Octal and Hexadecimal Integers

CS 350: Computer Organization & Assembler Language Programming

A. Why?

- Octal and hexadecimal numbers are useful for abbreviating long bitstrings.
- Some operations on octal and hexadecimal numbers correspond to operations on bitstrings.

B. Outcomes

At the end of today, you should know how

- Octal and hexadecimal numbers work
- To translate to and from binary, octal, hexadecimal, and decimal.
- To read octal and hexadecimal numbers as standing for binary 2's complement integers and to take their negative.

C. Converting Between Octal, Hexadecimal, and Binary

- Octal: base 8 (uses digits 0 – 7).
- Hexadecimal (a.k.a. "hex"): base 16 uses A – F as digits for 10–15.
- Octal and are hexadecimal useful for abbreviating bit strings, hex for bitstrings of lengths 4, 8, 12, 16, …, and octal for bitstrings of lengths 3, 6, 9, 12, ….

Converting Hexadecimal → Binary

- To convert a string of hex digits into binary, replace each hex digit by its 4-bit representation: $0_{16} = 0000_2$, …, $9_{16} = 1001_2$, $A_{16} = 10_{10} = 1010_2$; $B_{16} = 1011_2$, $C_{16} = 1100_2$, $D_{16} = 1101_2$, $E_{16} = 1110_2$, $F_{16} = 1111_2$.
  - E.g., $03FC_{16} = 0000\ 0011\ 1111\ 1100_2$.
- If we have k hex digits, we end up with $4\ k$ bits unless we're told explicitly that we want fewer than that. E.g., normally we would say that $5D_{16}$ represents the 8 bits 0101 1101, but if we say "the 7 bits represented by $5D_{16}$", we want 101 1101. We

 © James Sasaki, 2016

can't ask for "the 6 bits represented by $5D_{16}$" because we need at least 7 bits to get 101 1101.

- If we want the decimal number represented by $5D_{16}$, we have to know if it represents a signed or unsigned bitstring; if it represents a signed bitstring, we also need to know how many bits it represents (and what scheme we're using).
- **Example 1**: As an 8-bit number, $5D_{16} = 0101\ 1101_2 = 93_{10}$. (Since the sign bit is 0, this equality holds whether we read the number as unsigned, sign-magnitude, 1's complement, or 2's complement.)
- **Example 2**: Reading $5D_{16}$ as a 7-bit string, we get
  - Unsigned, $5D_{16} = 101\ 1101_2 = 93_{10}$.
  - In 2's complement, $5D_{16} = 101\ 1101_2 = -(010\ 0011_2) = -35_{10}$. (I.e., "7-bit (2's complement) $5D_{16} = -35_{10}$".)
  - In 1's complement, $5D_{16} = 101\ 1101_2 = -(010\ 0010_2) = -34_{10}$.
  - In sign-magnitude, $5D_{16} = 101\ 1101_2 = -(01\ 1101_2) = -29_{10}$.
- **Example 3**: Reading DD_{16} as the 8-bit string $1101\ 1101_2$
  - Unsigned, DD_{16} represents decimal 13*16 + 13 = 221
  - In 2's complement, $DD_{16} = 1101\ 1101_2 = -(0010\ 0011)_2 = -35_{10}$
  - In 1's complement, $DD_{16} = 1101\ 1101_2 = -(0010\ 0010)_2 = -34_{10}$
  - In sign-magnitude, $DD_{16} = 1101\ 1101_2 = -(101\ 1101)_2 = -93_{10}$
- Note that 8-bit $1101\ 1101_2$ and 7-bit $101\ 1101_2$ both represent -35 in 2's complement and -34 in 1's complement.
  - The operation that takes 7-bit 101 1101 to 8-bit 1101 1101 is **sign extension**; we copy the sign bit 1 leftward.
  - Note if the sign bit is 0, we copy that instead during sign extension: 001 1101 becomes 0001 1101.

 © James Sasaki, 2016

Converting Octal → Binary

- Reading an octal number as a bitstring is similar to the process for hexadecimal numbers except that each digit is replaced by 3 bits, since $8 = 2^3$. If we have k octal digits, we end up with $3\,k$ bits unless told otherwise.
- **Example 4**: 6-bit 31_8 = 011 001 = 25_{10}. 5-bit 31_8 = 11 001_2
  - In 2's complement, 31_8 = 11 001 = –(00 111) = –7
  - In 1's complement 31_8 = 11 001 = –(00 110) = –6
  - In sign-magnitude 31_8 = 11 001 = –(01001) = –9.

Converting Binary → Hexadecimal

- To convert a bitstring read as an unsigned binary number to hexadecimal
- First, if the bitstring length is not a multiple of 4, then pad the bitstring on the left with enough 0s to get length that's a multiple of 4. Then replace each 4-bit sequence by its equivalent hex digit. E.g., 011011_2 → 0001 1011_2 → $1B_{16}$.
- Alternatively, group the bitstring's bits into clumps of 4 going **right-to-left** and convert each clump into a hexadecimal digit.
  - E.g., 011011_2 = 01 1011 = $1B_{16}$. Don't try 011011_2 = 0110 11_2 = 63_{16}.

Converting Octal → Binary

  - Replace each octal digit (0–7) by its 3-bit representation.
  - E.g., 106_8 → 001 000 110_2.
- Converting Binary → Octal
  - Pad bitstring on left with 0s to get a string of length divisible by 3.
  - Replace each 3-bit sequence by its equivalent octal digit.
  - E.g., 11010_2 → 011 010_2 → 32_8. (Not 11010_2 → 110 10_2 → 62_8.)

D. Converting To Decimal

- We know that converting from base b to decimal involves some multiplication by powers of b. You can avoid calculating the powers of b by using factoring to get an expression that only multiplies by b.

 © James Sasaki, 2016

- E.g., $10110_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2 + 0$
  $= (1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2 + 1) \cdot 2 + 0$
  $= ((1 \cdot 2^2 + 0 \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 0$
  $= (((1 \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 0$
- At each step, the running total is the decimal equivalent of the bitstring seen so far: $10_2 = 2_{10}$; $101_2 = 5_{10}$, $1011_2 = 11_{10}$, and $10110_2 = 22_{10}$.
- This "multiply and add" technique generalizes to other bases.
- To convert an octal string to decimal, we multiply by 8 and add the next digit.
- To convert a hex string to decimal, we multiply by 16 and add the next digit.

E. Converting From Decimal

- When converting from decimal to base *b*, we repeatedly divide by *b* and capture the remainders to build the base *b* number from right to left.
- To convert a number n > 0 to base b:
  Start with result string S = the empty string.
  while $n > 0$
      Divide n by b to get quotient q and remainder r.
      If r > ten, convert it to the appropriate letter (A for 10, B for 11, etc).
      Append r to the **left** end of the string S.
      Set $n \leftarrow q$

F. Taking the (1 or 2's Complement) Negative of Bitstrings Represented Using Hexadecimal

- What is the relationship between the hex representations of a bitstring and its 1 or 2's complement negative?
- **Example 5**: $3A_{16}$ represents $0011\ 1010_2$, whose 2's complement negative is $1100\ 0110_2$, which is represented by $C6_{16}$. (Its 1's complement negative is $1100\ 0101_2$, which is $C5_{16}$.)
- **Example 6**: $FAB_{16} = 1111\ 1010\ 1011_2$, whose 2's complement negative = $0000\ 0101\ 0101_2 = 055_{16}$.

 © James Sasaki, 2016

- More generally, let N be a hex number that represents a bitstring X, and let M be the hex number that represents $-X$. How can we get from N to M? The obvious way is to take N, convert to X, calculate $-X$, and then convert to M.
- For 1's complement, we can go from hex M = bitstring X to hex N = bitstring $-X_2$ without calculating X by processing M digit by digit.
  - Let d be a hex digit, then we get the 1's complement of (the bitstring for) d by flipping each bit of (the bitstring for) d.
  - This corresponds to subtracting each bit of d from 1, which corresponds to subtracting $1111_2 - d = 15_{10} - d = \mathrm{F}_{16} - d$.
  - **Definition**: 15 - d is the **15's complement** of the hex digit d. The 15's complement of a hex string is the result of taking the 15's complement of each hex digit.
  - Recall that we were trying to take M, a hex representation of bitstring X and find N, the hex representation of $-X$.
  - Using 1's complement to interpret X and $-X$, it turns out that the 15's complement of M gives you N.
  - **Example 7**: $3\mathrm{A}_{16} = 0011\ 1010_2$ and the 15's complement of $3\mathrm{A}_{16}$ is $\mathrm{C5}_{16}$ = 1100 0101, and in 1's complement, $-(0011\ 1010) = 1100\ 0101$.
- To use 2's complement when going from X to $-X$, we take the 1's complement negative of X and add 1. So to go from $M = X$ to $N = -X$ using 2's complement negative, we can take the 15's complement of M and add 1.
  - **Definition**: The 16's complement of a hex string is its 15's complement plus 1.
  - **Example 8**: $3\mathrm{A}_{16} = 0011\ 1010_2$ and the 16's complement of $3\mathrm{A}_{16}$ is $\mathrm{C6}_{16}$ = 1100 0110, and in 1's complement, $-(0011\ 1010) = 1100\ 0110$.

G. Taking the (1 or 2's Complement) Negative of Bitstrings Represented Using Octal

- Now let's assume that M is an octal string that represents some bitstring X, and we want to calculate the octal string N that represents $-X$ when we take the negative in (a) 1's complement or (b) 2's complement.

- It turns out that operations involved are similar to those for hexadecimal numbers.
- To go from octal M = bitstring X to octal N = bitstring $-X$ using 1's complement negative, we take the **7's complement** of M by subtracting each octal digit from 7.
  - **Example 9**: The 7's complement of 765_8 is 012_8, $765_8 = 111\ 110\ 101_2$, and using 1's complement, $-(111\ 110\ 101) = 000\ 001\ 010 = 012_8$.
- To take the negative using 2's complement, we take the **8's complement** of M by adding 1 to the 7's complement of M.
  - **Example 10**: The 8's complement of 765_8 is 013_8, $765_8 = 111\ 110\ 101_2$, and using 2's complement, $-(111\ 110\ 101) = 000\ 001\ 011 = 013_8$.

 © James Sasaki, 2016

Octal and Hexadecimal Data

CS 350: Computer Organization & Assembler Language Programming

A. Why?

- Octal and hexadecimal numbers are useful for abbreviating long bitstrings.

B. Outcomes

After this activity, you should

- Be able to convert between binary, octal, hexadecimal, and decimal representations of numbers and to take the 1's and 2's complement of octal and hexadecimal numbers (when read as standing for bitstrings).

C. Problems

1. Let *X* be the bitstring 101101101100.
   a. What is the hex representation of *X*?
   b. What is the octal representation of *X*?
   c. What decimal value does *X* represent as an unsigned binary number?
   d. What decimal value does *X* represent in 2's complement?
2. Let *X* be the the bitstring 1101011101.
   a. What decimal value does *X* represent as an unsigned binary number?
   b. What decimal value does *X* represent in 2's complement?
   c. What the hex representation of *X*? (Pad *X* with leading 0's to get to the next multiple of 4 bits.)
   d. What is the octal representation of *X*? (Pad *X* with leading 0's to get to the next multiple of 3 bits.)
3. Let *X* be the bitstring represented by hexadecimal FAB.
   a. What is *X*?
   b. What decimal value does FAB_{16} represent as an unsigned integer?
   c. What is $-X$ in 2's complement and what does $-X$ equal in decimal?
   d. What decimal value does *X* represent in 2's complement?

 © James Sasaki, 2016

e. What is the hex representation of $-X$?

f. What are the 15's and 16's complement of FAB_8?
(The 16's complement should match your answer to part (e).)

4. Let X be the bitstring represented by 735_8.

a. What is X?

b. What decimal value does X represent as an unsigned integer?

c. What is $-X$ in 2's complement and what does $-X$ equal in decimal?

d. What decimal value does X represent in 2's complement?

e. What is the octal representation of $-X$?

f. What are the 7's and 8's complement of FAB_8?
(The 8's complement should match your answer to part (e).)

5. Let X be the 12-bit string represented by 1610_8. Let Y be the 10-bit string we get if we truncate the two leftmost bits from X.

a. What are X and Y?

b. What is $-Y$ in 2's complement and what does $-X$ equal in decimal?

c. What decimal value does Y represent in 2's complement?

d. Turning back to X, what are its 7's and 8's complements?

6. Let Z be the 12-bit "signed" extension of Y from the previous problem: To get Z, copy the sign bit of Y twice and prepend them to Y.

a. What is Z and what is its octal representation?

b. What is $-Z$ in 2's complement? What is its octal representation?

c. What are the 7's and 8's complement of the octal representation of Z?
(The 8's complement should match the octal representation of $-Z$.)

 © James Sasaki, 2016