# lec 2024/08/29

- binary: get more precision over n-ary or smth
- and (&), or (|), not (~), xor (^)
- shifts
  - `x << y`
    - throw away extra bits at left
    - fill with 0s on right
  - `x >> y`
    - throw away extra bits at left
    - logical shift: fill with 0s on left
    - arithmetic shift: replicate sign bit on left
    - *undefined*: shift amtn $< 0$ or $\geq$ word size
- `-x = ~x + 1` in twos complement
  - but if `x = Tmin` (most negative two's complement), you get back `Tmin`
- mix of signed and unsigned in expression eg ==? implicit casted and evaled in unsigned.