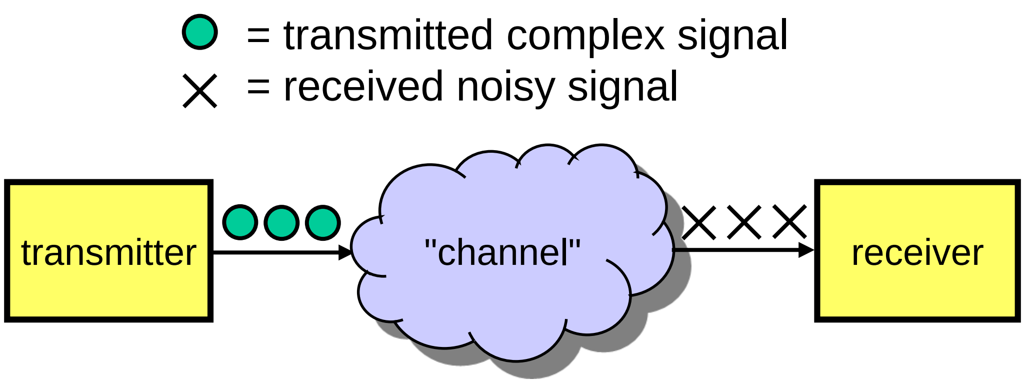
COMPLEX ADDITION, MULTIPLICATION, ROTATION, AND RECT ↔ POLAR CONVERSION

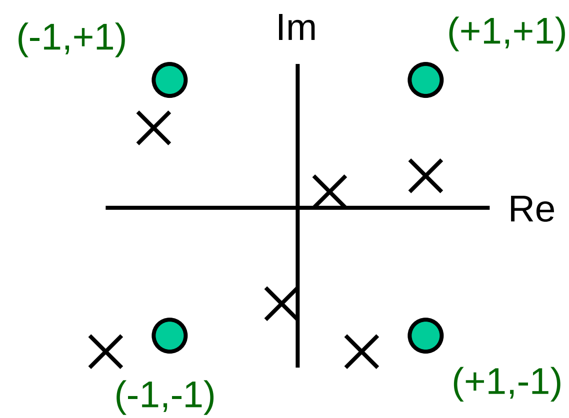
Complex Numbers

- Complex numbers are commonly used in DSP applications
- *real* + j *imag* (common in EE)
  real + i *imag* (common in math)
  $i = j = \text{sqrt}(-1)$
- rectangular notation: *Re* + j *Im*
- polar notation: *Mag*, *Angle* (θ)
- Re = I = "In-Phase"
  Im = Q = "Quadrature"
  Yes, it is unfortunate that Im ≠ I
- Complex numbers and negative frequencies are "in the math"
- (Convenient to draw in 2-dimensions with *real* and *imag* axes)

Polar Notation

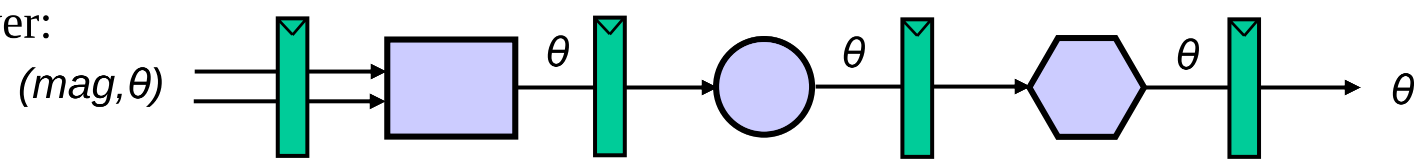
- Sometimes the phase of a signal is what is important
- Example: QPSK modulation (encodes 2 bits/symbol)





- As a rule, convert to a system which eliminates un-useful information as soon as possible

Receiver:



Complex Addition

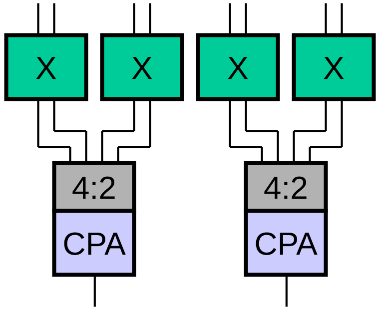
- Best done in rectangular notation
  - There isn't a way to add numbers in polar notation in a hardware-efficient manner
- $(A + jB) + (C + jD)$
  $$= (A + C) + j(B + D)$$
- 2 adds

Complex Multiplication, Polar

- Easiest in polar notation
- $(A_{mag}, \theta_A) \times (B_{mag}, \theta_B)$
  $= A_{mag} \times B_{mag}, (\theta_A + \theta_B)$
- 1 multiply, 1 add

Complex Multiplication, Rectangular

1. Straightforward approach
   - $(A + jB) \times (C + jD)$
     $= AC + j^2\, BD + j\, AD + j\, BC \qquad (j^2 = -1)$
     $= (AC - BD) + j(AD + BC)$
   - 4 multiplies, 2 "adds"
   - Critical path
     - product_real = 2 mults > ~(BD) > 4:2 > CPA
     - product_imag = 2 mults > 4:2 > CPA

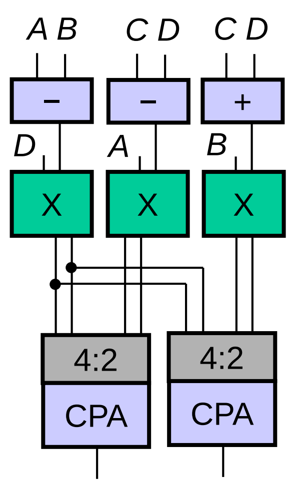


Complex Multiplication, Rectangular

2. A second approach
   - $real = (A - B)\,D + A\,(C - D)$
     $imag = (A - B)\,D + B\,(C + D)$
   - Check the equations to verify they are correct
     $real = (A - B)\,D + A\,(C - D)$
     $= AD - BD + AC - AD$
     $= AC - BD$ ☺ the same answer as method #1
     $imag = (A - B)\,D + B\,(C + D)$
     $= AD - BD + BC + BD$
     $= AD + BC$ ☺ the same answer as method #1
   - 3 multiplies, 5 "adds"

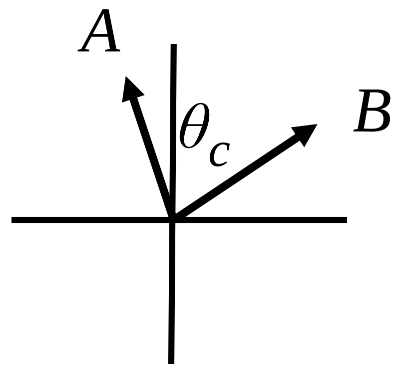
Complex Multiplication, Rectangular

2. A second approach (continued)
   – Drawbacks to this algorithm
     - There are two CPA adder/subtractors in the critical path
     - With *N*-bit inputs, all three multipliers are almost certainly going to need one input that is one bit wider than *N*; i.e., *N*-bit × (*N*+1)-bit multipliers



Complex Multiplication, Rotations

3. Multiplication by a (fixed) complex number C
   - $A = B \times C$ where C is magnitude 1 and fixed θ
   - $A = B \times 1.0\angle\theta$



   - $C = \cos\theta + j \sin\theta$
   - This changes only the *phase* of B
   - Functions of θ may be precomputed and stored

Complex Rotations

a) Straightforward method

- $A_{real} = B_{real} \cos\theta - B_{imag} \sin\theta$
  $A_{imag} = B_{imag} \cos\theta + B_{real} \sin\theta$
- 4 multiplies, 2 adds
- Critical path (see notes)

Complex Rotations

b) Golub's method

- $A_{real} = (B_{real} + B_{imag})(\cos\theta - \sin\theta) + B_{real}\sin\theta - B_{imag}\cos\theta$
  $A_{imag} = B_{imag}\cos\theta + B_{real}\sin\theta$
- 3 multiplies
  - All should have carry-save outputs
- 5 (4) additions/subtractions
  - Better yet, build it with:
    - A single carry-save adder followed by a carry-propagate adder for the three terms of A_{real}
    - 3 more carry-propagate adders
- Critical path (see notes)

Complex Rotations

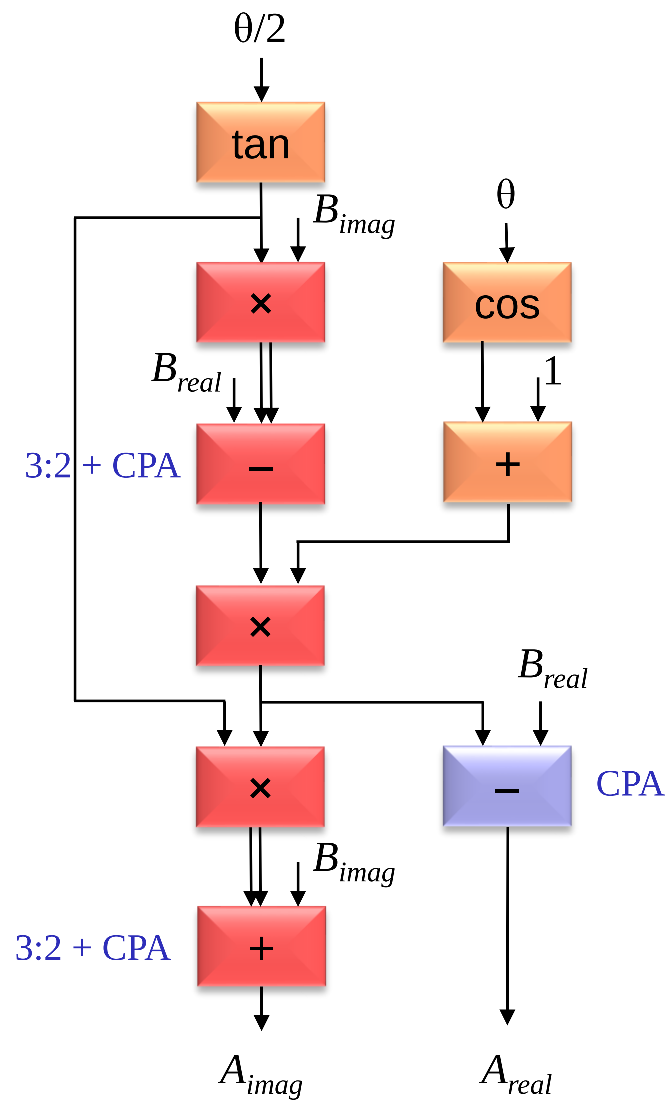
c) Buneman's method

- $A_{real} = [(1 + \cos\theta)(B_{real} - B_{imag} \tan\theta/2)] - B_{real}$
  $A_{imag} = [(1 + \cos\theta)(B_{real} - B_{imag} \tan\theta/2)]$
  $\times \tan\theta/2 + B_{imag}$
- 3 multiplies, 3 adds
- Best for angles where $\tan\theta/2$ is not too large

Complex Rotations

c) Buneman's method – critical path analysis

- Function on the critical path
- Function not on the critical path if θ is fixed
- Function not on the critical path
  - A more accurate hardware total for an efficient implementation with fixed θ:
    - 1 mult, single word output
    - 2 mults, carry-save output
    - 3 carry-propagate adds
    - 2 rows of 3:2 adders

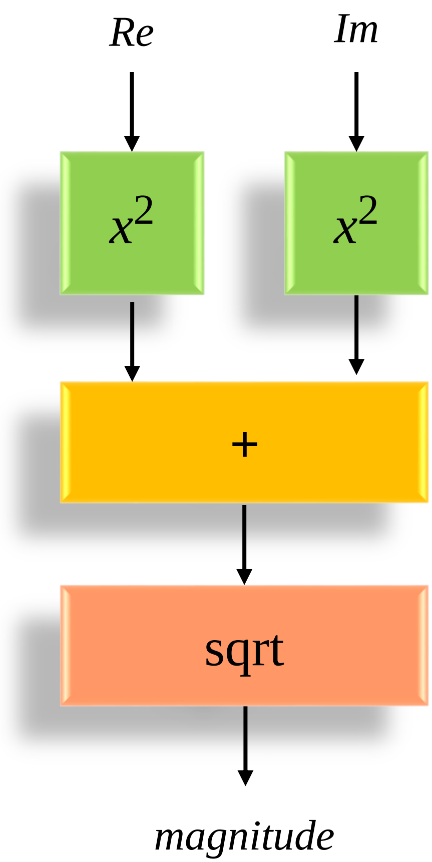


Polar to Rectangular Conversion

- $real = magnitude \times \cos(\theta)$
  $imag = magnitude \times \sin(\theta)$
- Hardware
  - 1 cos and 1 sin (could be expensive)
  - 2 multiplies

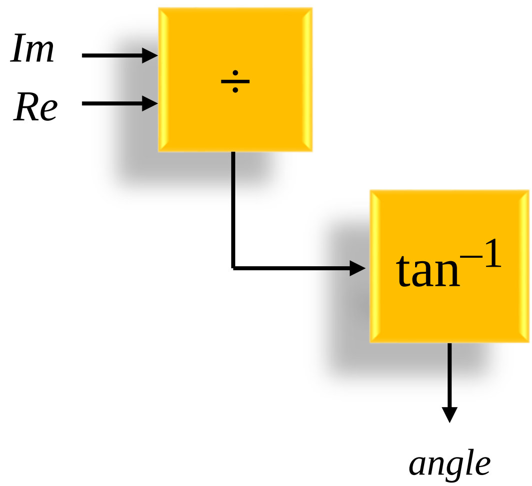
Rectangular to Polar Conversion

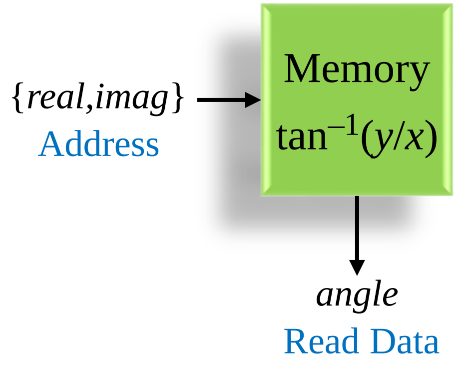
- *magnitude* = sqrt($real^2 + imag^2$)
  angle = $\tan^{-1}(imag/real)$
- *magnitude* hardware
  - 2 squares (~equivalent to 1 multiply if optimized)
  - 1 addition
  - 1 square root (could be expensive)



Rectangular to Polar Conversion

- $magnitude = \text{sqrt}(real^2 + imag^2)$
  $angle = \tan^{-1}(imag/real)$
- *angle* Hardware
  - 1 $\tan^{-1}$ (could be expensive)
  - 1 division possibly (probably not)
    - If implemented with lookup table(s) and the *imag/real* quotient is not needed elsewhere, the cascaded divide and $\tan^{-1}$ functions should be implemented as a single merged function: $\tan^{-1}(y/x)$ addressed by *real* and *imag*
      - $address = \{real, imag\}$





Rectangular to Polar Conversion

- The *magnitude* calculation can be simplified in a similar manner
  - If the square root will be implemented with a lookup table...then it is worth considering if the address into the table should be {*real*, *imag*} rather than sqrt($real^2 + imag^2$)
  - Need to think about word widths required

Rectangular vs. Polar Key Point Summary

- Addition/Subtraction easiest in rectangular
- Multiplication easiest in polar
- Non-trivial to convert between the two
- Rules of thumb
  - Keep in rectangular
  - If doing many complex multiplications or if only phase or magnitude is important, then consider moving to polar