RAKE Correlator

Roy Ha
Graham Hull
Lakshmi Sreekumar
Jason White

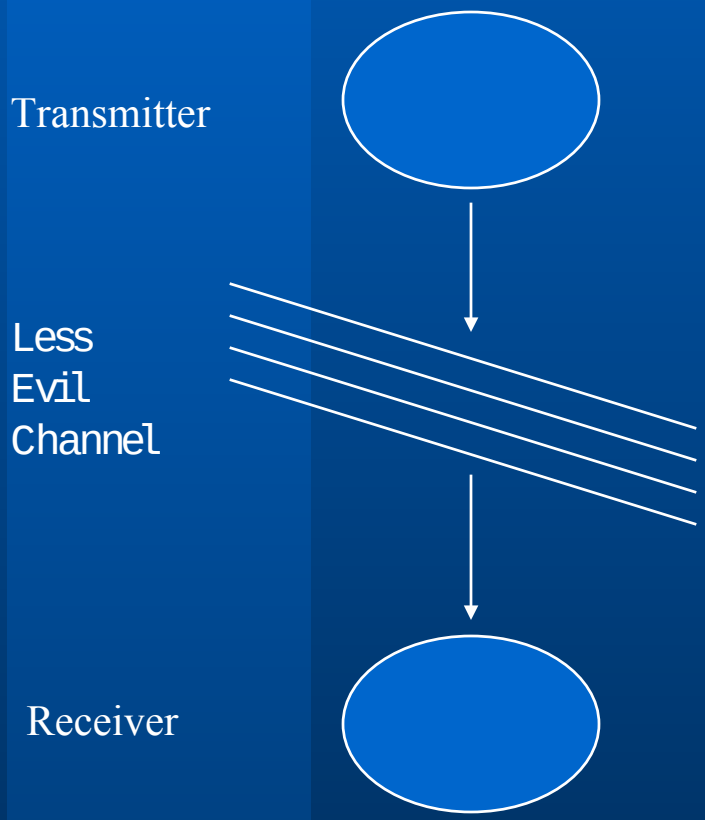
Overview

- Why should you care?
- Channel Model
- What is it? How does it work?
- Our RAKE finger Specs

Why should you care?

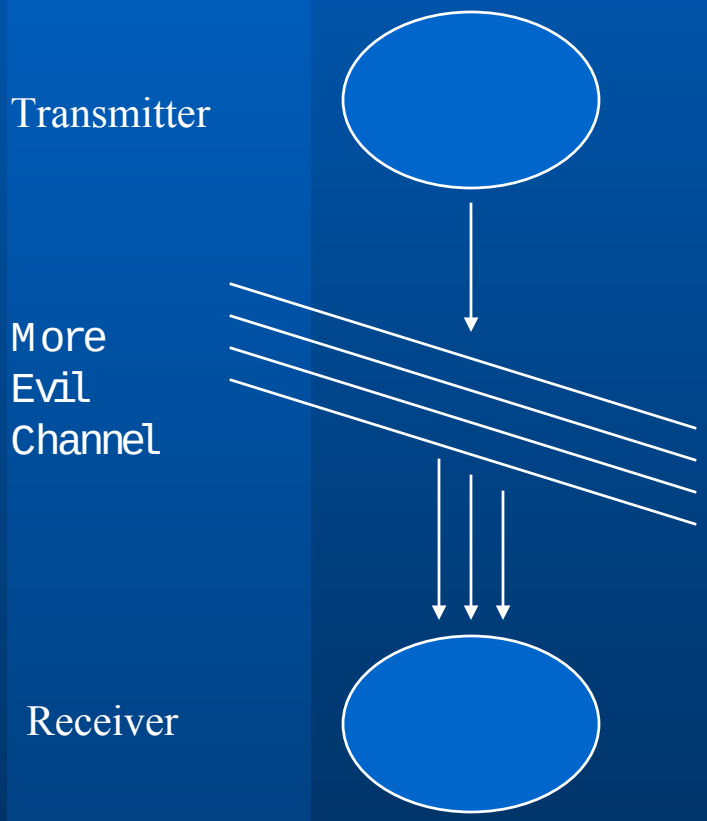
- **Wireless Devices**
  - **DSP**
  - **Vocoder**
  - **RAKE Correlator**

Channel Model



- **Standard, Basic Channel Model**
  - **Additive noise**
  - **Amplitude Changes**
  - **Phase Changes**

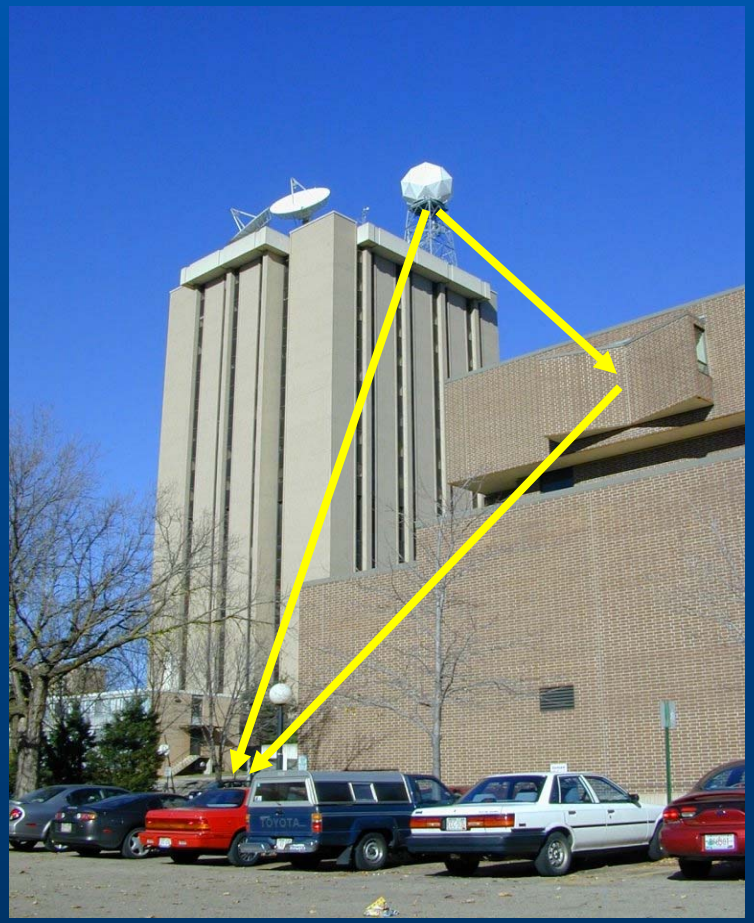
Channel Model



- More “realistic” channel model
  - Additive noise
  - Multiple Paths
    - Amplitude Changes
    - Phase Changes
  - Added Together at receiver

Channel Model

- **Reasons**
  - **Signals bounce off buildings**
  - **Amplitude, phase distortion**
  - **Delayed paths summed with direct path**



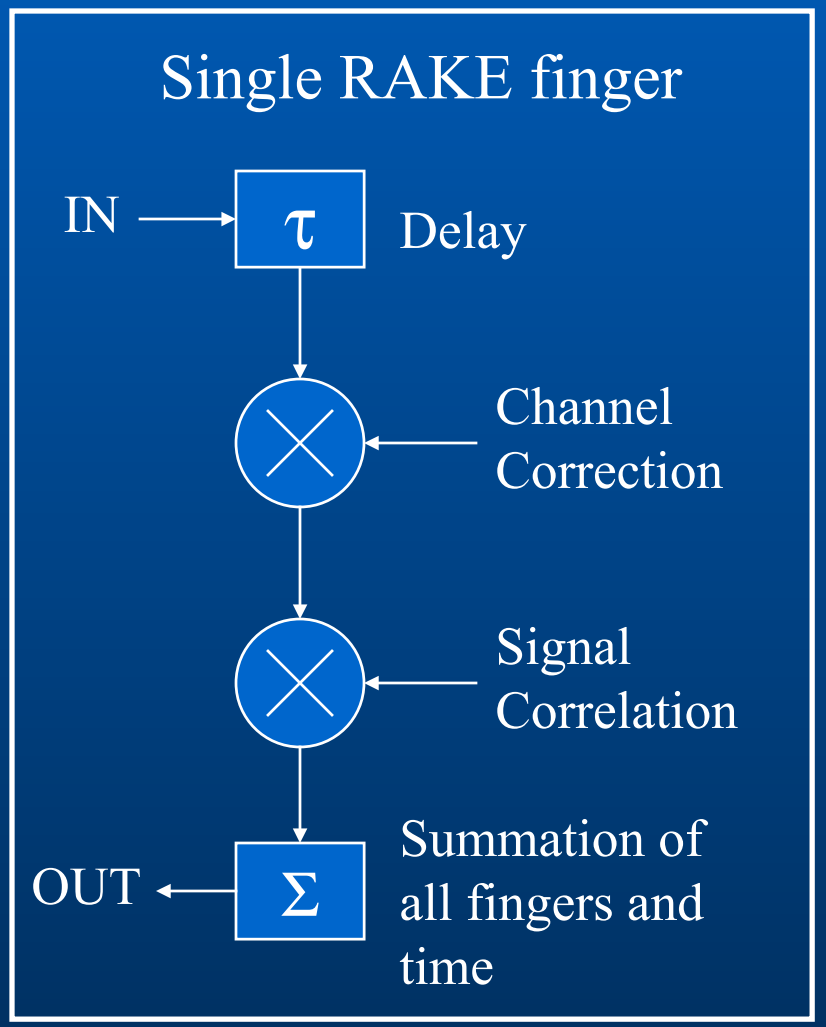
What is it? How does it work?

RAKE = Matched Filter for Channel Model

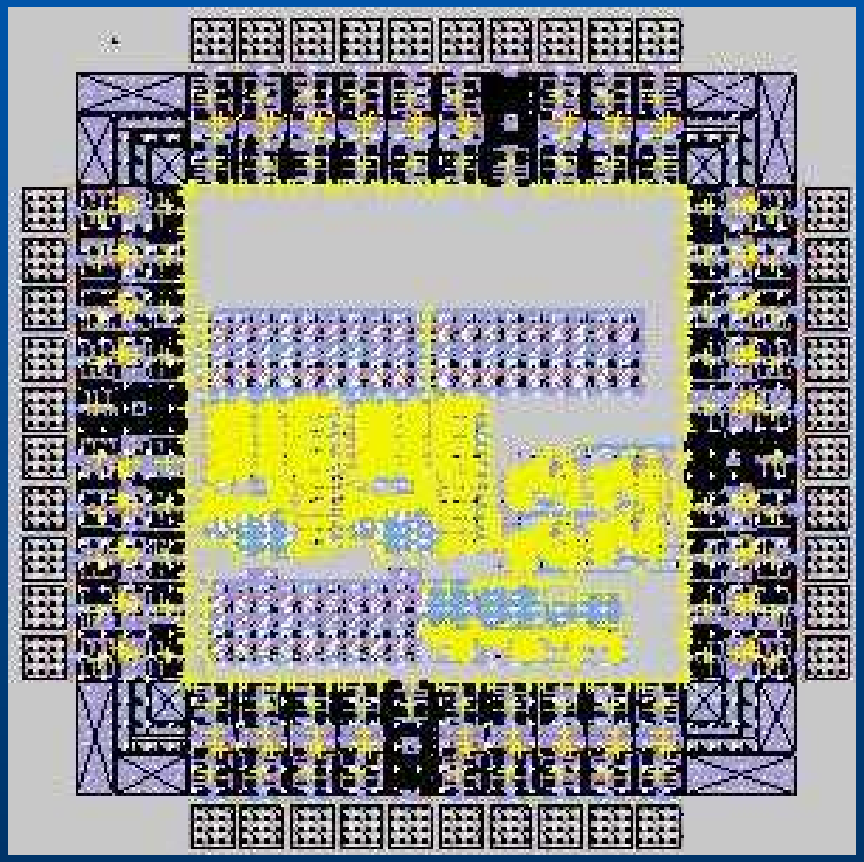
- **RAKE Correlator**
  - **Multiple paths increase performance**
  - **Acts like a set of mini-correlators, called RAKE fingers**
  - **Each finger is delayed and scaled for a given path (ie. A reflection from a building)**

Our specs

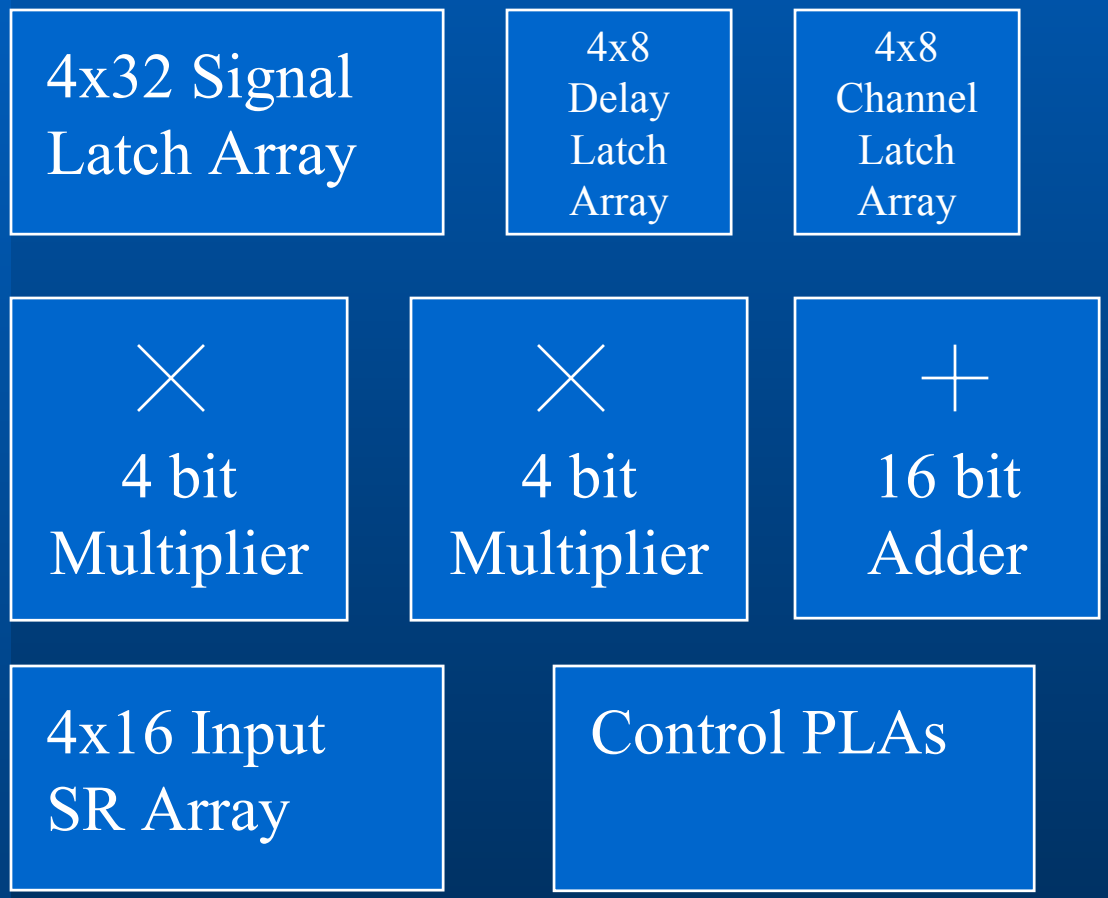
- **Up to 8 RAKE fingers, Max delay of 16 chips**
- **4 bit, 2's complement**
- **32 chips/symbol, 1 bit/symbol**
- **Assume no inter-signal interference**



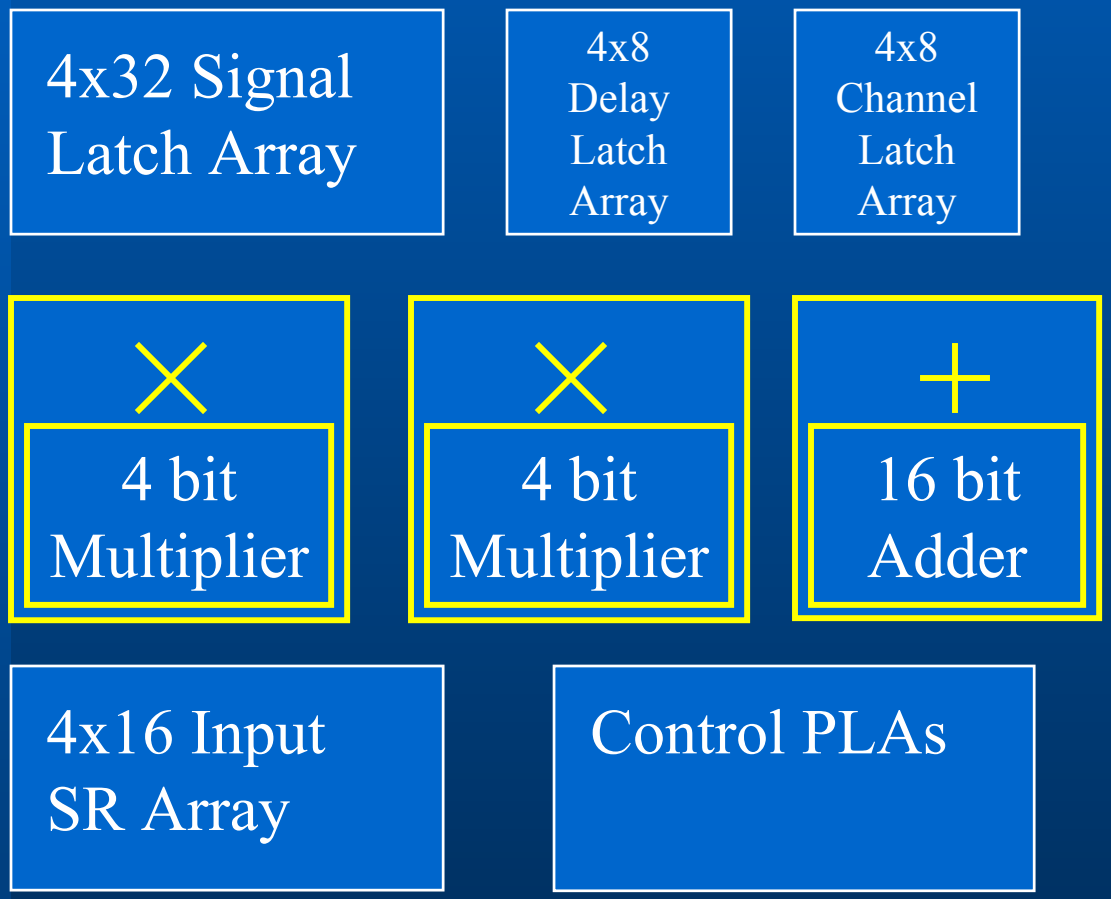
Full Chip



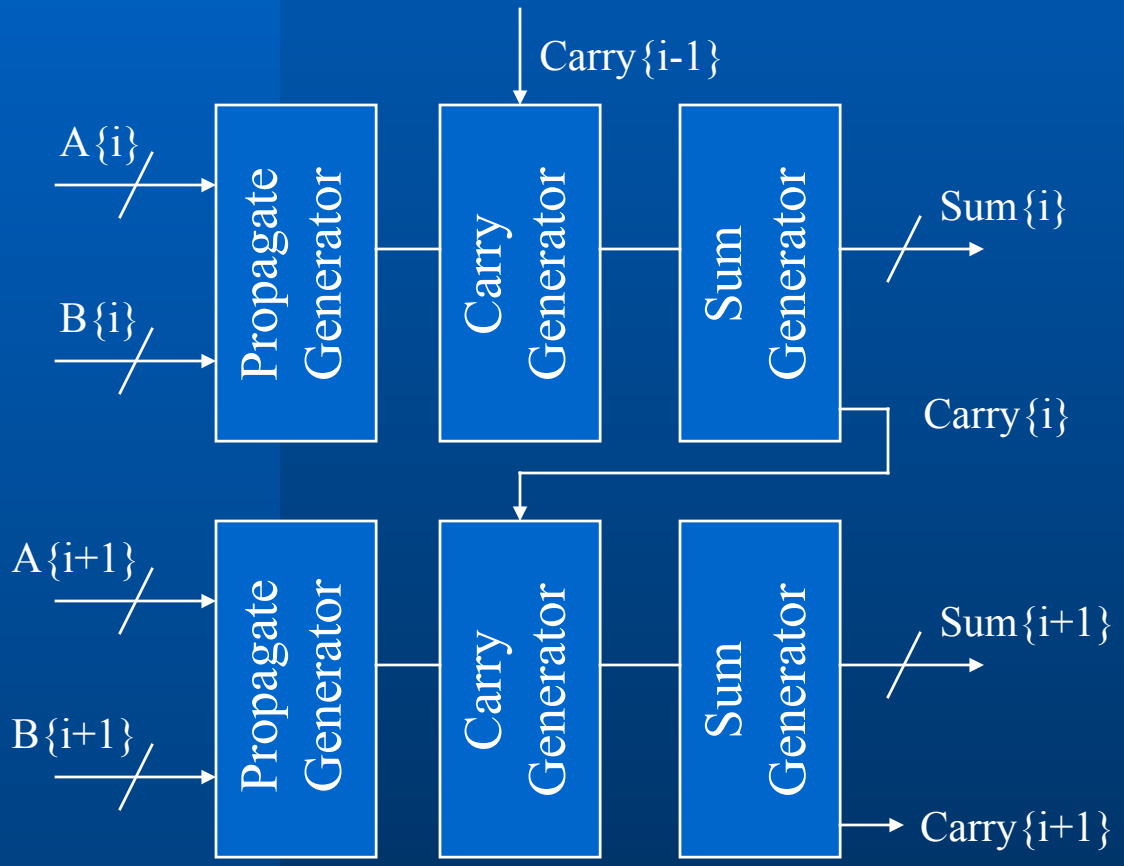
Basic Floorplan



Arithmetic


4x32 Signal Latch Array
4x8 Delay Latch Array
4x8 Channel Latch Array
×
4 bit Multiplier
×
4 bit Multiplier
+
16 bit Adder
4x16 Input SR Array
Control PLAs

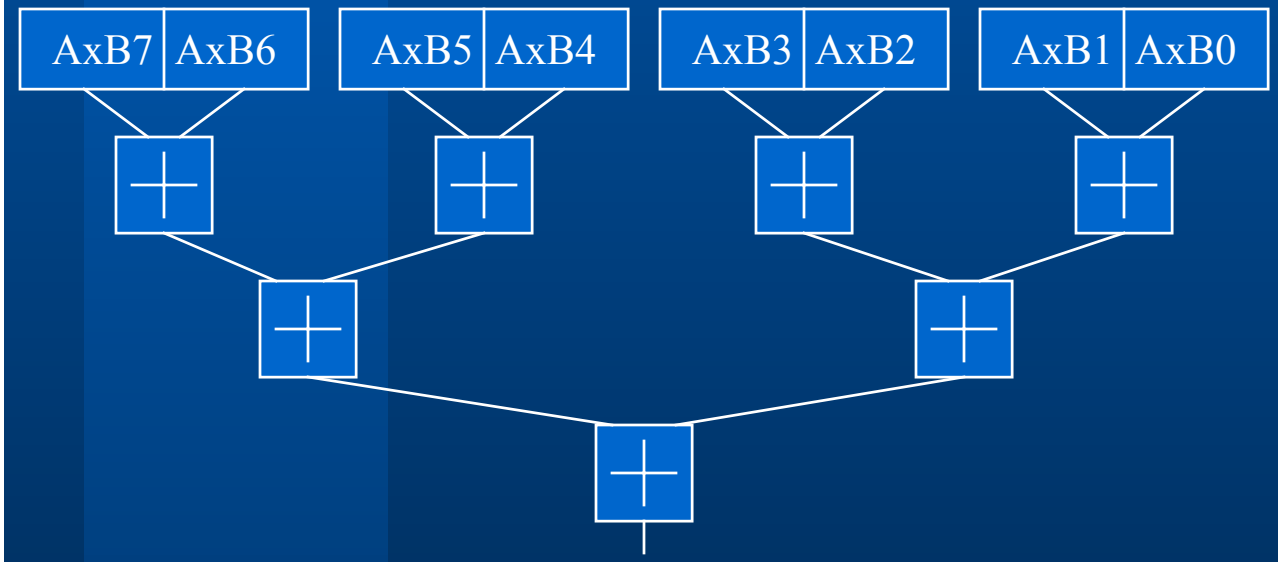
Carry Look-ahead Adder

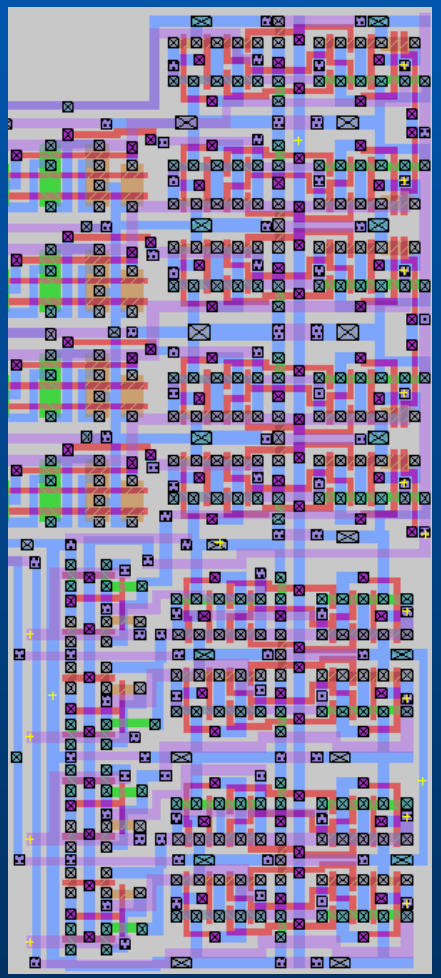


- Carry Look-Ahead is 3x faster than ripple carry
- Carry Eqns
  - Carry{i} = G{i}+P{i} • C{i-1}
  - G{i} = A{i} • B{i}
  - P{i} = A{i}+B{i}
- Used in Multiplier and Summation blocks

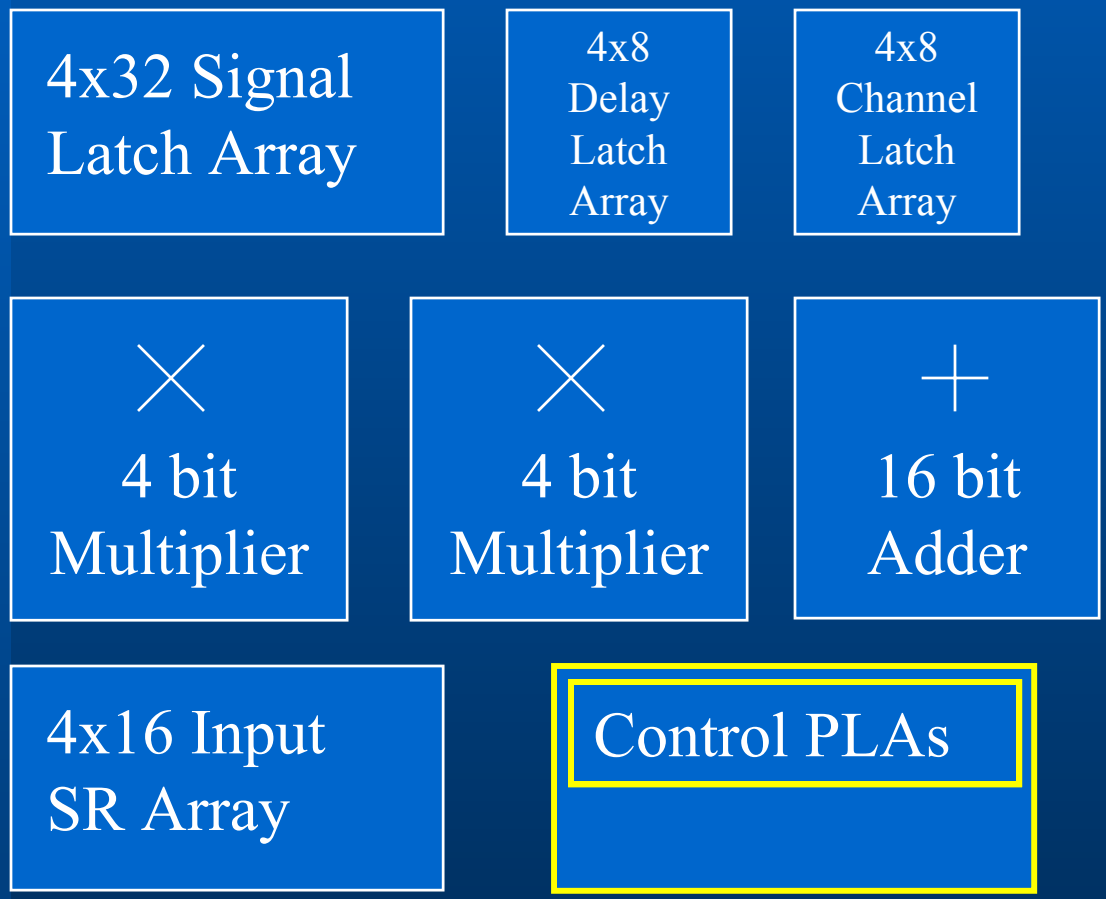
Multiplier

- 1st Multiplier with a Manchester Adder and rounding of bits
- 2nd Multiplier with a CLA

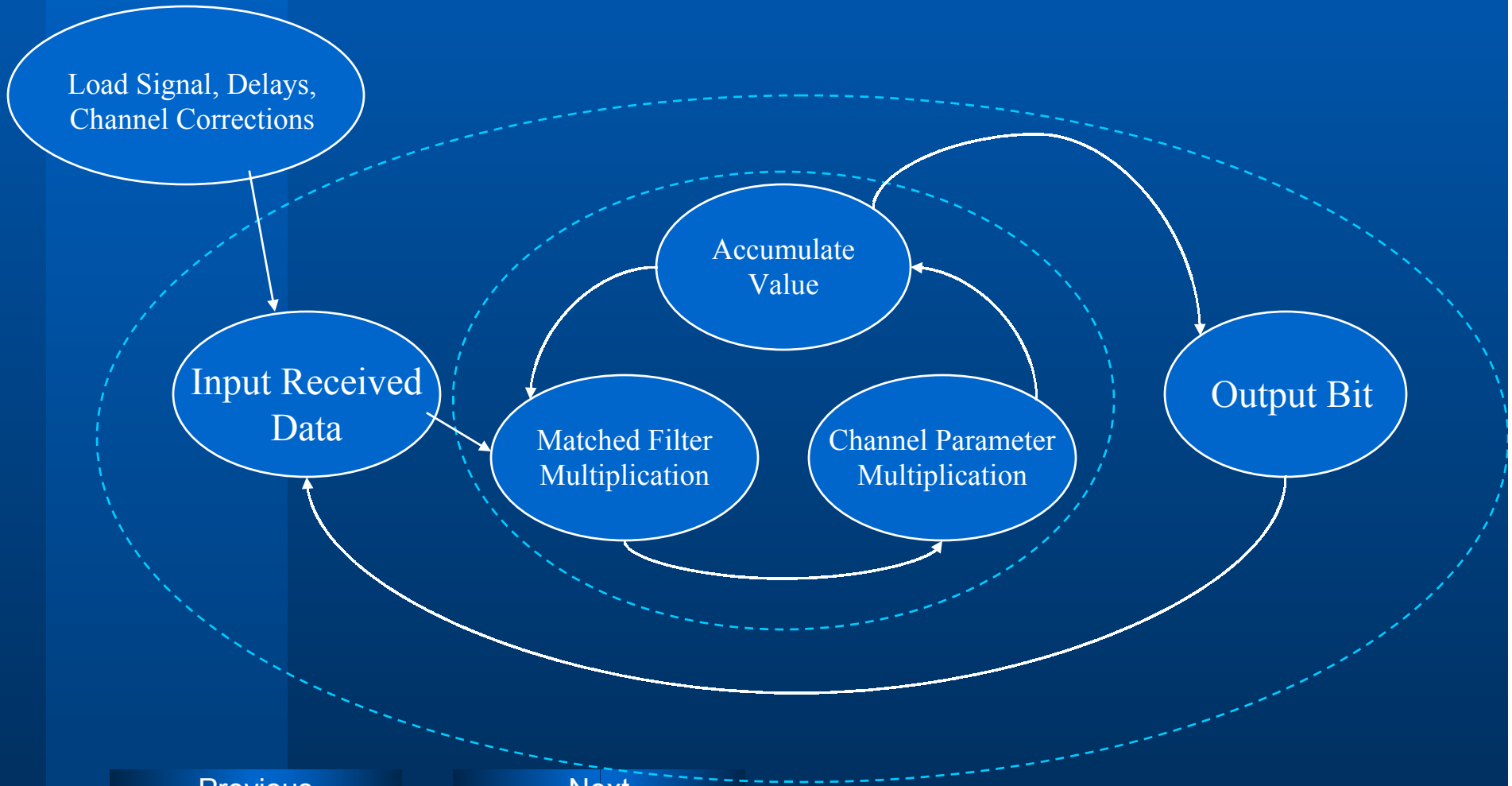




Control


4x32 Signal Latch Array
4x8 Delay Latch Array
4x8 Channel Latch Array
4 bit Multiplier
4 bit Multiplier
16 bit Adder
4x16 Input SR Array
Control PLAs

Control


Load Signal, Delays, Channel Corrections
Accumulate Value
Input Received Data
Matched Filter Multiplication
Channel Parameter Multiplication
Output Bit

Timing Control

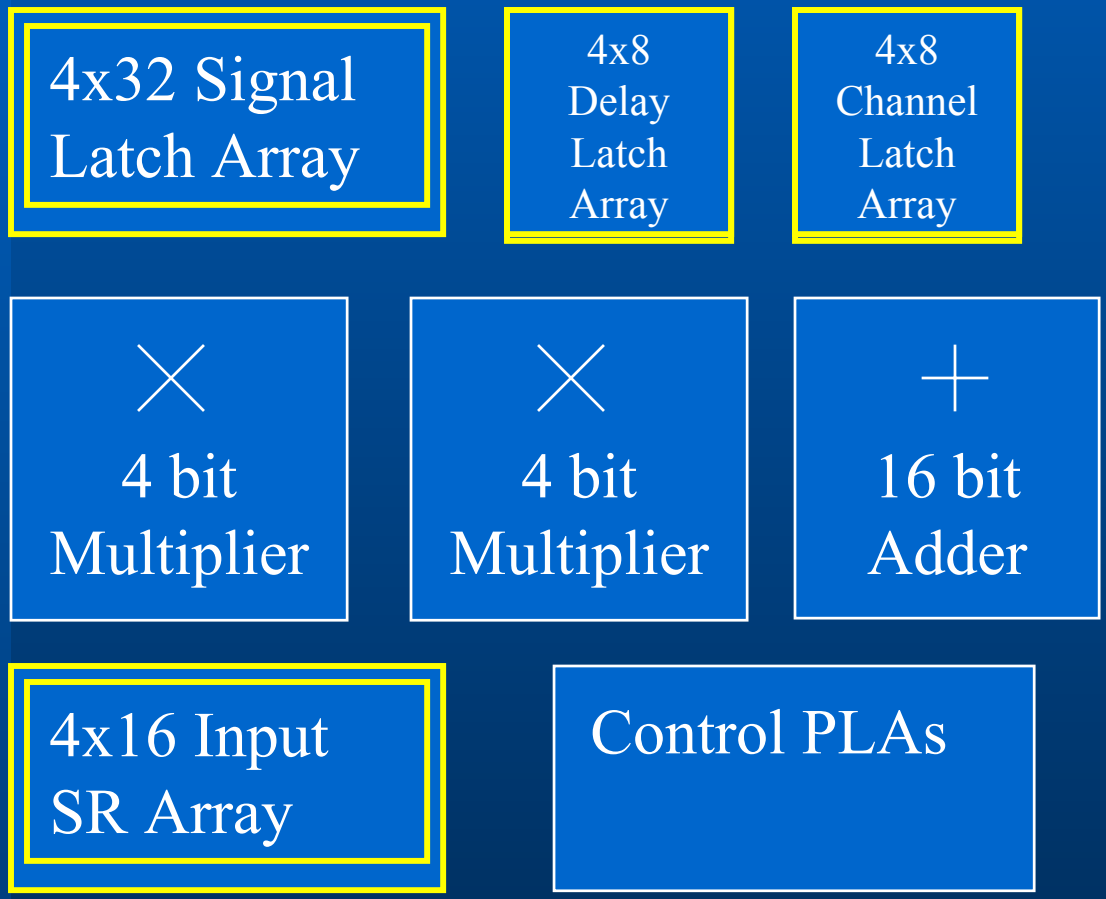
Major Points

- Followed 2-phase clocking discipline
  - Internal outputs all latched to clock B
  - Internal Inputs all latched to clock A

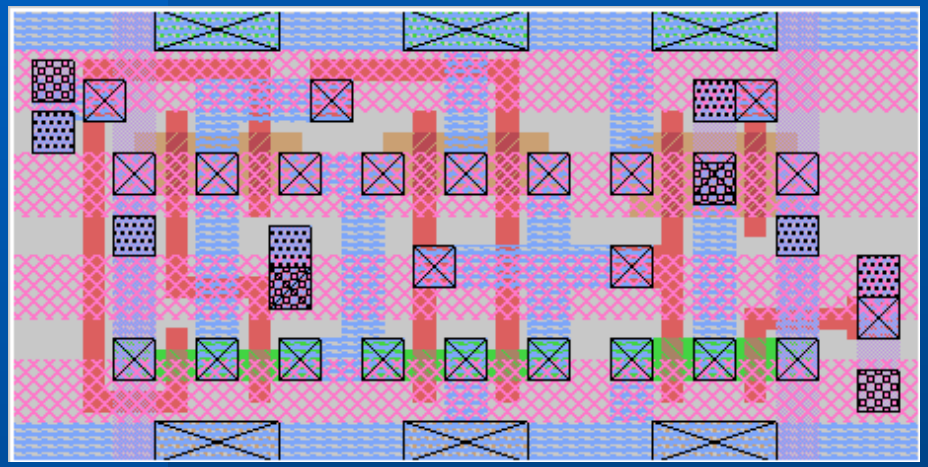
Speed Optimizations

- Split up main PLA into 4 sub PLAs
- Always full pipeline
- .5 μM process
- Carry Lookahead Adders
- Fast Tree Multipliers

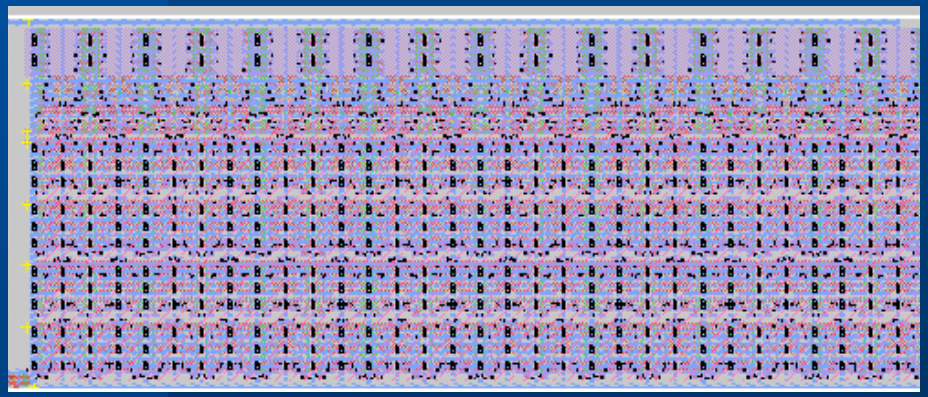
Memory


4x32 Signal Latch Array
4x8 Delay Latch Array
4x8 Channel Latch Array
4 bit Multiplier
4 bit Multiplier
16 bit Adder
4x16 Input SR Array
Control PLAs

Memory Layout



Basic Latch Cell



Latch Array

Memory Subcells

4x32 Signal Latch Array

4x8 Delay Latch Array

4x8 Channel Latch Array

4x16 Input SR Array

- Contains received data
- Shifts in new data

Memory Subcells

4x32 Signal Latch Array

4x8 Delay Latch Array

4x8 Channel Latch Array

- Contains signal we are comparing to input signal

4x16 Input SR Array

Memory Subcells

4x32 Signal Latch Array

4x8 Delay Latch Array

4x8 Channel Latch Array

- Contains amount of time to delay for each "finger"
- Up to 8 fingers

4x16 Input SR Array

Memory Subcells

4x32 Signal Latch Array

4x8 Delay Latch Array

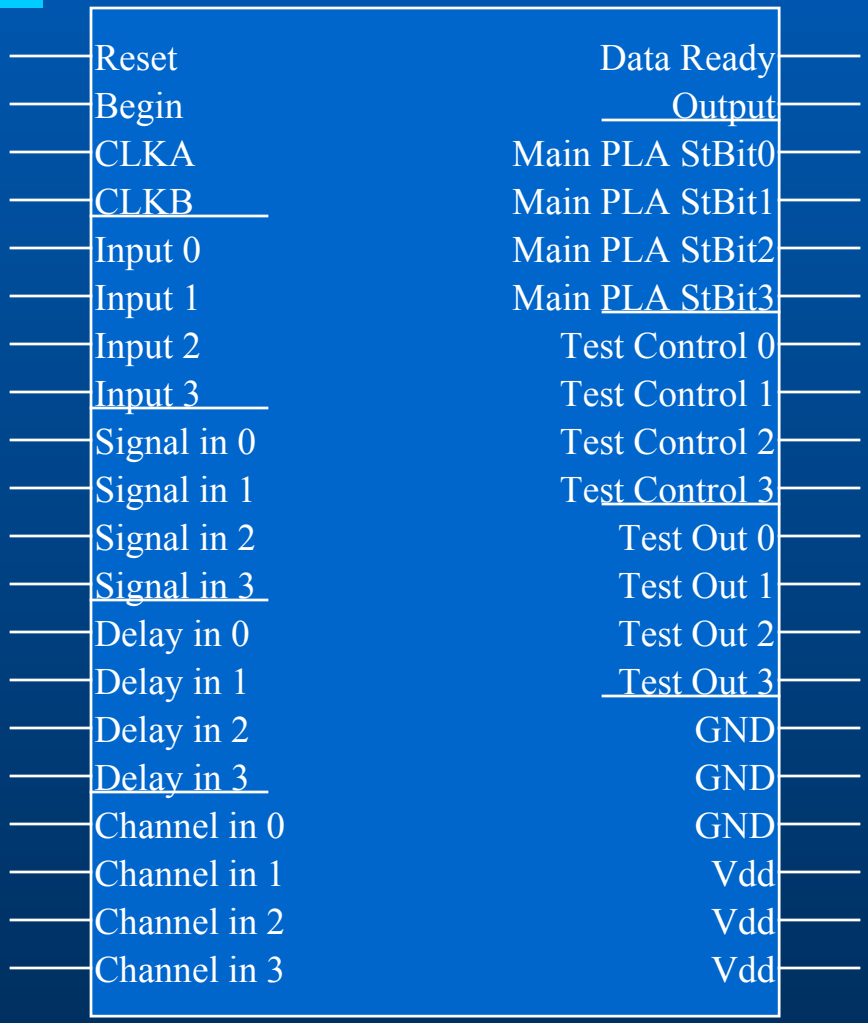
4x8 Channel Latch Array

- Contains data to reverse effects of channel for each “finger” or delay path

4x16 Input SR Array

Testability

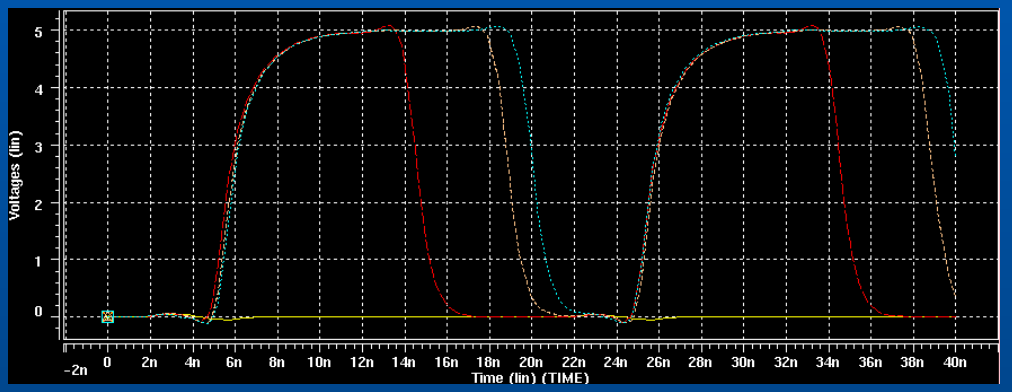
- "Test Control" chooses what appears on "Test Out"
- Final output is a 0 or 1, on "Output" pin
- Main PLA state bits are outputs



Spice Analysis

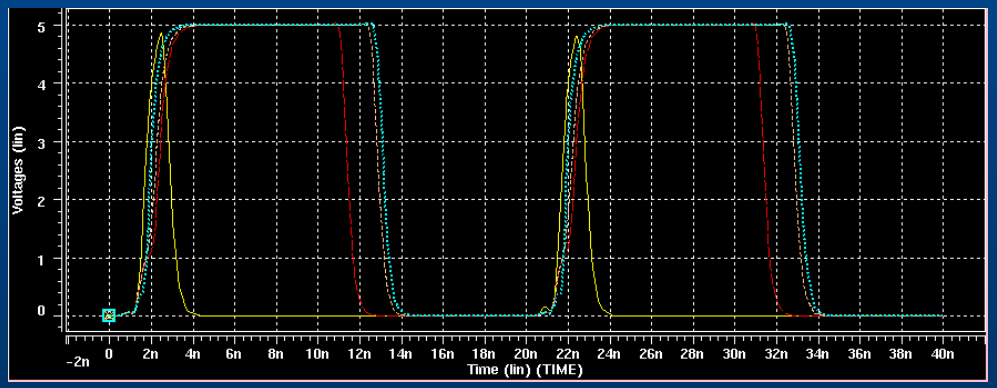
Multiplier plots

~8 ns



4 bit adder plots

~2.2 ns



Performance Summary

Max Delay Paths

- 15.5 ns longest path in PLA
- 8 ns multiplier
- 2.2 ns four bit CLA (8 ns 16 bit adder)

Theoretical Maxima

- Clock ~ 50 MHz
- Input ~ 6.4 Mchips/sec
- Output ~ 200 kbits/sec

Current Status

- Integrating and fitting components into pad frame
- High level Integration testing
- Developing test strategies

Future...