

Computer Systems Notes William Furber

1 Program Structure and Execution

1.1 Information Storage

Computers store information as a series of bits. These bits can be interpreted by users in either binary, hexadecimal, or floating point.

Computers also have a default word size, i.e. the largest continuous block of memory the computer can access. Currently, most computers are 32 bit, however more are becoming 64.

Going with word size, each data type also has a typical size in memory:

C Declaration	32 Bit	64 Bit
char	1	1
short int	2	2
int	4	4
long int	4	8
long long int	8	8
char*	4	8
float	4	4
double	8	8

Table 1: Size of C Data Types

Besides the bits themselves, the order also matters, which brings up the distinction between little-endian and big-endian*. Big Endian has the highest place values put in the lowest memory location, while Little Endian has the lowest place values put in the lowest memory location.

1.2 Integer Arithmetic

Depending on the type of numbers involved in addition, we can get strange or unexpected behavior.

If we're dealing with large numbers, ones that are near to the word size in length, we have to be concerned about overflow. Overflow occurs when the full integer result cannot fit within the most size limits of the given data type.

Unsigned integer addition results in something that resembles modular addition. If we have signed integers, we need to now concern ourselves with the negative numbers as well. Negative overflow often results in a positive number due to the definition of two's complement.*

*Note: arises from Gauss's Theorem
 *Computers represent numbers by essentially inverting the bits. Normal binary adds each place, the system subtracts each place from the highest set bit. For a 4 bit word size, the number 0111 would actually be -5 instead of 11.

CSCI 2400

Computer Systems Notes William Furber

1.3 Floating Point

The first method of expressing floating point numbers for a computer was through fractional binary numbers. These numbers had the form:

$$\frac{p}{b_0} + \frac{p-1}{b_0 b_1} + \frac{p-2}{b_0 b_1 b_2} + \dots + \frac{p-n}{b_0 b_1 b_2 \dots b_{n-1}}$$

The inherent issue with this method, is that it's not very good at dealing with larger numbers. This is where IEEE floating point standard comes in, which has the form:

$$(-1)^s \times M \times 2^E$$

Where:

- s determines sign
- M is a fractional binary number ranging from 1 and 2^{-e} or between 0 and $1 + 2^{-e}$
- E weighs the power of 2
- Final format looks as such: **S | E | M**

With these floating point numbers, we have three cases to deal with.

- Normalized Values** are the most common. This occurs when E is neither all ones nor all zeros.
- Denormalized Values** occur when the exponent field is all zeros. These values express zero, as well as numbers that are very close to zero in absolute value.
- Special Values** occur when the exponent field is all ones. This either indicates $\pm\infty$ or NaN when the fractional value is non-zero.

2 Machine Level Representation of Programs

Most computers primarily use assembly for a more human-readable machine code. This code is much more explicit than the C that it was derived from.

- Program counter, referred to as PC, and called `%rip`, refers to the address in memory of the next instruction to be executed.
- The integer register file contains eight named locations storing 32 bit values.
- Condition code registers
- Floating point registers

*The value of `%rip`, all memory addresses are for 32 bit platforms only.

CSCI 2400

Computer Systems Notes William Furber

3 Processor Architecture

3.1 X86 Instruction Set

In this instruction set some commands are split up into several.

3.2 Sequential Implementations vs. Pipelining

In a sequential implementation, all cycles occur one after another. No new operation can start until the old one has finished.

(a) Hardware: Unpipelined

(b) Pipeline diagram

Figure 2: Sequential Implementation

Conversely in a pipelined implementation, we split up the cycles into stages and begin operations before old ones have finished.

4 Optimization

Optimization is an art.

We can use the metric *cycles per element* (CPE) to express how effective a program is.

First step in code optimization is to reduce the number of bottlenecks. This is referred to as code motion.

After that's complete, we remove unnecessary memory referencing.

Followed by loop unrolling, the strategy that involves taking loops with a concrete number of steps and turning them into a set of similar commands.

Finalization can be utilized in certain cases.

Command	Concrete Syntax	Example	Description
<code>mov</code>	<code>mov(Reg, R, #)</code>	<code>mov(%eax, %edx, 4)</code>	Copy data from the source location to the destination.
<code>push</code>	<code>push(imm)</code>	<code>push \$0x0F</code>	Push an item onto the stack.
<code>pop</code>	<code>pop(imm)</code>	<code>pop %eax</code>	Pop an item from the stack.
<code>leaq</code>	<code>leaq(imm, %eax)</code>	<code>leaq 6(%eax), %edx</code>	Load Effective Address takes what's in the register and adds it to the value in the register.
<code>inc</code>	<code>inc(imm)</code>	<code>inc %eax</code>	Increment the value in the register.
<code>dec</code>	<code>dec(imm)</code>	<code>dec %eax</code>	Decrement the value in the register.
<code>neg</code>	<code>neg(imm)</code>	<code>neg %eax</code>	Negate the value in the register.
<code>not</code>	<code>not(imm)</code>	<code>not %eax</code>	Invert the bits in the register.
<code>add</code>	<code>add(imm, %eax)</code>	<code>add \$4, %eax</code>	Add the value in the register to the value in the register.
<code>sub</code>	<code>sub(imm, %eax)</code>	<code>sub \$4, %eax</code>	Subtract the value in the register from the value in the register.
<code>mul</code>	<code>mul(imm, %eax)</code>	<code>mul \$4, %eax</code>	Multiply the value in the register by the value in the register.
<code>div</code>	<code>div(imm, %eax)</code>	<code>div \$4, %eax</code>	Divide the value in the register by the value in the register.
<code>and</code>	<code>and(imm, %eax)</code>	<code>and \$4, %eax</code>	Bitwise AND the value in the register with the value in the register.
<code>or</code>	<code>or(imm, %eax)</code>	<code>or \$4, %eax</code>	Bitwise OR the value in the register with the value in the register.
<code>xor</code>	<code>xor(imm, %eax)</code>	<code>xor \$4, %eax</code>	Bitwise XOR the value in the register with the value in the register.
<code>shl</code>	<code>shl(imm, %eax)</code>	<code>shl \$4, %eax</code>	Shift the value in the register left by the value in the register.
<code>shr</code>	<code>shr(imm, %eax)</code>	<code>shr \$4, %eax</code>	Shift the value in the register right by the value in the register.
<code>cmpl</code>	<code>cmpl(imm, %eax)</code>	<code>cmpl \$4, %eax</code>	Compare the value in the register with the value in the register.
<code>jeq</code>	<code>jeq(imm, %eax)</code>	<code>jeq \$4, %eax</code>	Jump if equal.
<code>jneq</code>	<code>jneq(imm, %eax)</code>	<code>jneq \$4, %eax</code>	Jump if not equal.
<code>jmp</code>	<code>jmp(imm)</code>	<code>jmp \$4</code>	Jump to the address in the register.
<code>jmpnl</code>	<code>jmpnl(imm)</code>	<code>jmpnl \$4</code>	Jump if not taken.
<code>jmppl</code>	<code>jmppl(imm)</code>	<code>jmppl \$4</code>	Jump if taken.
<code>call</code>	<code>call(imm)</code>	<code>call \$4</code>	Call procedure for execution.
<code>leave</code>	<code>leave</code>	<code>leave</code>	Prepare stack for return. This sets the return address and restores the saved registers.
<code>ret</code>	<code>ret</code>	<code>ret</code>	Return from call.

Table 2: Assembly Reference

CSCI 2400

Computer Systems Notes William Furber

2.2 Arithmetic Logic Unit

The ALU performs arithmetic and logical operations. It takes two numbers and performs an operation on them. The result is then stored in a register.

Table 3: Y86 Instruction Set

5 Memory Hierarchy

5.1 RAM

RAM comes in two forms, static and dynamic. Static RAM is much more expensive, but much faster.

5.1.1 SRAM

Each bit is stored in a bistable memory cell. It can only be in one position or another, never both, or neither. It will also keep its state indefinitely as long as it's kept powered.

5.1.2 DRAM

Each bit is stored as a charge on a capacitor. They lose power relatively quickly, but are much cheaper.

5.2 Disk Storage

Disks have several platters that spin at a fixed rate. The capacity of a disk is determined by:

$$\text{Recording Density} \times \text{Track Density} = \text{Areol Density}$$

or

$$\text{Capacity} = \frac{\text{bits}}{\text{sector}} \times \frac{\text{sectors}}{\text{track}} \times \frac{\text{tracks}}{\text{surface}} \times \frac{\text{platters}}{\text{platter}} \times \text{disk}$$

There is an actuator arm responsible for reading and writing from and to the disk.

CSCI 2400

Computer Systems Notes William Furber

2.2 Arithmetic Logic Unit

The ALU performs arithmetic and logical operations. It takes two numbers and performs an operation on them. The result is then stored in a register.

Table 3: Y86 Instruction Set

5 Memory Hierarchy

5.1 RAM

RAM comes in two forms, static and dynamic. Static RAM is much more expensive, but much faster.

5.1.1 SRAM

Each bit is stored in a bistable memory cell. It can only be in one position or another, never both, or neither. It will also keep its state indefinitely as long as it's kept powered.

5.1.2 DRAM

Each bit is stored as a charge on a capacitor. They lose power relatively quickly, but are much cheaper.

5.2 Disk Storage

Disks have several platters that spin at a fixed rate. The capacity of a disk is determined by:

$$\text{Recording Density} \times \text{Track Density} = \text{Areol Density}$$

or

$$\text{Capacity} = \frac{\text{bits}}{\text{sector}} \times \frac{\text{sectors}}{\text{track}} \times \frac{\text{tracks}}{\text{surface}} \times \frac{\text{platters}}{\text{platter}} \times \text{disk}$$

There is an actuator arm responsible for reading and writing from and to the disk.

CSCI 2400

Computer Systems Notes William Furber

6 Linking

7 Virtual Memory

All Assembly code is executed using virtual memory. At no point is it aware of the actual hardware address.

The MMU is in charge of converting these physical addresses to virtual ones, and vice-versa. This can lead to problems however with fragmentation.

7.1 Page Tables

Figure 4: Sequential Implementation

Figure 5: Sequential Implementation

Figure 6: Sequential Implementation

Figure 7: Sequential Implementation

Figure 8: Sequential Implementation

Figure 9: Sequential Implementation

Figure 10: Sequential Implementation

Figure 11: Sequential Implementation

Figure 12: Sequential Implementation

Figure 13: Sequential Implementation

Figure 14: Sequential Implementation

Figure 15: Sequential Implementation

Figure 16: Sequential Implementation

Figure 17: Sequential Implementation

Figure 18: Sequential Implementation

Figure 19: Sequential Implementation

Figure 20: Sequential Implementation

Figure 21: Sequential Implementation

Figure 22: Sequential Implementation

Figure 23: Sequential Implementation

Figure 24: Sequential Implementation

Figure 25: Sequential Implementation

Figure 26: Sequential Implementation

Figure 27: Sequential Implementation

Figure 28: Sequential Implementation

Figure 29: Sequential Implementation

Figure 30: Sequential Implementation

Figure 31: Sequential Implementation

Figure 32: Sequential Implementation

Figure 33: Sequential Implementation

Figure 34: Sequential Implementation

Figure 35: Sequential Implementation

Figure 36: Sequential Implementation

Figure 37: Sequential Implementation

Figure 38: Sequential Implementation

Figure 39: Sequential Implementation

Figure 40: Sequential Implementation

Figure 41: Sequential Implementation

Figure 42: Sequential Implementation

Figure 43: Sequential Implementation

Figure 44: Sequential Implementation

Figure 45: Sequential Implementation

Figure 46: Sequential Implementation

Figure 47: Sequential Implementation

Figure 48: Sequential Implementation

Figure 49: Sequential Implementation

Figure 50: Sequential Implementation

Figure 51: Sequential Implementation

Figure 52: Sequential Implementation

Figure 53: Sequential Implementation

Figure 54: Sequential Implementation

Figure 55: Sequential Implementation

Figure 56: Sequential Implementation

Figure 57: Sequential Implementation

Figure 58: Sequential Implementation

Figure 59: Sequential Implementation

Figure 60: Sequential Implementation

Figure 61: Sequential Implementation

Figure 62: Sequential Implementation

Figure 63: Sequential Implementation

Figure 64: Sequential Implementation

Figure 65: Sequential Implementation

Figure 66: Sequential Implementation

Figure 67: Sequential Implementation

Figure 68: Sequential Implementation

Figure 69: Sequential Implementation

Figure 70: Sequential Implementation

Figure 71: Sequential Implementation

Figure 72: Sequential Implementation

Figure 73: Sequential Implementation

Figure 74: Sequential Implementation

Figure 75: Sequential Implementation

Figure 76: Sequential Implementation

Figure 77: Sequential Implementation

Figure 78: Sequential Implementation

Figure 79: Sequential Implementation

Figure 80: Sequential Implementation

Figure 81: Sequential Implementation

Figure 82: Sequential Implementation

Figure 83: Sequential Implementation

Figure 84: Sequential Implementation

Figure 85: Sequential Implementation

Figure 86: Sequential Implementation

Figure 87: Sequential Implementation

Figure 88: Sequential Implementation

Figure 89: Sequential Implementation

Figure 90: Sequential Implementation

Figure 91: Sequential Implementation

Figure 92: Sequential Implementation

Figure 93: Sequential Implementation

Figure 94: Sequential Implementation

Figure 95: Sequential Implementation

Figure 96: Sequential Implementation

Figure 97: Sequential Implementation

Figure 98: Sequential Implementation

Figure 99: Sequential Implementation

Figure 100: Sequential Implementation

Figure 101: Sequential Implementation

Figure 102: Sequential Implementation

Figure 103: Sequential Implementation

Figure 104: Sequential Implementation

Figure 105: Sequential Implementation

Figure 106: Sequential Implementation

Figure 107: Sequential Implementation

Figure 108: Sequential Implementation

Figure 109: Sequential Implementation

Figure 110: Sequential Implementation

Figure 111: Sequential Implementation

Figure 112: Sequential Implementation

Figure 113: Sequential Implementation

Figure 114: Sequential Implementation

Figure 115: Sequential Implementation

Figure 116: Sequential Implementation

Figure 117: Sequential Implementation

Figure 118: Sequential Implementation

Figure 119: Sequential Implementation

Figure 120: Sequential Implementation

Figure 121: Sequential Implementation

Figure 122: Sequential Implementation

Figure 123: Sequential Implementation

Figure 124: Sequential Implementation

Figure 125: Sequential Implementation

Figure 126: Sequential Implementation

Figure 127: Sequential Implementation

Figure 128: Sequential Implementation

Figure 129: Sequential Implementation

Figure 130: Sequential Implementation

Figure 131: Sequential Implementation

Figure 132: Sequential Implementation

Figure 133: Sequential Implementation

Figure 134: Sequential Implementation

Figure 135: Sequential Implementation

Figure 136: Sequential Implementation

Figure 137: Sequential Implementation

Figure 138: Sequential Implementation

Figure 139: Sequential Implementation

Figure 140: Sequential Implementation

Figure 141: Sequential Implementation

Figure 142: Sequential Implementation

Figure 143: Sequential Implementation

Figure 144: Sequential Implementation

Figure 145: Sequential Implementation

Figure 146: Sequential Implementation

Figure 147: Sequential Implementation

Figure 148: Sequential Implementation

Figure 149: Sequential Implementation

Figure 150: Sequential Implementation

Figure 151: Sequential Implementation

Figure 152: Sequential Implementation

Figure 153: Sequential Implementation

Figure 154: Sequential Implementation

Figure 155: Sequential Implementation

Figure 156: Sequential Implementation

Figure 157: Sequential Implementation

Figure 158: Sequential Implementation

Figure 159: Sequential Implementation

Figure 160: Sequential Implementation

Figure 161: Sequential Implementation

Figure 162: Sequential Implementation

Figure 163: Sequential Implementation

Figure 164: Sequential Implementation

Figure 165: Sequential Implementation

Figure 166: Sequential Implementation

Figure 167: Sequential Implementation

Figure 168: Sequential Implementation

Figure 169: Sequential Implementation

Figure 170: Sequential Implementation

Figure 171: Sequential Implementation

Figure 172: Sequential Implementation

Figure 173: Sequential Implementation

Figure 174: Sequential Implementation

Figure 175: Sequential Implementation

Figure 176: Sequential Implementation

Figure 177: Sequential Implementation

Figure 178: Sequential Implementation

Figure 179: Sequential Implementation

Figure 180: Sequential Implementation

Figure 181: Sequential Implementation

Figure 182: Sequential Implementation

Figure 183: Sequential Implementation

Figure 184: Sequential Implementation

Figure 185: Sequential Implementation

Figure 186: Sequential Implementation

Figure 187: Sequential Implementation

Figure 188: Sequential Implementation

Figure 189: Sequential Implementation

Figure 190: Sequential Implementation

Figure 191: Sequential Implementation

Figure 192: Sequential Implementation

Figure 193: Sequential Implementation

Figure 194: Sequential Implementation

Figure 195: Sequential Implementation

Figure 196: Sequential Implementation

Figure 197: Sequential Implementation

Figure 198: Sequential Implementation

Figure 199: Sequential Implementation

Figure 200: Sequential Implementation

Figure 201: Sequential Implementation

Figure 202: Sequential Implementation

Figure 203: Sequential Implementation

Figure 204: Sequential Implementation

Figure 205: Sequential Implementation

Figure 206: Sequential Implementation

Figure 207: Sequential Implementation

Figure 208: Sequential Implementation

Figure 209: Sequential Implementation

Figure 210: Sequential Implementation

Figure 211: Sequential Implementation

Figure 212: Sequential Implementation

Figure 213: Sequential Implementation

Figure 214: Sequential Implementation

Figure 215: Sequential Implementation

Figure 216: Sequential Implementation

Figure 217: Sequential Implementation

Figure 218: Sequential Implementation

Figure 219: Sequential Implementation

Figure 220: Sequential Implementation

Figure 221: Sequential Implementation

Figure 222: Sequential Implementation

Figure 223: Sequential Implementation

Figure 224: Sequential Implementation

Figure 225: Sequential Implementation

Figure 226: Sequential Implementation

Figure 227: Sequential Implementation

Figure 228: Sequential Implementation

Figure 229: Sequential Implementation

Figure 230: Sequential Implementation

Figure 231: Sequential Implementation

Figure 232: Sequential Implementation

Figure 233: Sequential Implementation

Figure 234: Sequential Implementation

Figure 235: Sequential Implementation

Figure 236: Sequential Implementation

Figure 237: Sequential Implementation

Figure 238: Sequential Implementation

Figure 239: Sequential Implementation

Figure 240: Sequential Implementation

Figure 241: Sequential Implementation

Figure 242: Sequential Implementation

Figure 243: Sequential Implementation

Figure 244: Sequential Implementation

Figure 245: Sequential Implementation

Figure 246: Sequential Implementation

Figure 247: Sequential Implementation

Figure 248: Sequential Implementation

Figure 249: Sequential Implementation

Figure 250: Sequential Implementation

Figure 251: Sequential Implementation

Figure 252: Sequential Implementation

Figure 253: Sequential Implementation

Figure 254: Sequential Implementation

Figure 255: Sequential Implementation

Figure 256: Sequential Implementation

Figure 257: Sequential Implementation

Figure 258: Sequential Implementation

Figure 259: Sequential Implementation

Figure 260: Sequential Implementation

Figure 261: Sequential Implementation

Figure 262: Sequential Implementation

Figure 263: Sequential Implementation

Figure 264: Sequential Implementation

Figure 265: Sequential Implementation

Figure 266: Sequential Implementation

Figure 267: Sequential Implementation

Figure 268: Sequential Implementation

Figure 269: Sequential Implementation

Figure 270: Sequential Implementation

Figure 271: Sequential Implementation

Figure 272: Sequential Implementation

Figure 273: Sequential Implementation

Figure 274: Sequential Implementation

Figure 275: Sequential Implementation

Figure 276: Sequential Implementation

Figure 277: Sequential Implementation

Figure 278: Sequential Implementation

Figure 279: Sequential Implementation

Figure 280: Sequential Implementation

Figure 281: Sequential Implementation

Figure 282: Sequential Implementation

Figure 283: Sequential Implementation

Figure 284: Sequential Implementation

Figure 285: Sequential Implementation

Figure 286: Sequential Implementation

Figure 287: Sequential Implementation

Figure 288: Sequential Implementation

Figure 289: Sequential Implementation

Figure 290: Sequential Implementation

Figure 291: Sequential Implementation

Figure 292: Sequential Implementation

Figure 293: Sequential Implementation

Figure 294: Sequential Implementation

Figure 295: Sequential Implementation

Figure 296: Sequential Implementation

Figure 297: Sequential Implementation

Figure 298: Sequential Implementation

Figure 299: Sequential Implementation

Figure 300: Sequential Implementation

Figure 301: Sequential Implementation

Figure 302: Sequential Implementation

Figure 303: Sequential Implementation

Figure 304: Sequential Implementation

Figure 305: Sequential Implementation

Figure 306: Sequential Implementation

Figure 307: Sequential Implementation

Figure 308: Sequential Implementation

Figure 309: Sequential Implementation

Figure 310: Sequential Implementation

Figure 311: Sequential Implementation

Figure 312: Sequential Implementation

Figure 313: Sequential Implementation

Figure 314: Sequential Implementation

Figure 315: Sequential Implementation

Figure 316: Sequential Implementation

Figure 317: Sequential Implementation

Figure 318: Sequential Implementation

Figure 319: Sequential Implementation

Figure 320: Sequential Implementation

Figure 321: Sequential Implementation

Figure 322: Sequential Implementation

Figure 323: Sequential Implementation

Figure 324: Sequential Implementation

Figure 325: Sequential Implementation

Figure 326: Sequential Implementation

Figure 327: Sequential Implementation

Figure 328: Sequential Implementation

Figure 329: Sequential Implementation

Figure 330: Sequential Implementation

Figure 331: Sequential Implementation

Figure 332: Sequential Implementation

Figure 333: Sequential Implementation

Figure 334: Sequential Implementation

Figure 335: Sequential Implementation

Figure 336: Sequential Implementation

Figure 337: Sequential Implementation

Figure 338: Sequential Implementation

Figure 339: Sequential Implementation

Figure 340: Sequential Implementation

Figure 341: Sequential Implementation

Figure 342: Sequential Implementation

Figure 343: Sequential Implementation

Figure 344: Sequential Implementation

Figure 345: Sequential Implementation

Figure 346: Sequential Implementation

Figure 347: Sequential Implementation

Figure 348: Sequential Implementation

Figure 349: Sequential Implementation

Figure 350: Sequential Implementation

Figure 351: Sequential Implementation

Figure 352: Sequential Implementation

Figure 353: Sequential Implementation

Figure 354: Sequential Implementation

Figure 355: Sequential Implementation

Figure 356: Sequential Implementation

Figure 357: Sequential Implementation

Figure 358: Sequential Implementation

Figure 359: Sequential Implementation

Figure 360: Sequential Implementation

Figure 361: Sequential Implementation

Figure 362: Sequential Implementation

Figure 363: Sequential Implementation

Figure 364: Sequential Implementation

Figure 365: Sequential Implementation

Figure 366: Sequential Implementation

Figure 367: Sequential Implementation

Figure 368: Sequential Implementation

Figure 369: Sequential Implementation

Figure 370: Sequential Implementation

Figure 371: Sequential Implementation

Figure 372: Sequential Implementation

Figure 373: Sequential Implementation

Figure 374: Sequential Implementation

Figure 375: Sequential Implementation

Figure 376: Sequential Implementation

Figure 377: Sequential Implementation

Figure 378: Sequential Implementation

Figure 379: Sequential Implementation

Figure 380: Sequential Implementation

Figure 381: Sequential Implementation

Figure 382: Sequential Implementation

Figure 383: Sequential Implementation

Figure 384: Sequential Implementation

Figure 385: Sequential Implementation

Figure 386: Sequential Implementation

Figure 387: Sequential Implementation

Figure 388: Sequential Implementation

Figure 389: Sequential Implementation

Figure 390: Sequential Implementation

Figure 391: Sequential Implementation

Figure 392: Sequential Implementation

Figure 393: Sequential Implementation

Figure 394: Sequential Implementation

Figure 395: Sequential Implementation

Figure 396: Sequential Implementation

Figure 397: Sequential Implementation

Figure 398: Sequential Implementation

Figure 399: Sequential Implementation

Figure 400: Sequential Implementation

Figure 401: Sequential Implementation

Figure 402: Sequential Implementation

Figure 403: Sequential Implementation

Figure 404: Sequential Implementation

Figure 405: Sequential Implementation

Figure 406: Sequential Implementation

Figure 407: Sequential Implementation

Figure 408: Sequential Implementation

Figure 409: Sequential Implementation

Figure 410: Sequential Implementation

Figure 411: Sequential Implementation

Figure 412: Sequential Implementation

Figure 413: Sequential Implementation

Figure 414: Sequential Implementation

Figure 415: Sequential Implementation

Figure 416: Sequential Implementation

Figure 417: Sequential Implementation

Figure 418: Sequential Implementation

Figure 419: Sequential Implementation

Figure 420: Sequential Implementation

Figure 421: Sequential Implementation

Figure 422: Sequential Implementation

Figure 423: Sequential Implementation

Figure 424: Sequential Implementation

Figure 425: Sequential Implementation

Figure 426: Sequential Implementation

Figure 427: Sequential Implementation

Figure 428: Sequential Implementation

Figure 429: Sequential Implementation

Figure 430: Sequential Implementation

Figure 431: Sequential Implementation

Figure 432: Sequential Implementation

Figure 433: Sequential Implementation

Figure 434: Sequential Implementation

Figure 435: Sequential Implementation

Figure 436: Sequential Implementation

Figure 437: Sequential Implementation

Figure 438: Sequential Implementation

Figure 439: Sequential Implementation

Figure 440: Sequential Implementation

Figure 441: Sequential Implementation

Figure 442: Sequential Implementation

Figure 443: Sequential Implementation

Figure 444: Sequential Implementation

Figure 445: Sequential Implementation

Figure 446: Sequential Implementation

Figure 447: Sequential Implementation

Figure 448: Sequential Implementation

Figure 449: Sequential Implementation

Figure 450: Sequential Implementation

Figure 451: Sequential Implementation

Figure 452: Sequential Implementation

Figure 453: Sequential Implementation

Figure 454: Sequential Implementation

Figure 455: Sequential Implementation

Figure 456: Sequential Implementation

Figure 457: Sequential Implementation

Figure 458: Sequential Implementation

Figure 459: Sequential Implementation

Figure 460: Sequential Implementation

Figure 461: Sequential Implementation

Figure 462: Sequential Implementation

Figure 463: Sequential Implementation

Figure 464: Sequential Implementation

Figure 465: Sequential Implementation

Figure 466: Sequential Implementation

Figure 467: Sequential Implementation

Figure 468: Sequential Implementation

Figure 469: Sequential Implementation

Figure 470: Sequential Implementation

Figure 471: Sequential Implementation

Figure 472: Sequential Implementation

Figure 473: Sequential Implementation

Figure 474: Sequential Implementation

Figure 475: Sequential Implementation

Figure 476: Sequential Implementation

Figure 477: Sequential Implementation

Figure 478: Sequential Implementation

Figure 479: Sequential Implementation

Figure 480: Sequential Implementation

Figure 481: Sequential Implementation

Figure 482: Sequential Implementation

Figure 483: Sequential Implementation

Figure 484: Sequential Implementation

Figure 485: Sequential Implementation

Figure 486: Sequential Implementation

Figure 487: Sequential Implementation

Figure 488: Sequential Implementation

Figure 489: Sequential Implementation

Figure 490: Sequential Implementation

Figure 491: Sequential Implementation

Figure 492: Sequential Implementation

Figure 493: Sequential Implementation

Figure 494: Sequential Implementation

Figure 495: Sequential Implementation

Figure 496: Sequential Implementation

Figure 497: Sequential Implementation

Figure 498: Sequential Implementation

Figure 499: Sequential Implementation

Figure 500: Sequential Implementation

Figure 501: Sequential Implementation

Figure 502: Sequential Implementation

Figure 503: Sequential Implementation

Figure 504: Sequential Implementation

Figure 505: Sequential Implementation

Figure 506: Sequential Implementation

Figure 507: Sequential Implementation

Figure 508: Sequential Implementation

Figure 509: Sequential Implementation

Figure 510: Sequential Implementation

Figure 511: Sequential Implementation

Figure 512: Sequential Implementation

Figure 513: Sequential Implementation

Figure 514: Sequential Implementation

Figure 515: Sequential Implementation

Figure 516: Sequential Implementation

Figure 517: Sequential Implementation

Figure 518: Sequential Implementation

Figure 519: Sequential Implementation

Figure 520: Sequential Implementation

Figure 521: Sequential Implementation

Figure 522: Sequential Implementation

Figure 523: Sequential Implementation

Figure 524: Sequential Implementation

Figure 525: Sequential Implementation

Figure 526: Sequential Implementation

Figure 527: Sequential Implementation

Figure 528: Sequential Implementation

Figure 529: Sequential Implementation

Figure 530: Sequential Implementation

Figure 531: Sequential Implementation

Figure 532: Sequential Implementation

Figure 533: Sequential Implementation

Figure 534: Sequential Implementation

Figure 535: Sequential Implementation

Figure 536: Sequential Implementation

Figure 537: Sequential Implementation

Figure 538: Sequential Implementation

Figure 539: Sequential Implementation

Figure 540: Sequential Implementation

Figure 541: Sequential Implementation

Figure 542: Sequential Implementation

Figure 543: Sequential Implementation

Figure 544: Sequential Implementation

Figure 545: Sequential Implementation

Figure 546: Sequential Implementation

Figure 547: Sequential Implementation

Figure 548: Sequential Implementation

Figure 549: Sequential Implementation

Figure 550: Sequential Implementation

Figure 551: Sequential Implementation

Figure 552: Sequential Implementation

Figure 553: Sequential Implementation

Figure 554: Sequential Implementation

Figure 555: Sequential Implementation

Figure 556: Sequential Implementation

Figure 557: Sequential Implementation

Figure 558: Sequential Implementation

Figure 559: Sequential Implementation

Figure 560: Sequential Implementation

Figure 561: Sequential Implementation

Figure 562: Sequential Implementation

Figure 563: Sequential Implementation

Figure 564: Sequential Implementation

Figure 565: Sequential Implementation

Figure 566: Sequential Implementation

Figure 567: Sequential Implementation

Figure 568: Sequential Implementation

Figure 569: Sequential Implementation

Figure 570: Sequential Implementation

Figure 571: Sequential Implementation

Figure 572: Sequential Implementation

Figure 573: Sequential Implementation