CSE 31
Computer Organization

Lecture 11 – Integer Representation (cont.)

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 02 (zyBooks 2.1 – 2.9) due **tonight**, 27-FEB and 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.

How Many Bits for Representation (review)

- Characters?
  - 26 letters → 5 bits ($2^5 = 32$)
  - upper/lower case + punctuation
    → 7 bits (in 8 bits) (“ASCII”)
  - standard code to cover all the world’s languages → 8-, 16-, 32-bits (“Unicode”) www.unicode.com

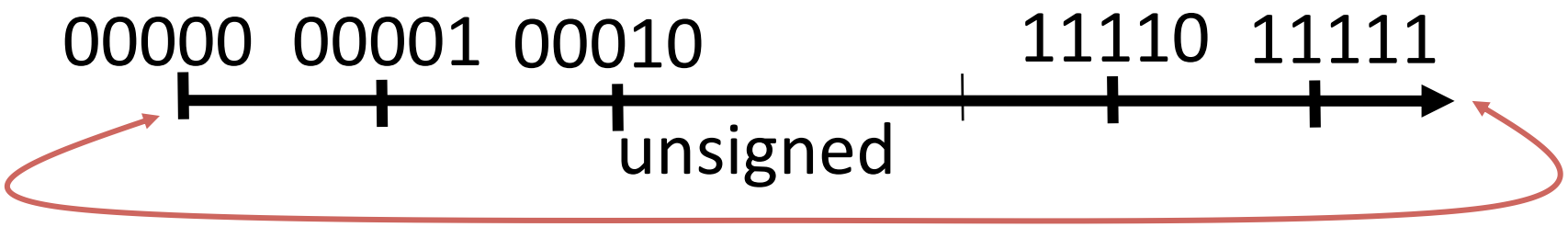
- Logical values?
  - 0 → False, 1 → True

- Color? Ex: *Red (00)* *Green (01)* *Blue (11)*

- Remember: N bits → at most 2^N things

What if too big?

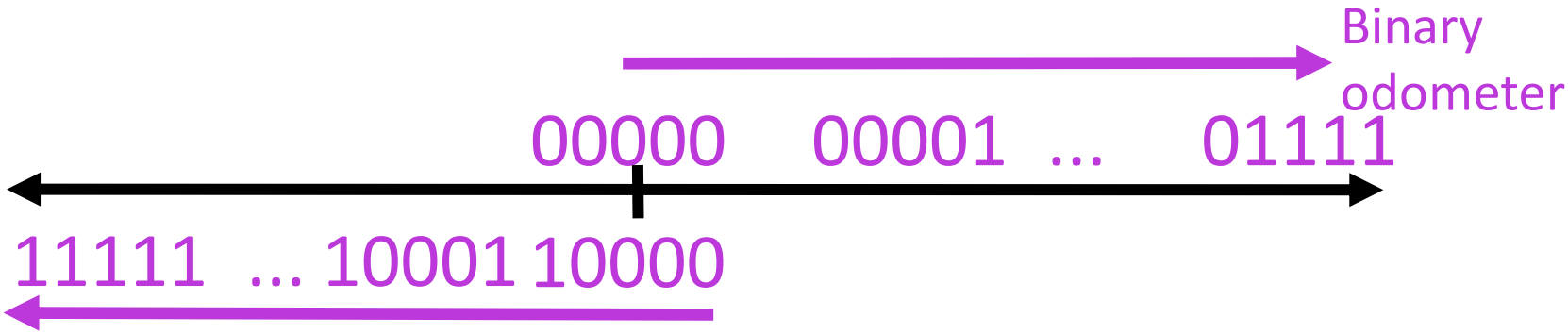
- Binary bit patterns are simply representatives of numbers. Strictly speaking they are called "numerals"
- Numbers really have an ∞ number of digits
  - with almost all being same (00…0 or 11…1) except for a few of the rightmost digits
  - Just don't normally show leading digits
- If result of add (or -, *, /) cannot be represented by these rightmost HW bits, ***overflow*** is said to have occurred.



Negative Numbers

- So far, ***unsigned numbers***

- Obvious solution: define leftmost bit to be sign!
  - 0 → + , 1 → –
  - Rest of bits can be numerical value of number
- Representation called ***sign and magnitude***

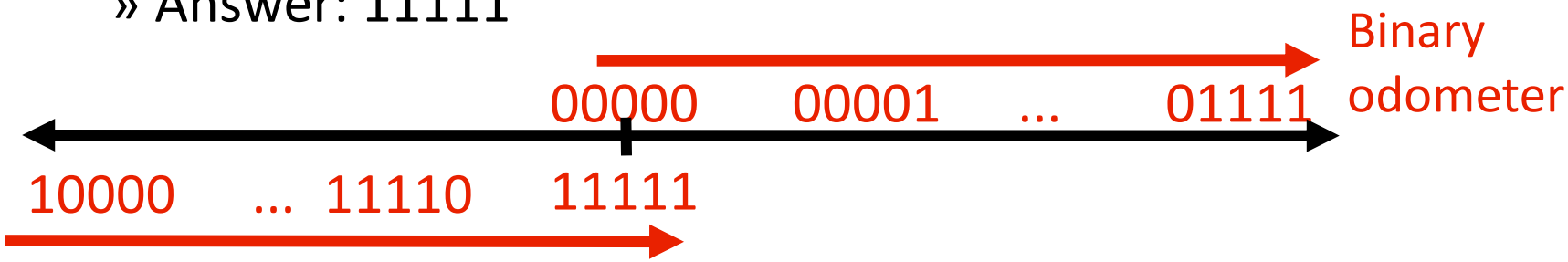


Shortcomings of Sign Magnitude?

- Arithmetic circuit complicated
  - Special steps depending whether signs are the same or not
- Also, two zeros
  - 0x00000000 = $+0_{ten}$
  - 0x80000000 = -0_{ten}
  - What would two 0s mean for programming?
- Also, incrementing "binary odometer", sometimes increases values, and sometimes decreases!
- Therefore, sign and magnitude abandoned

Another try

- Complement the bits
  - Example: $7_{10} = 00111_2$ $-7_{10} = 11000_2$
  - 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}