
MD3 register, 109, 113, 114
MD4 register, 109
medieval knight model, 17–18
behavioral model, 19–21
dataflow description, 27
logic level implementation, 26
state diagram, 18
structural description, 28, 29–30
Mem[x] notation, 132
memory
accessing, 45, 115–116
cache, 103
initial contents, 65
ISA definition, 133
Verilog modeling, 12, 60, 61
memory access pipeline stage (MEM), 95, 96, 109, 114–115
memory-to-memory architecture, 45–46
modules, in Verilog, 8–10
module operator, in Verilog, 14
MOV instruction, 47, 50, 56, 141, 149–152
multiplication instruction, 37
multiplication, in Verilog, 14
MUX, 105, 106, 110
N bit, 41, 42, 61, 112, 113
N flag 112, 113
name dependencies, 99
NAND instruction, 39
negative bit see N bit
negedge keyword, 21
net variables, 10
newlines, 148
nonblocking assignments, 23–24
NOP instruction, 47, 50, 56, 62, 76, 142
NOR instruction, 39
NOT instruction, 39, 50, 52, 62, 142
dependencies, 98–99
icekt, 103
models for, 105
number of stages, 116–117
performance, 96–97
stages, 95, 109–115
structure, 108–109
ports, 8, 10–11
posedge keyword, 21
power dissipation, 26
power nets, 10
print trace function, 79–81
program counter see PC
pseudo-instructions, 47
pseudorandom test vectors, 125–127
qvars.h file, 151
R[x] notation, 132
RAW dependencies and hazards, 106–108
read-after-write dependencies and hazards, 102, 106–108
rdat notation, 132
operations, in source code, 84
operators, in Verilog, 16–16
OR instruction in ISA, 39, 50, 62, 143
output dependencies, 99
see also register file access dependencies
output ports, 10, 11
overflow, arithmetic, 37–38
overflow bit see V bit
overridding of values, 12–13
parameter keyword/statement, 12, 59–60, 62
parsing, 86, 90, 150–151
PC2 register, 108, 109, 110
PC3 register, 109, 111, 113
PC4 register, 109, 113, 114, 115
PC-relative addressing, 40, 53, 91
physical design level, 2, 3
pipelines, 94–117
and bubbles, 106, 109
and condition codes, 41, 43
definition, 94
dependencies, 98–99
hazards, 99–104
interlock, 103
models for, 105
number of stages, 116–117
performance, 96–97
stages, 95, 109–115
structure, 108–109
ports, 8, 10–11
posedge keyword, 21
power dissipation, 26
power nets, 10
print trace function, 79–81
program counter see PC
pseudo-instructions, 47
pseudorandom test vectors, 125–127
qvars.h file, 151
R[x] notation, 132
RAW dependencies and hazards, 106–108
read-after-write dependencies and hazards, 102, 106–108
rdat notation, 132

real data type in Verilog, 10
realtime data type in Verilog, 10
reduction operators, 14–15
reg keyword/data type, 10, 12
register addressing, 45
register file access dependencies, 112
see also output dependencies
register file, 45, 112, 115
register identifiers, in VASM, 147
register variables, in Verilog, 10
registers
and condition codes, 77, 78, 79, 113
CC, 113
IR, 77, 78, 79, 109, 110, 113, 133
IR2, 108, 109, 110, 111
IR3, 108, 109, 111, 113
IR4, 109, 113
IR5, 109
ISA notation, 133
length of, 48
link, 40
MD3, 109, 113, 114
MD4, 109
modelling of, 60–62
number of, 49, 60
PC2, 108, 109, 110
PC3, 109, 111, 113
PC4, 109, 113, 114, 115
renaming, 99
result, 61–62
RUN, 61, 66
short-circuiting, 103
width, 60
X3, 108, 109, 112, 114
Y3, 108, 109, 112, 114
Z4, 109, 113, 114, 115
Z5, 109, 112, 113, 114, 115
register-transfer level of design, 2
relational operators, Verilog, 15
repeat loops, Verilog, 24
replication operator, Verilog, 16
representation of numbers, Verilog, 13–14
reset state, 127
result register, 61–62
rise delays, 28
rs1, rs2 notation, 132
rst notation, 132
RTL see register-transfer level of design
RUN register, 61, 66
self-test programs, 118, 127–130
semicolon char, 85, 87, 147
setec task, 77
setcc(x) notation, 132
setxx notation, 132
shift operators, 15
sign extension, 52, 55, 112
signal edges, 22
sized numbers, 13–14, 63
software compilation, 5
source code, assembler, 82, 91
space chars, 148
ST instruction, 47, 50, 55, 62, 76, 143–144
stack-based machines, 36
stalling, 103, 106, 109, 110, 113
state encoding, 25–26
state transition diagrams, 16, 18
structural descriptions, 25–28, 29–30
structural design/ modelling, 16
structural design level, 2, 3, 4, 26
STX instruction, 47, 50, 55–56, 62, 76, 144
SUB instruction, 37, 42–43, 50, 62, 145
subtract Verilog operator, 14
supply0, 10
supply1, 10
symbol table, 86–87, 90
symbolic labels, 84, 87, 89, 148
symbolic names, 86, 89, 148
syntactical analysis, 90
system-level testing, 127–130
initial functionality, 127–129
structure, 129–130
tab characters, 148
target languages, 83
tasks, in Verilog, 28, 30, 67, 68
test benches, 118, 119, 131
test vectors
automatic, 122–127
examples, 121–122
manually generated, 119–122
number of, 122
testbench setcc module, 120, 122
time data type, Verilog, 10, 12
timing
and behavioural model, 58
and pipelining, 115–117
modelling of, 21–22
tokens, 86, 90, 150
tracing execution, 79–81
triand keyword, 10
trioifes keyword, 10
tristable nets, 10
two-address machines, 36
unconditional branching, 34
un-sized numbers, 13–14
V bit, 41–42, 61, 112, 113
V flag see V bit
v.out file, 65
VASM, 88–91, 147–152
directives, 89, 148
example program, 85, 149
modifying, 140–152
syntax, 88–90, 147–148
see also assembly language and assemblers
vasm.lex file, 150
vasm.yacc file, 150, 151
vectors
in Verilog, 8, 11
test, 122–127
verification, 118
component-level, 118–127
formal, 130
system-level, 127–130
Verilog, via, 6
$time function, 10
arrays, 11–12
assignments, 22–24
behavioral modelling, 16–17, 18–25
compiler directives, 63
constants, 12–13
defparameter statement, 12
escape characters, 68
format specifications, 67
functions and tasks, 28–30, 67, 68, 69
integer variables, 10, 12, 67
loops, 24–25, 81
memory modelling, 60, 61
modules, 8, 9
net variables, 10
number formats, 13–14, 63
operators, 14, 70, 74–75
parameter statement, 12
ports, 8, 9, 10–11
real variables, 10
realtime data type, 10
register modelling, 61–66
register variables, 10, 12
structural modelling, 16–17, 25–28
time data type, 10, 12
timing controls, 21
vectors, 8, 11–12
wire variables, 8
Very Small Processor Architecture, 1
top-level definition, 59
VeSPA see Very Small Processor Architecture
VHDL, vii
wait statement, 22
wand keyword, 10
WAR see write-after-read hazards
WAW see write-after-write hazards
web site, viii
self-test programs, 130
structural model, 105
test benches, 127
while loops, 25
white-space chars, 148
wire data type/keyword, 8, 10
wir keyword, 10
write-after-read hazards, 102
write-after-write hazards, 102
write-back pipelining stage (WB), 95, 96, 109, 114–115
x value, 10, 15
X3 register, 108, 109, 112, 114
XOR instruction in ISA, 39, 50, 62, 146
Y3 register, 108, 109, 112, 114
yacc tool, 90, 149, 150–151

Z bit, 41–42, 61, 61, 112, 113
Z flag see Z bit

z value, 10, 15
Z4 register, 109, 113, 114, 115
Z5 register, 109, 112, 113, 114, 115
zero-address machines, 36
zero bit see Z bit