4) Complex rectangular x Complex [-1, +1]

- Similar to Hz

- Two special cases:
  × \([+1, 0]\)
  × \([0, +1]\)

- Solutions

1) Code in 2x format
   - waste ~1 bit

2) Use mixins to bypass nulls when \(\text{Re, Im} \geq 1.0\)

Control - Counters

- Data path \(a \times b\)
- Control \(v\)
- Memory
  - Small HW
  - Large amount of mixing
  - Don't care about area
  - Care about bags
Think

- Draw diagrams
  - block
  - pipeline
  - timing

- Determine registers
- State graph
- Think

Then

- Type verilog
- Test

Moore

![State Machine Diagram]

Not a valid pipelined block diagram
**Counter**

\[ 17 \rightarrow 16 \rightarrow 15 \ldots \rightarrow 0 \]

**Reset**

**FSM**

out = f(count, ...)

\[ 0 \rightarrow 0 \ldots \rightarrow 0 \]
5 Things in virtually every well-designed FSM

1) Default: state_c = state;

2) case (state)

3) $\text{STATE}_x$: begin
   end
   for each state

4) if (reset = 'b01) begin
   end
   at end of always block

5) Instantiate FF register 1) a separate always block
Generating Complex Functions

- sin, cos, tan
- divide
- tan^-1
- log
- e^x
- A/D correction values

1) High precision numerical
   - Often 1-2 bits of error per cycle - iterative
     - Long latency
   - may 4x throughput by parallel data paths
   - CORDIC
   - Polynomial expansions

2) Lookup table
   \[ \text{Total size} = k \cdot 2^{\text{addr-width}} \times \text{data-width} \]
Memories

I. Single bit

II. Array

View 2
1) Read-write
   - SRAM
   - DRAM

2) ROM
   - PROM - pro.
   - std. cells

3) Non-volatile
   - EPROM
   - EEPROM - flash

View 4 - ASIC mems
1) On-chip "macro"
   - Like a huge std cell

2) Synth from std cells (FFs)

3) Off-chip - DRAM, Flash