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

• Complex numbers are commonly used in DSP applications

• \(\text{real} + j\ \text{imag}\) (common in EE)
  \(\text{real} + i\ \text{imag}\) (common in math)
  \(i = j = \sqrt{-1}\)

• rectangular notation: \(\text{Re} + j\ \text{Im}\)

• polar notation: \(\text{Mag}, \text{Angle}\ (\theta)\)

• \(\text{Re} = \text{I} = \text{"In-Phase"}\)
  \(\text{Im} = \text{Q} = \text{"Quadrature"}\)
  Yes, it is unfortunate that \(\text{Im} \neq \text{I}\)

• Complex numbers and negative frequencies are “in the math”

• (Convenient to draw in 2-dimensions with \(\text{real}\) and \(\text{imag}\) 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 useless information as soon as possible

Receiver:

(mag,θ) — θ — θ — θ — θ

transmitter

"channel"

receiver

= transmitted complex signal
× = received noisy signal

© B. Baas
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
- \((A_{\text{mag} \theta_A}) \times (B_{\text{mag} \theta_B}) = A_{\text{mag}} \times B_{\text{mag}} (\theta_A + \theta_B)\)
- 1 multiply, 1 add
1. Straightforward approach
   - \((A + jB) \times (C + jD)\)
     \[= AC + j^2 BD + j AD + j BC\]  \( (j^2 = -1)\)
     \[= (AC - BD) + j(AD + BC)\]
   - 4 multiplies, 2 “adds”
   - Critical path
     • product_real = 2 mults > ~(BD) > 4:2 > CPA
     • product_imag = 2 mults > 4:2 > CPA
2. A second approach

- $\text{real} = (A - B) \, D + A \,(C - D)$
  
- $\text{imag} = (A - B) \, D + B \,(C + D)$

- Check the equations to verify they are correct

  $\text{real} = (A - B) \, D + A \,(C - D)$
  
  $= AD - BD + AC - AD$
  
  $= AC - BD \circ$ the same answer as method #1

  $\text{imag} = (A - B) \, D + B \,(C + D)$
  
  $= AD - BD + BC + BD$
  
  $= AD + BC \circ$ the same answer as method #1

- 3 multiplies, 5 “adds”
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 $\times$ (N+1)-bit multipliers
3. Multiplication by a (fixed) complex number $C$
   - $A = B \times C$ where $C$ is magnitude 1 and fixed $\theta$
   - $A = B \times 1.0 / \theta$

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

a) Straightforward method

\[ A_{\text{real}} = B_{\text{real}} \cos \theta - B_{\text{imag}} \sin \theta \]
\[ A_{\text{imag}} = B_{\text{imag}} \cos \theta + B_{\text{real}} \sin \theta \]

– 4 multiplies, 2 adds
– Critical path (see notes)
b) Golub’s method

- \[ A_{\text{real}} = (B_{\text{real}} + B_{\text{imag}}) \cos \theta - B_{\text{imag}} \sin \theta + B_{\text{real}} \sin \theta - B_{\text{imag}} \cos \theta \]
- \[ A_{\text{imag}} = B_{\text{imag}} \cos \theta + B_{\text{real}} \sin \theta \]
- 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 \( A_{\text{real}} \)
    - 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
c) Buneman’s method – critical path analysis

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

- \( \text{real} = \text{magnitude} \times \cos(\theta) \)
- \( \text{imag} = \text{magnitude} \times \sin(\theta) \)

- Hardware
  - 1 cos and 1 sin (could be expensive)
  - 2 multiplies
Rectangular to Polar Conversion

- **magnitude** = $\sqrt{\text{real}^2 + \text{imag}^2}$
  
- **angle** = $\tan^{-1}(\text{imag}/\text{real})$

- **magnitude** hardware
  - 2 squares (~equivalent to 1 multiply if optimized)
  - 1 addition
  - 1 square root (could be expensive)
Rectangular to Polar Conversion

- \( \text{magnitude} = \sqrt{\text{real}^2 + \text{imag}^2} \)
- \( \text{angle} = \tan^{-1}(\text{imag}/\text{real}) \)

- **angle Hardware**
  - 1 \( \tan^{-1} \) (could be expensive)
  - 1 division possibly (probably not)
  - If implemented with lookup table(s)
    and the \( \text{imag}/\text{real} \) 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 \( \text{real} \)
    and \( \text{imag} \)
    - \( \text{address} = \{\text{real}, \text{imag}\} \)
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