GENERATING COMPLEX FUNCTIONS
Generating Complex Functions

- Complex or “arbitrary” functions are not uncommon
- Examples
  - sin, cos, tan
  - tangent$^{-1}$
  - log
  - $e^x$
  - A/D converter correction values
  - RF mixer bias currents
Generating Complex Functions

1) High-precision Numerical Calculations

• Almost certainly requires many clock cycles per calculation
  – 1–2 bits per clock cycle is common. In some cases, more bits/cycle are possible by adding hardware
  – Can regain *throughput* by parallel implementations
  – However *latency* is unavoidable

• Ex: CORDIC (Coordinate Rotation Digital Computer)

• Ex: polynomial expansions, etc.
Generating Complex Functions

2) Lookup Table

A. ROM array memory

- “Real” memory with address decoder, wordlines, bitlines, sense amplifiers, etc.
- Frequently available as macros from the standard cell vendor
- Could be mask-defined at manufacture, one-time programmable with fuses or anti-fuses, or flash non-volatile memory
- Generally compares better with very large tables since ROM cells are among the densest of all CMOS structures and there is a significant amount of overhead circuitry for a small memory
B. Synthesized from standard cell combinational logic

- The “memory” block is implemented by a highly-optimized netlist of combinational logic gates
- Generally compares better with data that is less random (in an entropy information-theory sense) because it results in simpler and smaller logic equations
- See notes describing ROM memories
Input and Output Word Widths and Total Memory Size

• Total memory size
  \[ = 2 \text{address}\_\text{read} \times \text{data} \_\text{width} \]

• The overall best word widths are a complex function of factors such as:
  – Overall system accuracy (e.g., SNR) requirements
  – Effect of word widths of particular signals on the overall system accuracy
  – Choice of numerical algorithms (e.g., table lookup and/or numerical methods)
  – Available SRAM and ROM technologies
Input and Output Word Width Effects

- **Input word width**
  - A narrow-word-width lookup table input increases the quantization granularity
  - Example: \( \cos(\text{theta}[2:0]) \)
Input and Output Word Width Effects

- **Output word width**
  - A narrow-word-width lookup table input increases the quantization granularity
  - Example: $y[2:0] = \cos(\theta)$
matlab for previous plots

• copy, paste, and try it out
In many cases, computation is expressed or can be transformed into cascaded functions.

Example: The angle of a rectangular 2D vector = $\tan^{-1}(y/x)$

A straightforward implementation using lookup tables would use a table for division followed by a table for $\tan^{-1}()$

A better implementation would merge the cascaded functions into a single $\tan^{-1}(y/x)$ function implemented with a single memory.

- Assuming the intermediate result $y/x$ is not needed elsewhere.
- In both cases, the input address is the concatenated address = $\{x, y\}$ or $\{y, x\}$; in fact, the bits from $x$ and $y$ can be mixed arbitrarily although the two examples here are certainly the clearest.