CSE 31
Computer Organization

Lecture 12 – Integer Representation (wrap up)

Announcements

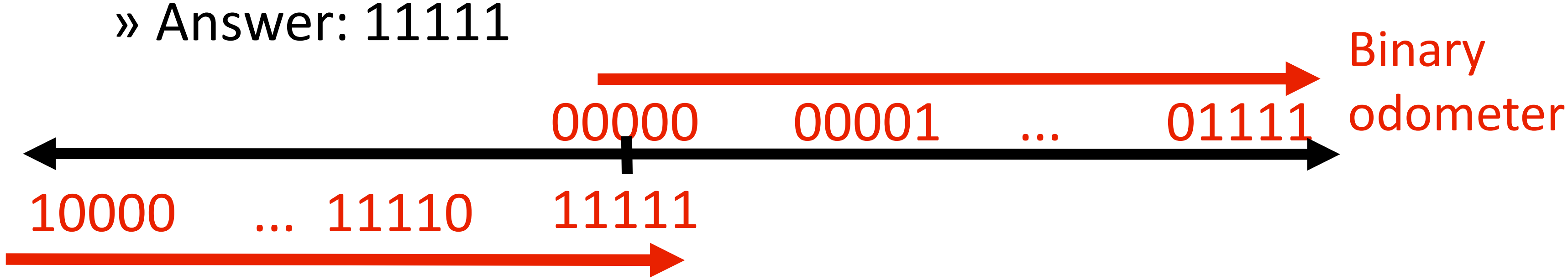
- Labs
  - Lab 4 grace period ends this week
  - Lab 5 out this week
    - » Due at 11:59pm on the same day of your next lab (with 7 days grace period after due date)
    - » You must demo your submission to your TA within 14 days from posting of lab
    - » Demo is REQUIRED to receive full credit
    - » **No penalty** for submission after due date but before end of grace period.
- Reading assignments
  - Reading 03 (zyBooks 3.1 – 3.7, 3.9) due 06-MAR
    - » Complete **Participation Activities** in each section to receive grade
    - » IMPORTANT: Make sure to submit score to CatCourses by using the link provided on CatCourses
- Homework assignment
  - Homework 03 (zyBooks 3.1 – 3.7, 3.9) due 13-MAR
    - » Complete **Challenge Activities** in each section to receive grade
    - » IMPORTANT: Make sure to submit score to CatCourses by using the link provided on CatCourses

Announcements

- Project 01
  - Due 17-MAR
  - Can work in teams of 2 students
    - » Each team member must identify teammate in “Comments…” text-box at the submission page
    - » If working in teams, each student must submit code (can be the same as teammate) and demo individually
    - » Grade can vary among teammates depending on demo
  - Demo required for project grade
    - » No partial credit for submission without demo
  - No grace period
    - » Must complete submission and demo by due date.
- Midterm 01
  - See Announcements 12 and 13 on CatCourses for details

One's Complement (review)

- Complement the bits
  - Example: $7_{10} = 00111_2 \quad -7_{10} = 11000_2$
  - Called ***One's Complement***
  - Note: positive numbers have leading 0s, negative numbers have leadings 1s.
  - What is -00000?
    - » Answer: 11111



  - How many positive (including +0) numbers in N bits? 2^{N-1}
  - How many negative (including -0) numbers? 2^{N-1}

Shortcomings of One’s complement?

- Arithmetic is less complicate than sign & magnitude.

- Still two zeros
  - 0x00000000 = $+0_{ten}$
  - 0xFFFFFFFF = -0_{ten}

- Although used for a while on some computer products, one’s complement was eventually abandoned because another solution was better.

Standard Negative # Representation

- Problem is the negative mappings “overlap” with the positive ones (the two 0s). Want to shift the negative mappings left by one.
  - Solution! For negative numbers, complement, then add 1 to the result
- As with sign and magnitude, & one’s complement, leading 0s → positive, leading 1s → negative
  - 000000...xxx is ≥ 0, 111111...xxx is < 0
  - except 1…1111 is -1, not -0
- This representation is ***Two’s Complement***
- This makes the hardware simple!

In C: short, int, long, intN_t (C99) are all signed integers.

Two's Complement Formula

- Can represent positive and negative numbers in terms of the bit value times a power of 2:

  $d_{31} * -(2^{31}) + d_{30} * 2^{30} + \ldots + d_2 * 2^2 + d_1 * 2^1 + d_0 * 2^0$

- Example: 1101_{two}

  $= 1x\text{-}(2^3) + 1*2^2 + 0*2^1 + 1*2^0$

  $= -2^3 + 2^2 + 0 + 2^0$

  $= -8 + 4 + 0 + 1$

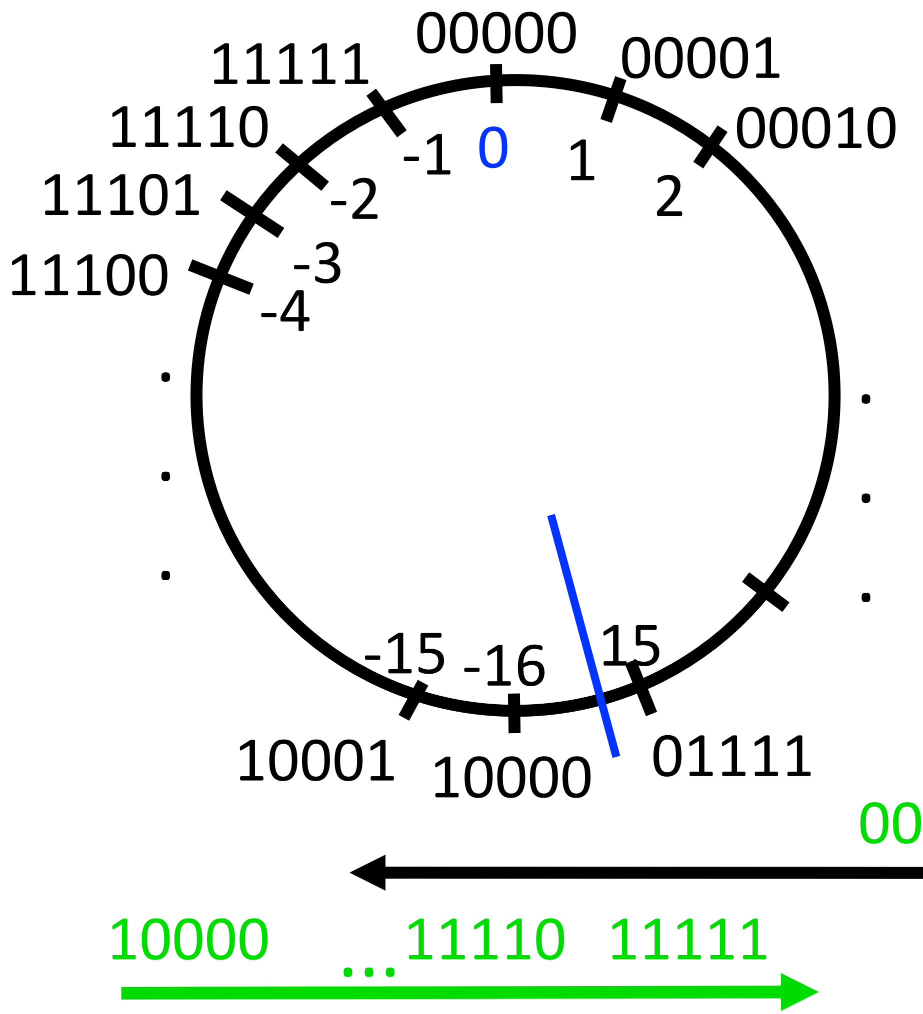
  $= -8 + 5$

  $= -3_{ten}$

Example: -3 to +3 to -3:

x :	1101_{two}	(-3)
x':	0010_{two}	
+1:	0011_{two}	(3)
()':	1100_{two}	
+1:	1101_{two}	(-3)

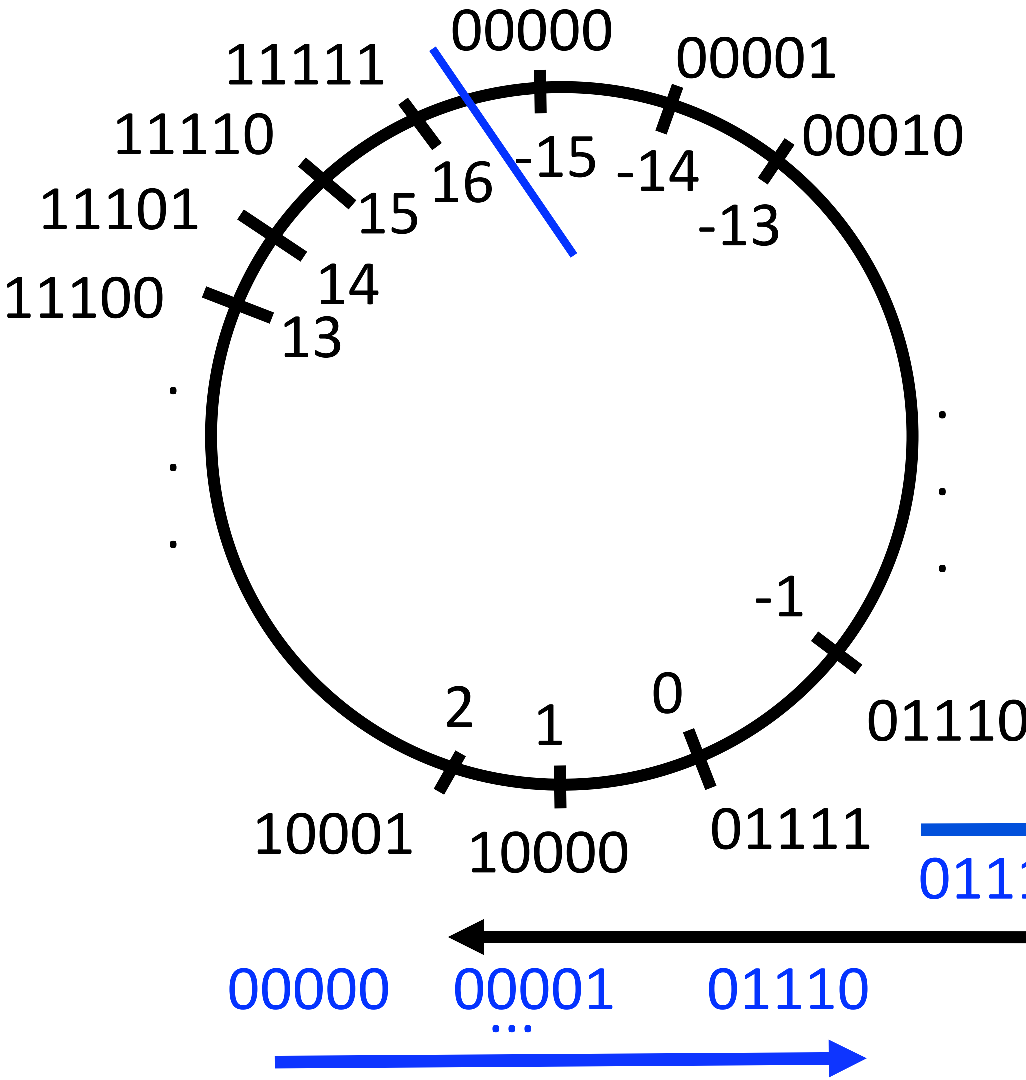
2’s Complement Number “line”: N = 5



- 2^{N-1} non-negatives (includes zero)
- 2^{N-1} negatives
- one zero
- how many positives?
  - $2^{N-1} - 1$

Note: 1’s and 2’s complements are used to represent negative numbers only!

Bias Encoding: N = 5 (bias = 15)

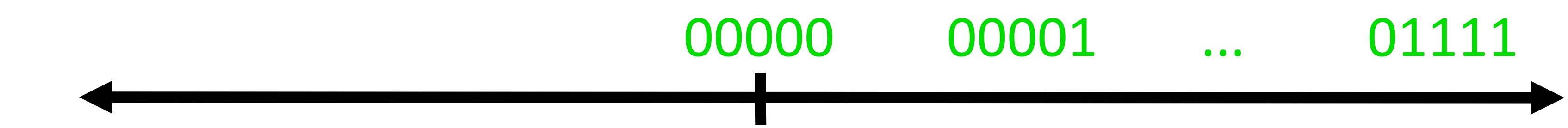


- Want 00… to represent the smallest number
- value = unsigned - bias
- Bias for N bits = $2^{N-1} – 1$
- one zero
- how many positives?
  - 2^{N-1}
  - (more than 2's complement)

Summary

- We represent "things" in computers as particular bit patterns:
  - N bits → 2^N things
- Different integer encodings have different benefits; 1s complement and sign/mag have most problems.
- unsigned (C99's `uintN_t`) :

- 2's complement (C99's `intN_t`): universal, learn it!



- Overflow: numbers ∞ ; computers finite → errors!