

# Wireless Systems-on-a-chip Design

---



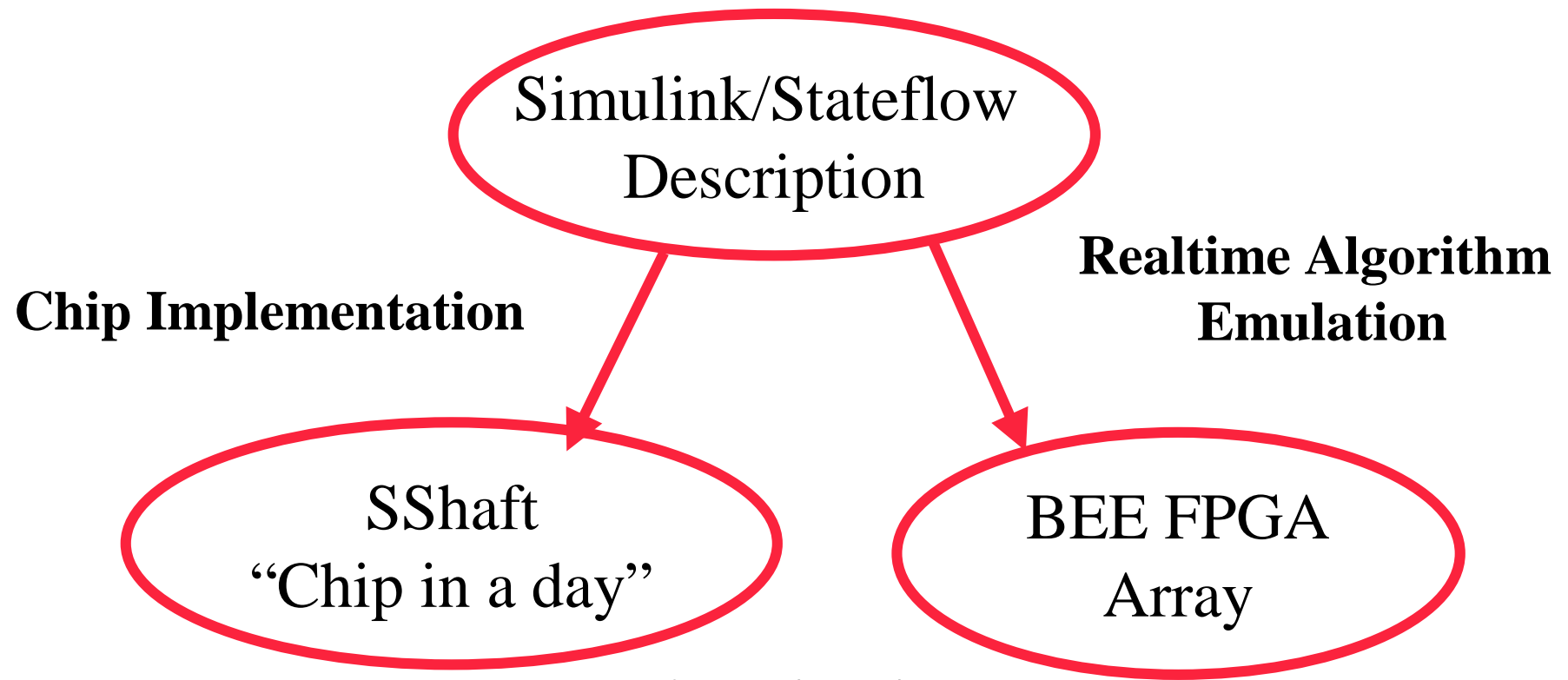
Bob Brodersen  
Dept. of EECS  
Univ. of Calif.  
Berkeley

<http://bwrc.eecs.berkeley.edu>

**Berkeley Wireless Research Center**

# BWRC Activities to be presented

---



## Applications

*UltraWideband Transmission (if there is time)*

*Multiple Antenna Systems*

*Sensor Networks*

## *(From Wednesday)* 3) Difficulty in getting dedicated SOC chips designed

---

- The dominant ASIC design flow is poorly suited for heterogeneous block level design
- This has resulted from a lack of research in providing new tools and architectures to attack the SOC design problem
- However, the foundry model is alive and well
  - multiproject runs in advanced technology are available with <1 month turn-around

## *(From Wednesday)* 3) Difficulty in getting dedicated SOC chips designed (cont.)

---

- There is no *fundamental* reason that power optimized, mixed signal designs can't be designed and implemented rapidly with low risk.
  - Many practical reasons
    - » Tools and methodologies not suited
    - » Inertia of previous experience
    - » Requires close integration of system and chip designers (analog and digital)

# *(From Wednesday)* An Energy/Area Efficient System Design Approach

---

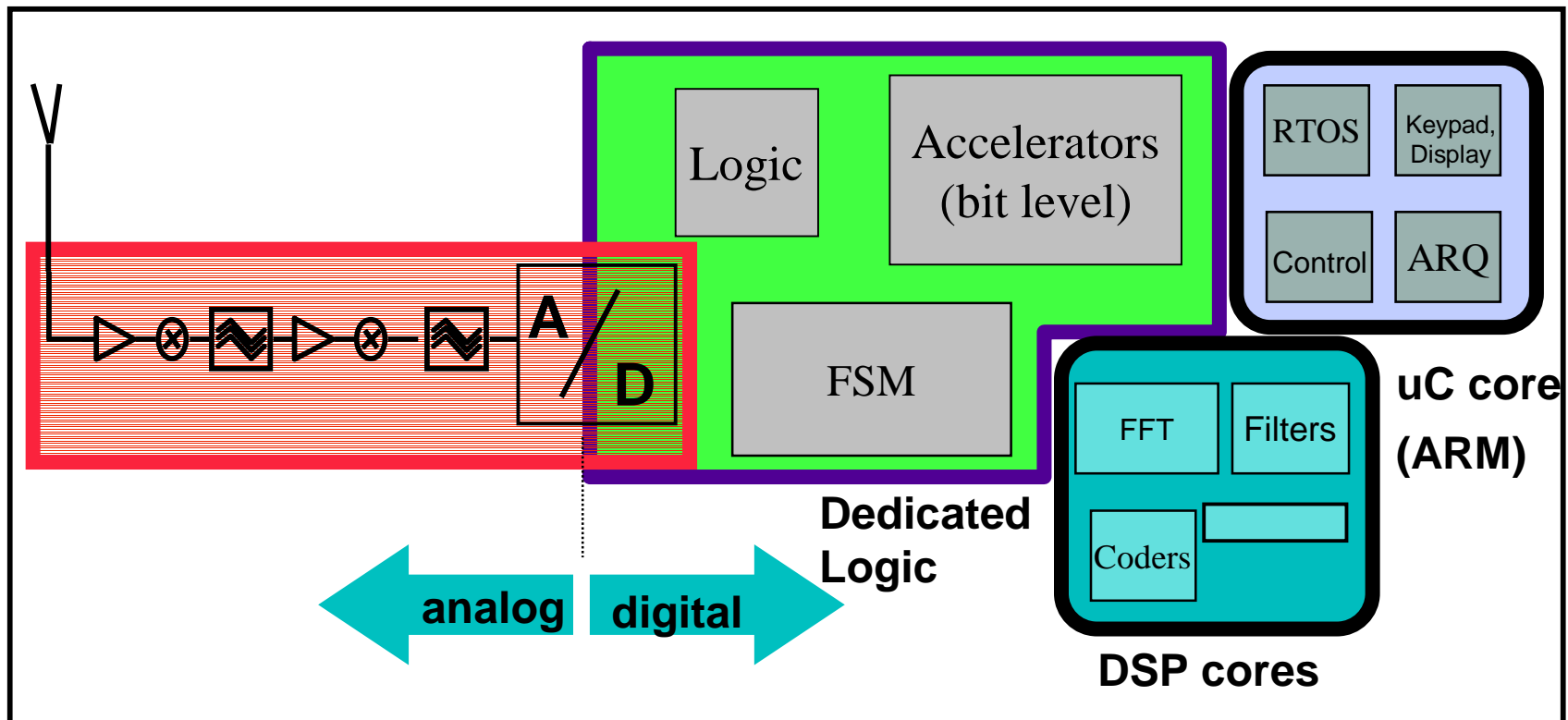
- Use a system level description to drive the design process
- Use dedicated hardwired (highly parallel) solutions if at all possible
- Provide hardware flexibility as needed for specific functions
- Use software only as the *last* resort when the computation is trivial and/or the application is sequential in nature (e.g. user interface, I/O).
- Solve the complete system problem – analog and digital together

# A Complete Wireless System

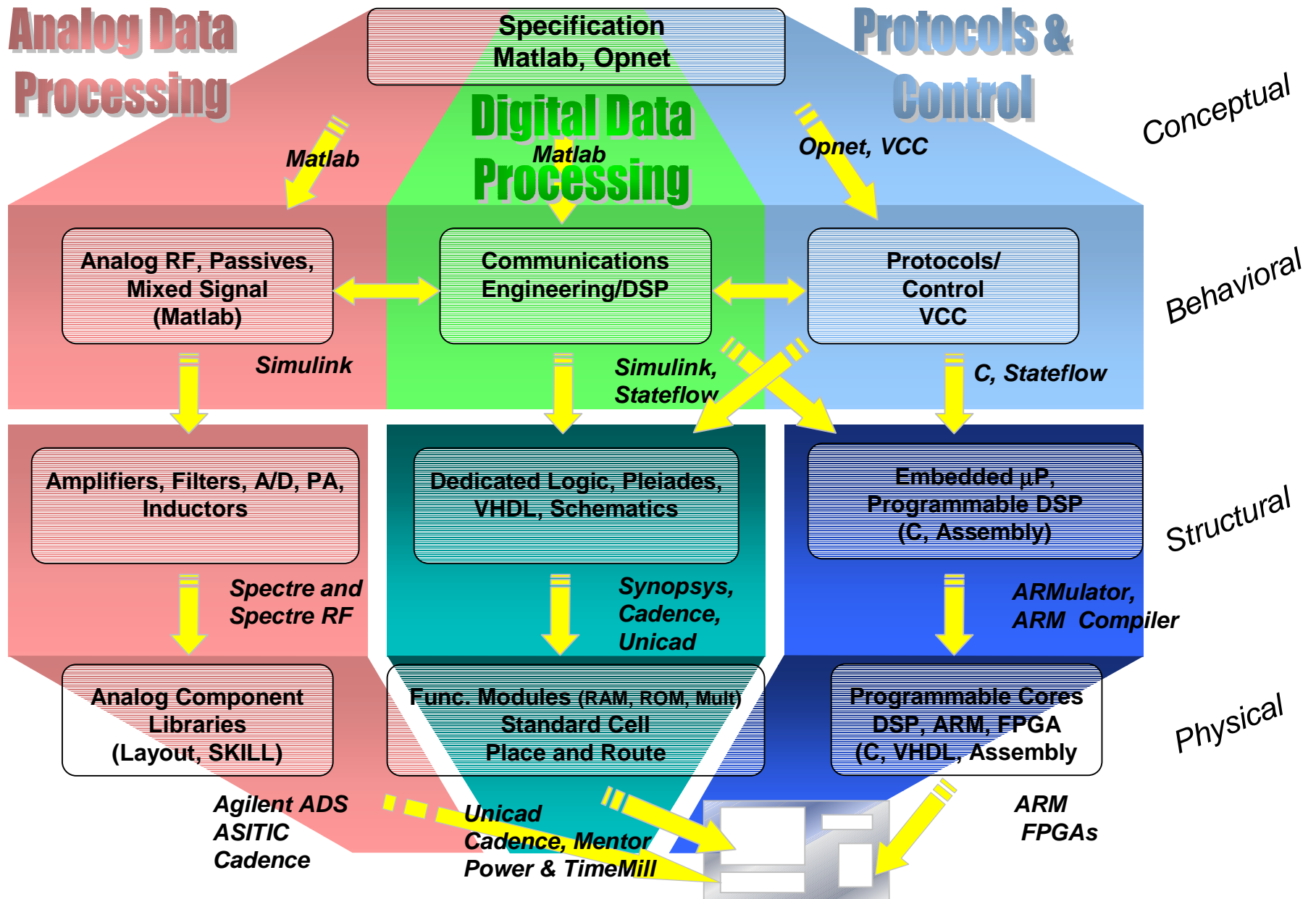
Analog Baseband  
and RF Circuits

Communication  
Algorithms

Protocols



# A Design Environment for Wireless Systems



# Simulink is a good Starting Point

---

Algorithm Exploration (Matlab)



***Simulink***



Analog Circuit Design

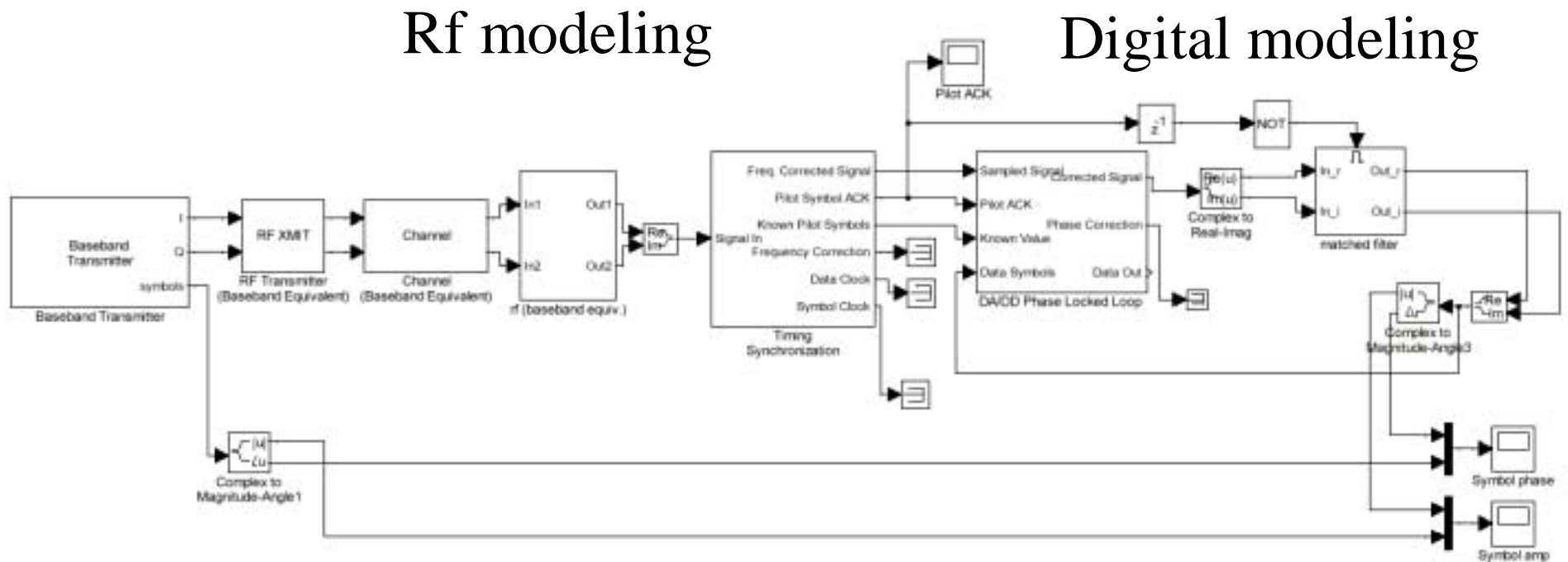
- Baseband Models of Components
- Fixed, Oversampled Timestep

Digital Circuit Design

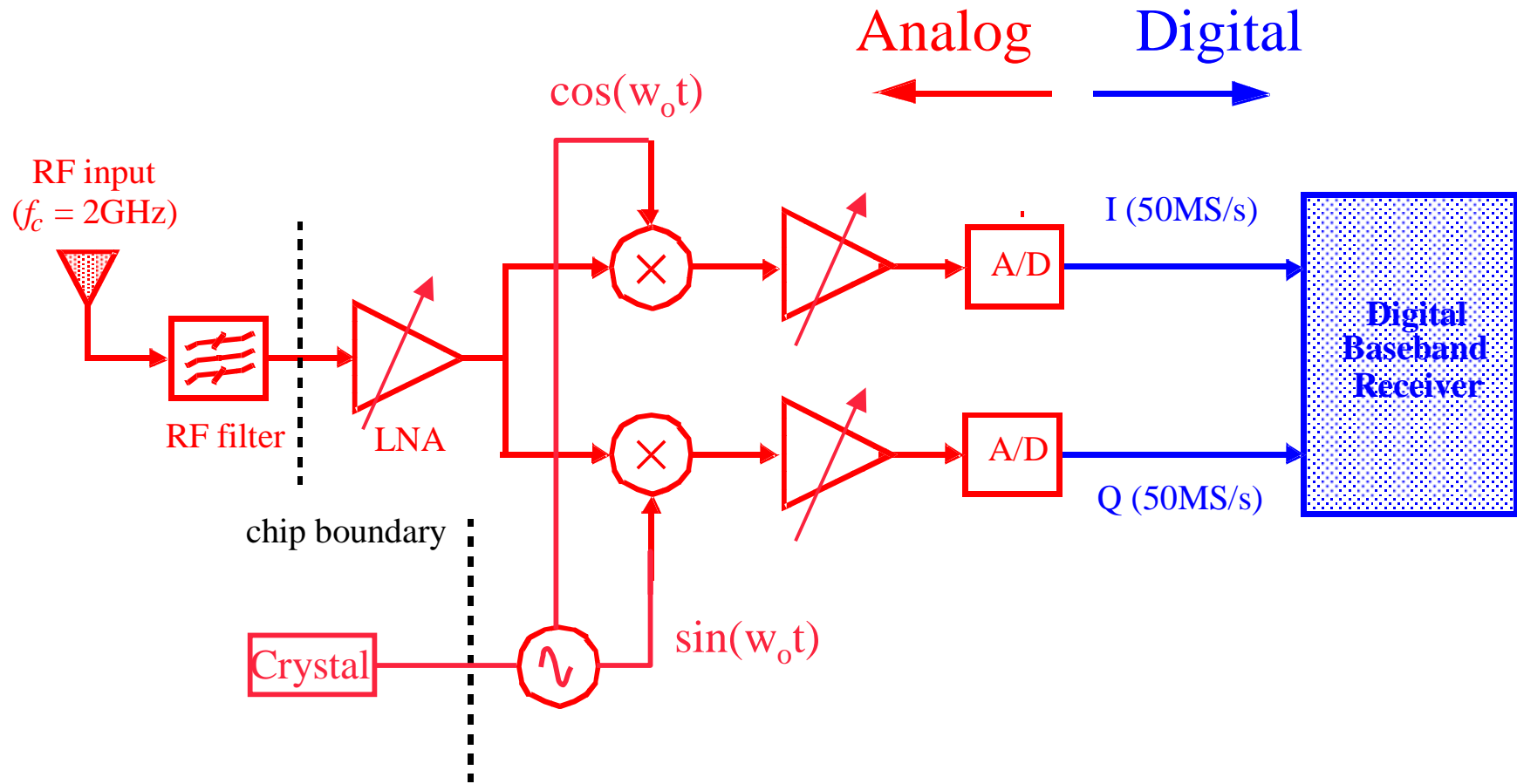
- DSP w/ Datapath
- Control/Protocol w/ Stateflow
- Microprocessors

Simulink is widely used by the Communications community

# Complete description of a radio system

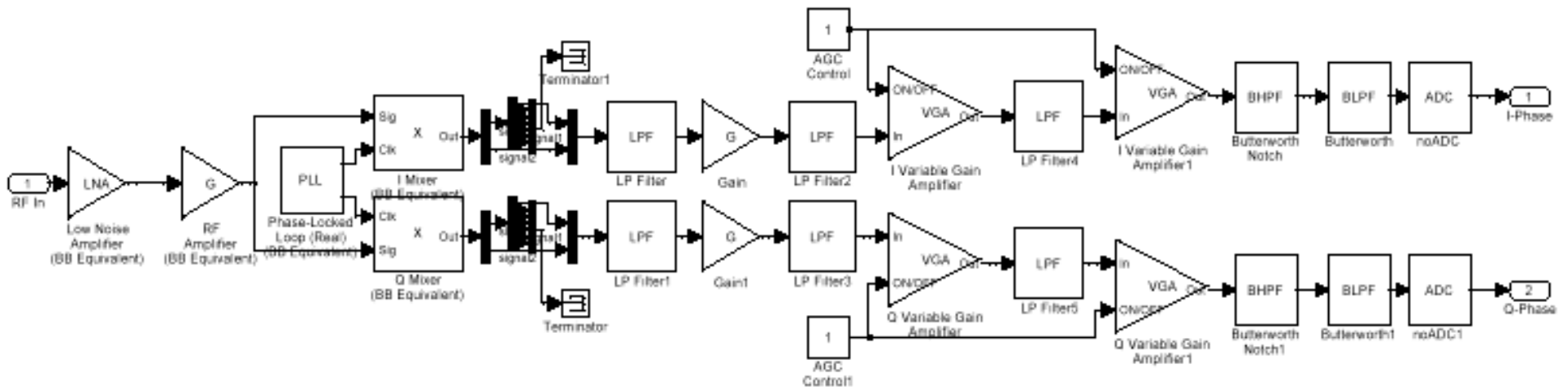
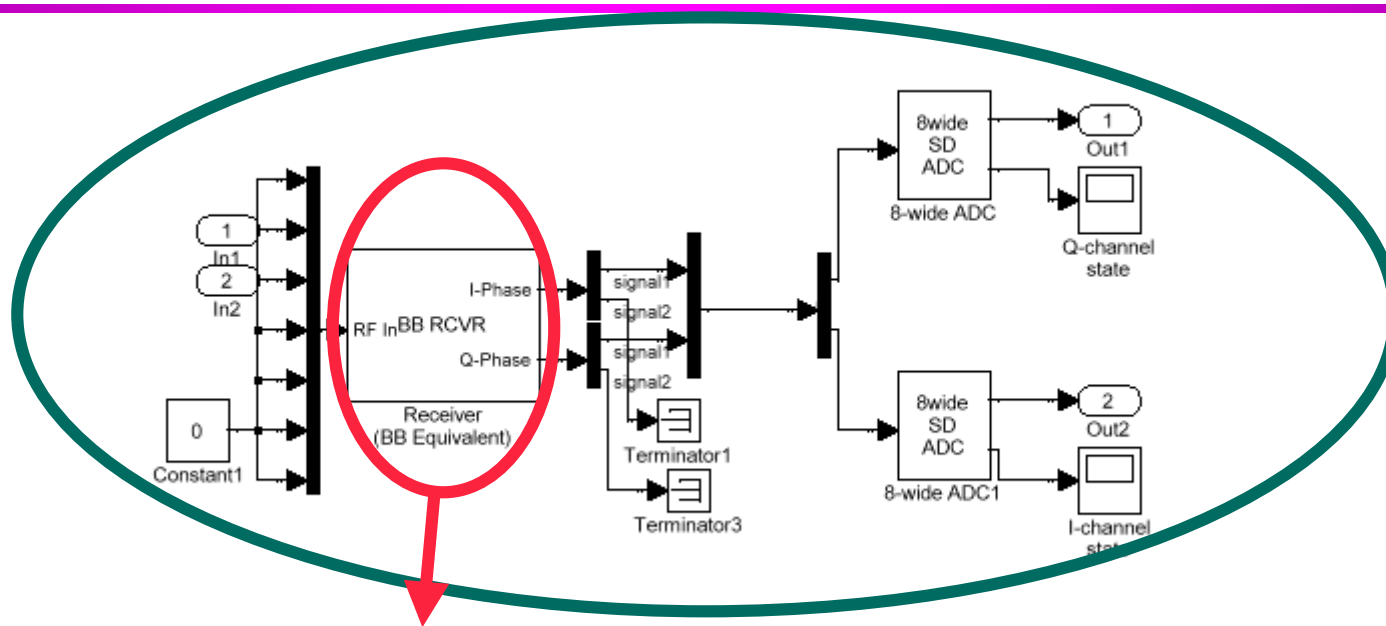


# Minimizing the Analog Components



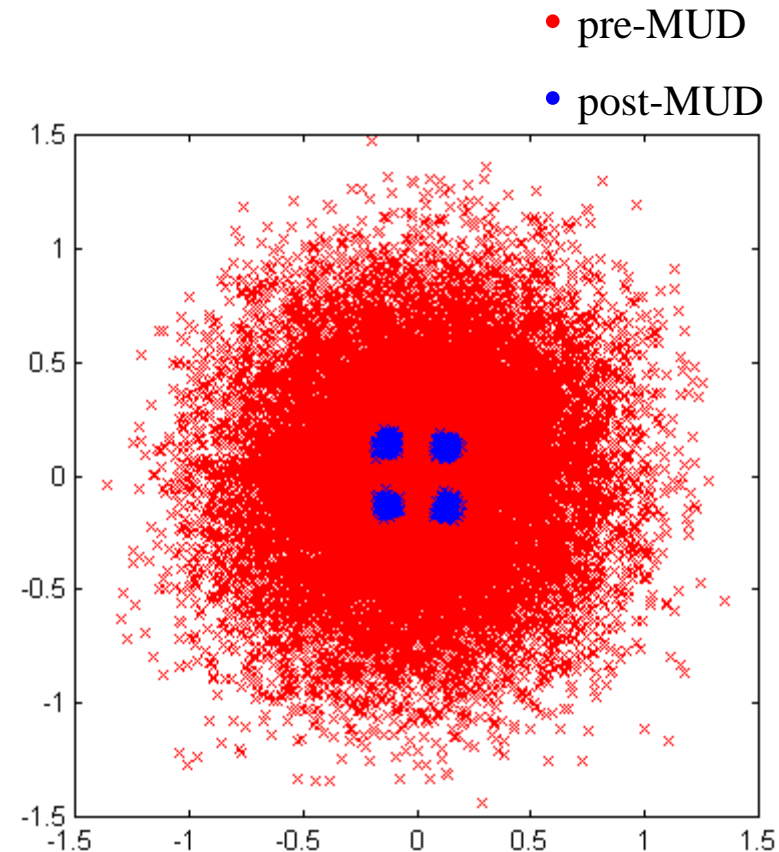
A zero IF (direct conversion) receiver

# Baseband equivalent analog modeling



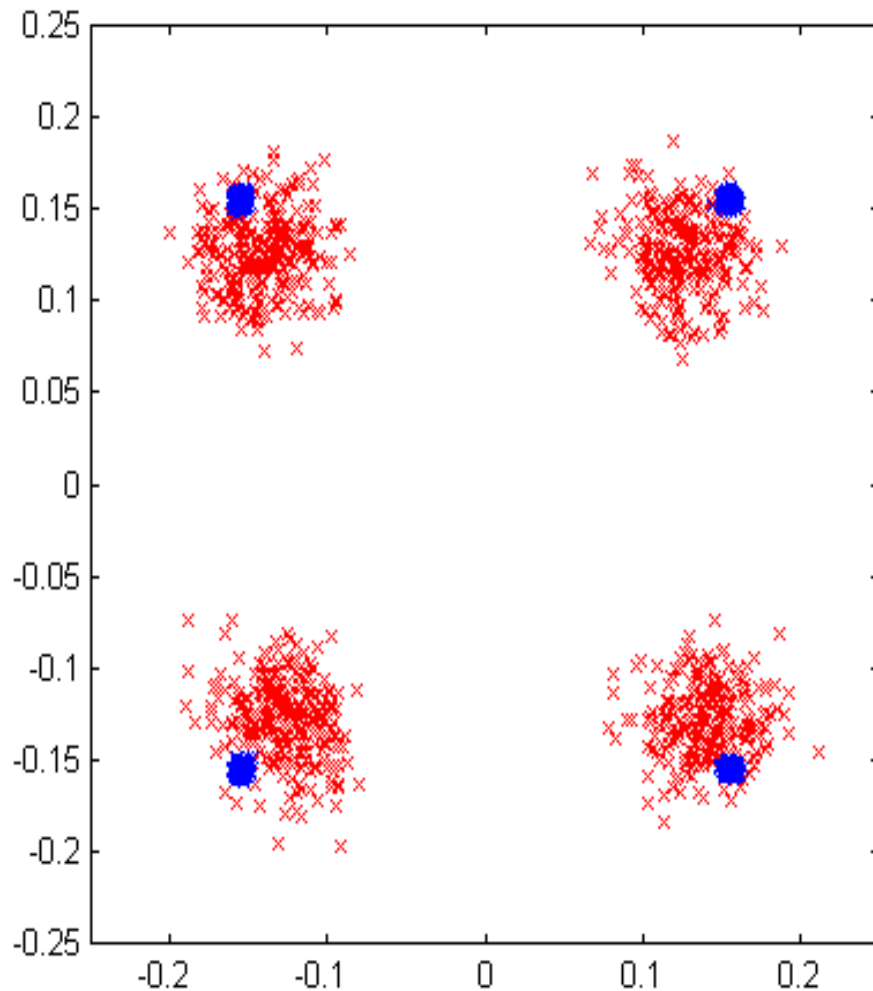
# System Simulation of Zero-IF Receiver

- 10 users (equal power)
- 13.5dB receiver NF
- PLL: -80dBc/Hz @ 100kHz
- 2.5° I/Q phase mismatch
- 82dB gain
- 4% gain mismatch
- IIP2 = -11dBm
- IIP3 = -18dBm
- 500kHz DC notch filter
- 20MHz Butterworth LPF
- 10-bit, 200MHz  $\Sigma$ - $\Delta$  ADC



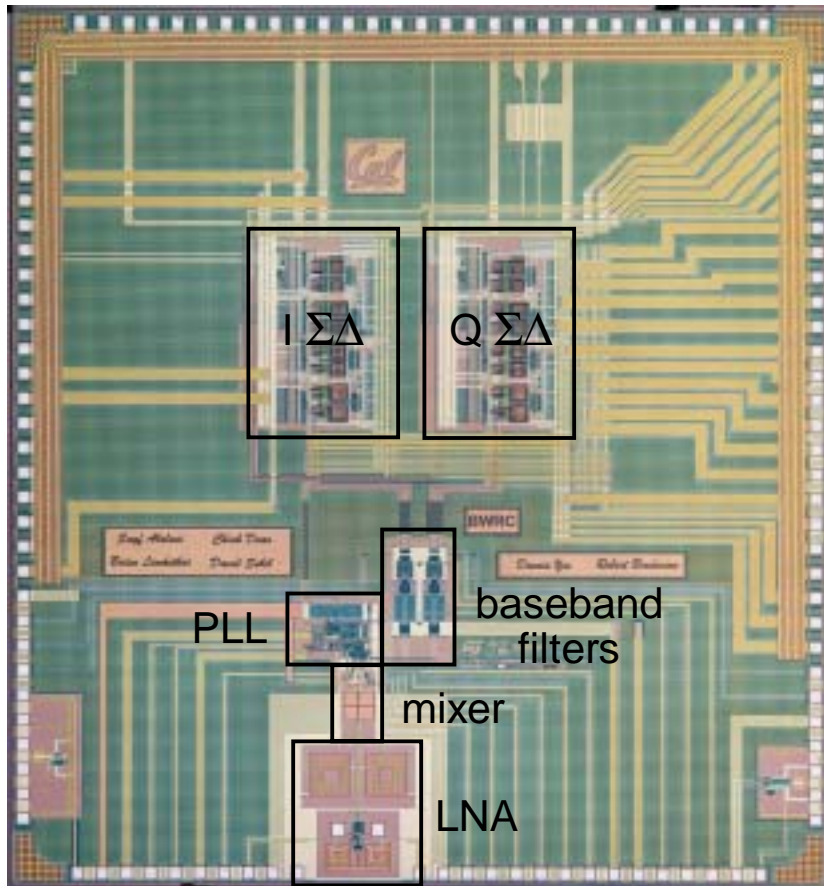
**Output SNR  $\approx$  15dB**

# With Analog Impairments



- 10 users (equal power)
- 20MHz Butterworth LPF
- 500kHz DC notch filter
- 13.5dB receiver NF
- 82dB gain
- 4% gain mismatch
- 2.5° I/Q phase mismatch
- IIP2 = -11dBm
- IIP3 = -18dBm
- PLL: -80dBc/Hz @ 100kHz
- 10-bit, 200MHz S-D ADC

# Receiver Prototype



0.25- $\mu\text{m}$ , 6-metal CMOS process

Active Area = 4 mm<sup>2</sup>

Noise Figure (DSB) = 8.5 dB

S11 < -30 dB

Voltage Gain = 41 dB

-3-dB Bandwidth: 90 kHz <  $f$  < 18 MHz

-1-dB Compression = -31.1 dBm

IIP2 (27 MHz, 37MHz) = -6.7 dBm

IIP3 (35 MHz, 60MHz) = -18.3 dBm

PLL Phase Noise: -85 dBc/Hz @ 2.5 MHz

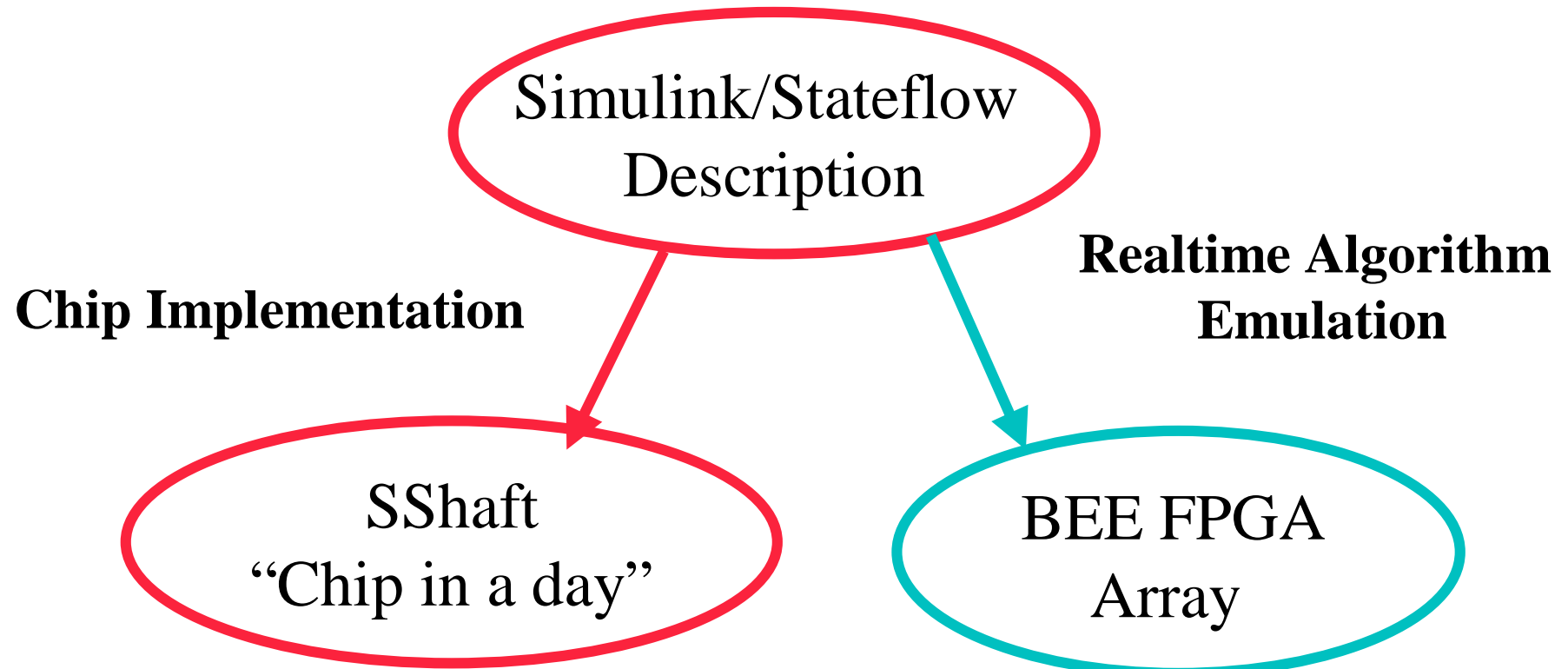
LO-to-RF Leakage = -81 dBm

$\Sigma\Delta$  Dynamic Range = 42 dB @ 200 MHz

Power Dissipation = 106 mW

# Outline

---



## Applications

*UltraWideband Transmission*

*Multiple Antenna Systems*

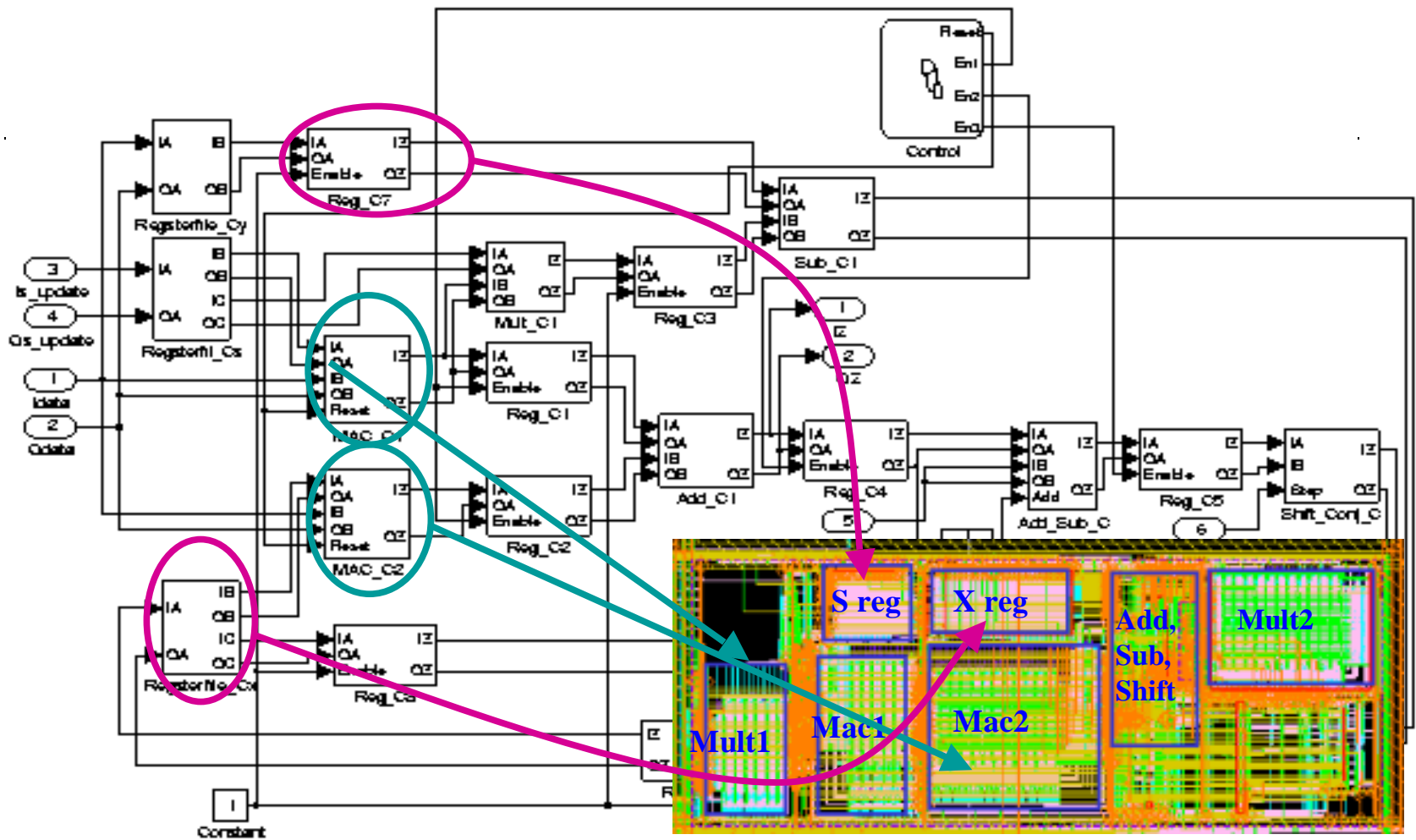
*Sensor Networks*

# Outline

---

- **Chip-in-a-Day Flow**
  - » **Design Decisions**
  - » **Estimation**
  - » **Automation**
- **Enabling Factors**
- **Design Examples**
  - » **Design Effort**
  - » **Problems**

# Direct Mapping of the Algorithm into Hardware



# Results in fully parallel solutions

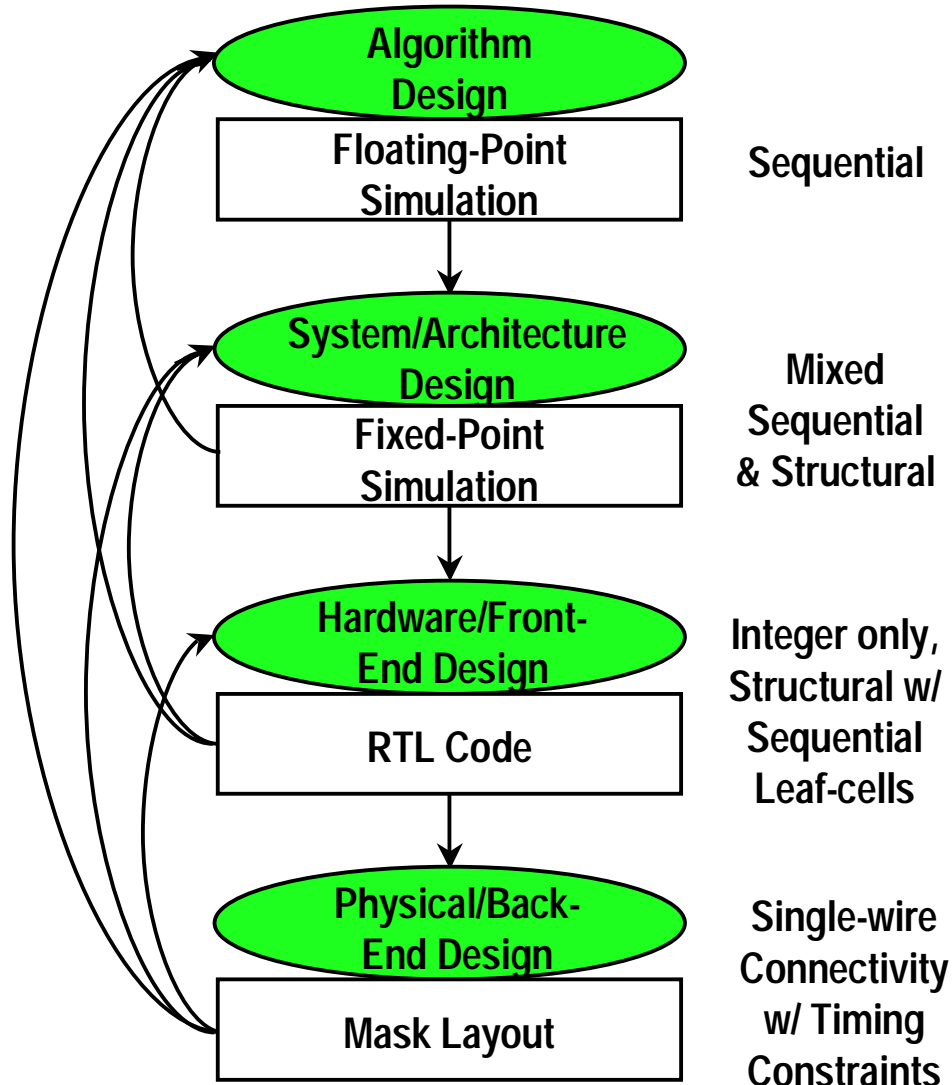
Reducing supply voltage saves energy:  $E = CV^2$

	Energy		Area	
	64-point FFT Energy per Transform (nJ)	16-State Viterbi Decoder Energy per Decoded bit (nJ)	64-point FFT Transforms per second per unit area (Trans/ms/mm <sup>2</sup> )	16-State Viterbi Decoder Decode rate per unit area (kb/s/mm <sup>2</sup> )
Direct-Mapped Hardware	1.78	0.022	2,200	200,000
FPGA	683	5.5	1.8	100
Low-Power DSP	436	19.6	4.3	50
High-Performance DSP	1700	108	10	150

(numbers taken from vendor-published benchmarks)

**Orders of magnitude lower efficiency  
even for an optimized processor architecture**

# Standard DSP-ASIC Design Flow

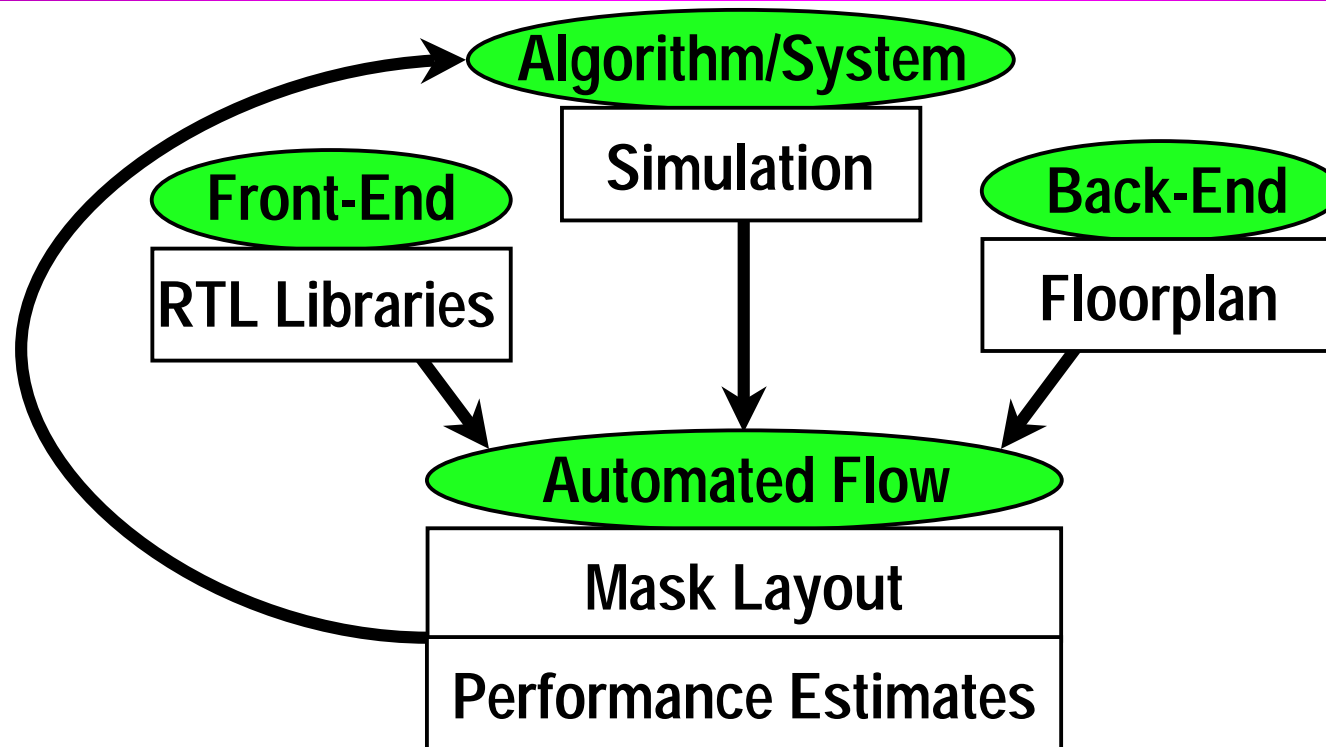


## Problems:

- Three translations of design data
- Requirements for re-verification at each stage
- Uncontrolled looping when pipeline stalls

**Prohibitively Long Design Time for Direct Mapped Architectures**

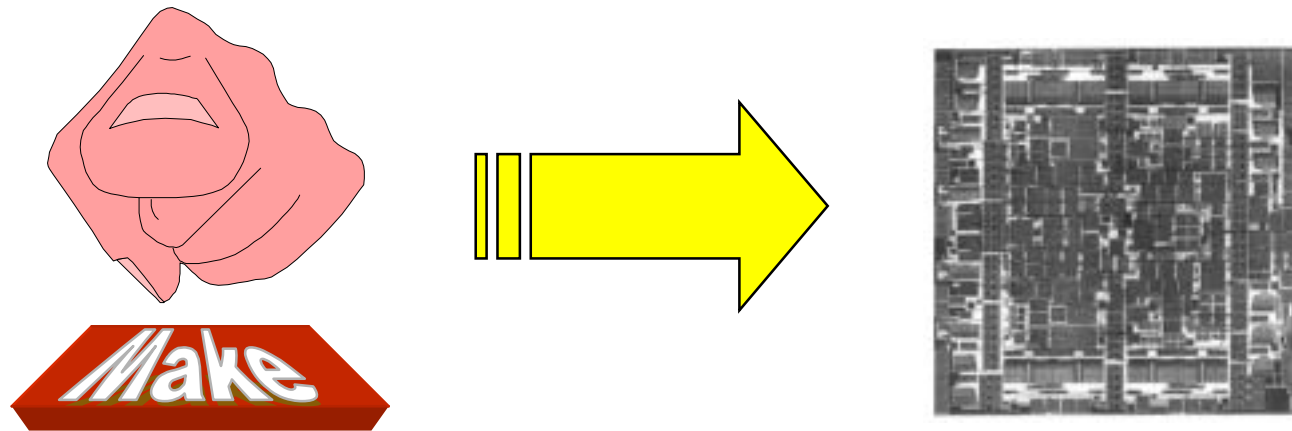
# Direct Mapping Design Flow



- Encourages iterations of layout
- Controls looping
- Reduces the flow to a single phase
- Depends on fast automation

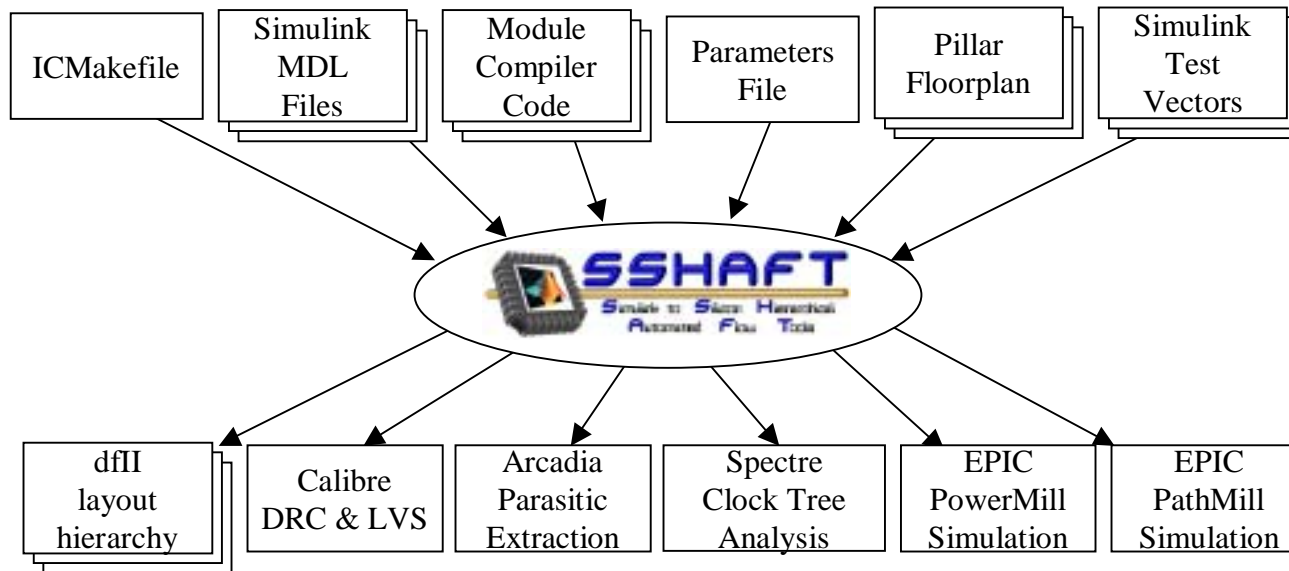
# “Push-Button” Automation

---



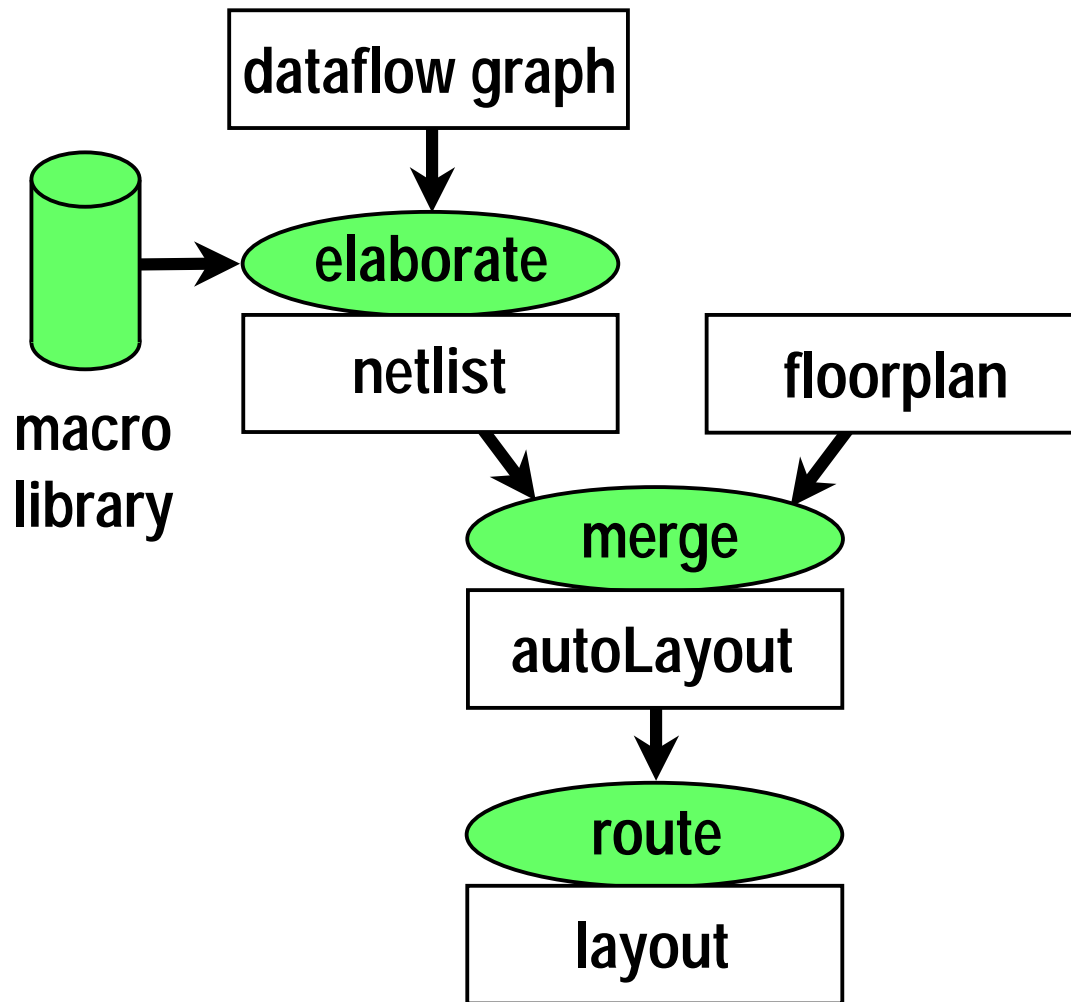
- Automation similar to MAKE
  - » No decisions made after button is pressed
- No translation of design data
  - » No decisions expressed more than once
- *Goal: “Chip in a Day”*

# Chip-in-a-Day Design Flow: A User Perspective



- Allow regeneration and re-analysis of the design for small changes at the push of a button
- Uses flow dependency graphs to manage large projects

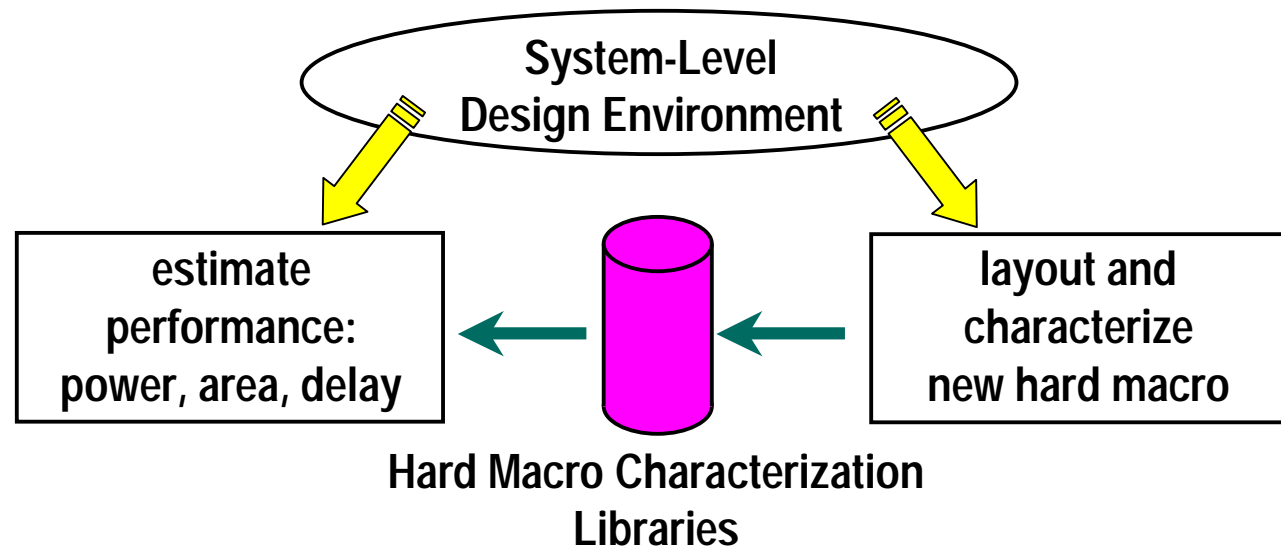
# Simplified View of the Flow



## New Software:

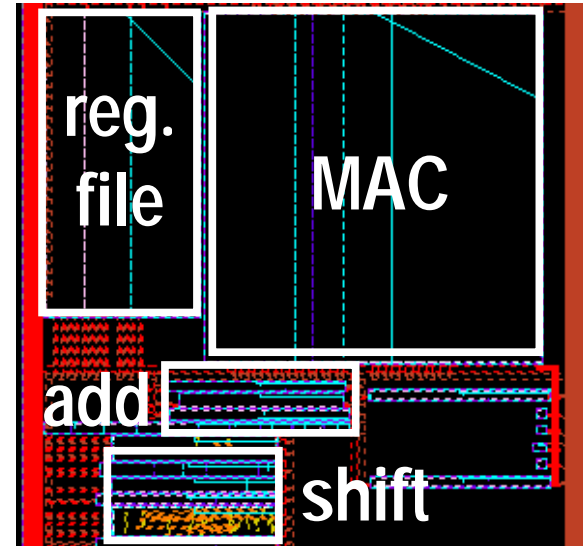
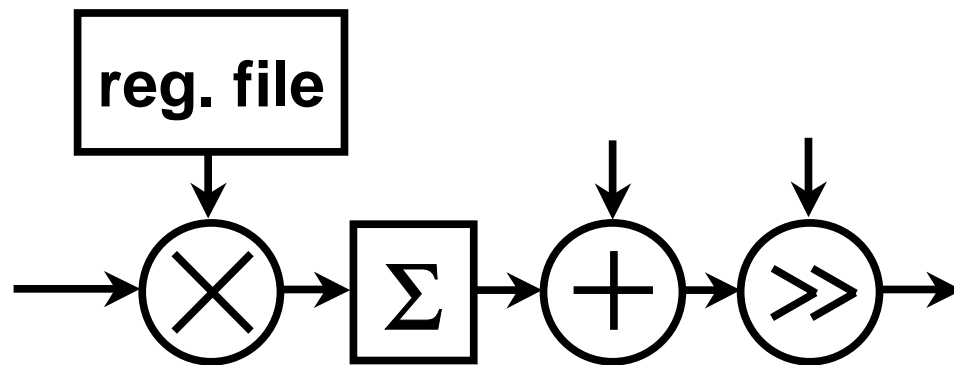
- Generation of netlists from a dataflow graph
- Merging of floorplan from last iteration
- Automatic routing and performance analysis
- Automation of flow as a dependency graph (UNIX MAKE program)

# Hierarchy Hardened Progressively



- Macro characterization saved for fast estimates
- Each level of hierarchy becomes a new hard macro
- Higher levels of hierarchy are adjusted
- When top level of hierarchy is hardened, the design is done

# Capturing Design Decisions



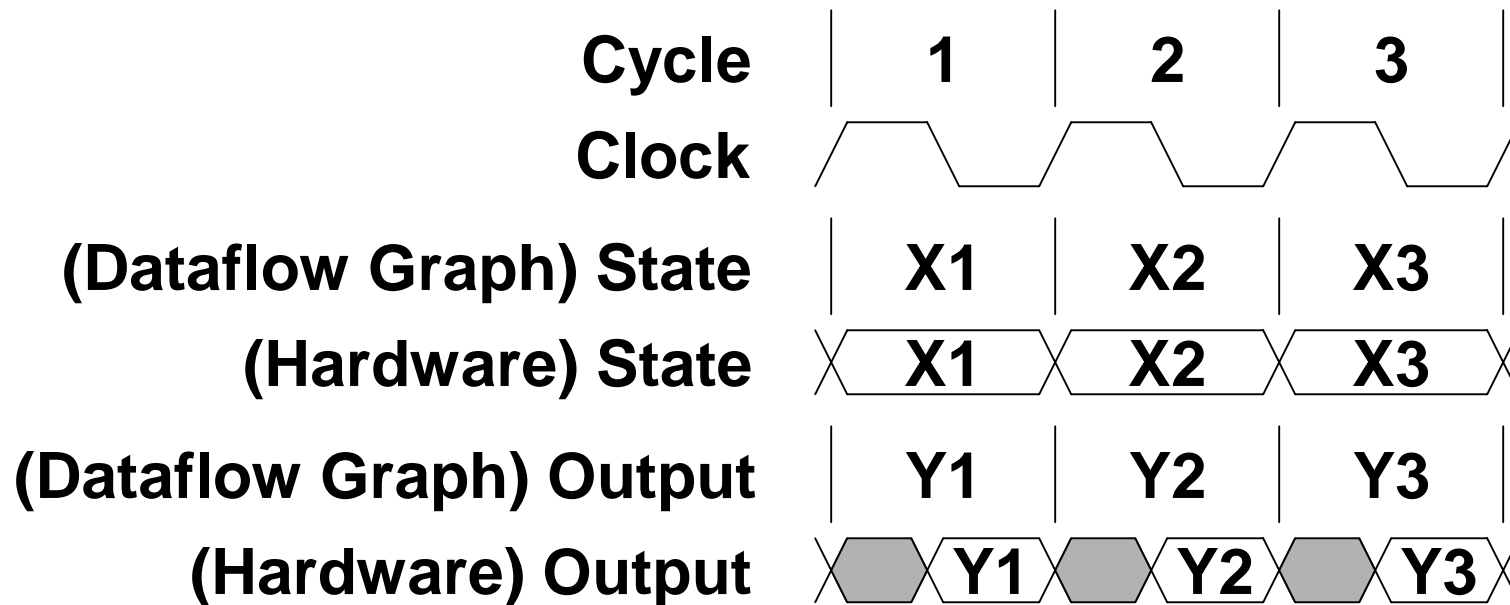
## Categories:

- Function - basic input-output behavior (user input)
- Signal - physical signals and types (user input)
- Circuit - transistors (in libraries)
- Floorplan - physical positions (user input)

**Layout and performance estimates**

# Choice of Computation Model

- Balance between conflicting needs
  - » Abstraction of Hardware (simulation speed)
  - » Clear Verification Strategy (detailed hardware)



# Fixed Mapping to Abstracts

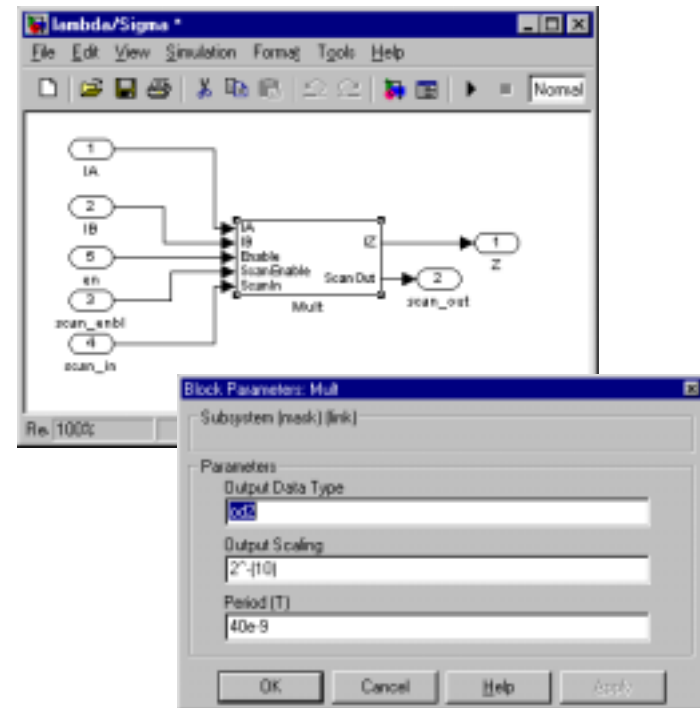
---

- Elaboration does more than specify transistors and circuits, it specifies mask-layout, or more specifically, **abstracts**
- Each Macro type will have a **fixed mapping** to abstracts
  - » **Block/Module Macros** - single abstract
  - » **VHDL/Stateflow Macros** - netlist of standard cell abstracts
  - » **Microprocessor Macros** - single abstract for processor and memory, netlist of standard cells for interface

This allows us to add as much **heterogeneity** as we like

# Block & Module Macros

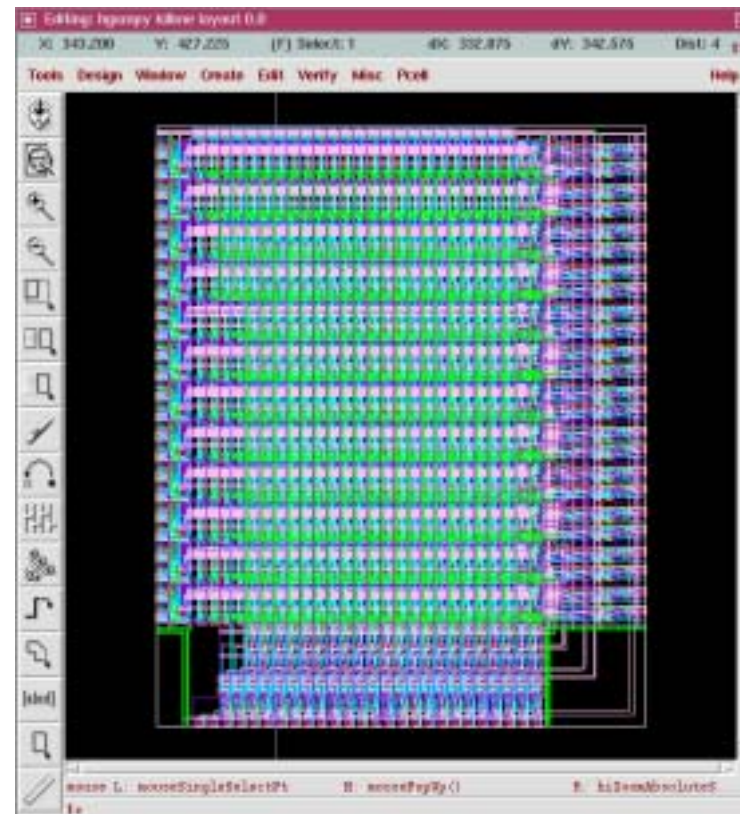
- Block
  - » Simple Layout and Schematic
- Module
  - » Parameterized
  - » Tiled Layout
  - » Generated Schematic



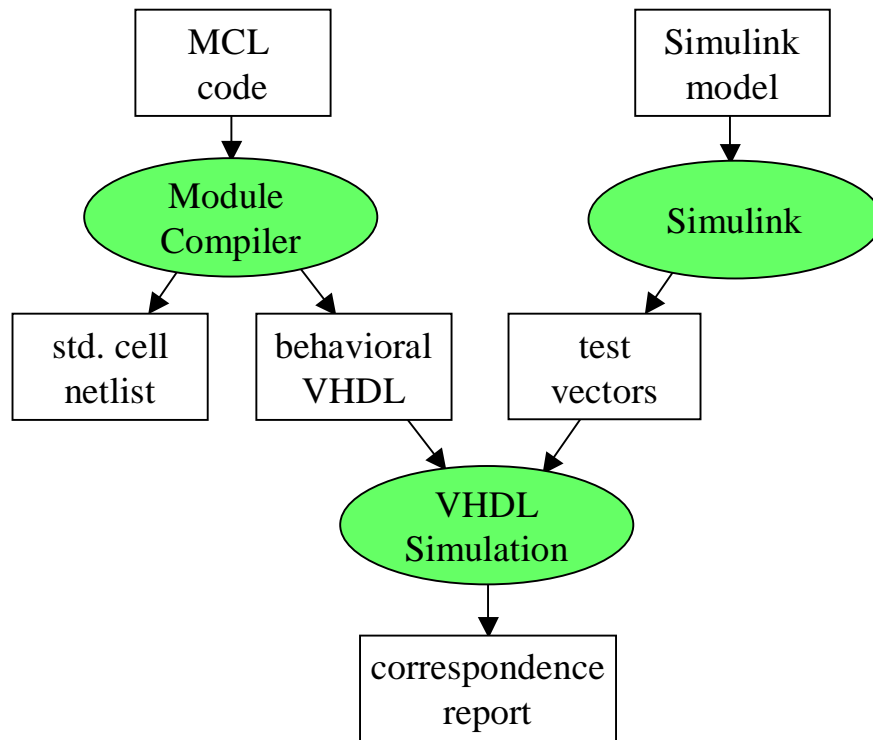
Block & Module Macros map to a Single Abstract

# Module generation

- Take parameters (e.g. bit width) from block diagram as input and generate layout
- Allows deterministic area, power and delay estimates
- Retains optimized density, speed and power of custom design
- Allows reuse



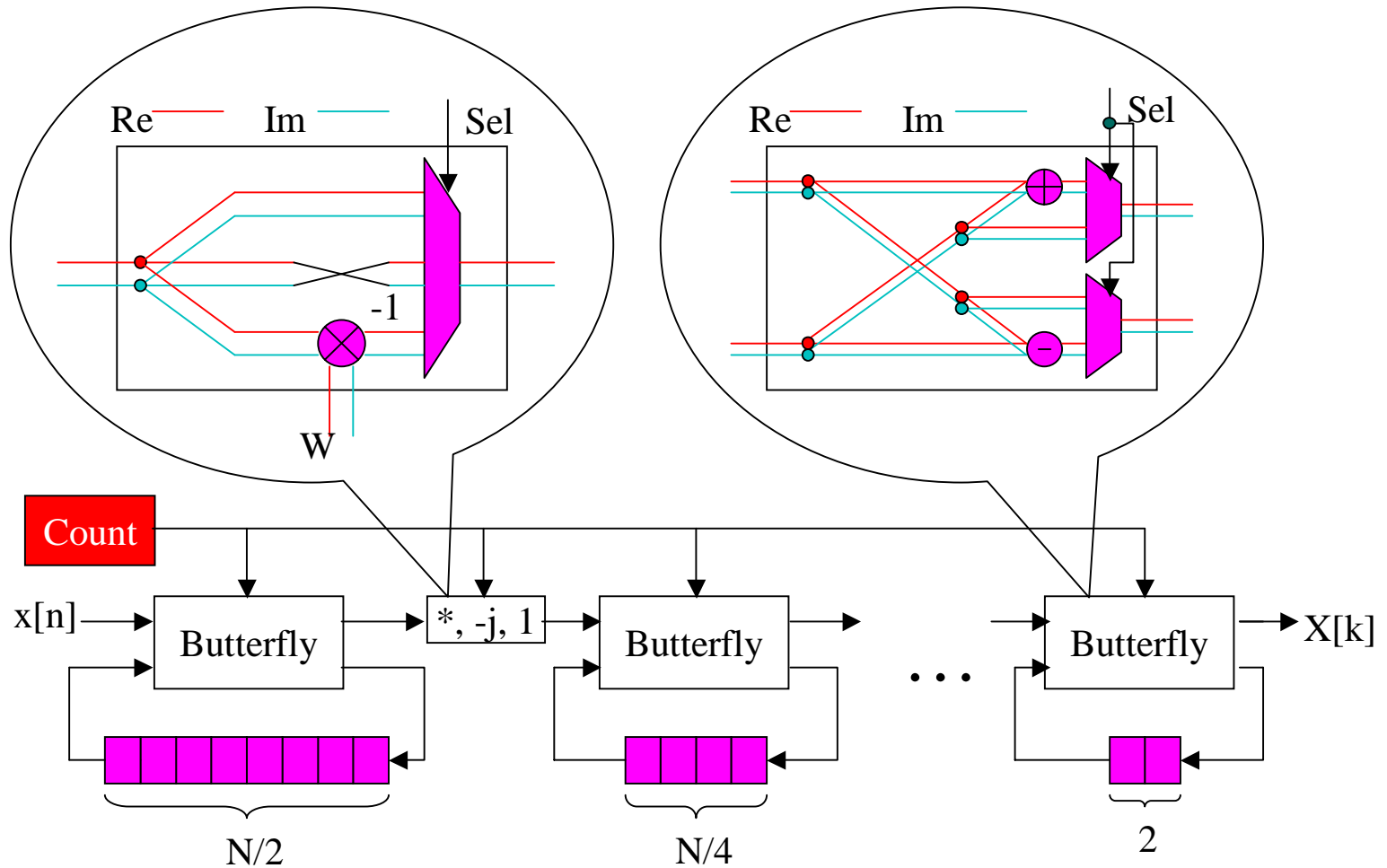
# Module Compiler for datapaths



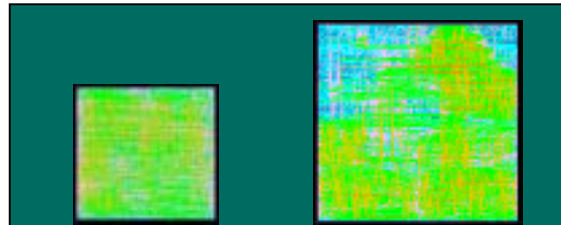
- Easy to create Simulink models which are bit accurate simulations of Module Compiler designs
- Allows arbitrarily complex modules (limited by ability to deal with flat netlists) – How big to make them????
- Datapath optimized floorplans

EECS 225C students successfully applied  
this flow to all of their designs

# Pipelined FFT Architecture



# Automated Flow Characterizes Macros



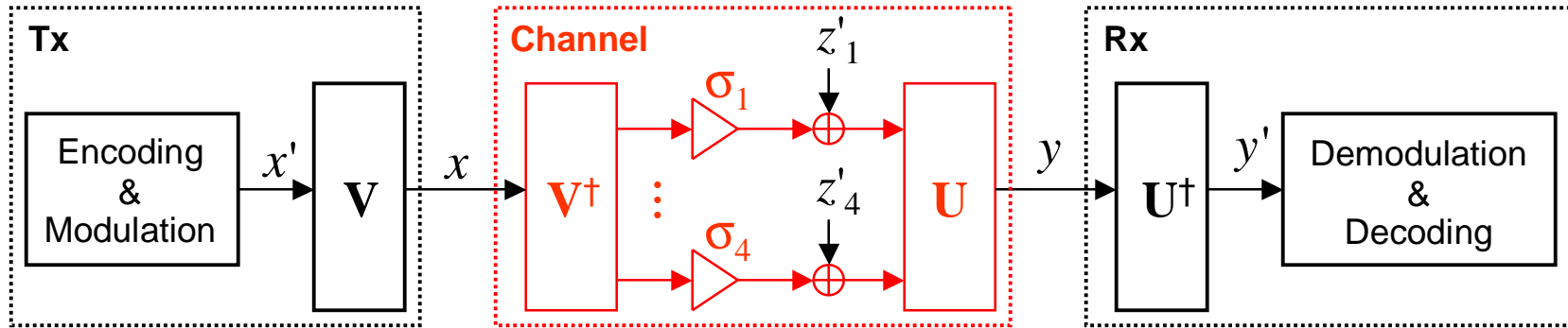
	viterbi decoder	fft
area in 0.25 $\mu\text{m}$	0.71 mm <sup>2</sup>	1.4 mm <sup>2</sup>
power @ 25 MHz (2.5 V)	69 mW	150 mW
(1.0 V)	7.0 mW	16 mW
critical path delay (2.5 V)	4.8 ns	20 ns
(1.0 V)	15 ns	63 ns
cells	10 k	19 k
transistors	130 k	270 k
execution time	18 hours	34 hours
disk space	831 MB	1.66 GB

Data collected after

- » synthesis
- » placement
- » routing
- » SPICE netlisting
- » Calibre DRC & LVS
- » Arcadia parasitic extraction
- » EPIC PowerMill simulation
- » EPIC PathMill timing analysis

Characterization flow currently supported for both  
0.25  $\mu\text{m}$  and 0.18  $\mu\text{m}$

# SVD



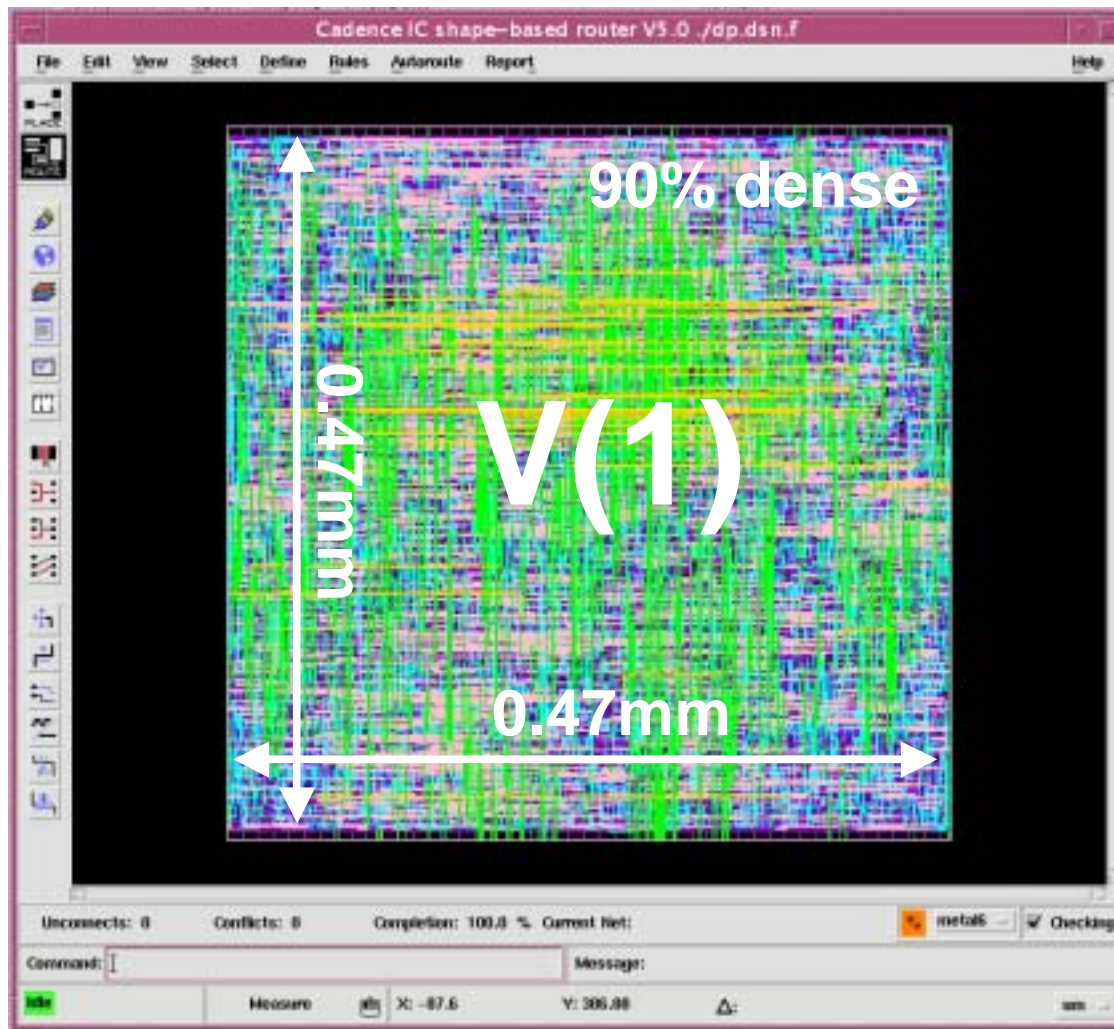
BB-equivalent channel models

1.	$y = \mathbf{H} \cdot x + z$	
2.	$y = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^\dagger \cdot x + z$	SVD
3.	$y' = \mathbf{\Sigma} \cdot x' + z'$	Unitary X-form

$$\left\{ \begin{array}{l} x' = \mathbf{V}^\dagger \cdot x \\ y' = \mathbf{U}^\dagger \cdot y \\ z' = \mathbf{U}^\dagger \cdot z \end{array} \right.$$

Need to know  $\mathbf{V}$  at Tx and  $\mathbf{U}, \mathbf{\Sigma}$  at Rx

# SVD Hardware Realization

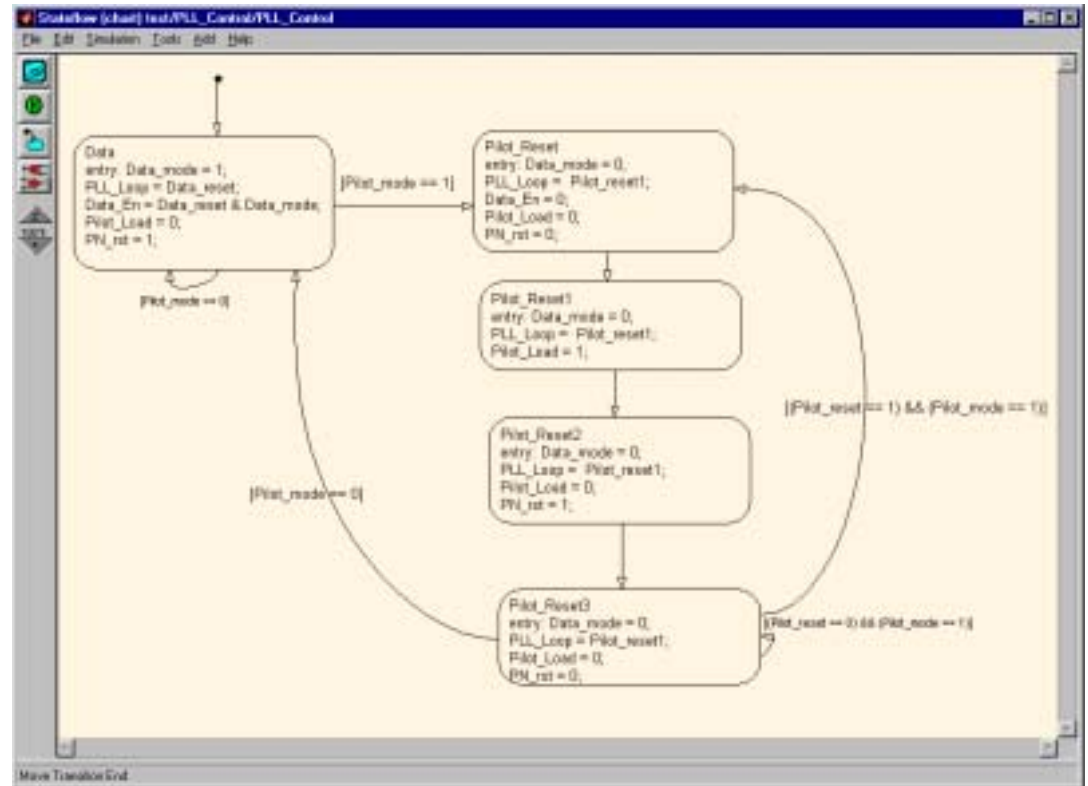


- 0.18 $\mu$ m process

Parameter	L [m]
Manhattan	0.26
• Horizontal	0.12
• Vertical	0.14
Routed	0.28
• Horizontal	0.13
• Vertical	0.15
# nets	7032
# vias	5870
# wire jncts	31285

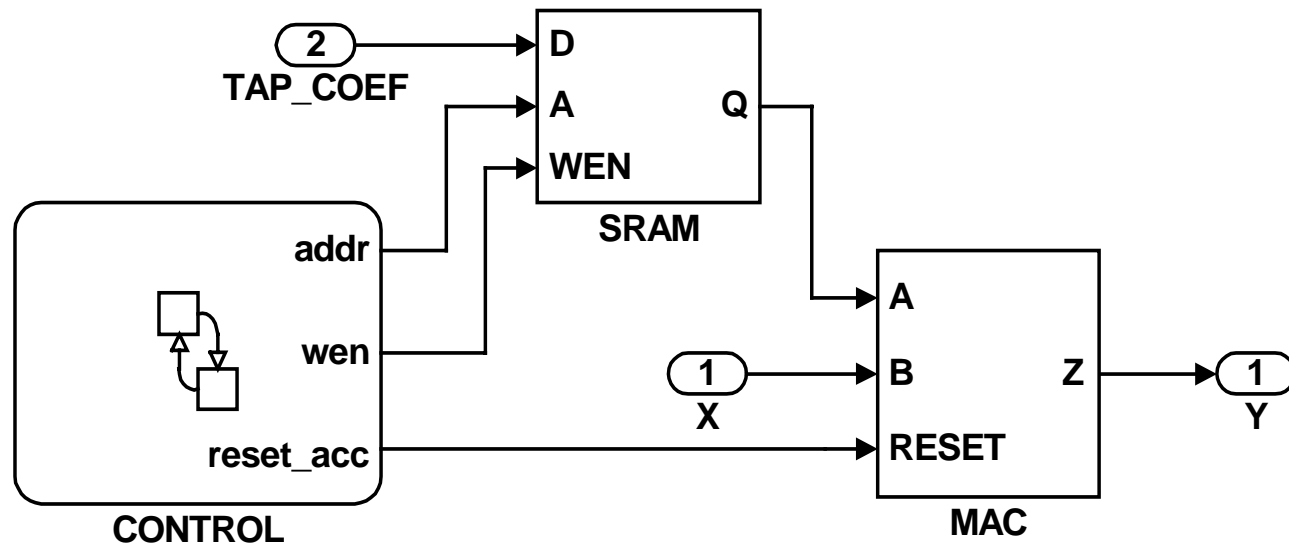
# Control

- Stateflow
  - » Extended Finite State Machine
  - » Subset of Syntax
  - » Converted to VHDL
  - » Synthesized
- VHDL
  - » Synthesized directly



VHDL & Stateflow Macros map to a netlist of Standard Cells using standard synthesis

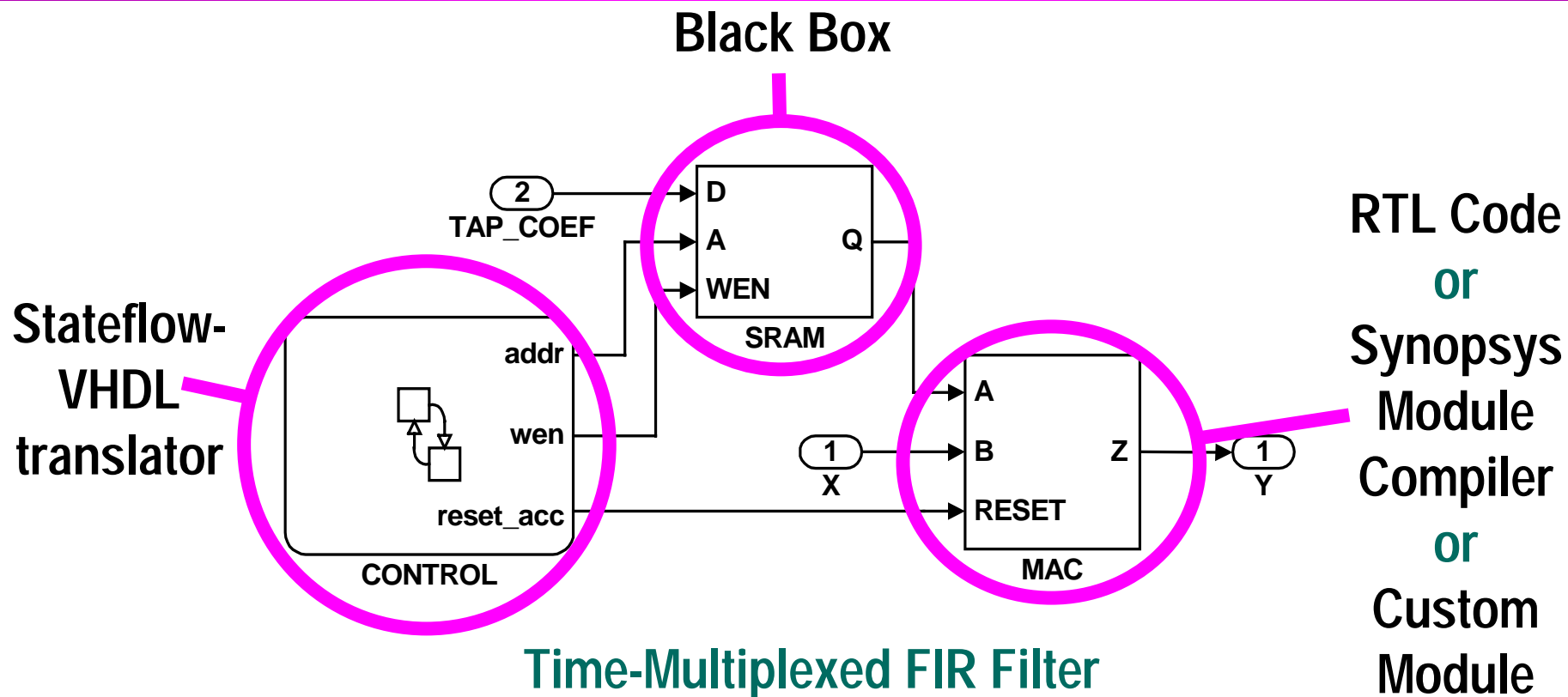
# Hierarchical Dataflow Graphs



## Time-Multiplexed FIR Filter

- Want the hardware hierarchy to match dataflow graph hierarchy
- Grouping needed to hide complexity
- Referencing needed to save design effort

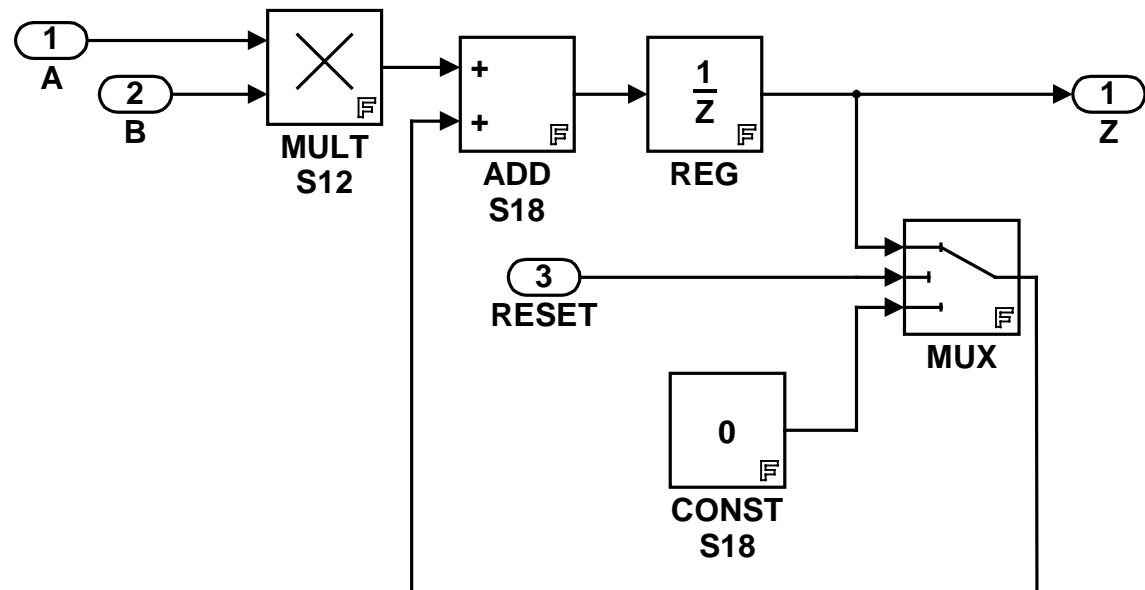
# Specifying Circuit Decisions



- Macro choices embedded in dataflow graph
- Cross-check simulations required

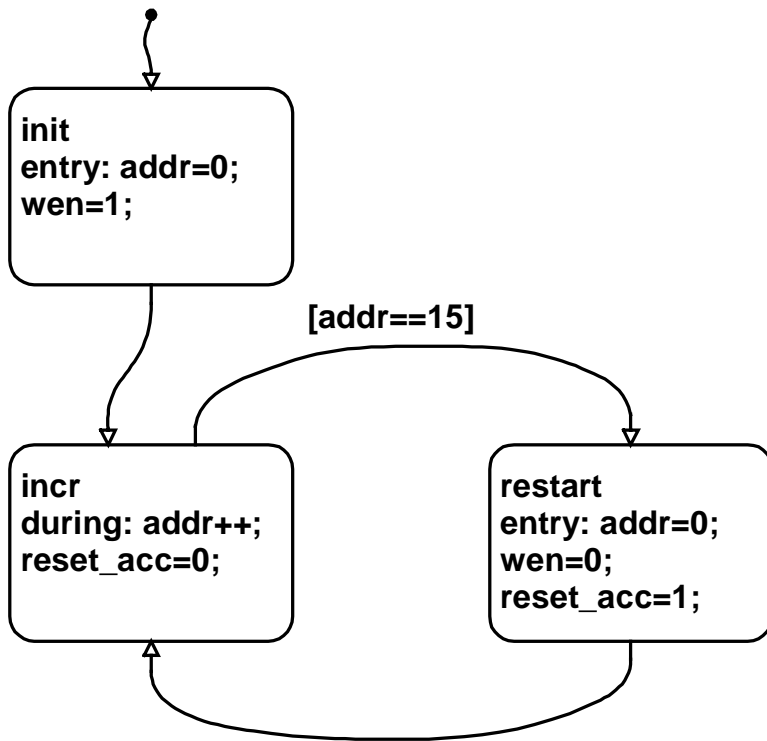
# MAC Dataflow Graph

- Discrete-Time  
(cycle accurate)
- Fixed-Point Types  
(bit true)
- No need for RTL  
simulation
- Embed macro choices
- Simulink (from  
Mathworks)



Multiply / Accumulate

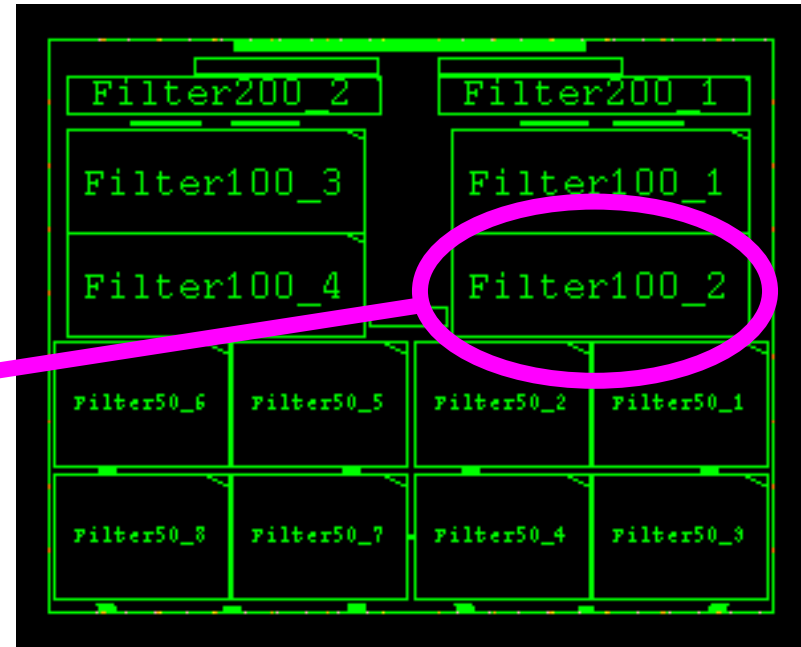
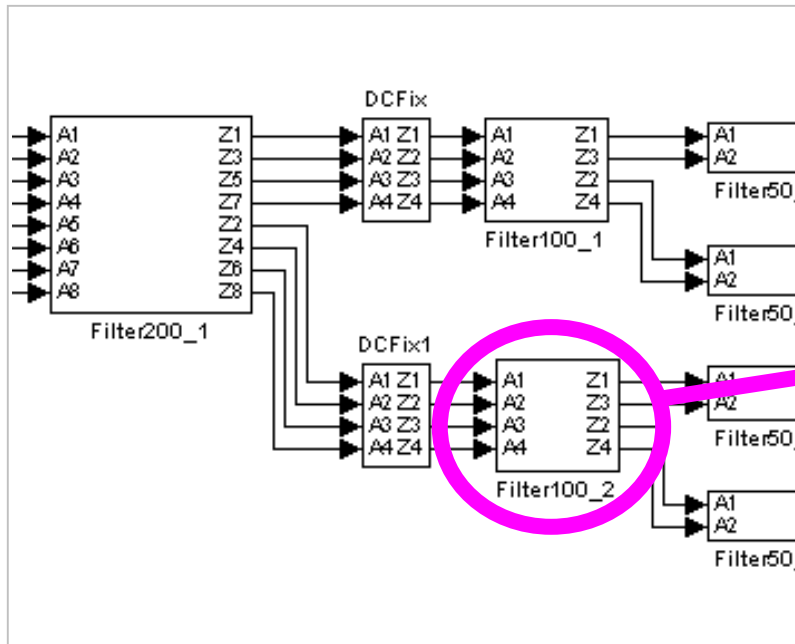
# Modeling Control Logic



## Address Generator / MAC Reset

- Extended finite state-machine editor
- Co-simulation with dataflow graph
- **New Software:** Stateflow-VHDL translator
- No need for RTL

# Capturing Floorplan Decisions



## Parallel Pipelined FIR Filter

- Commercial physical design tools used
- Instance names in floorplan match dataflow graph
- Placements merged on each iteration
- Manhattan distance can be used for parasitic estimates

# Floorplan Specifications

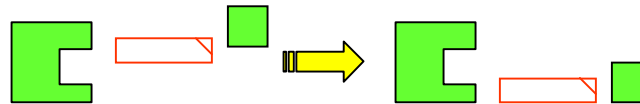
---

The Simulink Designer is responsible for creating the floorplan in *DesignPlanner* with the following functions:

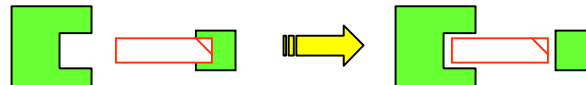
1) Draw Standard Cell Rows



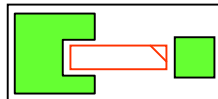
2) Align



3) Distribute/Compact

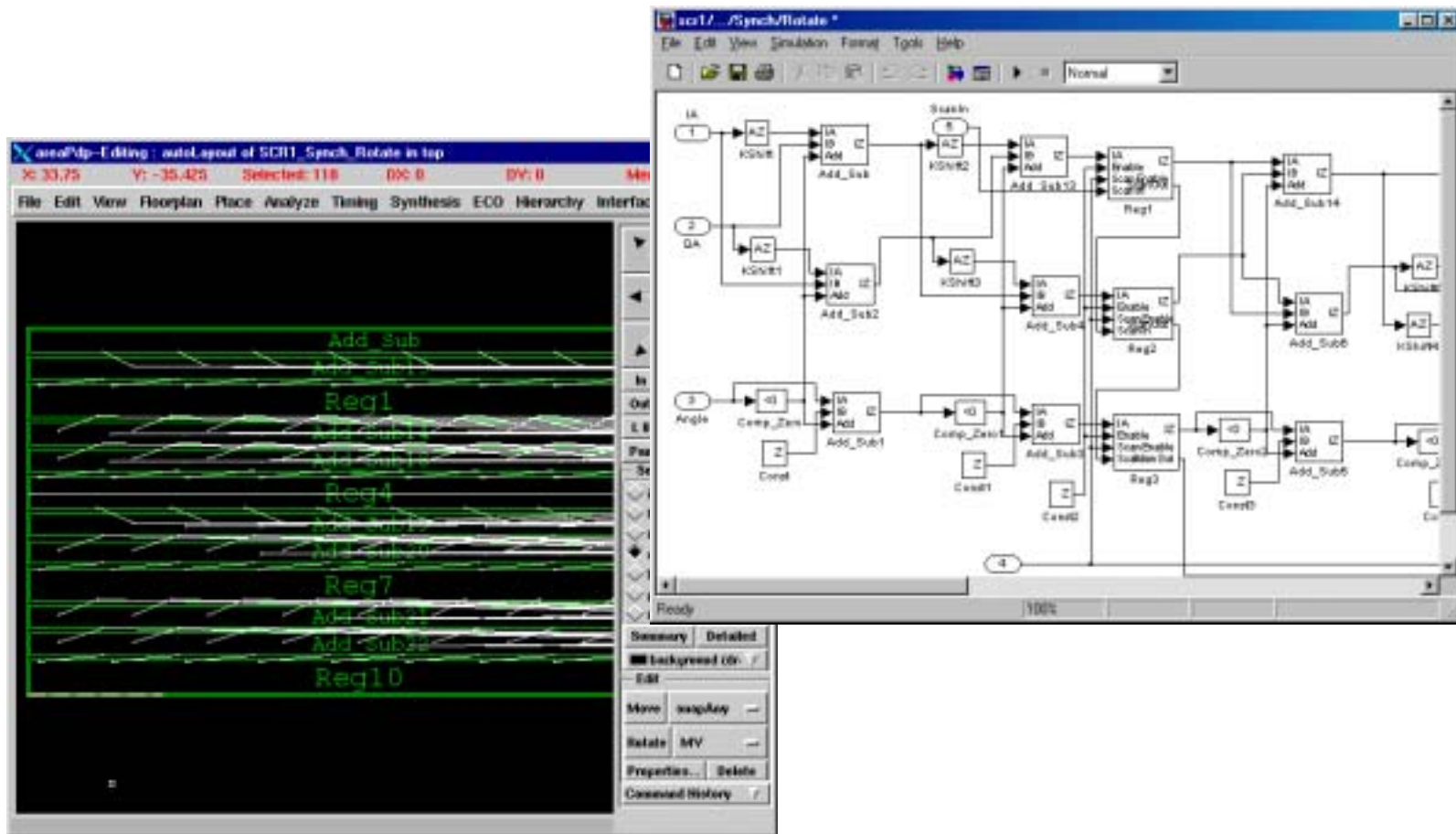


4) Boundary Compaction

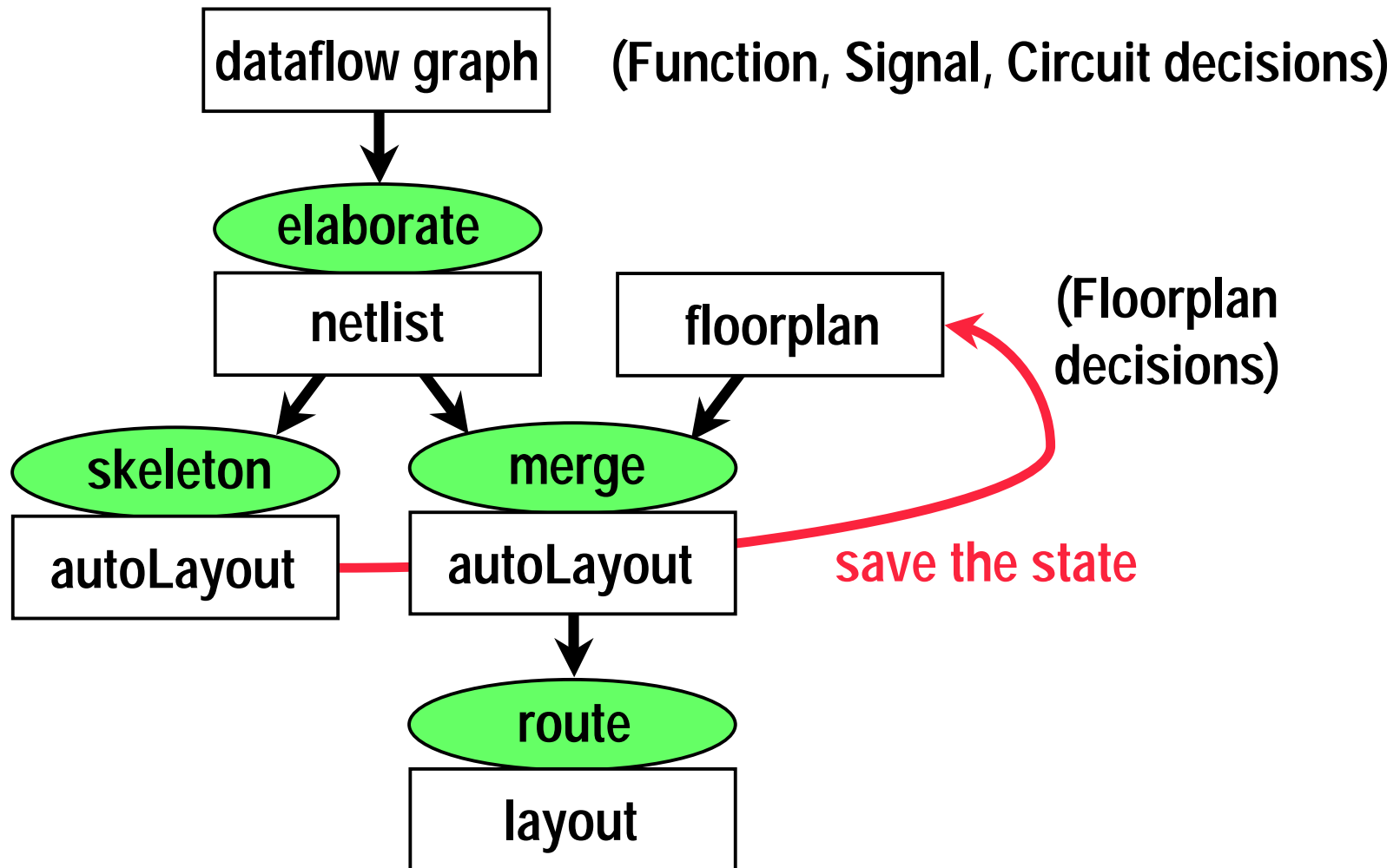


The floorplan contains placement information **only**

# Example Floorplan



# Floorplan Merged on Each Iteration

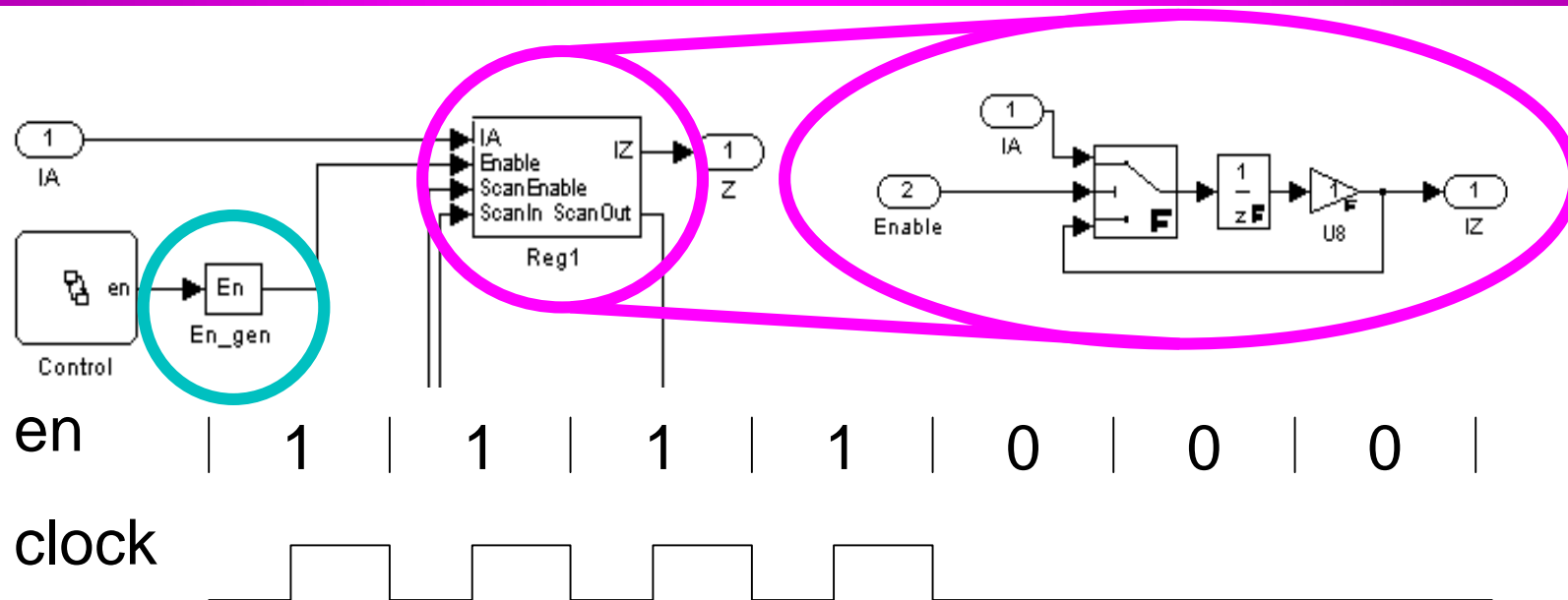


# Outline

---

- **Chip-in-a-Day Flow**
  - » Design Decisions
  - » Estimation
  - » Automation
- **Enabling Factors**
- **Design Examples**
  - » Design Effort
  - » Problems

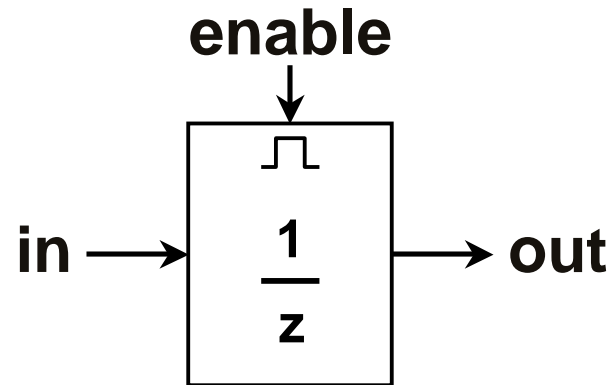
# Gated Clocks for Low Power



- Clock gating is modeled with **Enable** signals which can freeze the state of a register at the architecture level
- **Enable Generators** become gated clock buffers in the physical design

Entered at the algorithmic/architecture level as a Simulink subsystem

# Issues in making Simulink cycle accurate and bit true.... Example

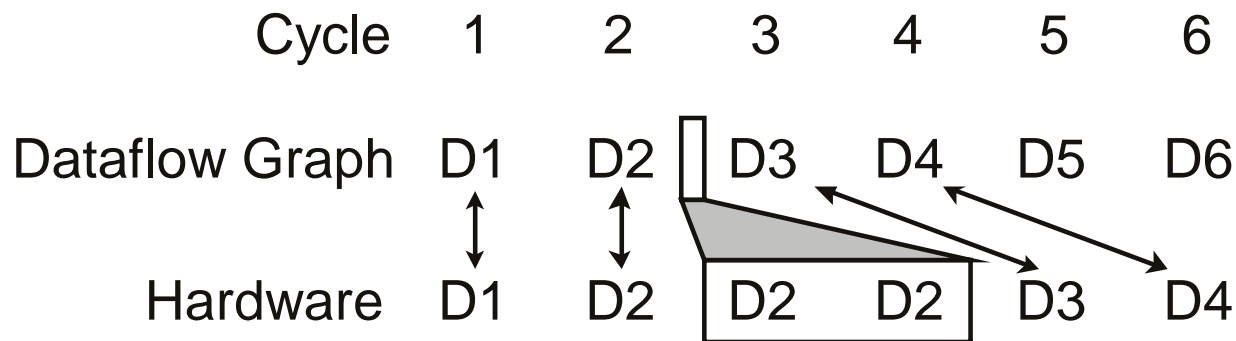


Cycle	1	2	3	4	5	6
enable	0	1	0	1	0	1
in	D1	D2	D3	D4	D5	D6
(Hardware) out	-	-	D2	D2	D4	D4
(Dataflow Graph) out	-	-	-	D2	D2	D4
"Don't Care" Cycles			X		X	

- State is updated but output is held when disabled
- Solved by specifying certain cycles as "don't care"

# Moving Beyond Cycle-Accuracy

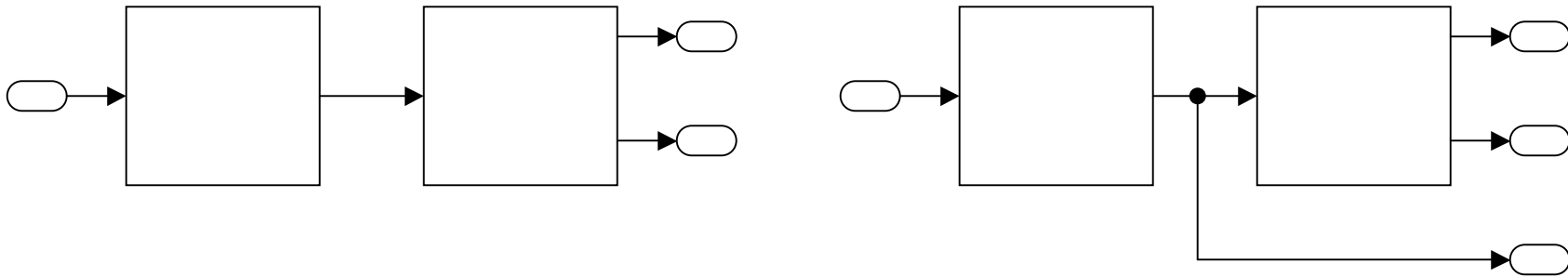
---



- Using “don’t care” windows could relax cycle-accuracy constraint
- Must be automated into verification process

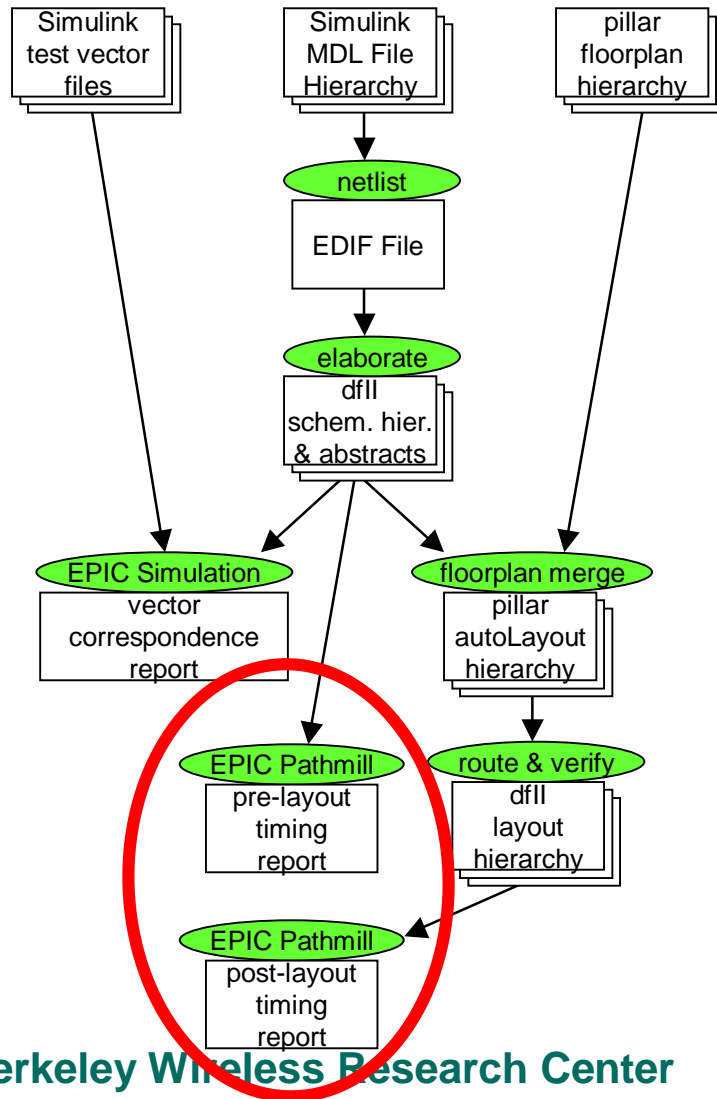
# Simulation only primitives

---



- Early versions inferred pads from ports
- Another copy of design was maintained to examine internal signals (decisions expressed twice)
- Today's version allows "SimOnly" ports

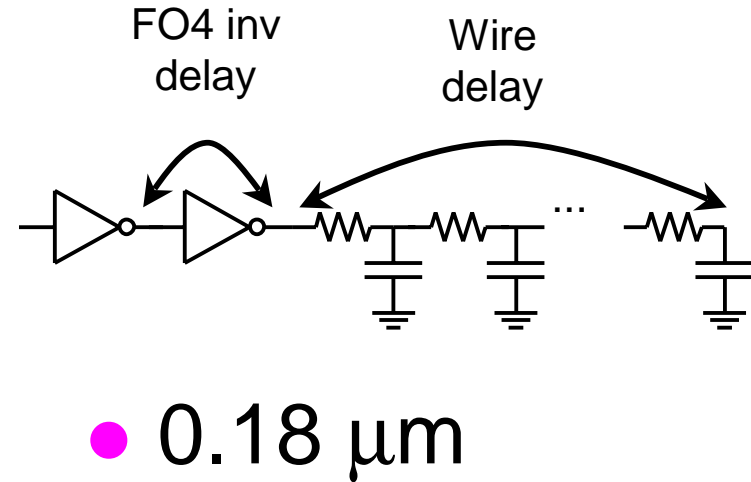
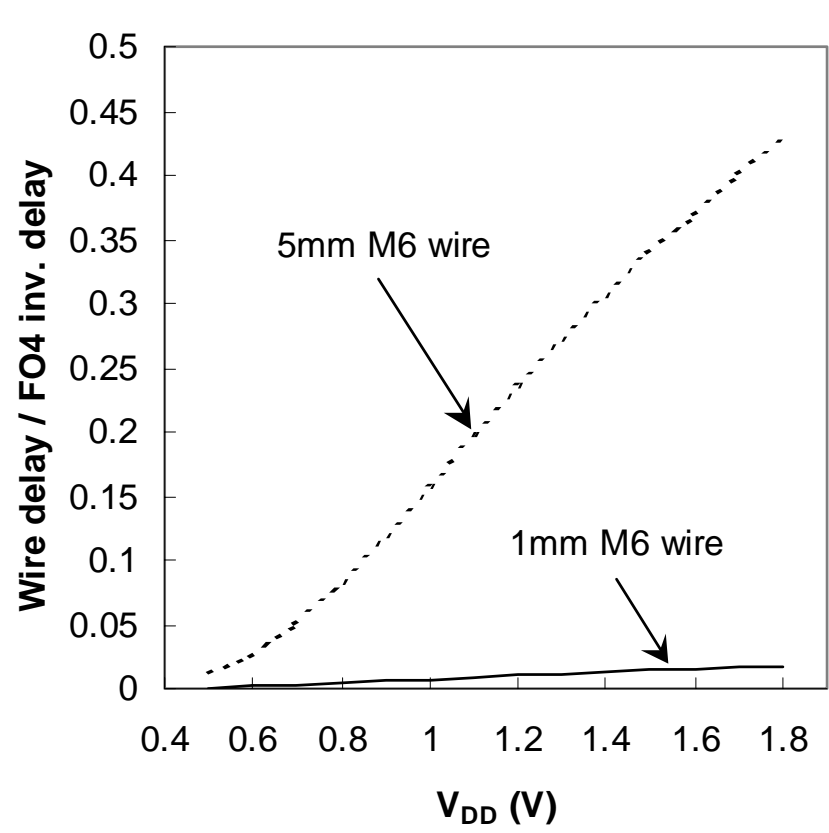
# Avoiding Timing Closure Problems



- Delay Characterization of Modules
- Buffer Macros
- Better Wire-Load Models since floorplan is known
- Clock and voltage domains corresponding to Simulink subsystems
- Final check with Pathmill

Concise Timing Methodology is still being developed

# Reduced Impact of Interconnect



Long wires can be modeled as lumped capacitances

# Race-Immune Clock Tree Synthesis

$$t_{\text{skew(max)}} < t_{\text{clk-Q(min)}} - t_{\text{hold(max)}}$$

Hierarchical Clock Tree Synthesis

Example Clock Tree	
Stages:	22
Sinks:	7650
Skew:	320 ps
Clock Power:	2.8 mW
Logic Power:	21 mW

← Race margin  
= 580 ps

- 0.18  $\mu\text{m}$
- $V_{\text{DD}} = 1 \text{ V}$

**Demonstrated on a 600k transistor design**

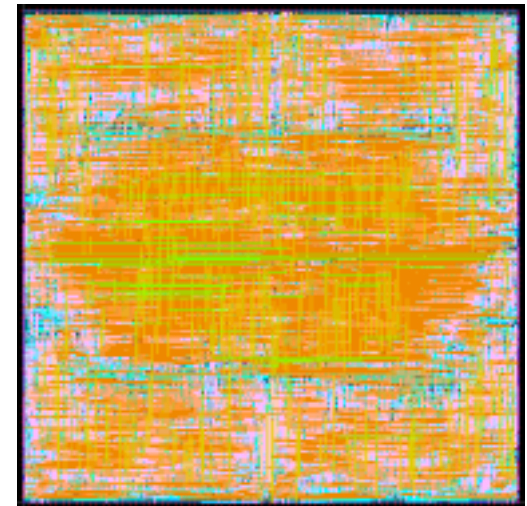
# Outline

---

- **Chip-in-a-Day Flow**
  - » Design Decisions
  - » Estimation
  - » Automation
- **Enabling Factors**
- **Design Examples**

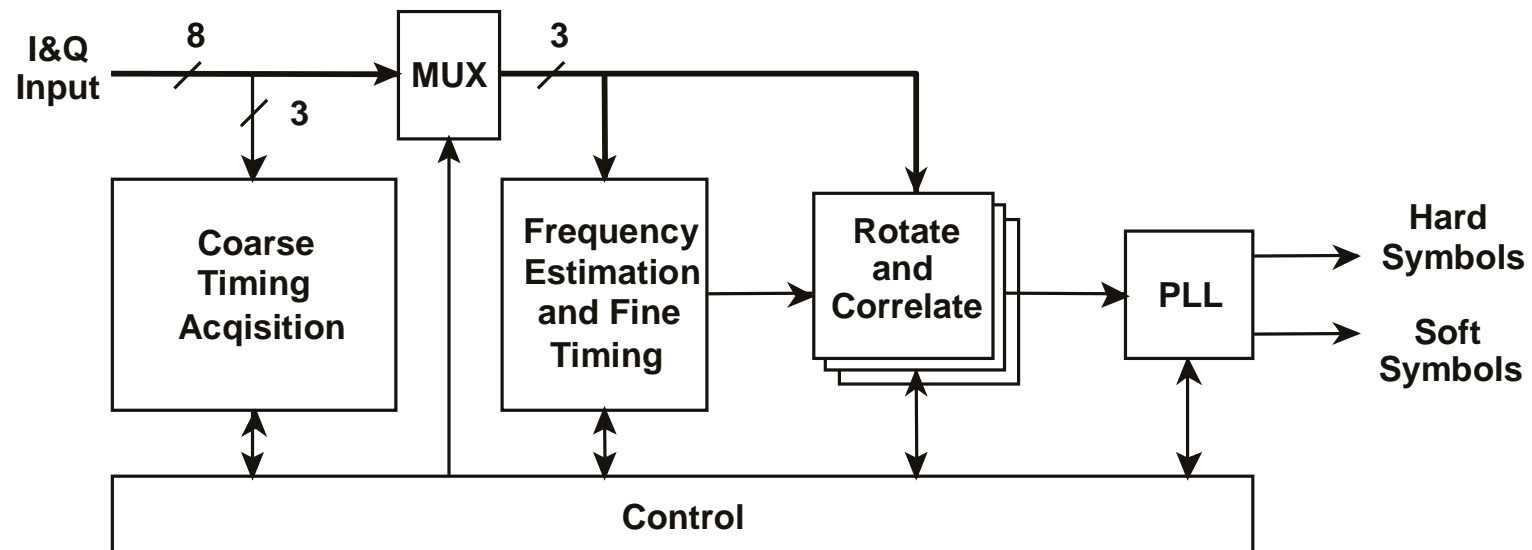
# Example 1: Macro hardening

parallel pipelined FIR filter		
area in 0.25 $\mu\text{m}$		1.4 $\text{mm}^2$
power @ 25 MHz (1 V, PowerMill)		13.0 mW
critical path delay (1 V, PathMill)		18.0 ns
cells		21 k
transistors		240 k
execution time	(elaborate / route)	3 hours
	(characterization)	9 hours
disk space	(elaborate / route)	180 MB
	(characterization)	1.5 GB

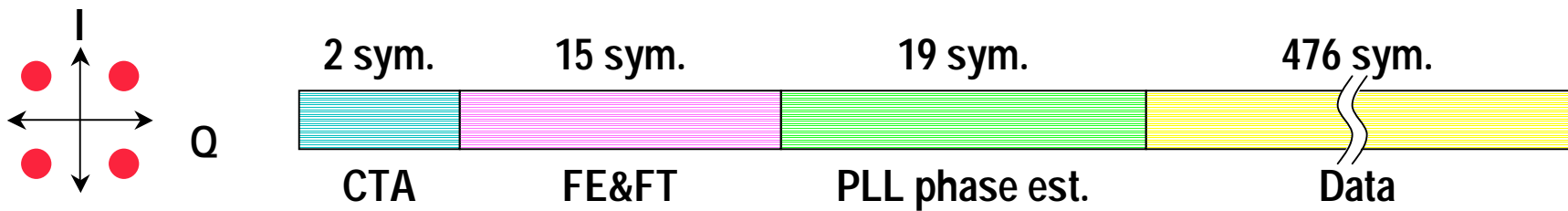


**Most time/disk space spent on extraction and power simulation**

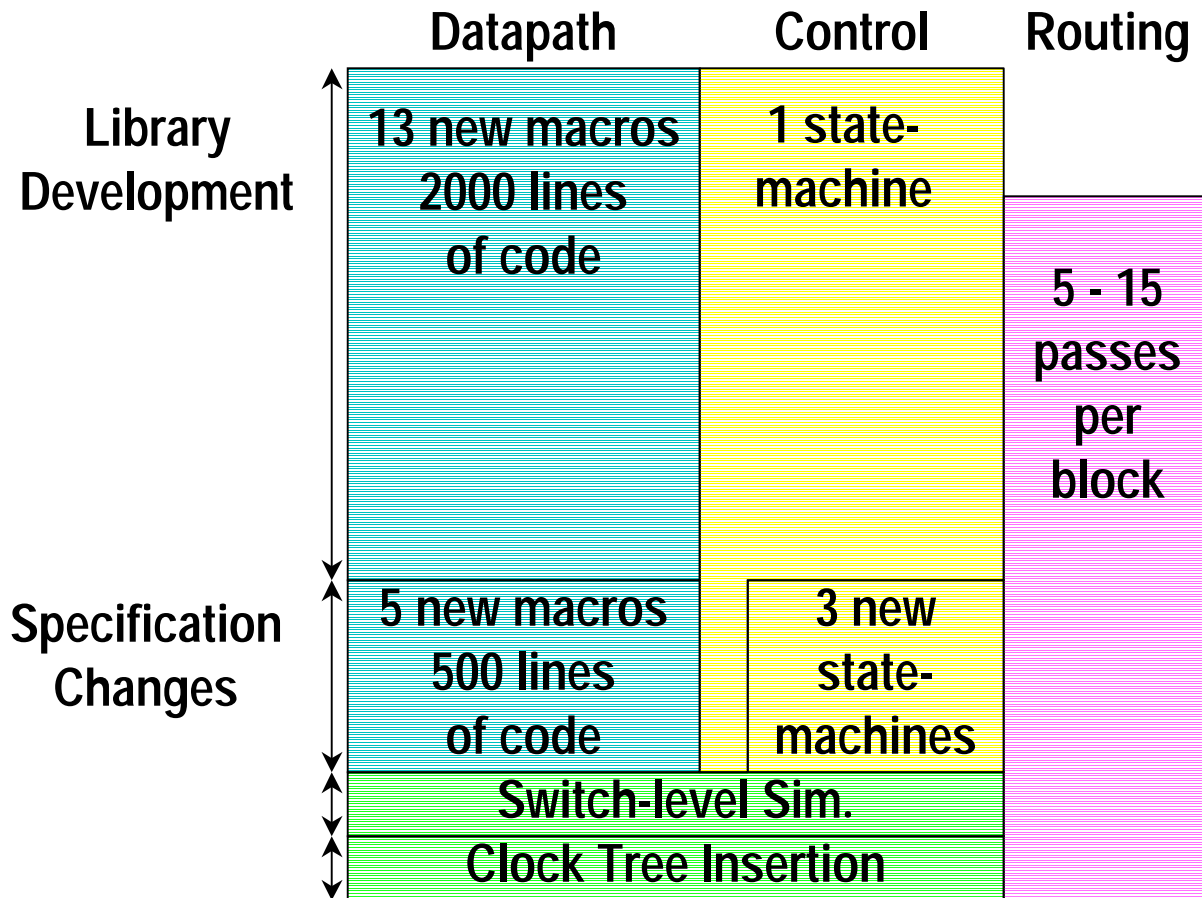
# Example 2: CDMA Baseband Receiver



- DSSS CDMA w/ length 31 spreading code & 25 MHz chip rate
- 806 kHz symbol rate, w/ QPSK gives 1.6 Mb/s data rate
- 7 bit I & Q streams at 200 MHz, 8 parallel streams at 25 MHz



# Design Effort



- Spec. Changes required no modification of datapath macros
- Routing began after 2 months
- No modification to dataflow graph from switch-level sims.
- Flow under development
- Reuse is crucial

# Automation Statistics

---

	Execution Time	Disk Space
Synthesis	1 hour	11 MB
Routing	13 hours	330 MB
DRC & LVS	3 hours	480 MB
Clock Tree Verif.	13 hours	1.2 GB
Other	11 minutes	350 MB
Total	30 hours	2.4 GB

- Assuming automated flow and libraries are debugged, design time is little more than a day

# CDMA Baseband Receiver Layout



- 600k transistors
- 0.18  $\mu\text{m}$
- 1.0 V
- 25 MHz
- 1.1  $\text{mm}^2$
- 21 mW
- single phase clock
- 5 clock domains
- 2 layers of P&R hierarchy

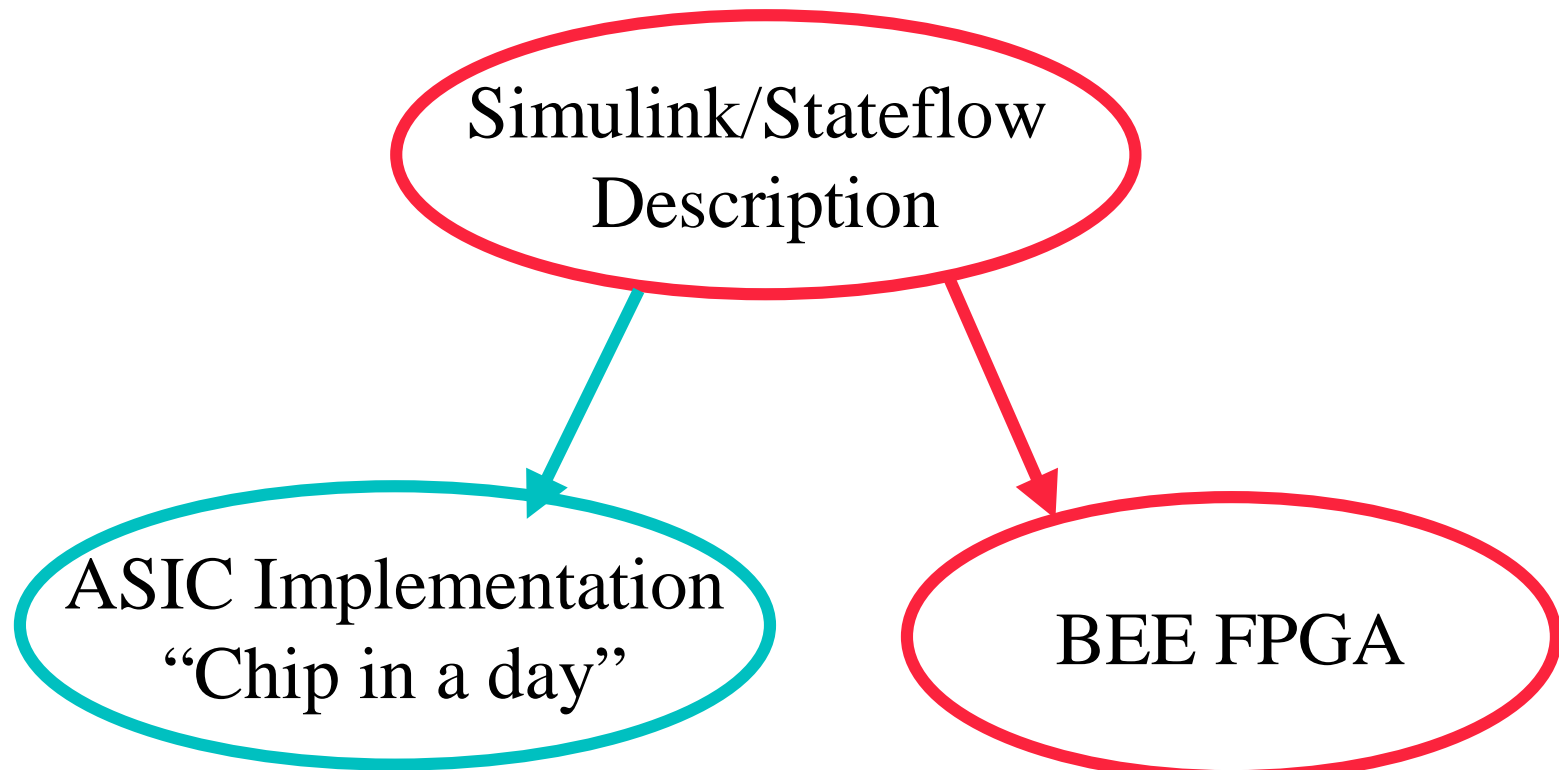
# SShaft Conclusions

---

- **Direct-Mapped hardware is the most efficient use of silicon**
- **Design with dataflow graphs, not sequential code**
- **Don't translate design data, refine it**
- **Verification methodology determines level of abstraction**
- **Chip-in-a-Day design flows are feasible**

# Outline

---



**Chip Implementation**

**Realtime Algorithm Emulation**

# BEE Outline

---

- Motivation
- System Overview
- Special Topic Discussions:
  - » Design Flow
  - » Interconnections
  - » Physical Design

# Algorithm Verification

---

- Simulation – can be slow and inaccurate
- Prototyping – can be slow and expensive
- HW Emulation – fast, accurate, & inexpensive (after construction)

# BEE Objectives

---

- Application to PicoRadio test bed, MCMA, UWB
- Real-time algorithm emulation
  - » 500 billion operations per second
  - » Customized I/O boards
- Fully integrated with existing Simulink-to-Silicon chip design flow (SSHAFT)

# What's BEE?

---

- A real time hardware emulator built with multiple high density Field Programmable Gate Arrays (FPGAs)
- Designed to directly emulate the digital portion of the chip and interface with the analog front-end.
- Fully automated design flow from Simulink to FPGA configuration bit stream.

# BEE Architecture

---

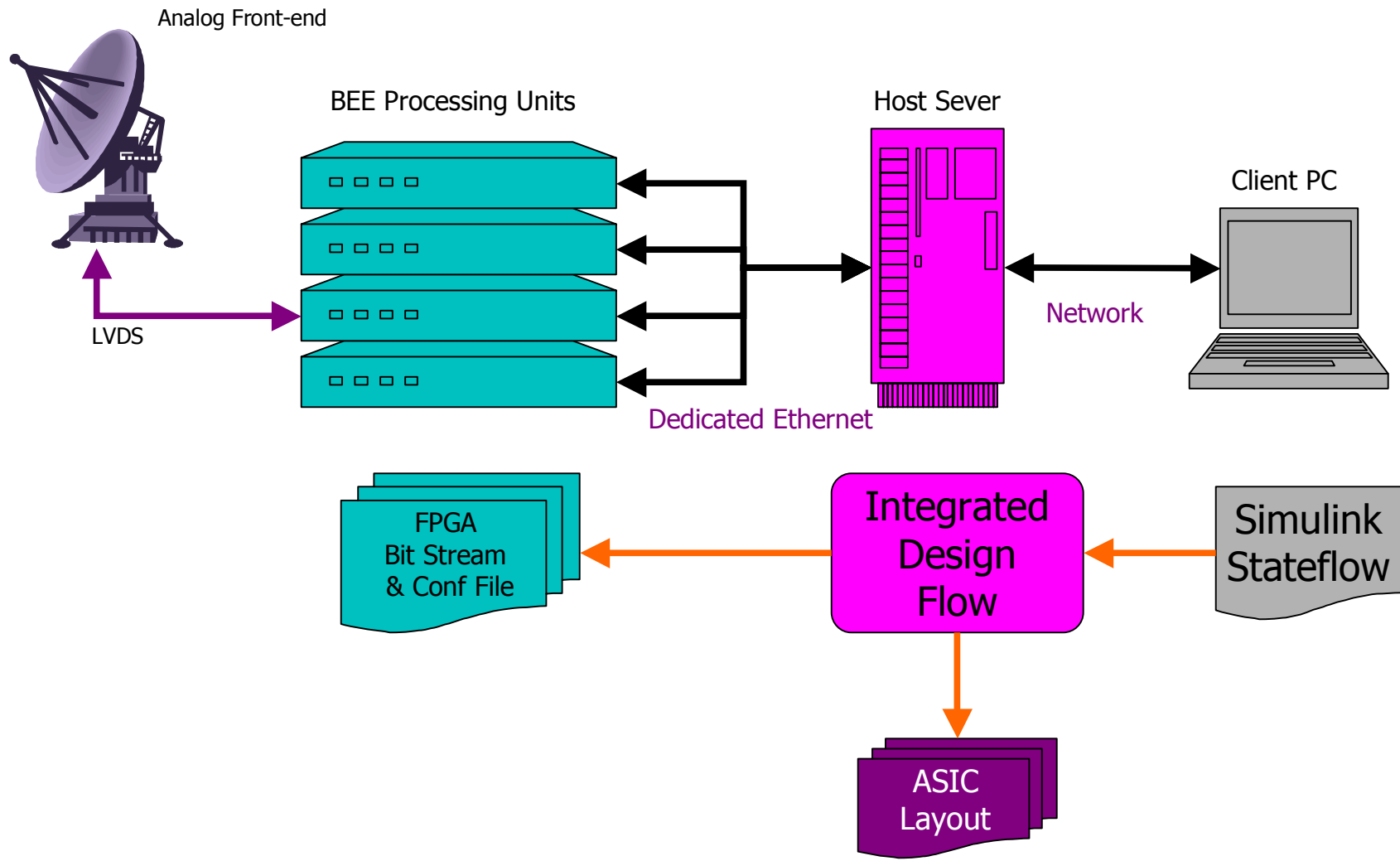
- Processing Board
  - » Total 20 Xilinx VirtexE 2000 chips, 16 for processing, 4 for interchip routing; 16 ZBT SRAM chips, 1MB each.
- Control module
  - » Intel StrongARM 1110, on board 10 Base-T Ethernet, Linux OS
- Radio Rx/Tx Board
  - » 2.4 GHz transceiver
- Design Flow
  - » From Simulink MDL to FPGA bit stream

# Comparison of FPGA and equivalent ASIC

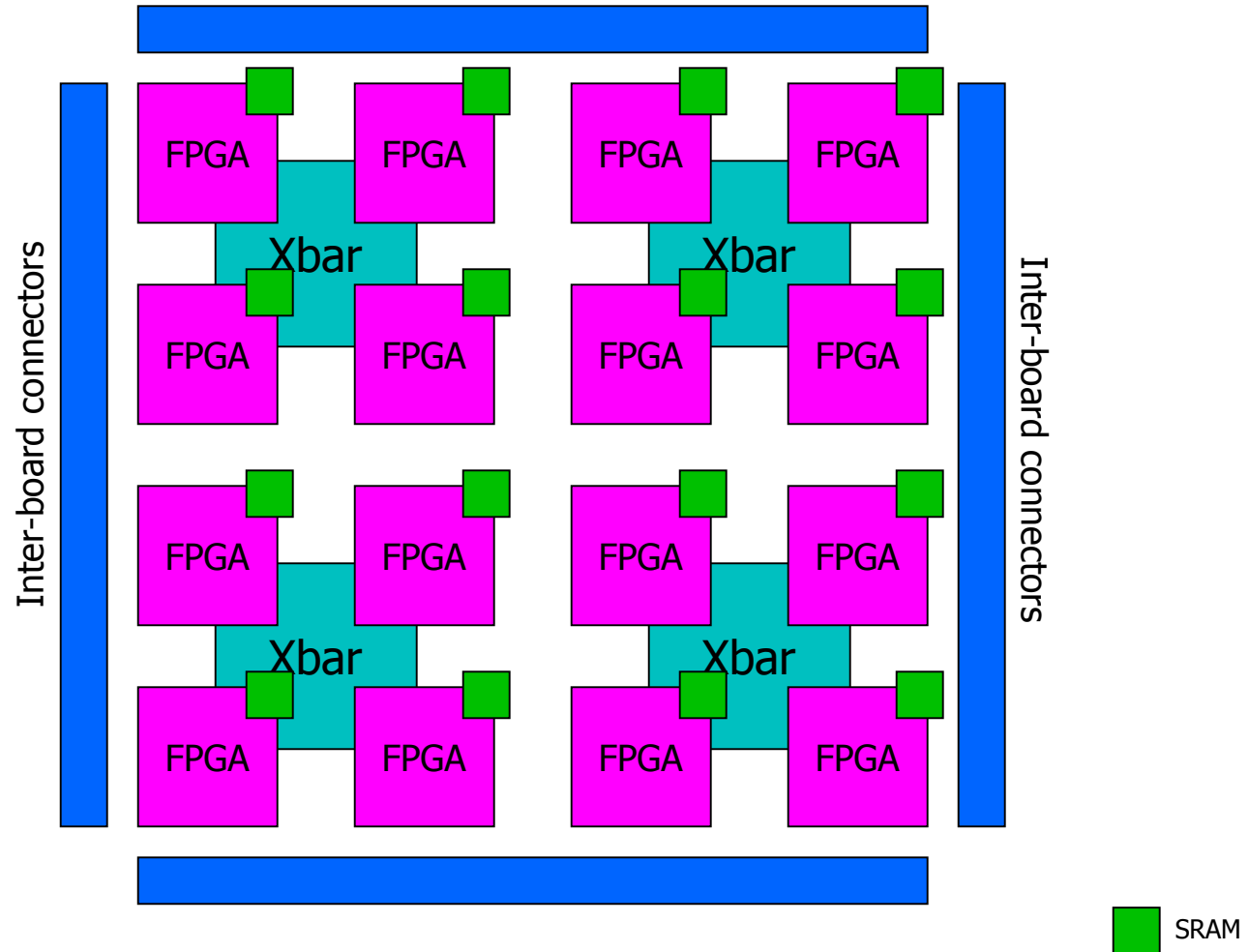
---

- 20 FPGA's each contain "2 Million FPGA gates"
  - » Actually about 100 multipliers per chip or 1000 Adders
  - » BEE clock rate 50 MHz
- $(20 \text{ FPGA's}) * (600 \text{ operators}) * (50 * 10^6) = 500 * 10^9$   
Operations/Second
- Chip metrics 5000 MOPs/mm<sup>2</sup>, 1000 MOPs/mW =>  
Equivalent chip size of 50 mm<sup>2</sup>, power of 500 mW.
- 400 Watts/500 mW => 1000 times more power
- $(20 * 2 \text{ cm}^2) / .5 = 100$  times more area

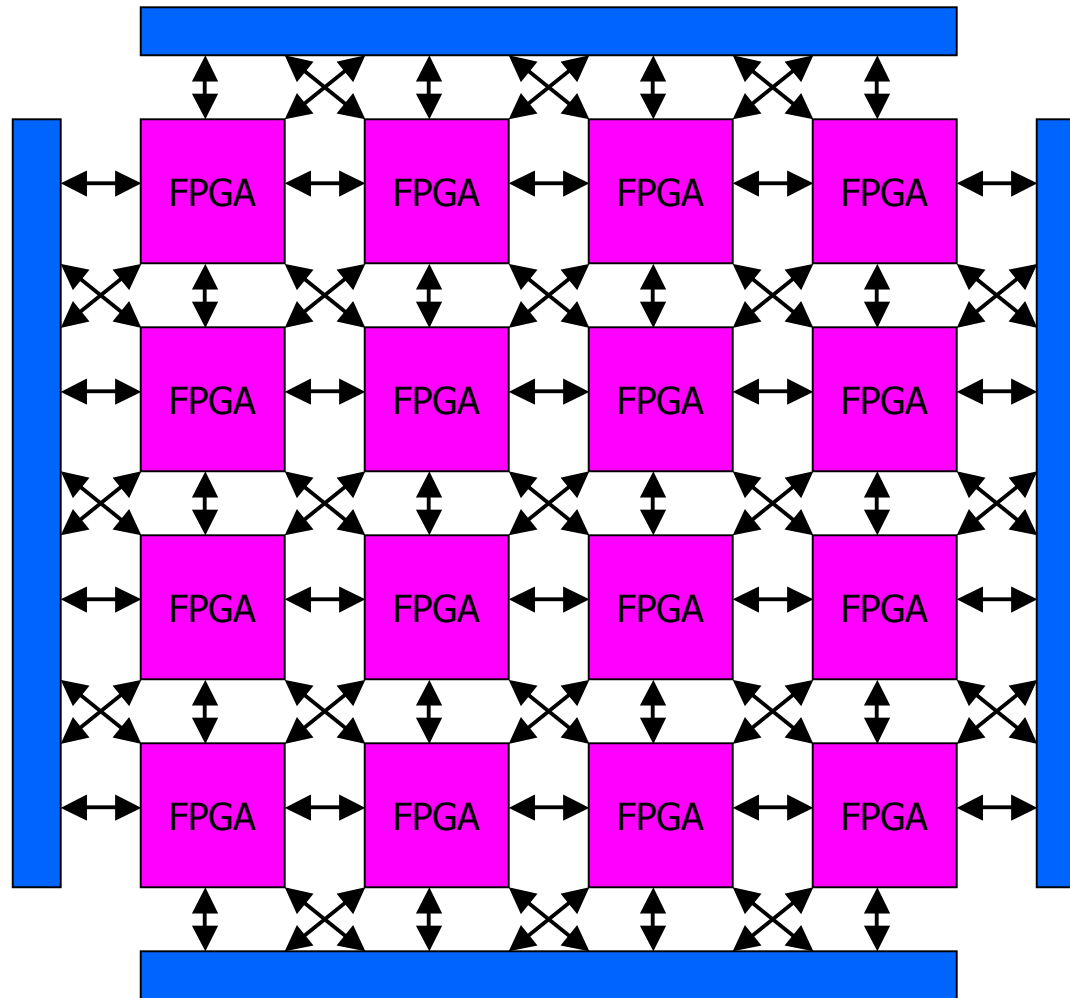
# BEE System Design



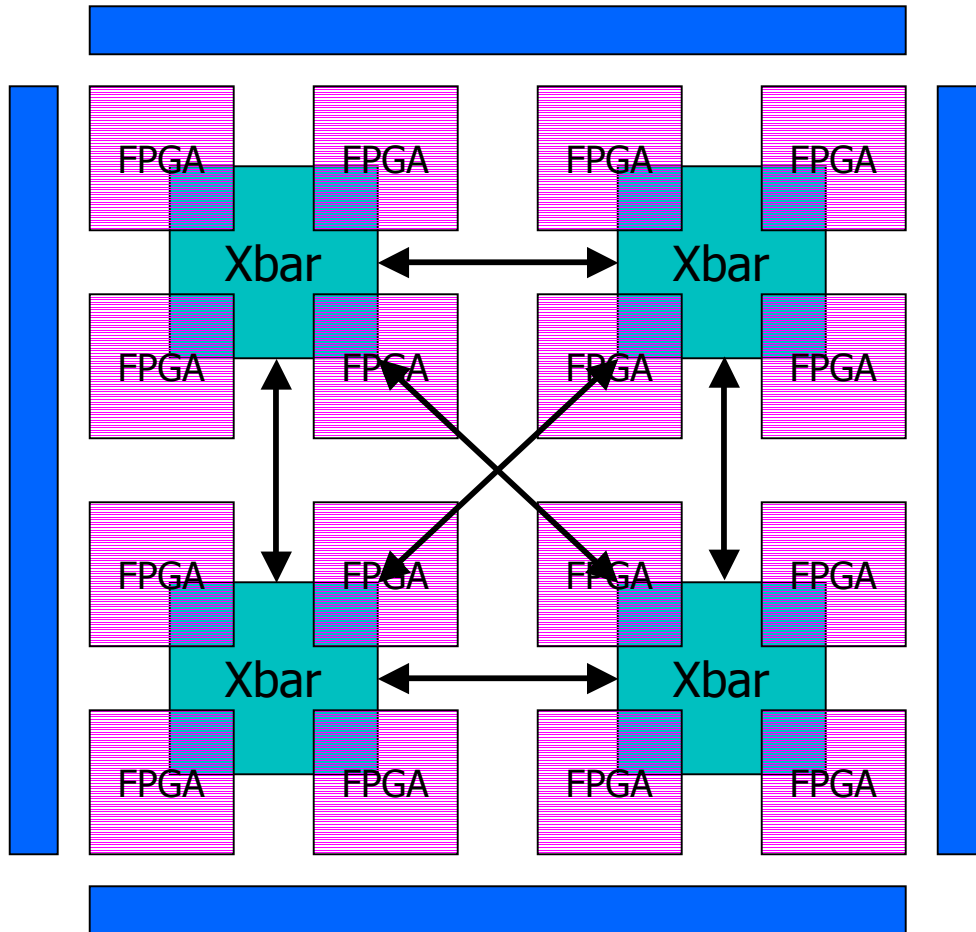
# BEE Processing Board



# Interconnect: Local Mesh

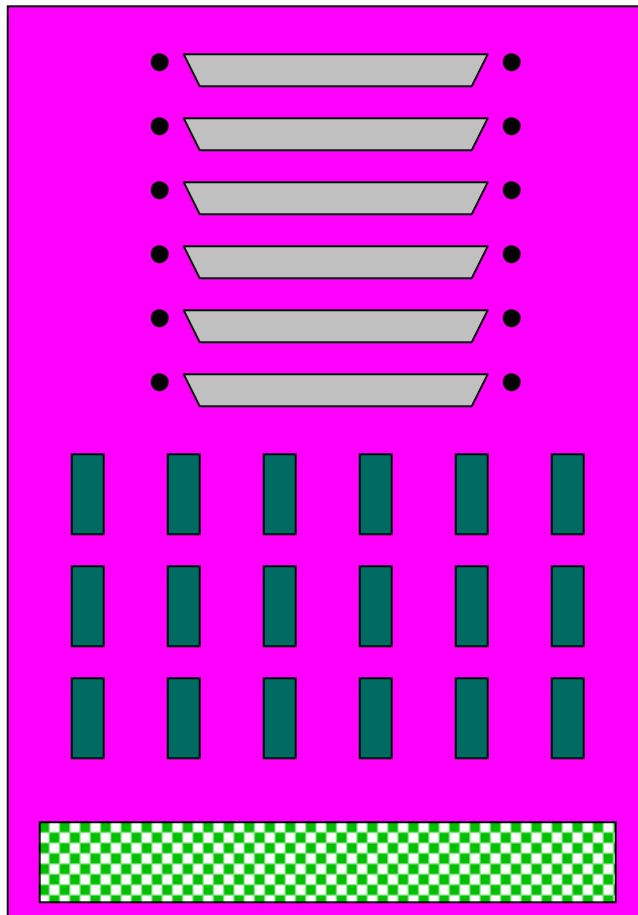


# Interconnect: On Board



- Dedicated FPGA as interconnect routing chips (Xbar)
- Support signal switching over 50MHz between any two FPGA

# Off Module Riser I/O Cards

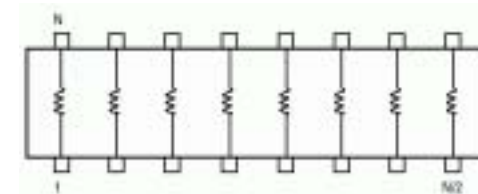


68 Pin  
HD SCSI  
Connectors

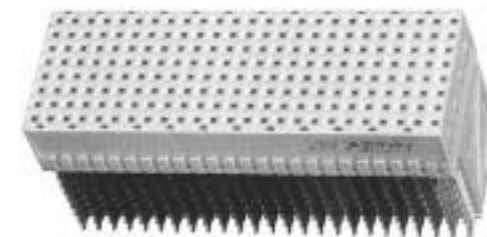
48 signals  
per connector



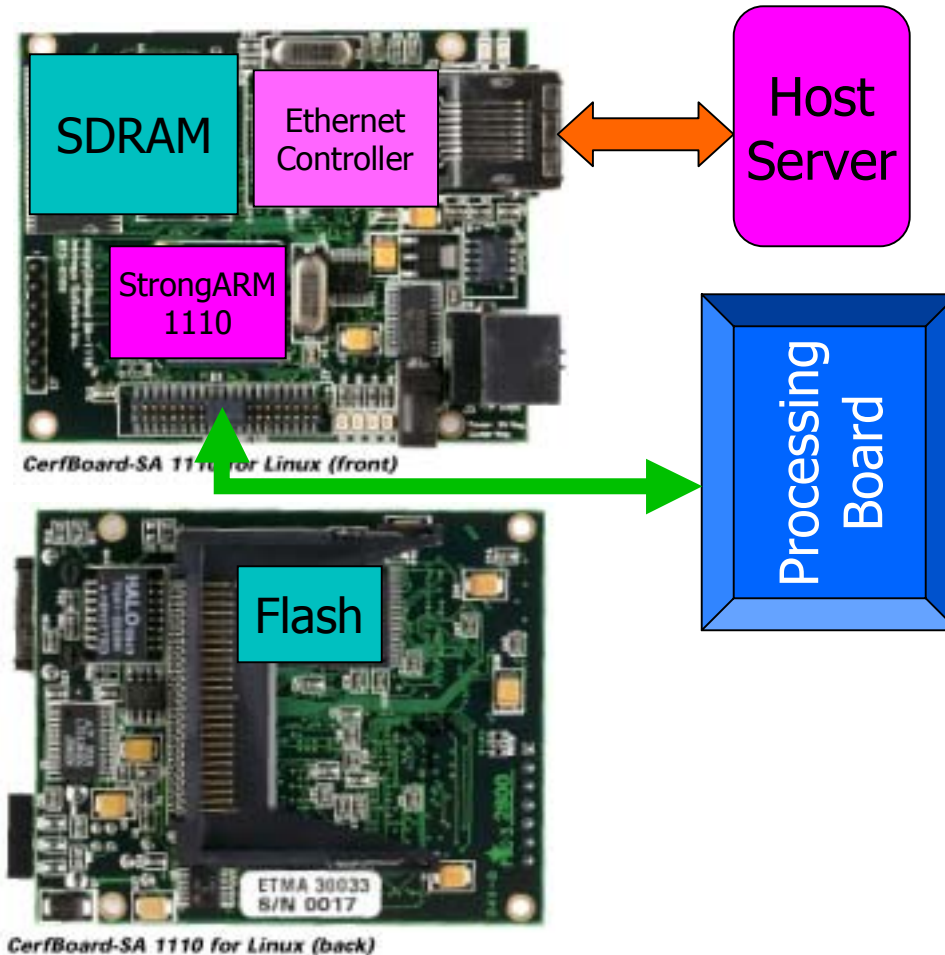
DIP Socket  
Termination  
Resistor  
Array



400 pin HD  
Right Angle  
PCB Connector

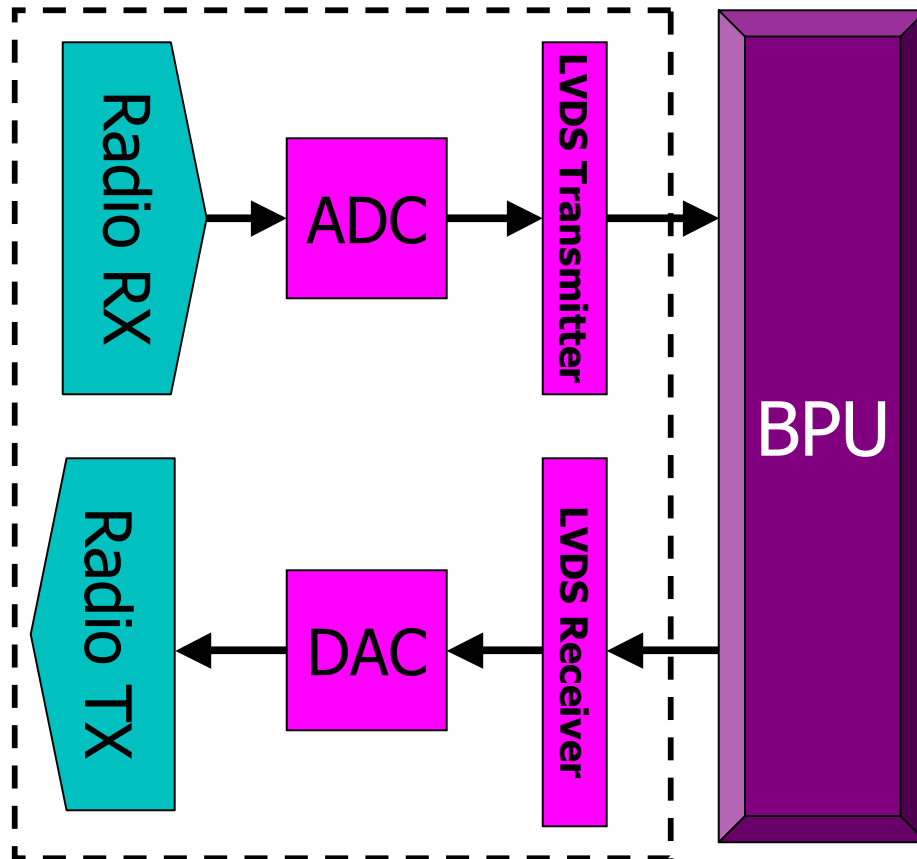


# Controller Module



- 206MHz StrongARM 1110 Processor
- 32MB SDRAM
- 16MB Flash ROM
- 10Base-T Ethernet with RJ45 jack
- Compact Flash slot for expandability
- Linux Kernel 2.4 as OS
- Remote FPGA configuration and read-back through GPIO

# RF Front-End



- Analog circuits along with ADC/DAC are very application dependent
- Separate Analog Front-end maximizes the reusability of BEE
- High speed inter-board connection using Low Voltage Differential Signals (LVDS)

# Special Topic 1: Design Flow

---

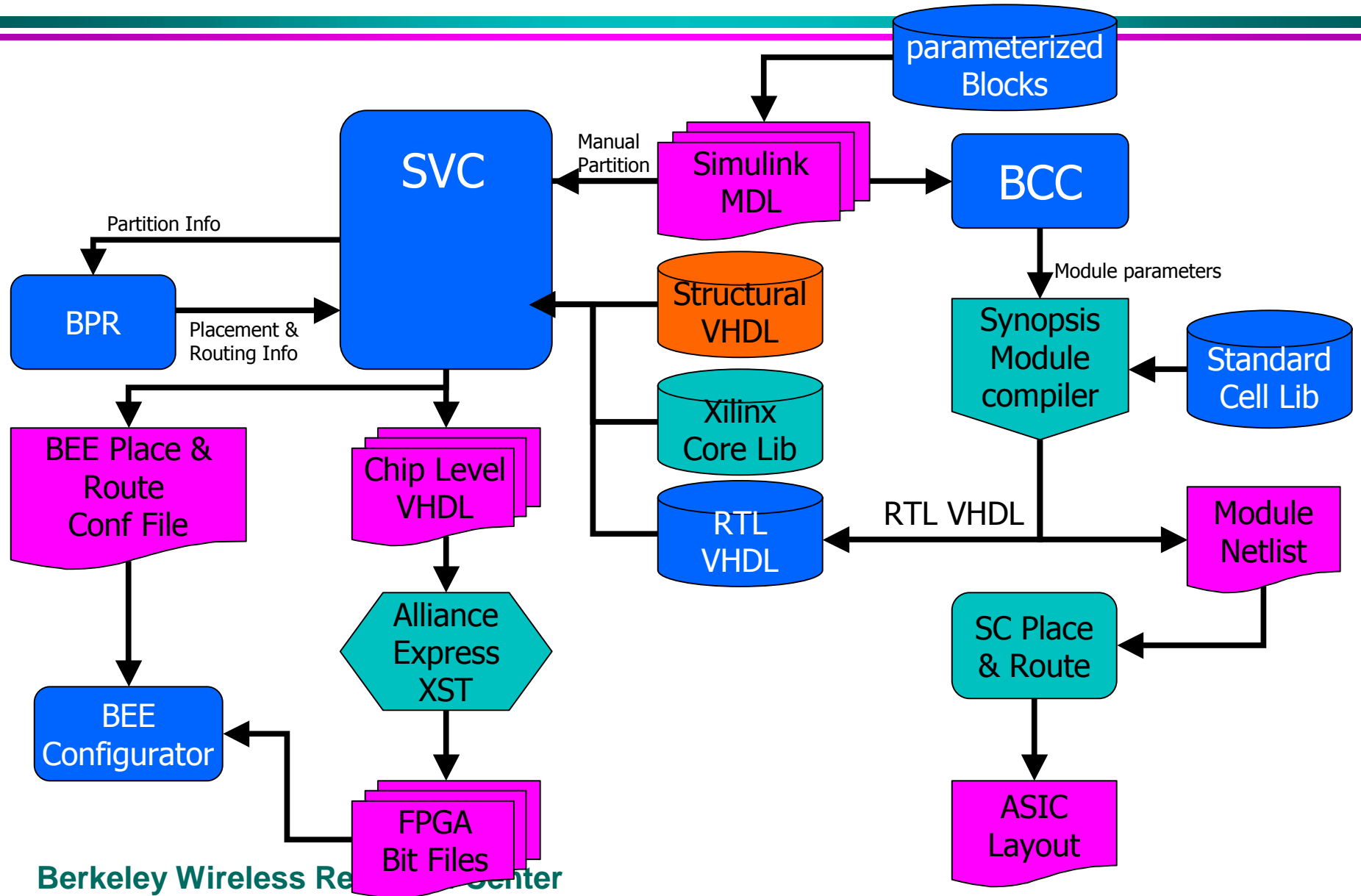
- Design Flow Goals
- BEE Design Flow
- System Level Block Library
- SSHAFT Compliance

# Design Flow Goals

---

- Auto-generation of FPGA bit stream and inter-chip place-and-routing configuration from system level design (i.e., Simulink MDL)
- Full integration with SSHAFT flow
- Computational component level equivalency between ASIC & BEE implementation

# BEE Design Flow

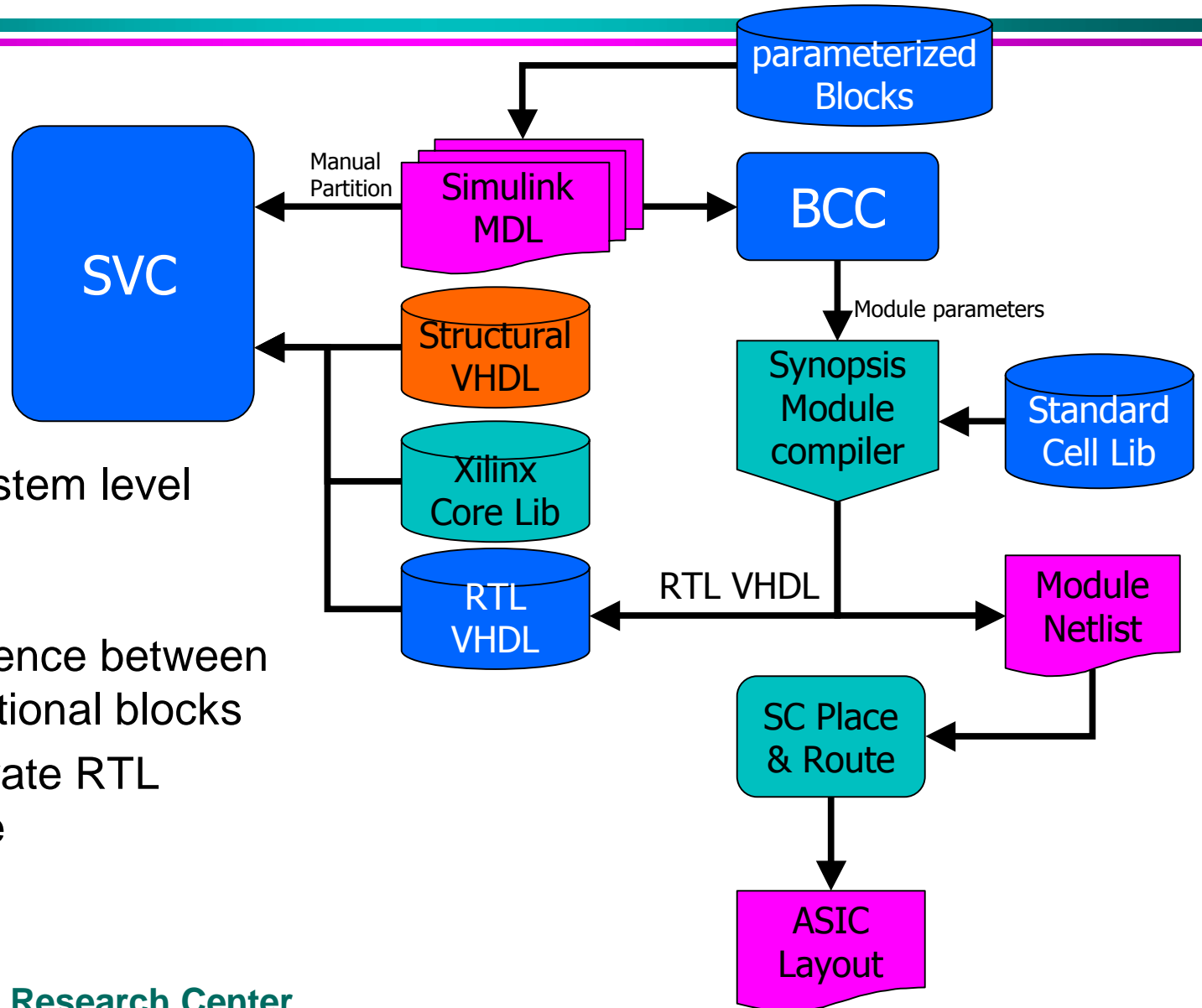


# System Level Block Library

---

- Parameterized system level blocks
  - » Bit-width
  - » Pipeline stages (latency)
  - » Output bits truncation
  - » Architecture
- Built-in flexibility to support multiple hardware descriptions
  - » Structural VHDL
  - » Module compiler parameter file
- Specialized Debug Blocks to record the states of the system

# SSHAFt Compliance



- Identical system level description
- Functional correspondence between all computational blocks
- Cycle accurate RTL equivalence

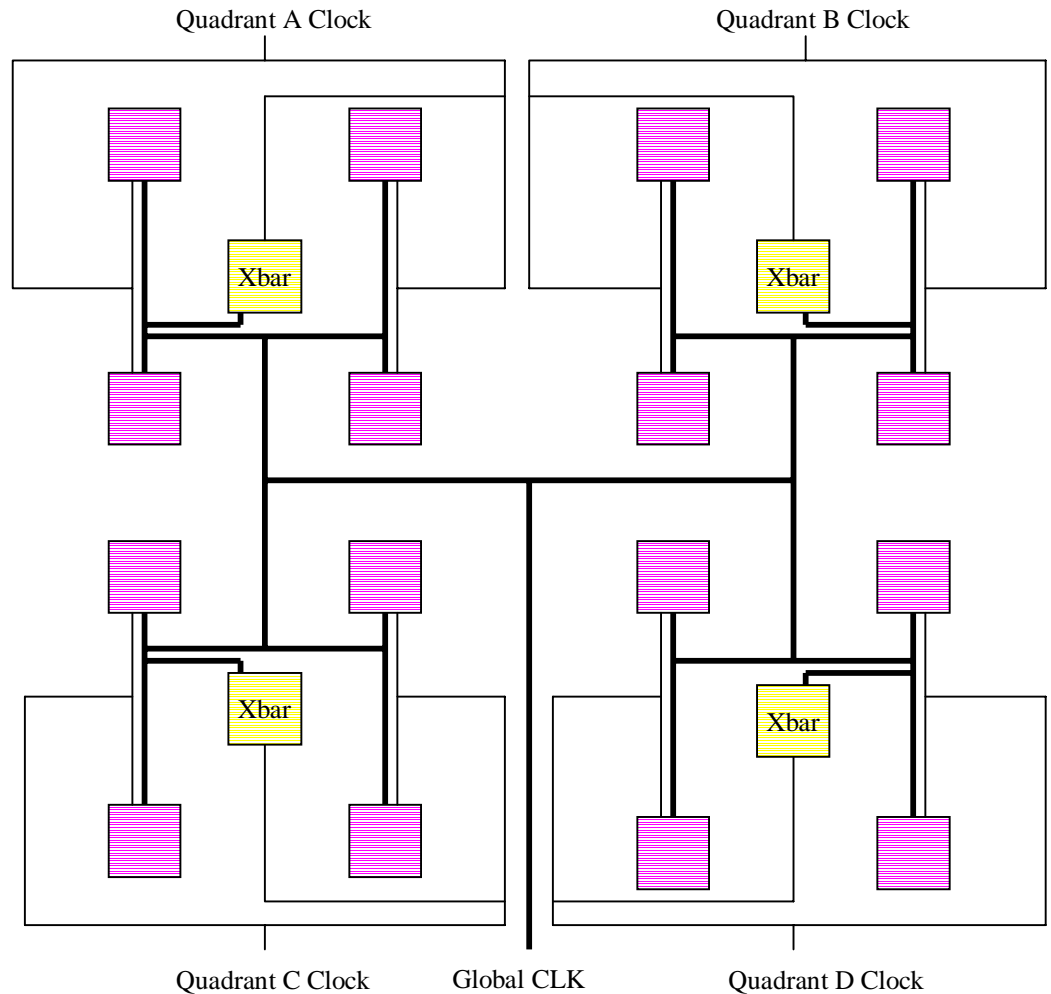
# Special Topic 2: Interconnects

---

- Global signals
- Signal layout
- On-board connections
- Off-board connections
- Termination

# Global Signals

- Global Clock
- Quadrant Clocks
- Global Reset
- JTAG
- FPGA Configuration Bus

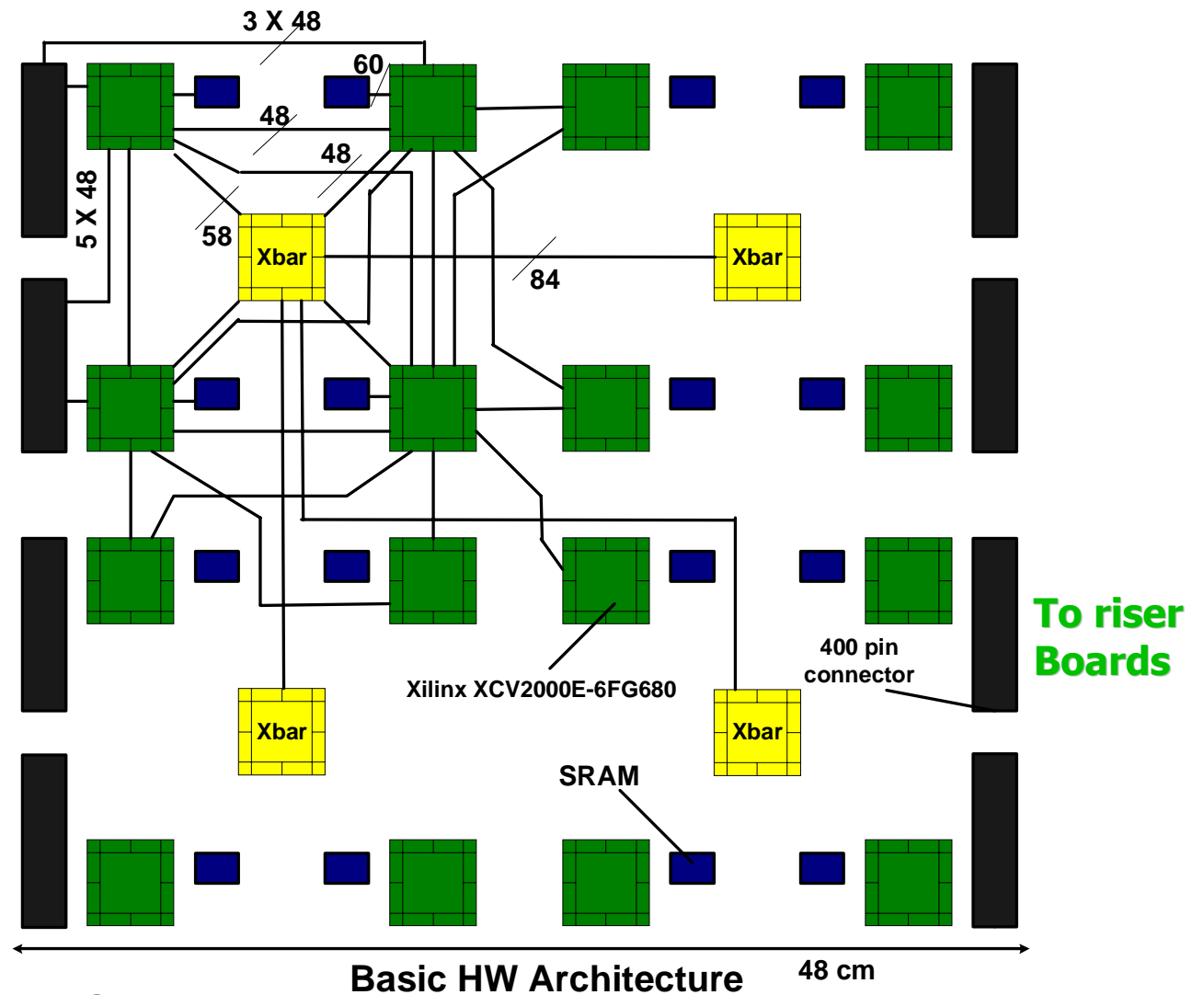


# Signal Layout

Routing shown  
Only for the top  
left quadrant

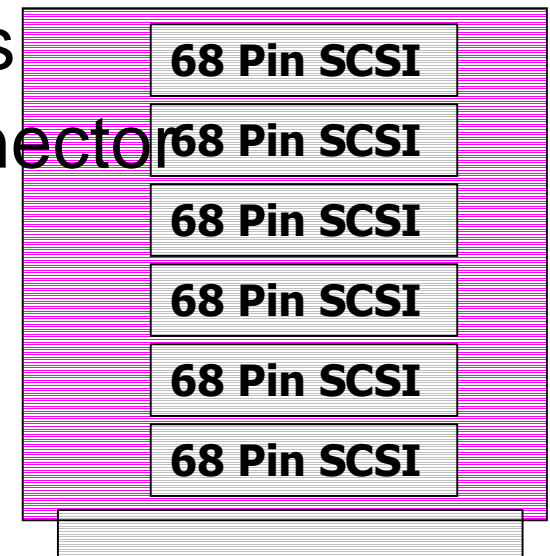
In addition, the  
following routing is  
required:

- Clock Tree
- Quadrant Clocks
- Global Reset
- JTAG
- FPGA Programming



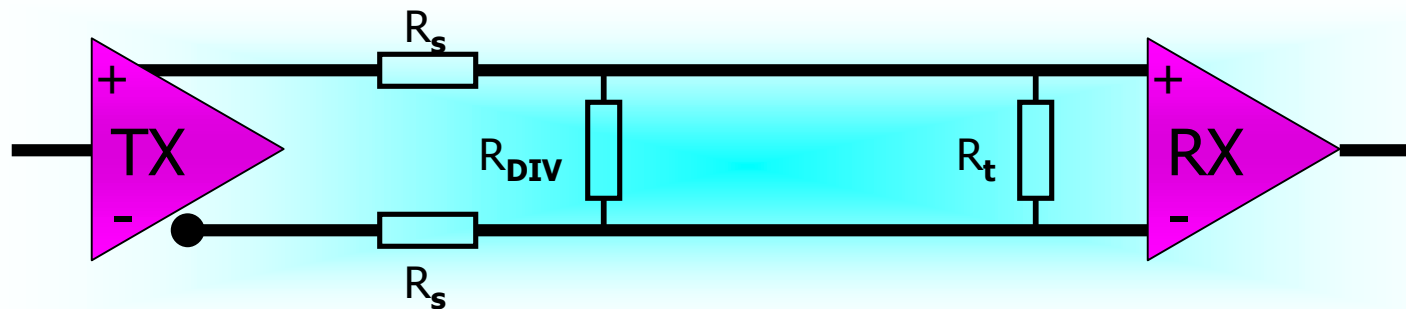
# Off-board Connections

- FPGA-to-Riser Board: 288 signals (per Riser Board) (LVDS & LVTTTL)
- Riser Board (8 Boards):
  - » 6 X 68 Pin SCSI = 408 Connector Pins / Board
  - » 6 X 48 = 288 Actual Signal Pins
  - » 20 Ground Pins per SCSI Connector
  - » Required since there is not enough space on the main board



# Termination

- FPGA-to-FPGA: No, trace length 6~12 cm
- FPGA-to-XBAR: No, trace length 5 cm
- XBAR-to-XBAR: Series, trace length 40 cm
- FPGA-to-SRAM: No, trace length 3 cm
- FPGA-to-Riser Board: LVDS about 1 m



# Special Topic 3: Testing

---

- BEE Board Assembly Test
- Runtime Hardware Test
- Design Debugging

# BEE Board Assembly Test

---

- Test Interface
  - » SCSI test module
  - » Test clips for memories
  - » LEDs
  - » FPGA I/O permitting:
    - User switches
    - Test points

# BEE Board Assembly Test

---

- Electrical test from the manufacturer
- Testable items:
  - » JTAG FPGA configuration
  - » Ethernet configuration
  - » Signals: FPGA-to-Connectors
  - » Signals: FPGA-to-FPGA-to-Connectors
  - » Signals: Interconnect speed test
  - » Single FPGA, Quadrant, BPU, BPU + FE, All

# Special Topic 4: Physical Design

---

- Power Consumption
- Distributed Power Supply
- Heat Dissipation/Ventilation
- Chassis Design

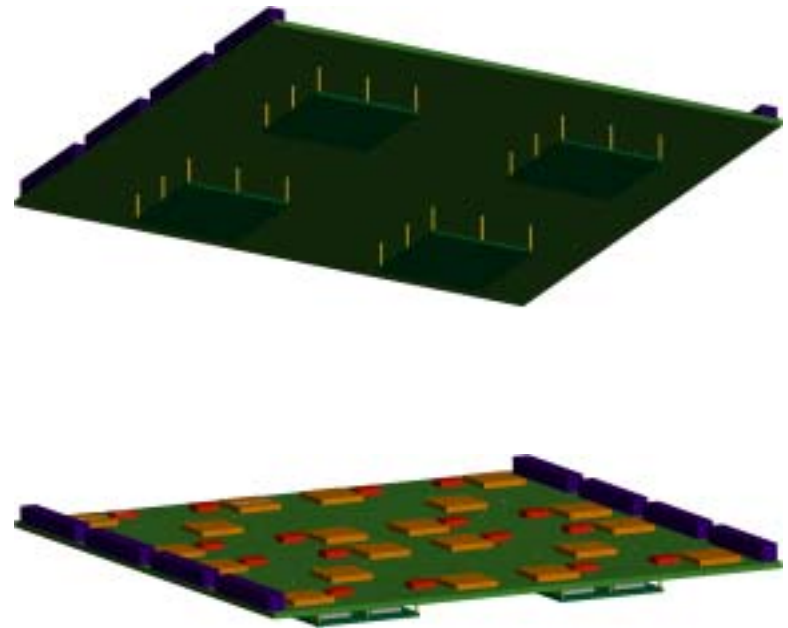
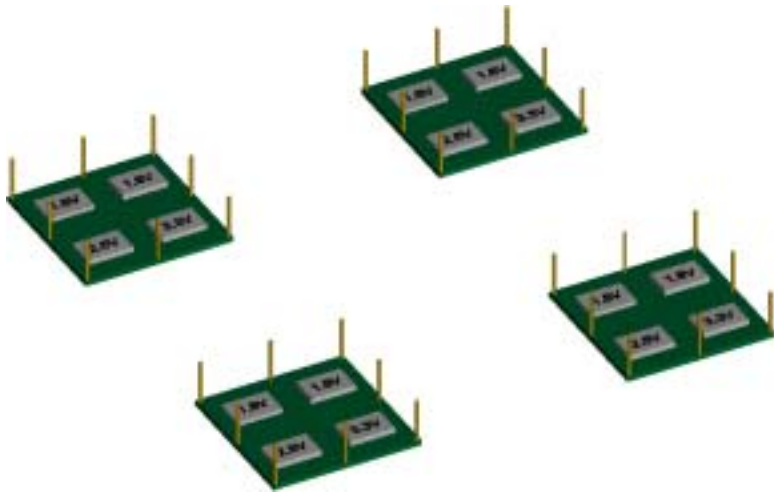
# Power Consumption

---

- Each FPGA consume up to about 20Watt
- 75% of each is consumed by the 1.8V core
- Total of 20 FPGA chips per BPU
- **166 Amps** at 1.8V

# Distributed Power Supply

---



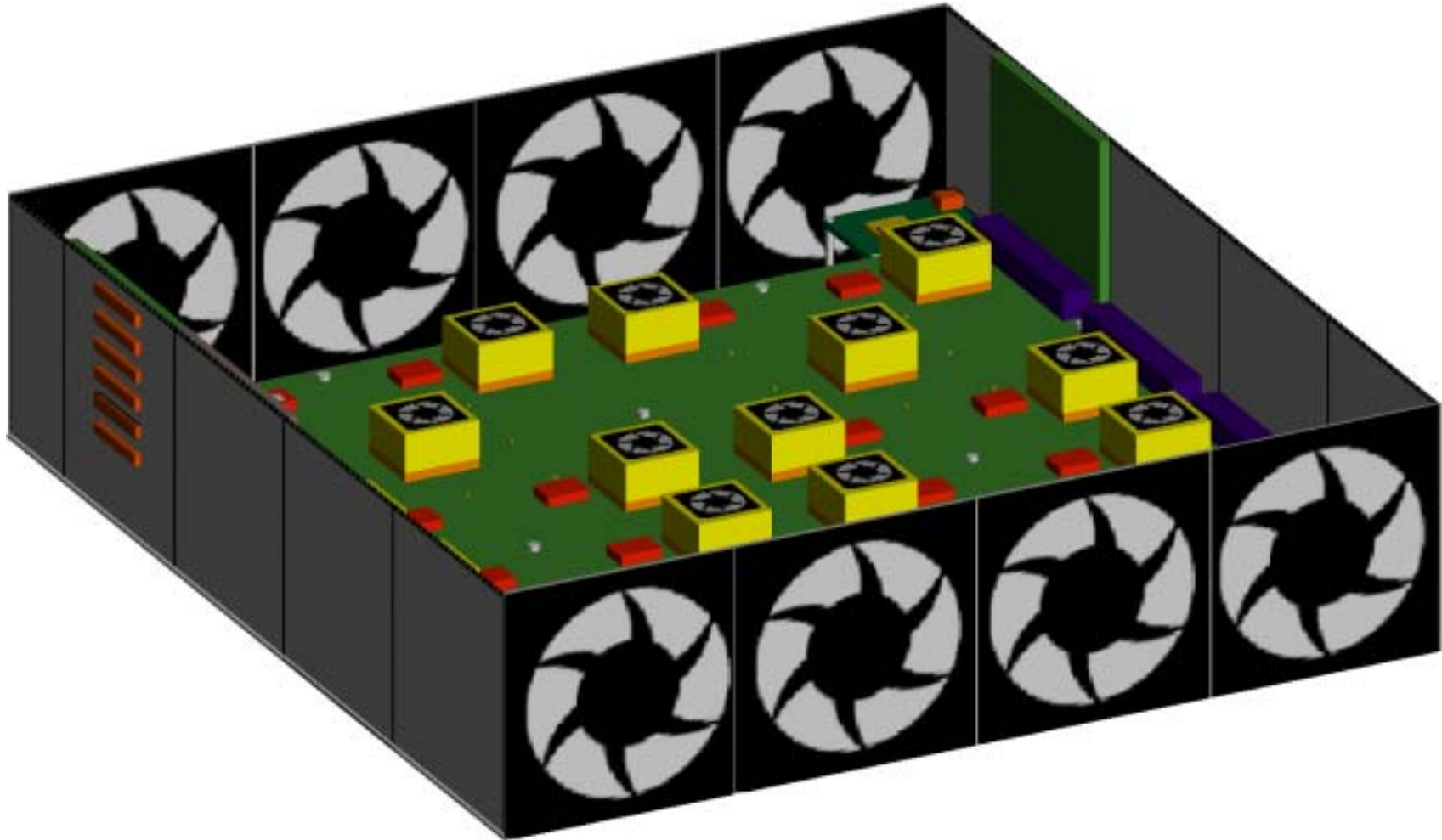
# Heat Dissipation/Ventilation

---

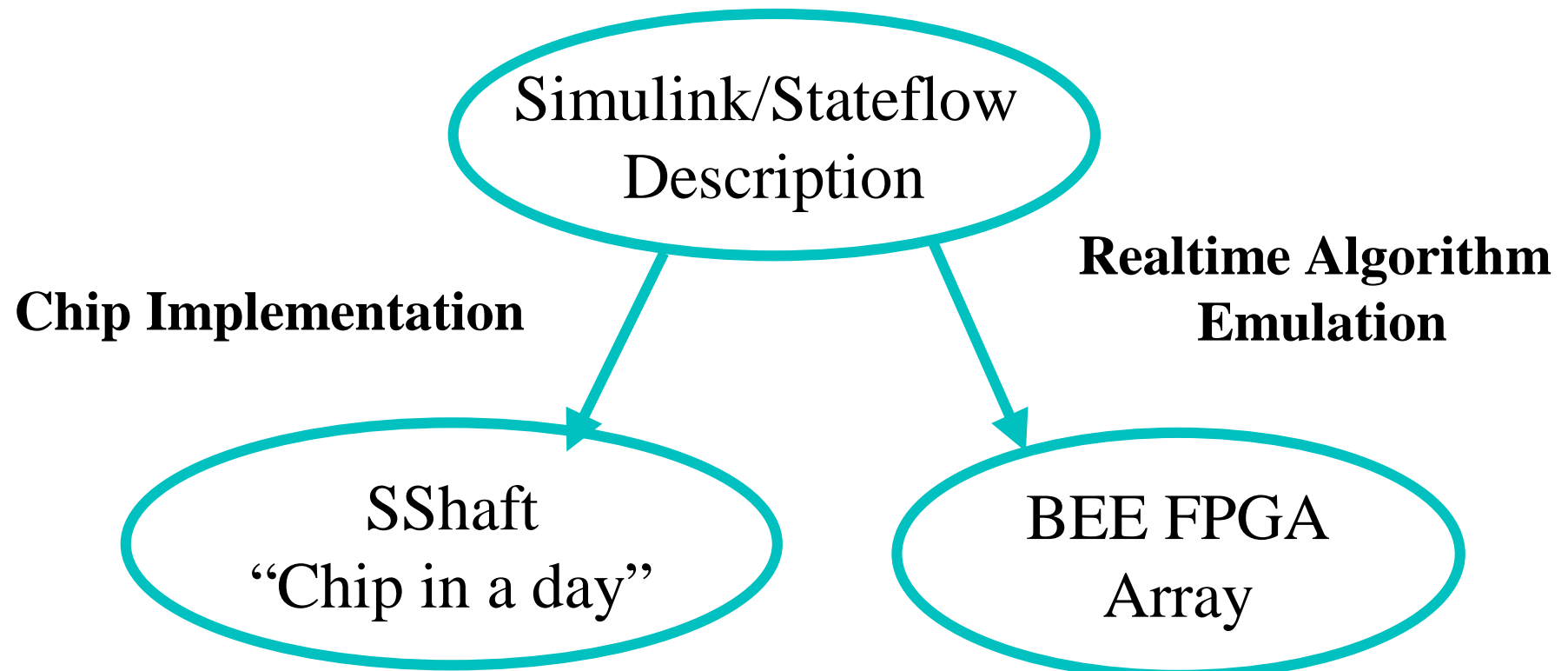
- Ambient temperature at 40 °C
- FPGA junction temperature max at 80 °C
- At 20Watt per chip, we need 2 °C/W active heatsink.

# Chassis Design: fans

---



# Outline



**Chip Implementation**

**Realtime Algorithm Emulation**

SShaft  
“Chip in a day”

BEE FPGA  
Array

**Applications**

*UltraWideband Transmission*

*Multiple Antenna Systems*

*Sensor Networks*

# Ultra Wideband Transmission

---

- Basic idea is to transmit narrow baseband pulses that have energy spread over GHz of bandwidth at spectral densities sufficiently low that it doesn't interfere with narrowband systems
- Potentially provides locationing information as well as communications
- FCC is allowing experimentation and will have a rule making in the near future

# Background

---

- Harmuth -- '60s pioneer in nonsinusoidal transmission
- Recent system demonstrations
  - » Multispectral Systems Inc. military communication systems
  - » Time domain Inc. "Impulse Radio"
  - » Aetherwire
- Good web sites
  - » Ultralab at USC //ultra.usc.edu/ulab
  - » //www.aetherwire.com

# Applications

---

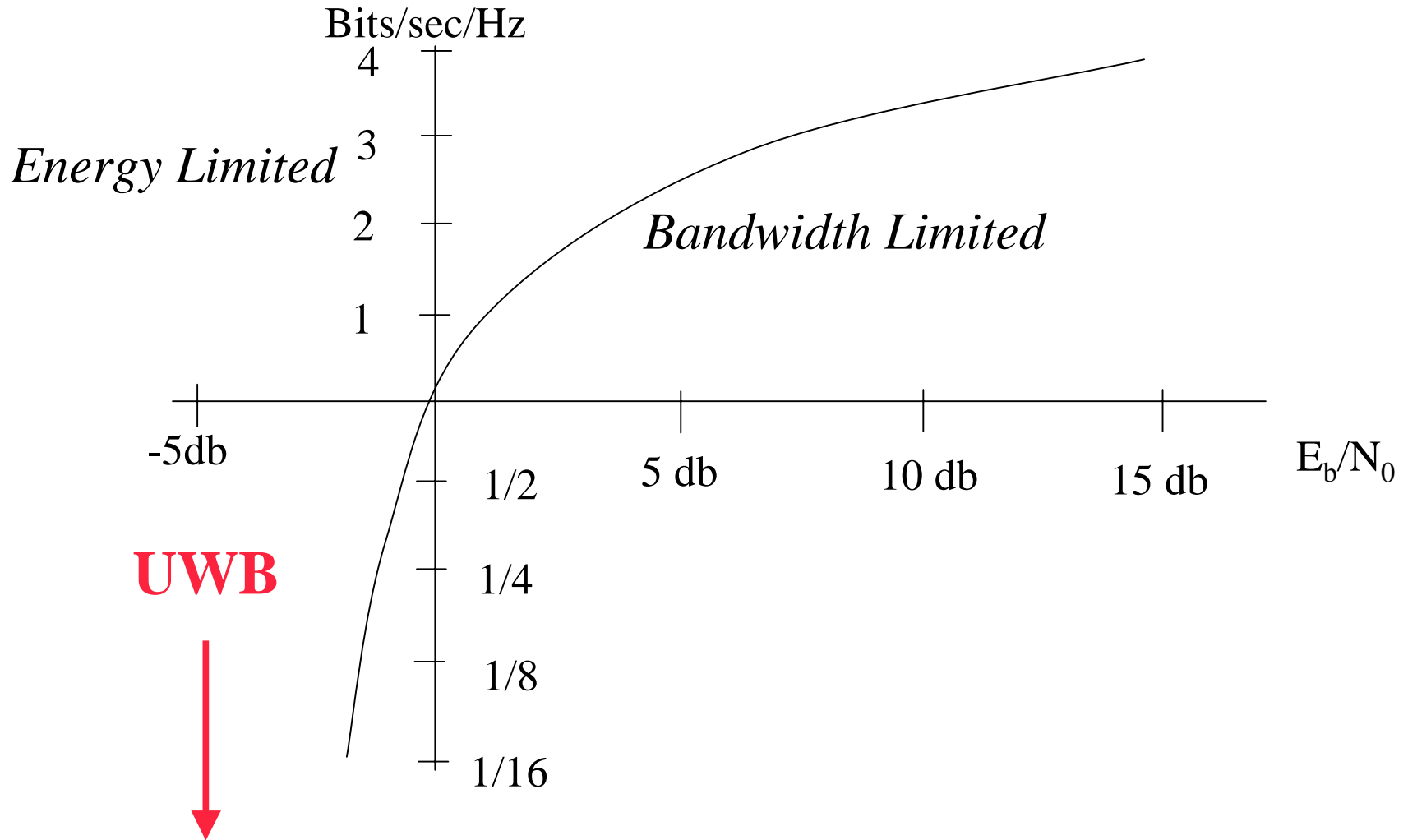
- Item tracking with location
  - » Logistics
    - Cargo containers
    - Truck
  - » Store produce – inventory and checkout
  - » Expensive or sensitive items
  - » Children
- Short range communication
  - » High rate
  - » Low data rates (sensor networks)

# Why Ultra Wide Band?

---

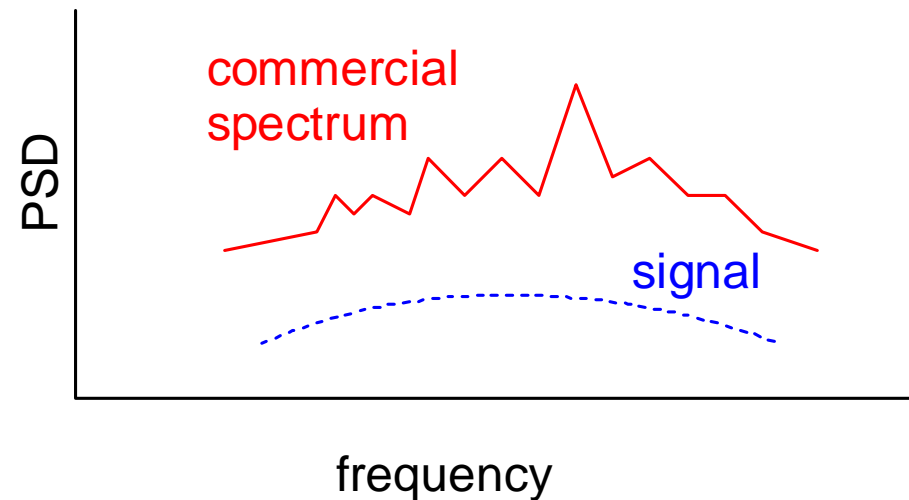
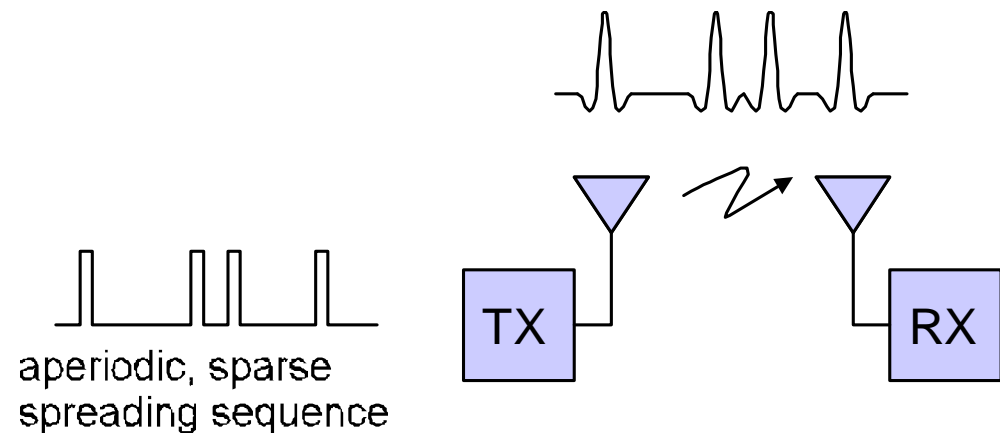
- Fine time resolution for accurate ranging and imaging
- Material penetration
- Potentially simple transmit and receive structures – lower power, less expensive
- Potential new FCC allocation

# Shannon likes UWB!



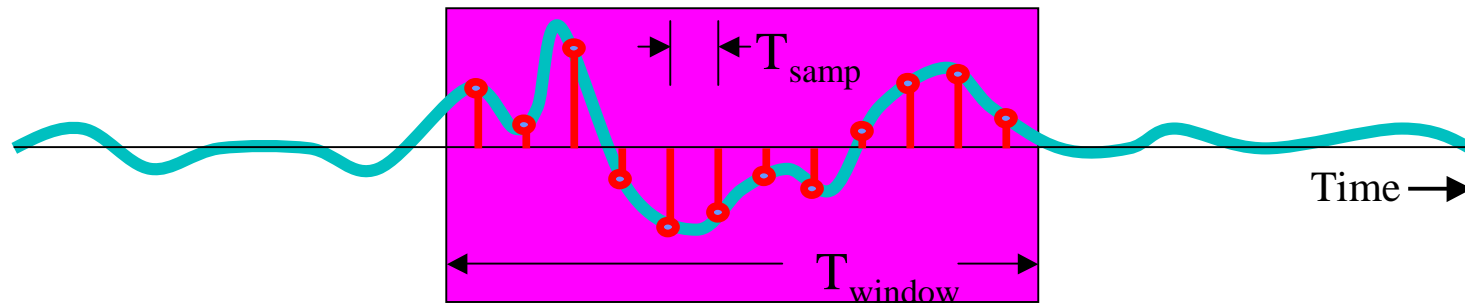
# Impulse Radio Transmission

- Sparse codes keep power low.
- Long codes give high processing gain.
- Aperiodic sequence is relatively immune to multipath.



# Analog Circuits - Pulse Reception

**Energy of Pulse is Contained in Small Time Window**



**Only Need Limited Amount of Fast Sampling**

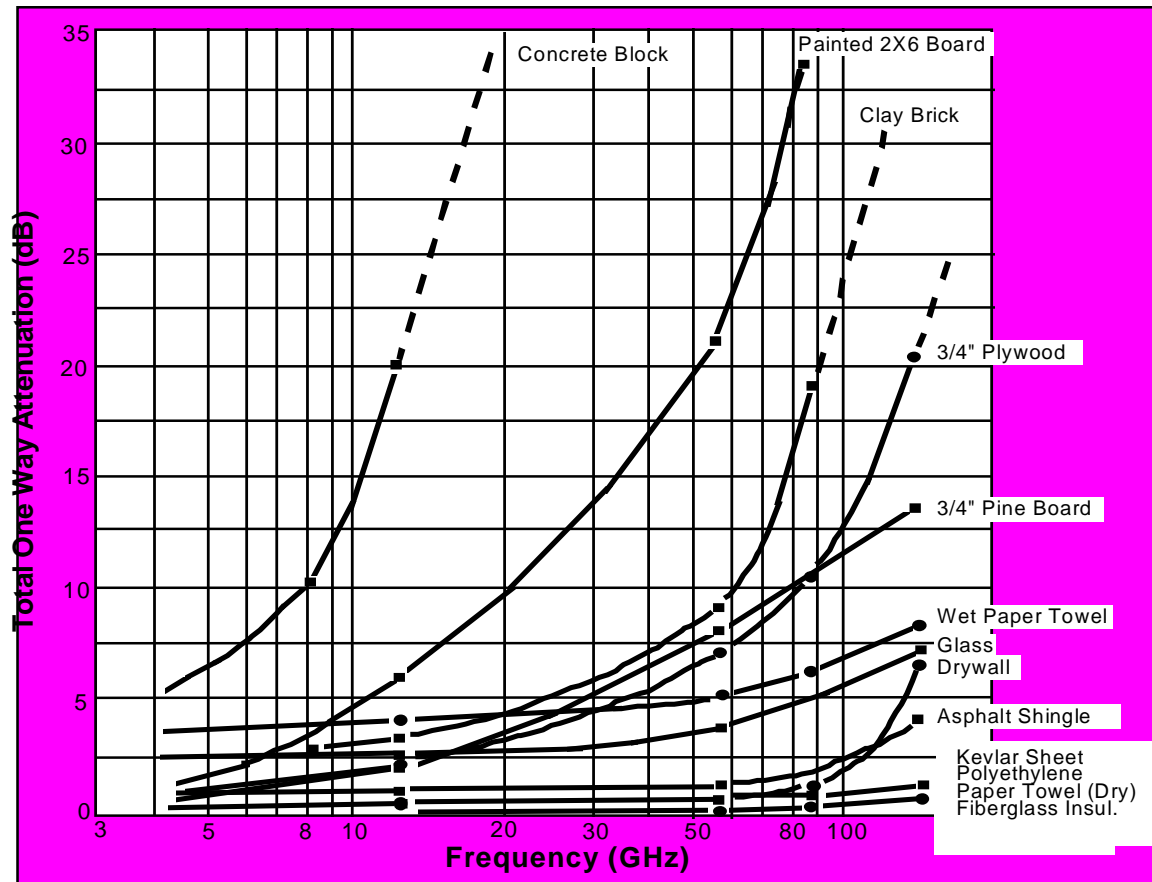
**Use Parallel Sampling Blocks**

Have Rest of Time in Cycle to Process Samples

**Do Digital Correlation**

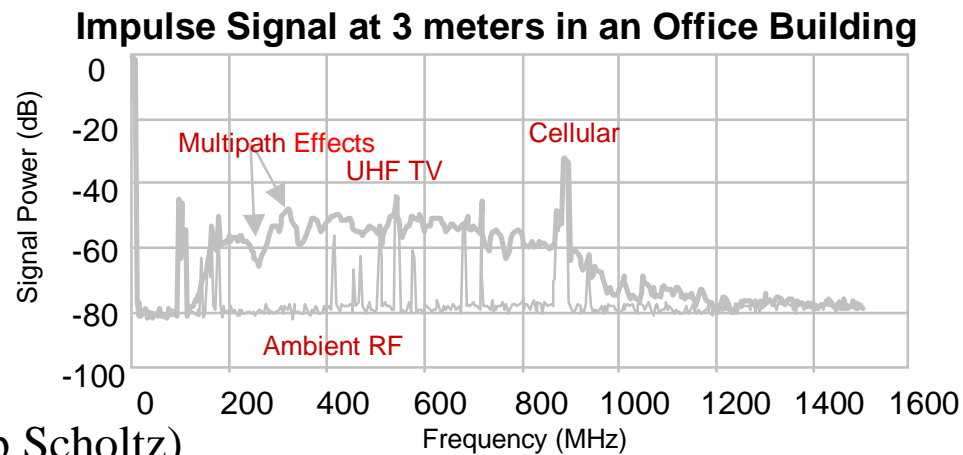
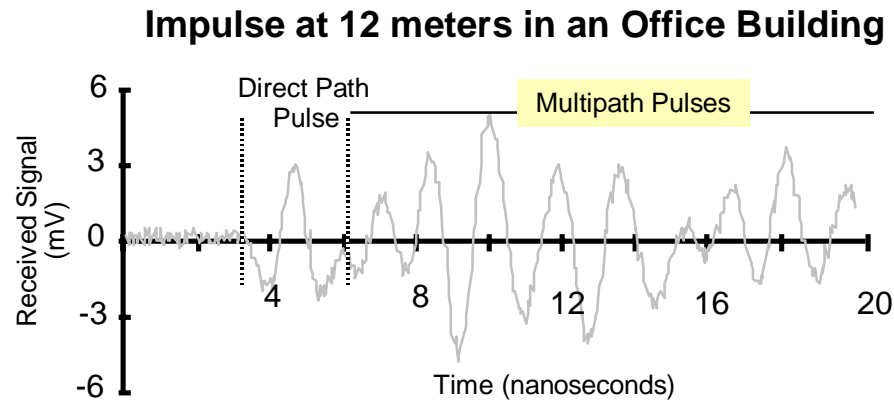
Minimum of Analog Blocks Run at Speed

# Material penetration



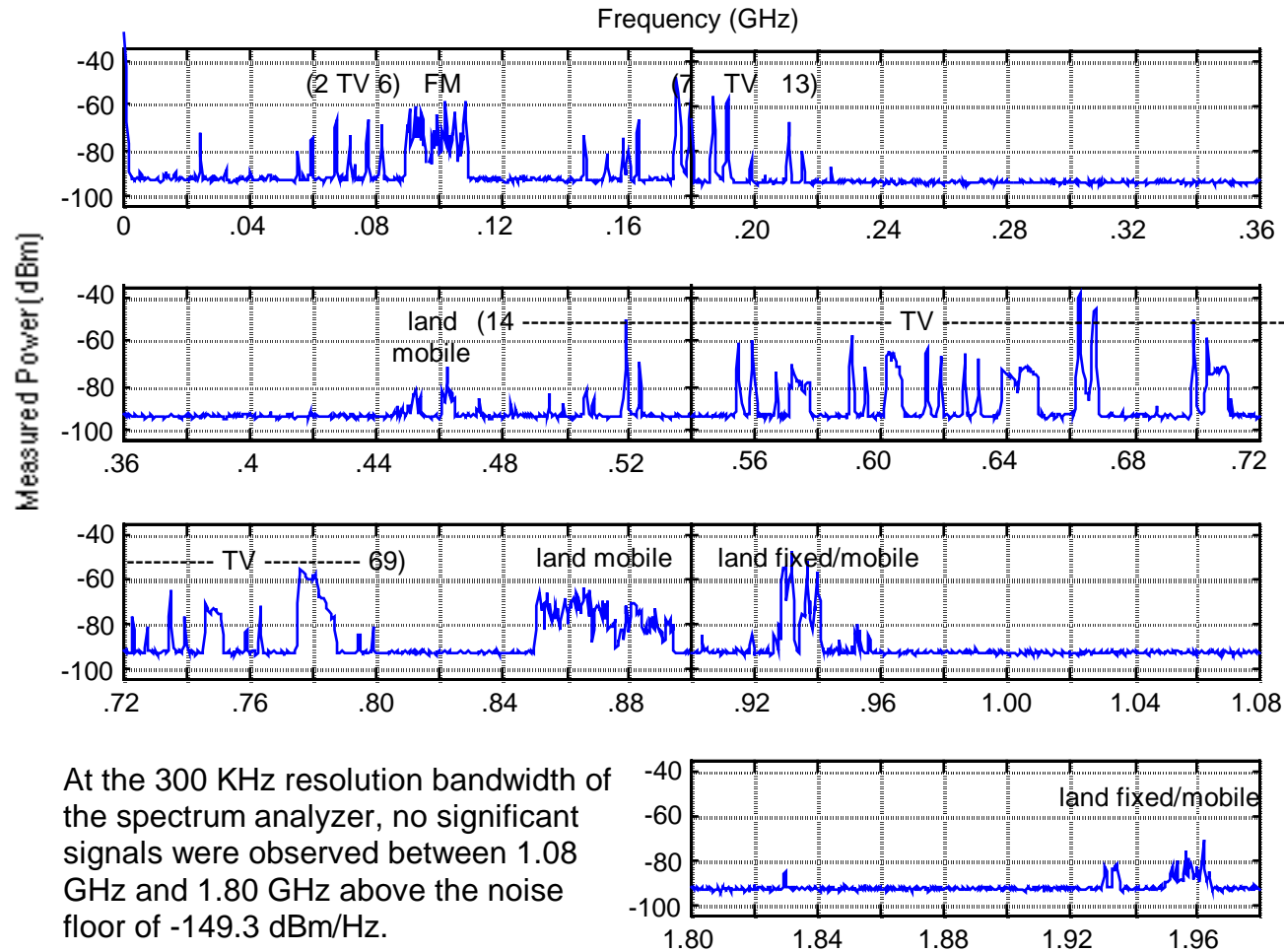
(from Bob Schlotz)

# Example time waveforms



(from Bob Scholtz)

# The interference...



(from  
Bob Scholtz)

At the 300 KHz resolution bandwidth of the spectrum analyzer, no significant signals were observed between 1.08 GHz and 1.80 GHz above the noise floor of -149.3 dBm/Hz.

# Present regulation

---

**Part 15 of Section 47 of the Code of Federal Regulations states that for signals above 960 MHz, the unintentional radiated emission limit for all but Class A devices is 500 microvolts/meter at 3 meters.**

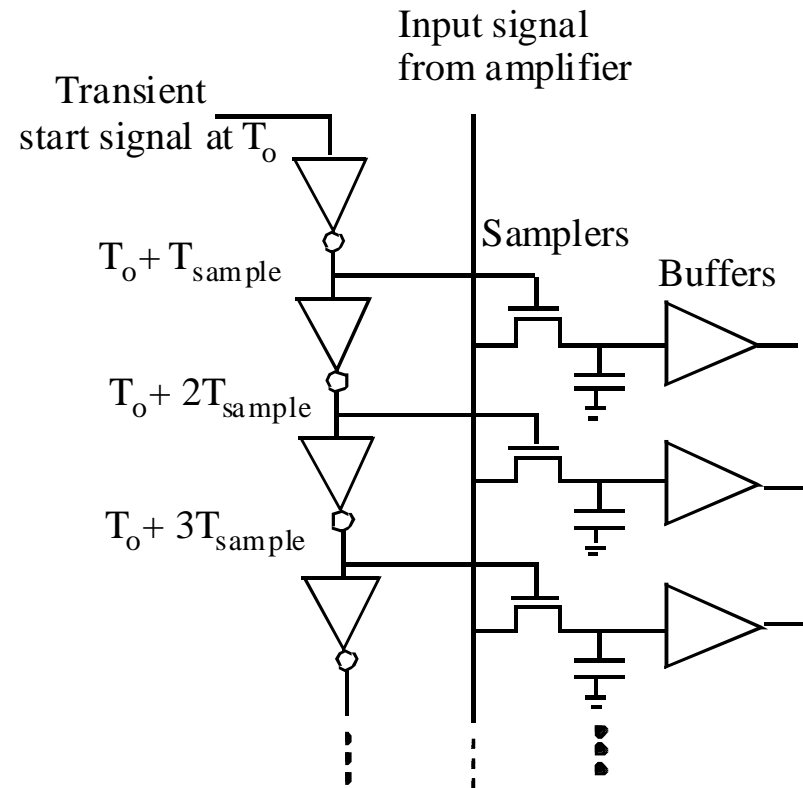
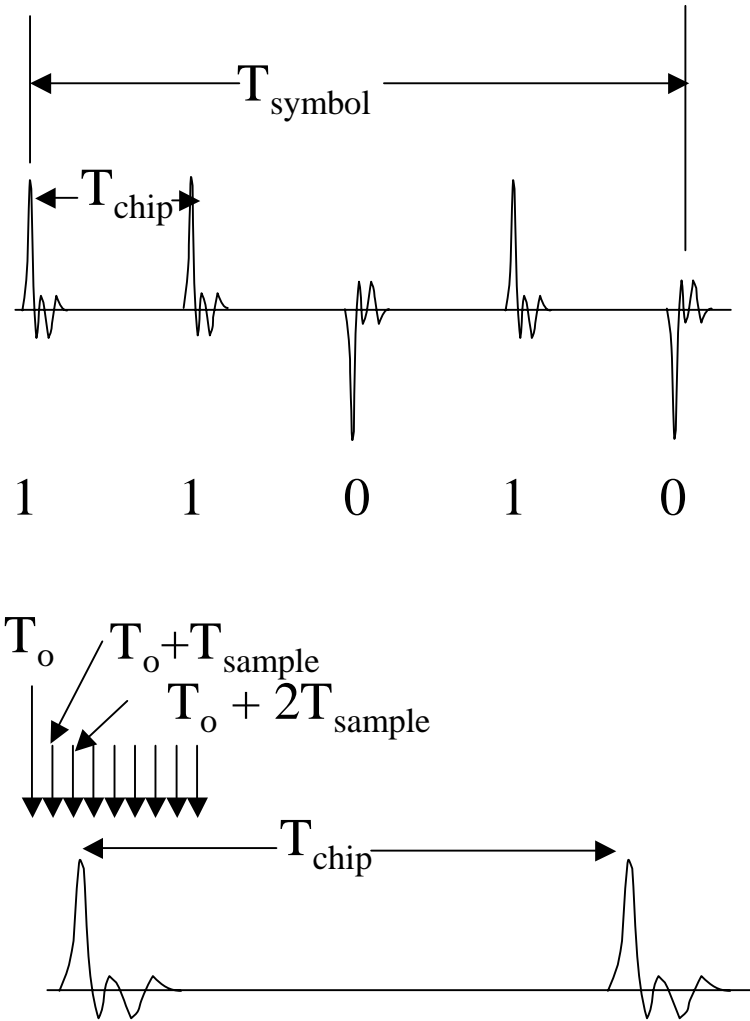
- New regulations for intentional radiation
- Need a specification that specifies the energy as a function of frequency

# The Future

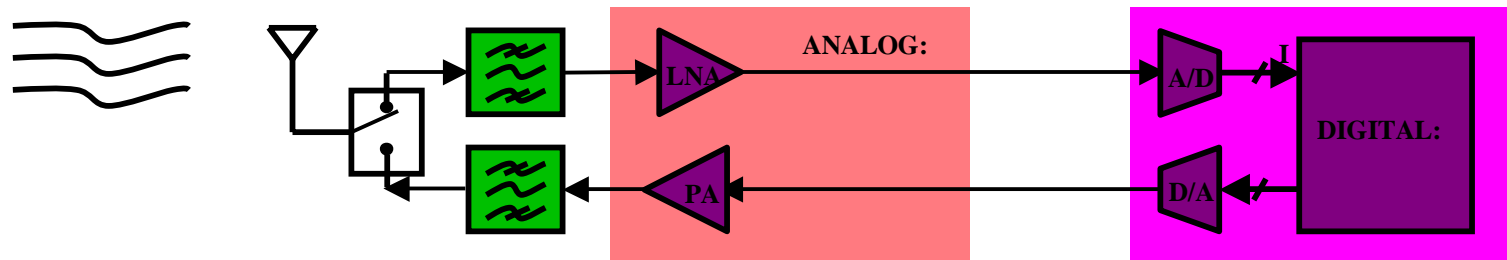
---

- FCC is actively considering various proposals for regulation – seems likely something is going to happen
- Significant increase in research activity may move the discussion from PR and hype to the actual issues
- Enormous application opportunity but tag applications are extremely cost sensitive

# UWB Time Waveforms



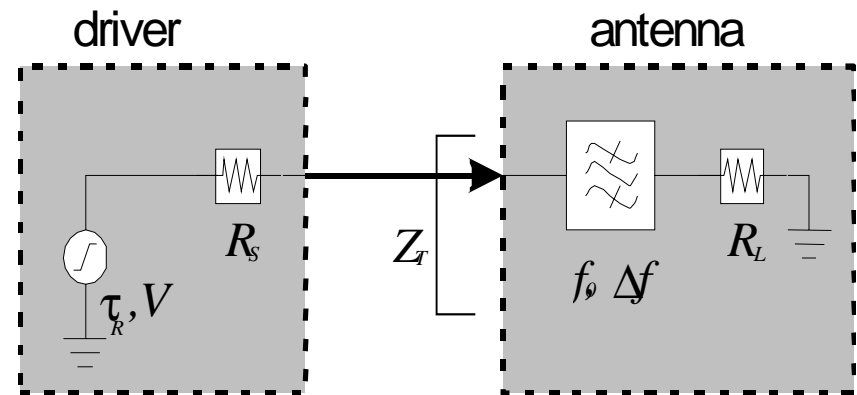
# System Modeling



- Develop models for antenna, packaging and CMOS circuitry appropriate for UWB (wideband vs. narrowband approximations)
- Simulink model of complete baseline system to drive SShaft and Bee design flows

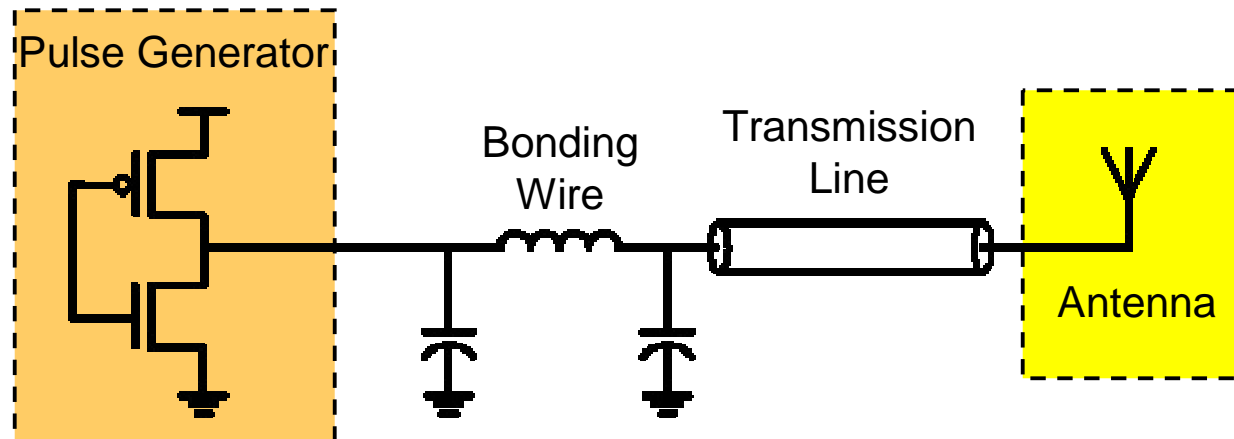
# CMOS UWB Pulser

- Use antenna/filter characteristics to shape pulse
- Use switching transient edge as broadband excitation
- Transceiver requires minimal analog circuitry.

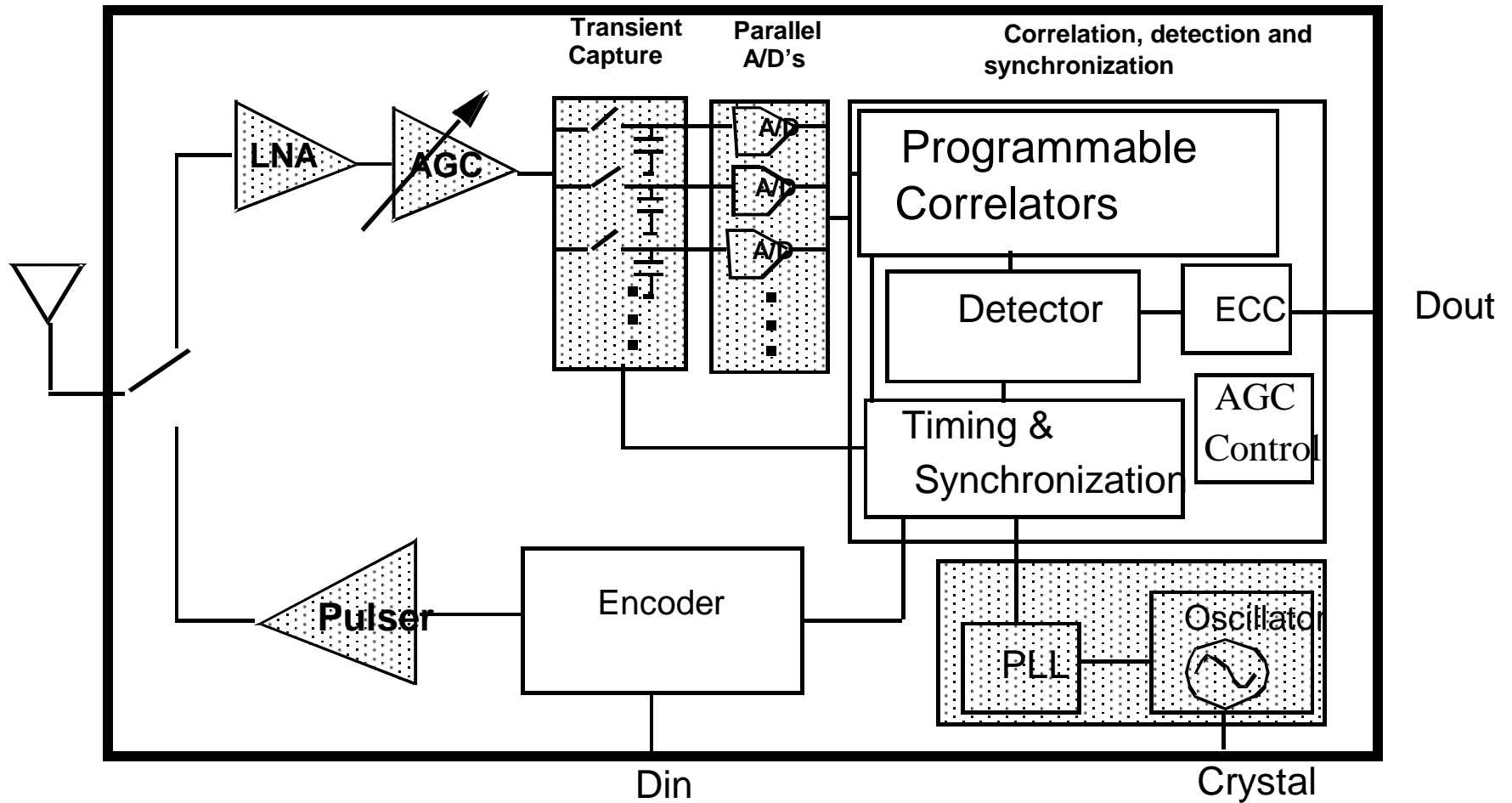


# Transmitter Modeling

- UWB antenna is not likely to be a purely resistive load and may strongly influence the transmitter circuits
  - Antenna/circuit co-design is necessary
- Efficient pulse-shape design
  - Taking pulse-shape design into account adds one more dimension to improve the performance of the antenna



# UWB Transceiver



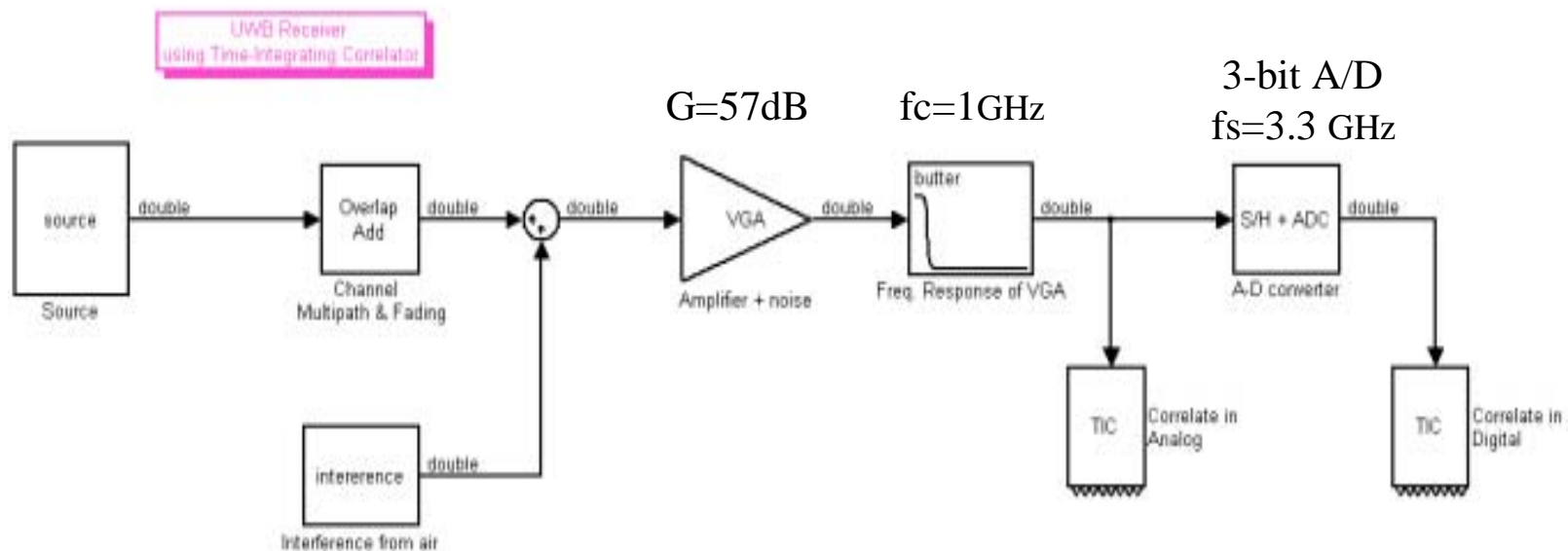
# Our Approach

---

- Transmission at below thermal noise levels
  - » use matched filter with processing gain to improve SNR
- Highly optimized analog and digital architectures
- Design and implement radio at frequencies up to 60 GHz

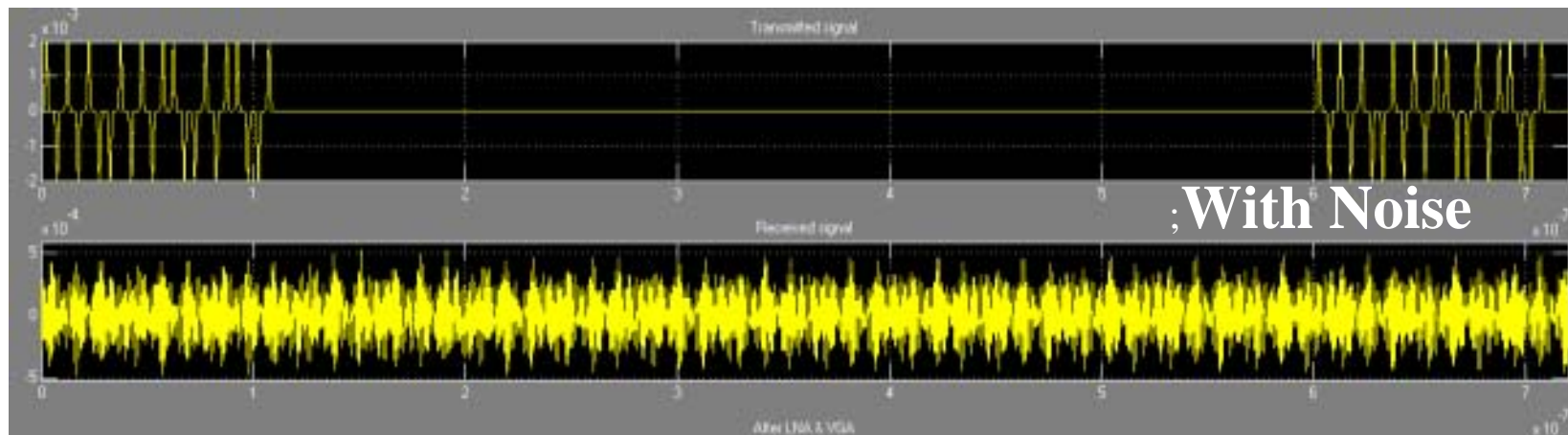
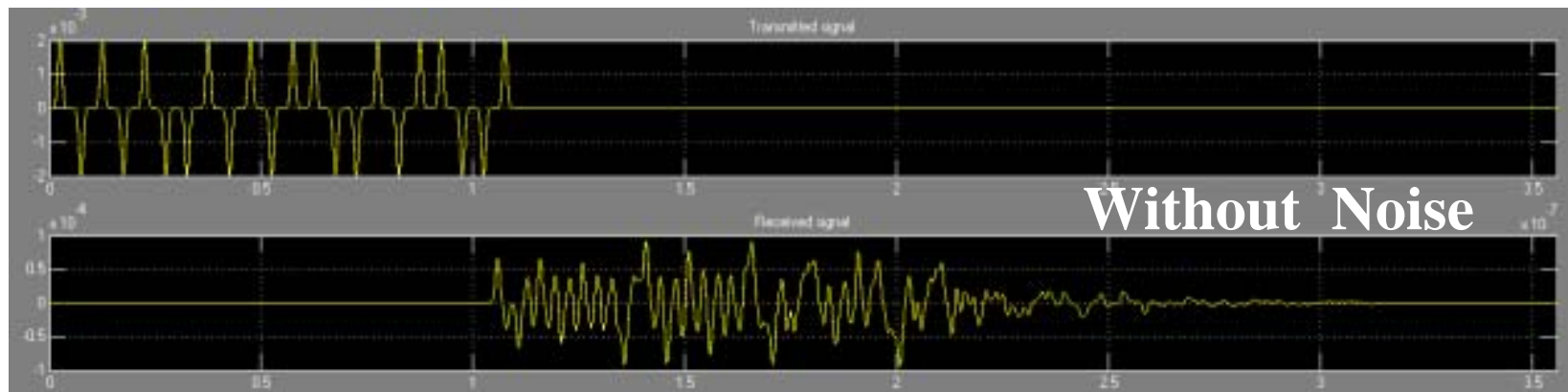
# Simulink System Simulation

- Transmit ideal Gaussian doublets modulated with a length 2200 code, whose chip time is 10ns at a 2MHz rate
- Reception simple correlator



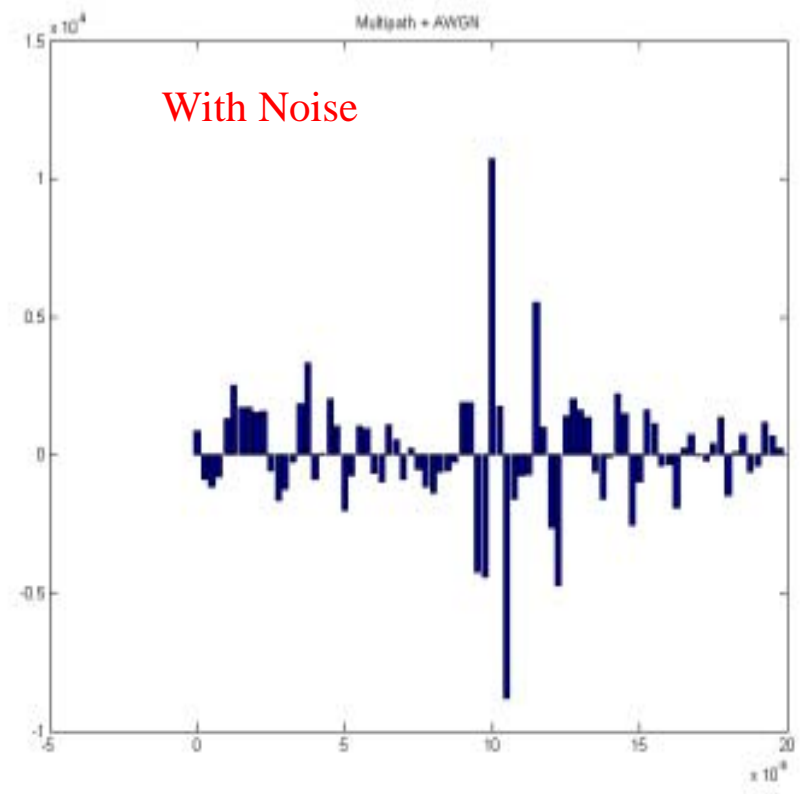
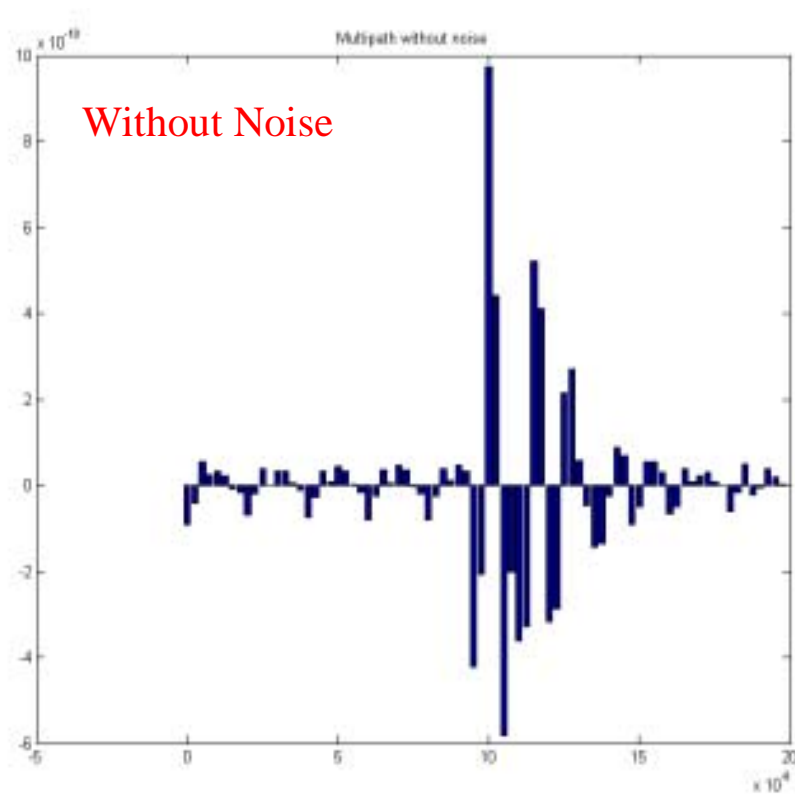
# Undetectable UWB

Transmit and receive waveforms in multipath with and without noise:



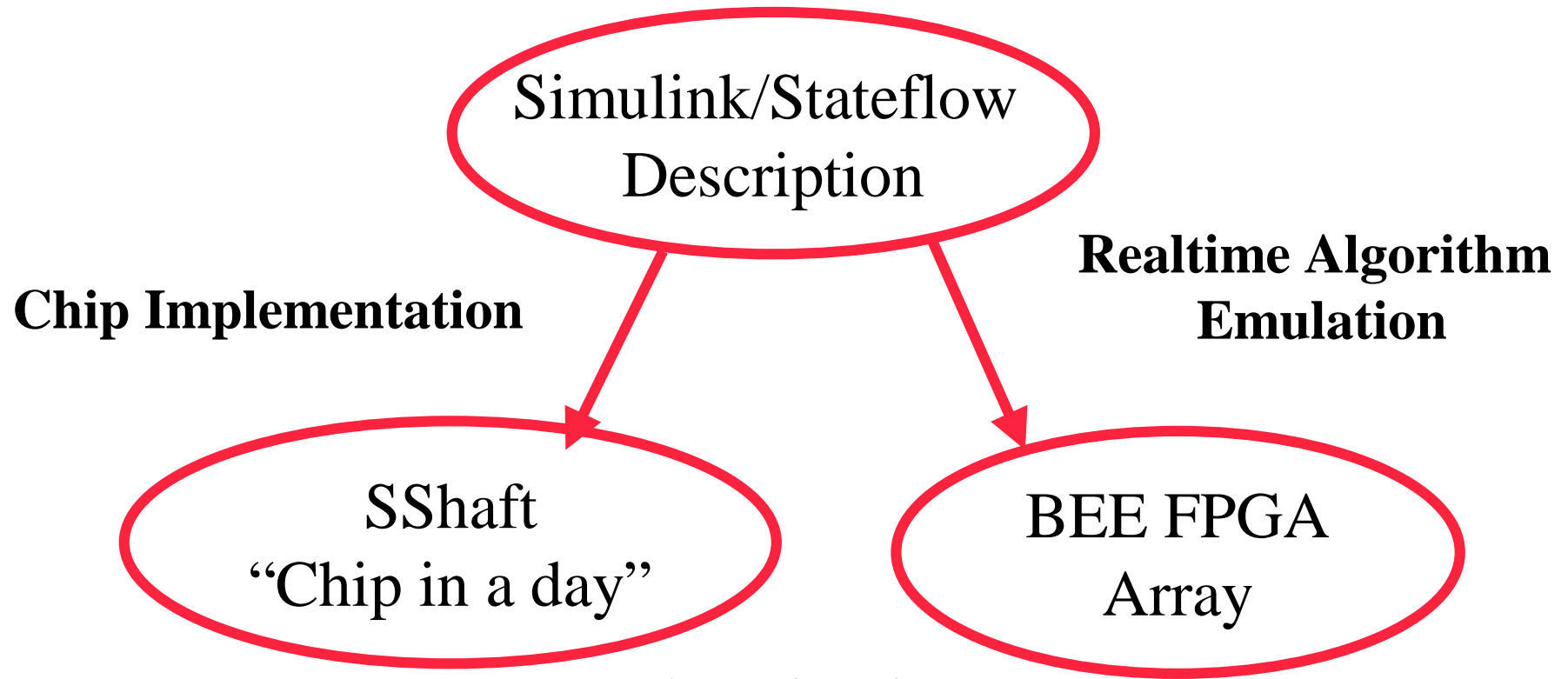
# Undetectable UWB

- **Correlation profiles in multipath with and without noise:**  
First arrival signal arrives at 103 ns



# Summary

---



## Applications

*UltraWideband Transmission*

*Multiple Antenna Systems*

*Sensor Networks*