Lecture 15

Topics:

1. Logic of Integer Representation
2. Decimal and Hexadecimal Representations
3. Binary Representation
4. Binary Addition Algorithm
5. Binary Multiplication Algorithm

4.2 Integer Representations and Algorithms

- Integers can be expressed using any integer greater than one as a base, as we will show in this section. Although we commonly use decimal (base 10), representations, binary (base 2), octal (base 8), and hexadecimal (base 16) representations are often used, especially in computer science. Given a base b and an integer n, we will show how to construct the base b representation of this integer. We will also explain how to quickly convert between binary and octal and between binary and hexadecimal notations.

4.2.2 Representations of Integers

In everyday life we use decimal notation to express integers. In decimal notation, an integer n is written as a sum of the form $a_k \cdot 10^k + a_{k-1} \cdot 10^{k-1} + \cdots + a_1 \cdot 10 + a_0$, where a_j is an integer with $0 \le a_j \le 9$ for $j = 0,1, \ldots, k$. For example, 965 is used to denote $9 \cdot 10^2 + 6 \cdot 10 + 5$. However, it is often convenient to use bases other than 10.

THEOREM 1: Let b be an integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form $n = a_k b_k + a_{k-1} b_{k-1} + \cdots + a_1 b + a_0$, where k is a nonnegative integer, $a_0, a_1, \ldots, a_k$ are nonnegative integers less than b, and $a_k \neq 0$.

4.2.2 Representations of Integers (Continued)

EXAMPLE 1: What is the decimal expansion of the integer that has $(1\,0101\,1111)_2$ as its binary expansion?
Solution: We have $(1\,0101\,1111)_2 = 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 351$.

EXAMPLE 2: What is the decimal expansion of the number with octal expansion $(7016)_8$?
Solution: Using the definition of a base b expansion with $b = 8$ tells us that $(7016)_8 = 7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8 + 6 = 3598$.

Sixteen different digits are required for hexadecimal expansions. Usually, the hexadecimal digits used are $0,1,2,3,4,5,6,7,8,9, A, B, C, D, E$, and F, where the letters A through F represent the digits corresponding to the numbers 10 through 15 (in decimal notation).

4.2.2 Representations of Integers (Continued)

EXAMPLE 3: What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?

Solution: Using the definition of a base b expansion with $b = 16$ tells us that $(2AE0B)_{16} = 2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16 + 11 = 175627$.

EXAMPLE 4: Find the octal expansion of $(12345)_{10}$.

Extra Solution: First, divide 12345 by 8 to obtain $12345 = 8 \cdot 1543 + 1 \cdot 262$. Successively dividing quotients by 8 gives $1543 = 8 \cdot 192 + 7$, $192 = 8 \cdot 24 + 0$, $24 = 8 \cdot 3 + 0$, $3 = 8 \cdot 0 + 3$. The successive remainders that we have found, 1,7,0,0, and 3, are the digits from the right to the left of 12345 in base 8. Hence, $(12345)_{10} = (30071)_{8}$.

4.2.2 Representations of Integers (Continued)

EXAMPLE 5: Find the hexadecimal expansion of $(177130)_{10}$.

<u>Solution:</u> First divide 177130 by 16 to obtain $177130 = 16 \cdot 11070 + 10$. Successively dividing quotients by 16 gives $11070 = 16 \cdot 691 + 14$, $691 = 16 \cdot 43 + 3$, $43 = 16 \cdot 2 + 11$, $2 = 16 \cdot 0 + 2$. The successive remainders that we have found, $10, 14, 3, 11, 2$, give us the digits from the right to the left of 177130 in the hexadecimal (base 16) expansion of $(177130)_{10.}$ It follows that $(177130)_{10} = (2B3EA)_{16}$.

4.2.2 Representations of Integers (Continued)

EXAMPLE 6: Find the binary expansion of $(241)_{10}$.

Solution: First divide 241 by 2 to obtain $241 = 2 \cdot 120 + 1$. Successively dividing quotients by 2 gives $120 = 2 \cdot 60 + 0,\ 60 = 2 \cdot 30 + 0,\ 30 = 2 \cdot 15 + 0,\ 15 = 2 \cdot 7 + 1,\ 7 = 2 \cdot 3 + 1,\ 3 = 2 \cdot 1 + 1,\ 1 = 2 \cdot 0 + 1$. The successive remainders that we have found, $1, 0, 0, 0, 1, 1, 1, 1$, are the digits from the right to the left in the binary (base 2) expansion of $(241)_{10.}$ Hence, $(241)_{10} = (1111\ 0001)_2$.

4.2.2 Representations of Integers (Continued)

ALGORITHM 1: Constructing Base b Expansions.

procedure base b expansion(n, b: positive integers with $b > 1$)

$q := n$

$k := 0$

while $q \neq 0$

$a_k := q \bmod b, \;\; q := q \operatorname{div} b, \;\; k := k + 1$

return $(a_{k-1}, \ldots, a_1, a_0)$ $\{(a_{k-1} \ldots a_1 a_0)_b$ is the base b expansion of $n\}$

4.2.2 Representations of Integers (Continued)

ALGORITHM 2: Addition of Integers.

procedure add(a, b: positive integers) {the binary expansions of a and b are $(a_{n-1}a_{n-2} \dots a_1 a_0)_2$ and $(b_{n-1}b_{n-2} \dots b_1 b_0)_2$, respectively}

$c := 0$

for $j := 0$ to $n - 1$

    $d := \lfloor (a_j + b_j + c)/2 \rfloor$

    $s_j := a_j + b_j + c - 2d$

    $c := d$

    $s_n := c$

return $(s_0, s_1, \dots, s_n)$ {the binary expansion of the sum is $(s_n s_{n-1} \dots s_0)_2$

MULTIPLICATION ALGORITHM: Next, consider the multiplication of two n-bit integers a and b. The conventional algorithm (used when multiplying with pencil and paper) works as follows.
Using the distributive law, we see that $ab = a(b_0 2^0 + b_1 2^1 + \cdots + b_{n-1} 2^{n-1}) = a(b_0 2^0) + a(b_1 2^1) + \cdots + a(b_{n-1} 2^{n-1})$

4.2.2 Representations of Integers (Continued)

ALGORITHM 3: Multiplication of Integers.

procedure multiply(a, b: positive integers) {the binary expansions of a and b are $(a_{n-1}a_{n-2} \ldots a_1 a_0)_2$ and $(b_{n-1}b_{n-2} \ldots b_1 b_0)_2$, respectively}

for $j := 0$ to $n - 1$

if $b_j = 1$ then $c_j := a$ shifted j places

    else $c_j := 0$ {$c_0, c_1, \ldots, c_{n-1}$ are the partial products}

    $p := 0$

    for $j := 0$ to $n - 1$

        $p :=$ add(p, c_j)

return p {p is the value of ab}

4.2.2 Representations of Integers (Continued)

EXAMPLE 7: Find the octal and hexadecimal expansions of $(11\ 1110\ 1011\ 1100)_2$ and the binary expansions of $(765)_8$ and $(A8D)_{16}$.

<u>Solution:</u> To convert $(11\ 1110\ 1011\ 1100)_2$ into octal notation we group the binary digits into blocks of three, adding initial zeros at the start of the leftmost block if necessary. These blocks, from left to right, are $011, 111, 010, 111$, and 100, corresponding to $3, 7, 2, 7$, and 4, respectively. Consequently, $(11\ 1110\ 1011\ 1100)_2 = (37274)_8$. To convert $(11\ 1110\ 1011\ 1100)_2$ into hexadecimal notation we group the binary digits into blocks of four, adding initial zeros at the start of the leftmost block if necessary. These blocks, from left to right, are $0011, 1110, 1011$, and 1100, corresponding to the hexadecimal digits $3, E, B$, and C, respectively. Consequently, $(11\ 1110\ 1011\ 1100)_2 = (3EBC)_{16}$. To convert $(765)_8$ into binary notation, we replace each octal digit by a block of three binary digits. These blocks are $111, 110$, and 101. Hence, $(765)_8 = (1\ 1111\ 0101)_2$. To convert $(A8D)_{16}$ into binary notation, we replace each hexadecimal digit by a block of four binary digits. These blocks are $1010, 1000$, and 1101. Hence, $(A8D)_{16} = (1010\ 1000\ 1101)_2$.

Integer Operations

ALGORITHM 2: Addition of Integers.

procedure add(a, b: positive integers) {the binary expansions of a and b are $(a_{n-1}\, a_{n-2} \ldots a_1 a_0)_2$ and $(b_{n-1} b_{n-2} \ldots b_1 b_0)_2$}

$c := 0$

for $j := 0$ to $n - 1$

$\quad d := \lfloor (a_j + b_j + c)/2 \rfloor$

$\quad s_j := a_j + b_j + c - 2d$

$\quad c := d$

$\quad s_n := c$

return $(s_0, s_1, \ldots, s_n)$ {the binary expansion of the sum is $(s_n s_{n-1} \ldots s_0)_2$}

Integer Operations (Continued)

ALGORITHM 3: Multiplication of Integers.

procedure multiply(a, b: positive integers) {the binary expansions of a and b are $(a_{n-1}a_{n-2} \ldots a_1a_0)_2$ and $(b_{n-1}b_{n-2} \ldots b_1b_0)_2$}

for $j := 0$ to $n - 1$

    if $b_j = 1$ then $c_j := a$ shifted j places

    else $c_j := 0$ {$c_0, c_1, \ldots, c_{n-1}$ are the partial products}

$p := 0$

for $j := 0$ to $n - 1$

    p := add(p, c_j)

return p {p is the value of ab}

Self Study and Homework

4.2.4 Modular Exponentiation

- Before we present an algorithm for fast modular exponentiation based on the binary expansion of the exponent, first observe that we can avoid using large amount of memory if we compute $b_n \bmod m$ by successively computing $b_k \bmod m$ for $k = 1, 2, \ldots, n$ using the fact that $b^{k+1} \bmod m = b\left(b^k \bmod m\right) \bmod m$ (by Corollary 2 of Theorem 5 of Section 4.1). (Recall that $1 \leq b < m$.) However, this approach is impractical because it requires $n - 1$ multiplications of integers and n might be huge.

Self Study and Homework

4.2.4 Modular Exponentiation (Continued)

- To motivate the fast modular exponentiation algorithm, we illustrate its basic idea. We will explain how to use the binary expansion of n, say $n = (a_{k-1} \dots a_1 a_0)_2$, to compute b^n. First, note that $b^n = b^{a_{k-1} \cdot 2^{k-1} + \cdots + a_1 \cdot 2 + a_0} = b^{a_{k-1} \cdot 2^{k-1}} \cdots b^{a_1 \cdot 2} \cdot b^{a_0}$

4.2.4 Modular Exponentiation (Continued)

For example, to compute 311 we first note that $11 = (1011)_2$, so that $3^{11} = 383231$. By successively squaring, we find that $3^2 = 9, 3^4 = 9^2 = 81$, and $3^8 = (81)^2 = 6561$.

Consequently, $3^{11} = 3^8 3^2 3^1$

Self Study and Homework

4.2.4 Modular Exponentiation (Continued)

ALGORITHM 5: Fast Modular Exponentiation.

procedure modular exponentiation(b: integer, $n = (a_{n-1}a_{n-2} \ldots a_1 a_0)_2$, m: positive integers)

$x := 1$

$\text{power} := b \bmod m$

for $i := 0$ to $k - 1$

    if $a_i = 1$ then $x := (x \cdot \text{power}) \bmod m$

    $\text{power} := (\text{power} \cdot \text{power}) \bmod m$

return x $\{x$ equals $b_n \bmod m\}$

Self Study and Homework

4.2.4 Modular Exponentiation (Continued)

EXAMPLE 12: Use Algorithm 5 to find $3^{644} \bmod 645$.

<u>Solution:</u> Algorithm 5 initially sets $x = 1$ and $\text{power} = 3 \bmod 645 = 3$. In the computation of $3^{644} \bmod 645$, this algorithm determines $3^{2^j} \bmod 645$ for $j = 1, 2, \ldots, 9$ by successively squaring and reducing modulo 645. If $a_j = 1$ (where a_j is the bit in the jth position in the binary expansion of 644, which is $(1010000100)_2$, it multiplies the current value of x by $3^{2^j} \bmod 645$ and reduces the result modulo 645. Here are the steps used:

Self Study and Homework

4.2.4 Modular Exponentiation (Continued)

$i = 0$: Because $a_0 = 0$, we have $x = 1$ and $\text{power} = 3^2 \bmod 645 = 9 \bmod 645 = 9$;

$i = 1$: Because $a_1 = 0$, we have $x = 1$ and $\text{power} = 9^2 \bmod 645 = 81 \bmod 645 = 81$;

$i = 2$: Because $a_2 = 1$, we have $x = 1 \cdot 81 \bmod 645 = 81$ and $\text{power} = 81^2 \bmod 645 = 6561 \bmod 645 = 111$;

$i = 3$: Because $a_3 = 0$, we have $x = 81$ and $\text{power} = 111^2 \bmod 645 = 12{,}321 \bmod 645 = 66$;

$i = 4$: Because $a_4 = 0$, we have $x = 81$ and $\text{power} = 66^2 \bmod 645 = 4356 \bmod 645 = 486$;

$i = 5$: Because $a_5 = 0$, we have $x = 81$ and $\text{power} = 486^2 \bmod 645 = 236{,}196 \bmod 645 = 126$;

$i = 6$: Because $\mathrm{a}_6 = 0$, we have $x = 81$ and $\text{power} = 12^{62} \bmod 645 = 15{,}876 \bmod 645 = 396$;

$i = 7$: Because $a_7 = 1$, we find that $x = (81 \cdot 396) \bmod 645 = 471$ and $\text{power} = 39^{62} \bmod 645 = 156{,}816 \bmod 645 = 81$;

$i = 8$: Because $a_8 = 0$, we have $x = 471$ and $\text{power} = 81^2 \bmod 645 = 6561 \bmod 645 = 111$;

$i = 9$: Because $a_9 = 1$, we find that $x = (471 \cdot 111) \bmod 645 = 36$.