A

accumulator register, 361
accuracy, 5
active-low signals, 151, 247
ADC. See analog-to-digital converter
adders, 141
  full, 144
  half, 141
address decoder, 243
address decoding, 250
address latch, 333
address lines, 245
addressing
  immediate, 392
  pointer, 392
  register, 391
addressing modes, 391
AF. See auxiliary carry flag
aliasing, 31
ALU. See arithmetic logic unit
analog, 3, 26
analog-to-digital converter, 6, 259
AND gate, 72, 73, 74, 90, 109, 114, 153
AND rules, 97, 98
application layer, 303
arithmetic logic unit, 360
arithmetic overflow, 67
arrays, 383
assembler, 375
assembler directive, 378
assembly language, 338, 339, 344
  comment field, 378
  instruction field, 377
  label field, 376
  operand field, 377
Associative Law, 95
auxiliary carry flag, 364
AX. See accumulator register

B

base address, 367
base pointer, 362
base register, 361
Basic Input/Output System, 245, 248, 261
BCD. See Binary Coded Decimal
BCD addition, 64
BEDO. See Burst EDO
binary addition, 43, 141
Binary Coded Decimal, 36
binary conversion, 23, 67
binary pulse, 9
binary signals, 8
binary subtraction, 45
binary system, 7
BIOS. See Basic Input/Output System
bit, 20, 17
bitwise operations, 166
  AND, 167
  OR, 171
  XOR, 171
BIU. See bus interface unit
boolean algebra, 89
  laws of, 95
  simplification, 101
BP. See base pointer
buffer, 329
Burst EDO, 266
bus, 244, 325
bus contention, 246
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bus interface unit, BX. See base register</td>
<td>365</td>
</tr>
<tr>
<td>byte</td>
<td>20</td>
</tr>
<tr>
<td>cache</td>
<td></td>
</tr>
<tr>
<td>block</td>
<td>286, 290</td>
</tr>
<tr>
<td>direct mapping</td>
<td>290, 295</td>
</tr>
<tr>
<td>fully associative mapping</td>
<td>290, 295</td>
</tr>
<tr>
<td>hit</td>
<td>289</td>
</tr>
<tr>
<td>L1, L2</td>
<td>285</td>
</tr>
<tr>
<td>line</td>
<td>286</td>
</tr>
<tr>
<td>mapping function</td>
<td>290</td>
</tr>
<tr>
<td>miss</td>
<td>289</td>
</tr>
<tr>
<td>set associative mapping</td>
<td>290, 297</td>
</tr>
<tr>
<td>size</td>
<td>290</td>
</tr>
<tr>
<td>split</td>
<td>286</td>
</tr>
<tr>
<td>tag</td>
<td>286</td>
</tr>
<tr>
<td>write back policy</td>
<td>300</td>
</tr>
<tr>
<td>write policy</td>
<td>290, 299</td>
</tr>
<tr>
<td>write through policy</td>
<td>299</td>
</tr>
<tr>
<td>cache replacement algorithm</td>
<td>290, 295</td>
</tr>
<tr>
<td>First In First Out</td>
<td>296</td>
</tr>
<tr>
<td>Least Frequently Used</td>
<td>296</td>
</tr>
<tr>
<td>Least Recently Used</td>
<td>296, 298</td>
</tr>
<tr>
<td>Random</td>
<td>296</td>
</tr>
<tr>
<td>capacitor</td>
<td>262</td>
</tr>
<tr>
<td>carry flag</td>
<td>364</td>
</tr>
<tr>
<td>CAV. See constant angular velocity</td>
<td></td>
</tr>
<tr>
<td>central processing unit</td>
<td>332</td>
</tr>
<tr>
<td>CF. See carry flag</td>
<td></td>
</tr>
<tr>
<td>checksum</td>
<td>175</td>
</tr>
<tr>
<td>l's complement</td>
<td>177</td>
</tr>
<tr>
<td>2's complement</td>
<td>177</td>
</tr>
<tr>
<td>chip select</td>
<td>242, 246, 256</td>
</tr>
<tr>
<td>clock</td>
<td>210, 220</td>
</tr>
<tr>
<td>code segment</td>
<td>369</td>
</tr>
<tr>
<td>collisions</td>
<td>310</td>
</tr>
<tr>
<td>combinational logic</td>
<td>80, 92</td>
</tr>
<tr>
<td>Commutative Law</td>
<td>95</td>
</tr>
<tr>
<td>compiler</td>
<td>375</td>
</tr>
<tr>
<td>conditional branching</td>
<td>327, 388</td>
</tr>
<tr>
<td>configuration registers</td>
<td>333</td>
</tr>
<tr>
<td>constant angular velocity</td>
<td>281</td>
</tr>
<tr>
<td>constants</td>
<td>383</td>
</tr>
<tr>
<td>control lines</td>
<td>245</td>
</tr>
<tr>
<td>counter</td>
<td>213</td>
</tr>
<tr>
<td>counter register</td>
<td>362</td>
</tr>
<tr>
<td>CPU. See central processing unit</td>
<td></td>
</tr>
<tr>
<td>CRC. See cyclic redundancy check</td>
<td></td>
</tr>
<tr>
<td>crosstalk</td>
<td>305, 308</td>
</tr>
<tr>
<td>CS. See code segment</td>
<td></td>
</tr>
<tr>
<td>CX. See counter register</td>
<td></td>
</tr>
<tr>
<td>cyclic redundancy check</td>
<td>179</td>
</tr>
<tr>
<td>cylinder</td>
<td>281</td>
</tr>
<tr>
<td>data buffer</td>
<td>333</td>
</tr>
<tr>
<td>data lines</td>
<td>244</td>
</tr>
<tr>
<td>data register</td>
<td>362</td>
</tr>
<tr>
<td>data segment</td>
<td>369</td>
</tr>
<tr>
<td>datagrams</td>
<td>310</td>
</tr>
<tr>
<td>datalink layer</td>
<td>304, 306, 308</td>
</tr>
<tr>
<td>datasum</td>
<td>175</td>
</tr>
<tr>
<td>DDR SDRAM. See Double Data Rate SDRAM</td>
<td></td>
</tr>
<tr>
<td>decode cycle</td>
<td>346, 370</td>
</tr>
<tr>
<td>decoders</td>
<td>154</td>
</tr>
<tr>
<td>DeMorgan's Theorem</td>
<td>104, 110, 119</td>
</tr>
</tbody>
</table>
demultiplexers, 157
destination index, 363
DF. See direction flag
DI. See destination index
digital signal processing, 7
direct memory access, 356
direction flag, 363
directive. See assembler directive
Distributive Law, 96
divide-by-two circuit, 212
DMA. See direct memory access
don't cares, 137
Double Data Rate SDRAM, 267
double word, 20
DRAM. See Dynamic RAM
DS. See data segment
DSP. See digital signal processing
duty cycle, 13
DX. See data register
dynamic RAM, 262

EDO. See Extended Data-Out
encoding, 39
endian, big/little, 345
ES. See extra segment
Ethernet frame, 308
CRC, 309
data, 309
destination address, 309
filler bytes, 309
length, 309
preamble, 308
source address, 309
start delimiter, 309

EU. See execution unit
exclusive-OR gate, 74, 142
execute cycle, 346, 370
execution unit, 360
Extended Data-Out, 266
extra segment, 369

F
falling edge, 9, 203
Fast Page Mode, 265
fetch cycle, 346, 370
flags, 327, 360
floating-point, 58
formatting, 283
FPM. See Fast Page Mode frame, 306
frequency, 12
frequency modulation, 272

G
Gray code, 39

H
Hamming Code, 188
header, 306
hexadecimal, 35
hexadecimal addition, 61
http, 313

I
IC. See integrated circuits
ICANN. See Internet Corporation for Assigned Names and Numbers
IEEE Std-754, 58
IEEE Std-802.3, 321, 304, 308
IEEE. See Institute of Electrical and Electronics Engineers
IF. See interrupt flag
Institute of Electrical and Electronics Engineers, 321

instruction pointer, 344, 362
instruction queue, 370
integrated circuits, 159

Intel assembly
ADC, 386
ADD, 386
AND, 386
CALL, 389
clearing bits, 390
CMP, 389
DEC, 386
DIV, 386
IN, 385
INC, 386
INT, 391
IRET, 391
JMP, 387
Jxx, 388
LOOP, 389
MOV, 385, 386
MUL, 386
NEG, 386
NOP, 391
NOT, 386
OR, 386
OUT, 385
PULL, 390
PUSH, 390
RET, 389
SAL, 387
SAR, 387
setting bits, 390
SHL/SHR, 387
SUB, 386
XOR, 386

Intel directives
.CODE, 380, 394

.DATA, 380, 394
.MODEL, 380
.STACK, 380
DB, DW, DD, DQ, 382
DUP, 383
END, 382, 394
EQU, 383
PROC, 381, 394
SEGMENT, 378

Internet Corporation for Assigned Names and Numbers, 321

internet protocol, 307, 310
interrupt driven I/O, 354
interrupt flag, 363
interrupt service routine, 354
interrupts, 391, 394
intersector gap, 280
intertrack gap, 279
inverter, 72, 91, 205
I/O channels/processors, 356
I/O ports, 333, 371
IP. See instruction pointer or internet protocol
IP address, 169, 254
IP header

address fields, 313
fragment offset, 312
header checksum, 312
identification, 311
length, 310
options, 313
padding, 313
time to live, 312
total length, 311
type of service, 311
version, 310
ISR. See interrupt service routine
K
Karnaugh map, 126
Karnaugh map rules, 131

L
latches
   D latch, 209, 223, 242, 262
edge-triggered, 210
S-R latch, 209
   transparent latches, 211
leakage current, 263
least significant bit, 20, 34, 165
LED. See light emitting diode
LIFO, 330
light emitting diode, 13, 147, 162
linker, 375
logic gates, 71
low level formatting, 283
LSB. See least significant bit

M
MAC address, 309, 321
machine code, 338
maximum, 55
Mealy machine, 237
memory
   address, 242
   asynchronous, 266
cell, 203
hierarchy, 269
magnetic core, 241
map, 248, 259, 352
model, 380
processor, 332
space, 249
synchronous, 267
volatile, 245
minimum, 55
modified frequency
   modulation, 273
Moore machine, 237
most significant bit, 20
MP3, 7
MSB. See most significant bit
multiplexer, 156

N
NAND gate, 120, 160, 205, 256
NAND-NAND Logic, 119
negative-going pulse, 10
network interface card, 309
network layer, 304, 310, 313
next state truth table, 231
nibble, 20, 34
NIC. See network interface card
noise, 6
non-periodic pulse trains, 10
NOT gate. See inverter
NOT rule, 96
Nyquist Theorem, 33

O
object file, 375
OF. See overflow flag
offset address, 367
one's complement, 46
one's complement
   checksum/datasum, 176, 312, 319
Open Systems Interconnection
   Model, 303, 307
OR gate, 73, 74, 90, 109, 114
OR rules, 96
O/S level formatting, 283
OSI model. See Open Systems
   Interconnection Model
output truth table, 231
overflow flag, 364

P

packet, 306
Packetizer, 321
parallel port, 214
parity, 174, 190, 193
parity flag, 364
partitioning, 283
pattern detection, 234
period, 11
periodic pulse trains, 11
PF. See parity flag
physical address, 368
physical layer, 304
pipelining, 347
platter, 270
polling, 353
POS. See product-of-sums
positive-going pulse, 10
powers of 2, multiplication and
division by, 65
preamble, 306
precedence, 92
prefix, 15
presentation layer, 303
principle of locality, 285
processor status register, 327
product-of-sums, 114
program counter, 344
protocol, 306
protocol analyzer, 321
protocol stack, 307
pull-up resistors, 163
pulses, 9, 11

R

RAM. See Random Access
Memory
RAM cache, 285
random access memory, 260
read enable, 242
read only memory, 261
read-write head, 270
refresh circuitry, 263
register, 300, 326, 360
registered ports, 313
request for comments, 320
return address, 389
RFC. See request for comments
rising edge, 9, 203
ROM. See Read Only Memory
rotational latency, 275, 276
roundoff error, 31
run length limited, 273

S

sampling, 5, 6, 31
SDRAM. See Synchronous
DRAM
sectors, 279
seek time, 275
segment, 369
addressing, 366, 367
registers, 366
Self-Monitoring Analysis and
Reporting Technology, 278
sequential access, 278
session layer, 303
seven-segment display, 147
SF. See sign flag
SI. See source index
sign bit, 50
sign flag, 364
signed magnitude, 51, 56
quesuing time, 275
SMART. See Self-Monitoring Analysis and Reporting Technology
SOP. See sum-of-products source index, 363
SP. See stack pointer
SRAM. See static RAM
SS. See stack segment
stack, 330
pointer, 362
segment, 369
state, 217
state diagram, 218
errors, 222
reset condition, 221, 226
transitions, 218, 222, 226
state machine, 217, 222
static RAM, 262
strings, 383
substrate, 270
sum-of-products, 109, 125, 129, 153
switch circuit, 163
Synchronous DRAM, 267

TCP. See transmission control protocol
TCP header
acknowledgement, 315
checksum field, 315
control bits, 315
data offset, 315
destination port, 314
option field, 316
sequence number, 314
source port, 314
urgent pointer field, 316
window field, 315
TCP ports, 313
TF. See trap flag
thrashing, 295
timing diagram, 79
track, 279
trailer, 306, 307
transfer time, 275, 276
transistors, 7
transmission control protocol, 307, 313
transport layer, 303, 310, 313
trap flag, 363
tristate output, 247
truth table, 75, 83, 110, 112, 115, 118, 126
two's complement, 47

U
undefined values, 204
unsigned binary, 17, 55

W
Winchester head, 272, 279
word, 20
write enable, 242

X
XOR compare, 173
XOR gate. See exclusive-OR
XOR subtraction, 182

Z
ZBR. See zone bit recording
zero flag, 364
ZF. See zero flag
zone bit recording, 282
ABOUT THE AUTHOR

David Tarnoff is an assistant professor in the Computer and Information Sciences Department at East Tennessee State University where he teaches computer hardware, embedded system design, and web technologies. He holds a bachelors and masters of science in electrical engineering from Virginia Tech. In 1999, David started Internation, Inc., a business that develops software for remote data collection and automation. His research interests include embedded system design and the application of web technologies to teaching and research. David lives in Tennessee with his wife and their son.

NOTE TO THE READER

This textbook was developed after years of teaching computer organization to students of computer science. It incorporates the feedback from hundreds of students and dozens of faculty members and industry professionals. The success of this textbook is a direct result of its users. Therefore, it is important that there always be a direct link between the author and the readers. Please send any feedback you have regarding errors, updates to the material, or suggestions for new material to tarnoff@etsu.edu.

In addition, one of the purposes of this book is to put the concepts of computer organization into the hands of anyone who wants to learn about the topic. As a result, electronic versions of this book should be freely downloadable from the Internet. If you cannot find a version for download, please e-mail the author at tarnoff@etsu.edu, and you will be directed to the proper resources.

Thank you for supporting this work.