

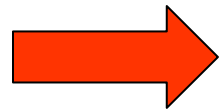
**Workshop on  
Transport Triggered Architecture  
Nov 25 & 26, Tampere, Finland**

**Presented by Steve Pekarich  
Texas Instruments  
Dallas, Texas**

**An Experiment with TTA  
&  
Requirements for TTA Tools**



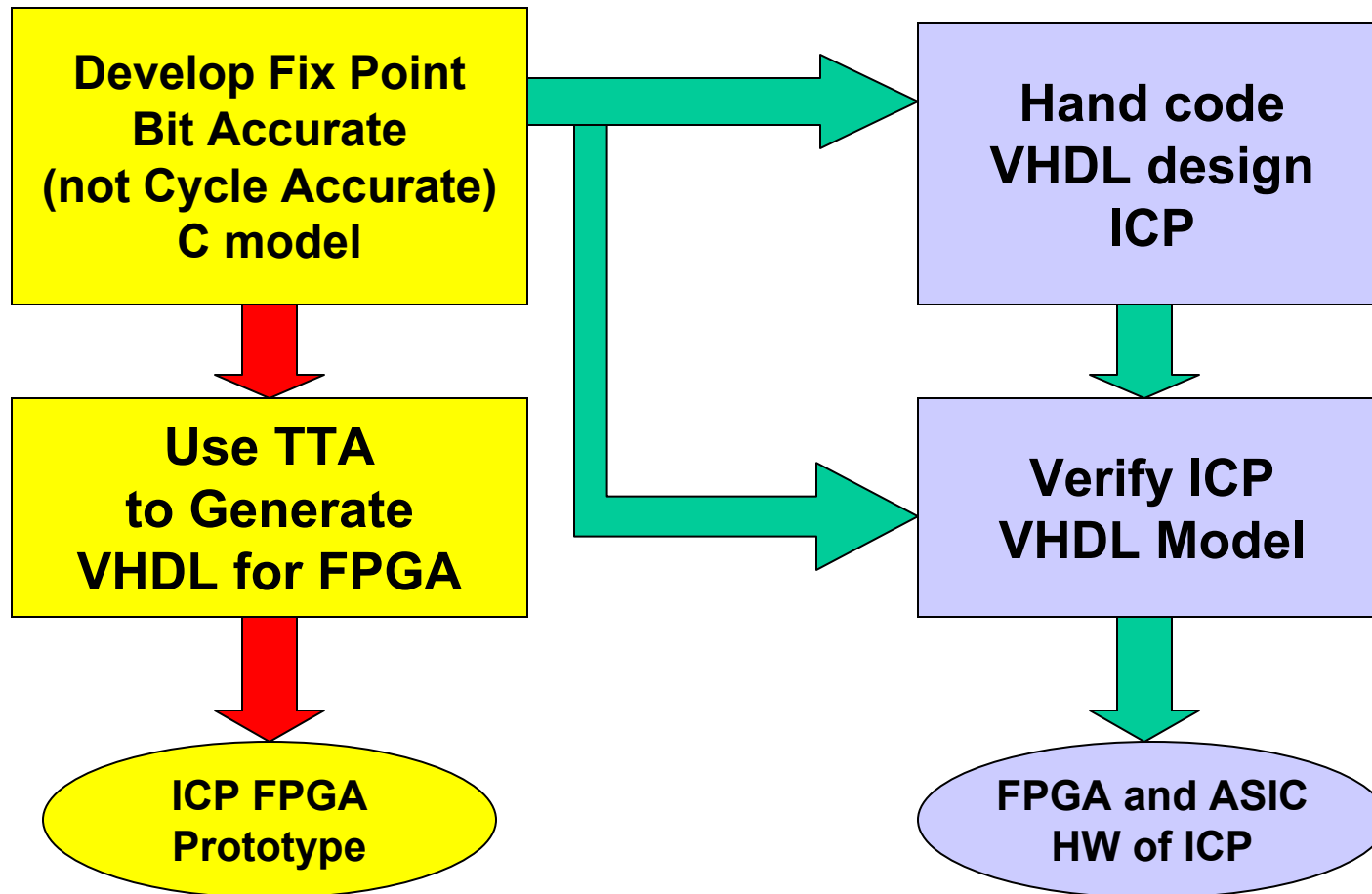
# An Experiment with TTA



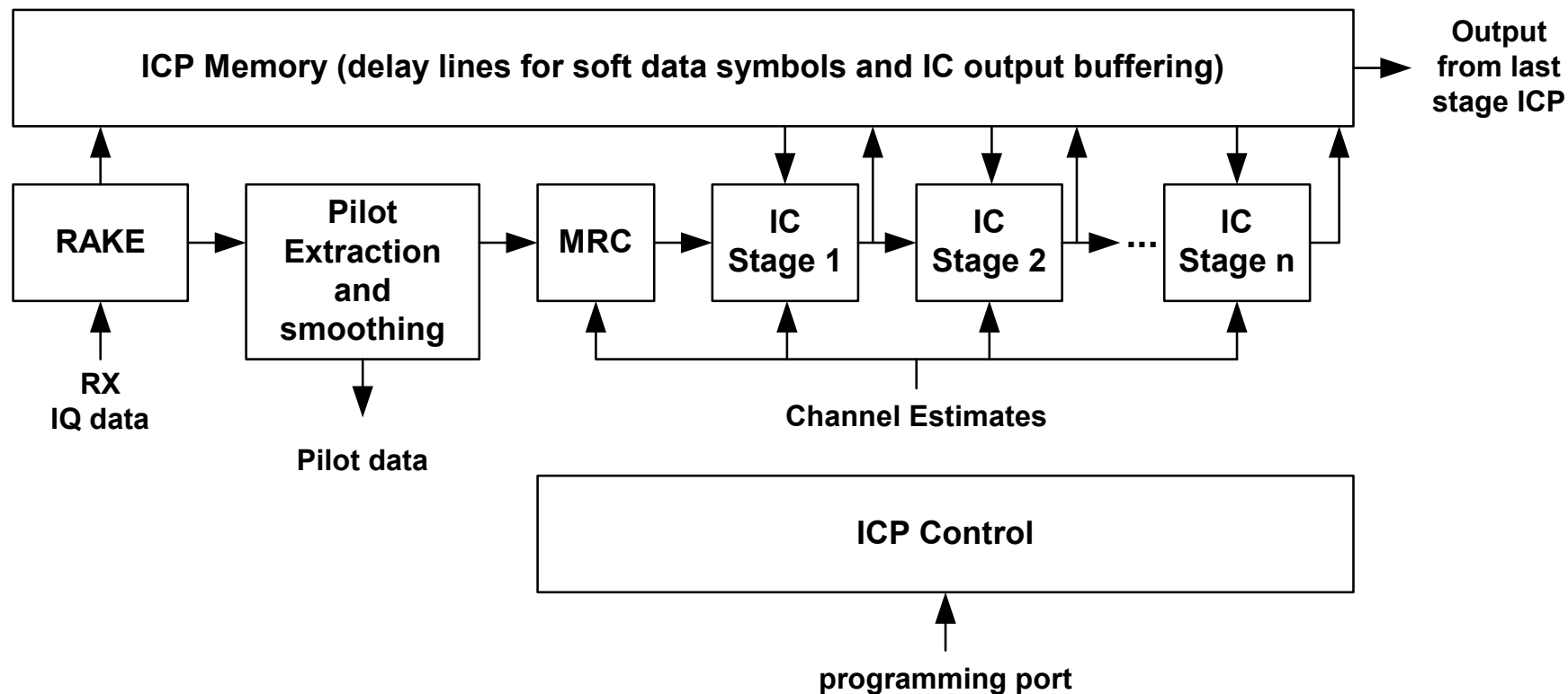
- Interference Cancellation Project
- MOVE Tool Set
- Chip Processing Engine
- Design Flow for Generating CPE



# IC Project: Design Approach



# Conceptual View of ICP Architecture



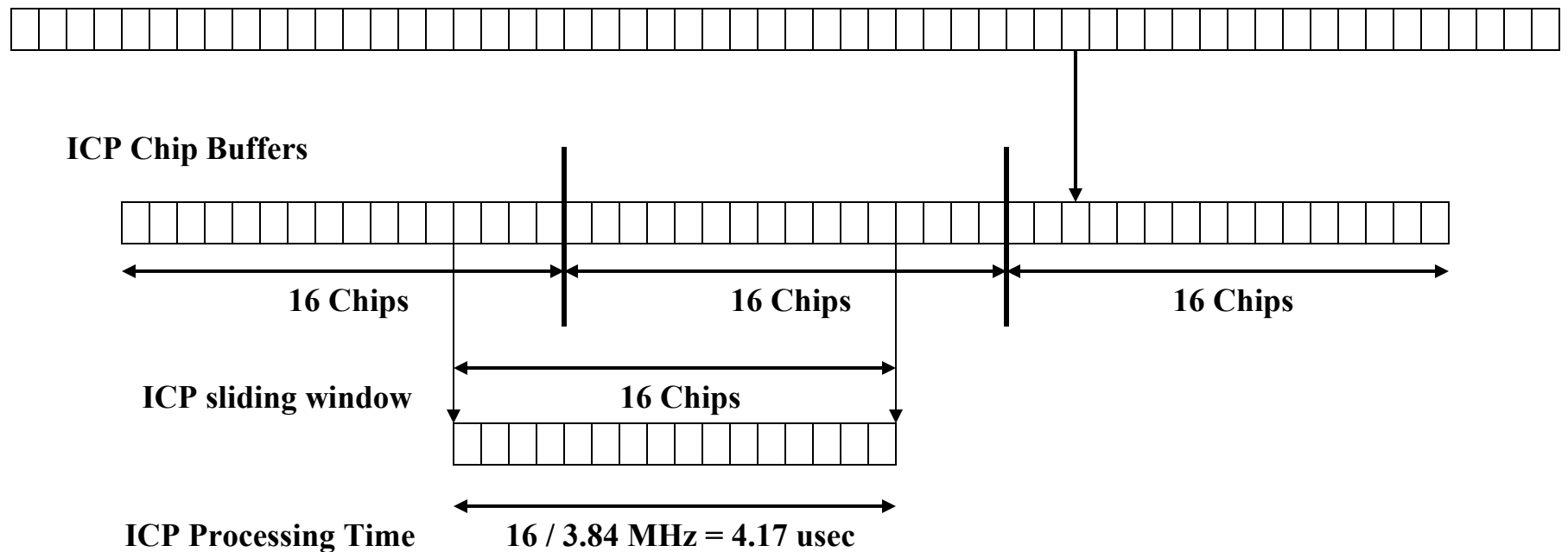
# Features of ICP

- Support Features:
  - High Speed Packet Data (1.2288MHz and 3.84 MHz) chip rates
  - Voice
  - 16-QAM and QPSK Modulation
- Programmable Options:
  - Number of Interference Cancellation Stages
  - Number of fingers per IC Stage
  - Sampling Rate
  - Path delay time per finger

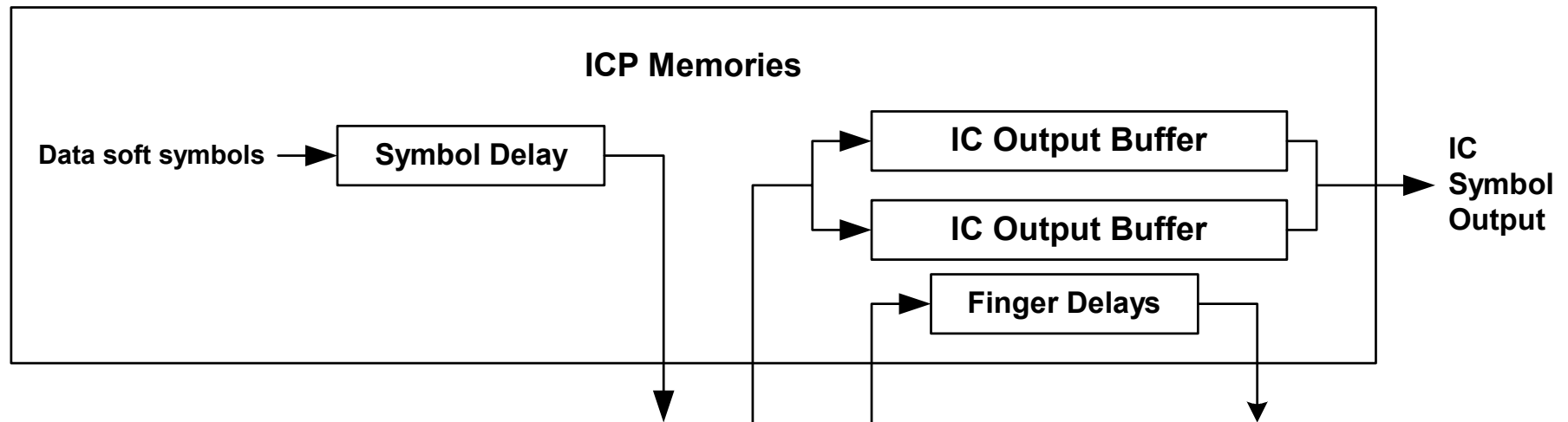


# ICP Chip Processing with Sliding Window

In coming chip stream, at 3.84 MHz rate



# Internal ICP Memory Subsystem



## Memory Requirements

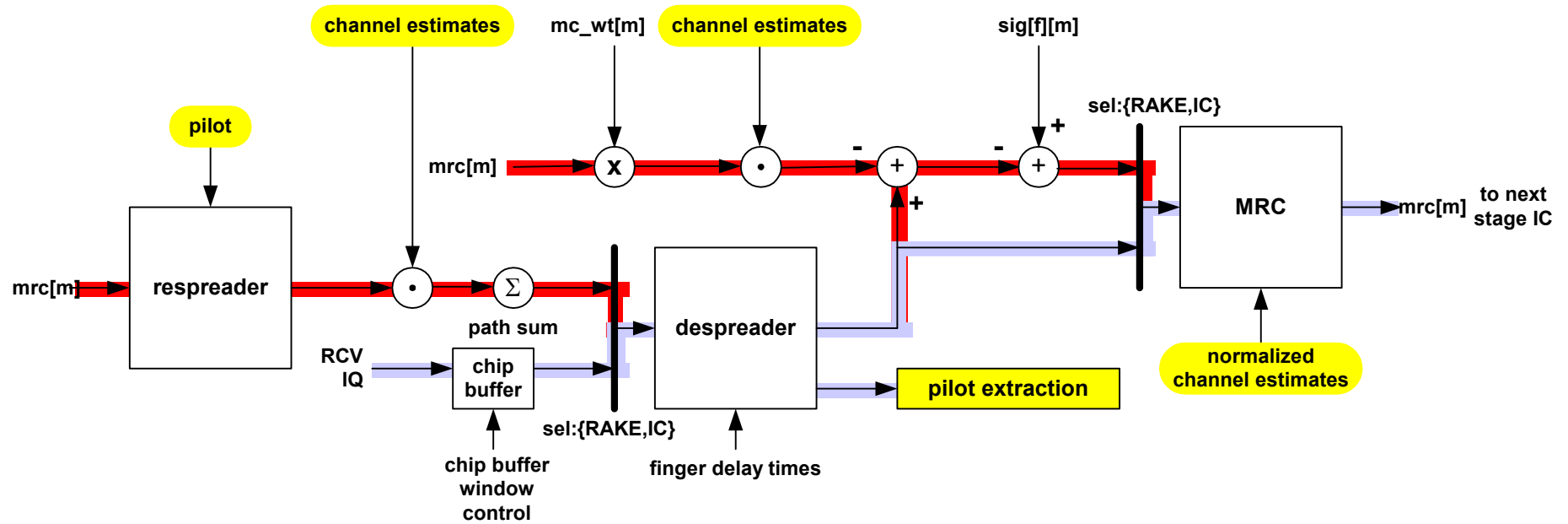
Symbol Delay:

RAKE Finger Delays:

IC Finger Delays:

IC Output Buffers:

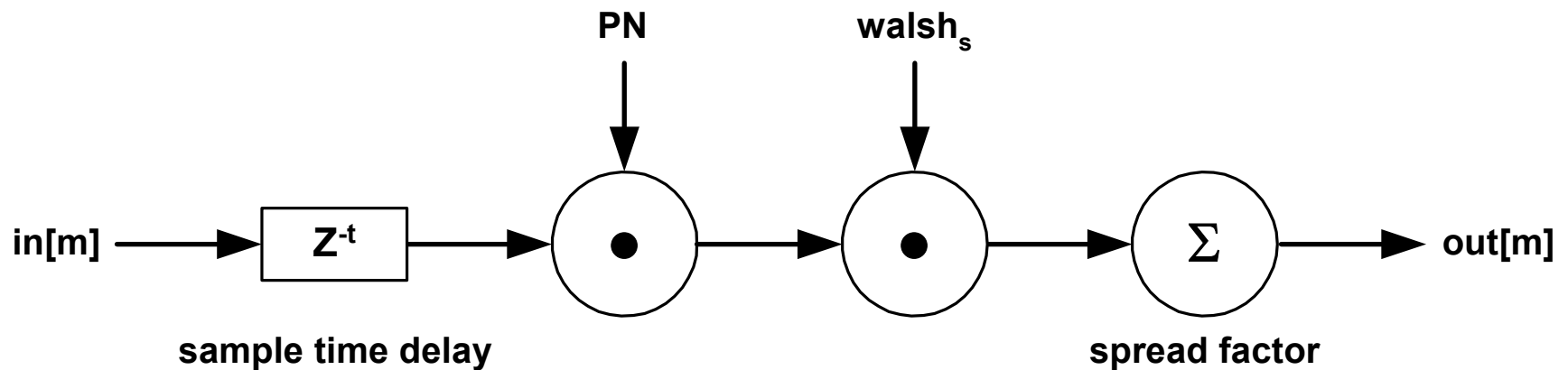
# IC Architecture



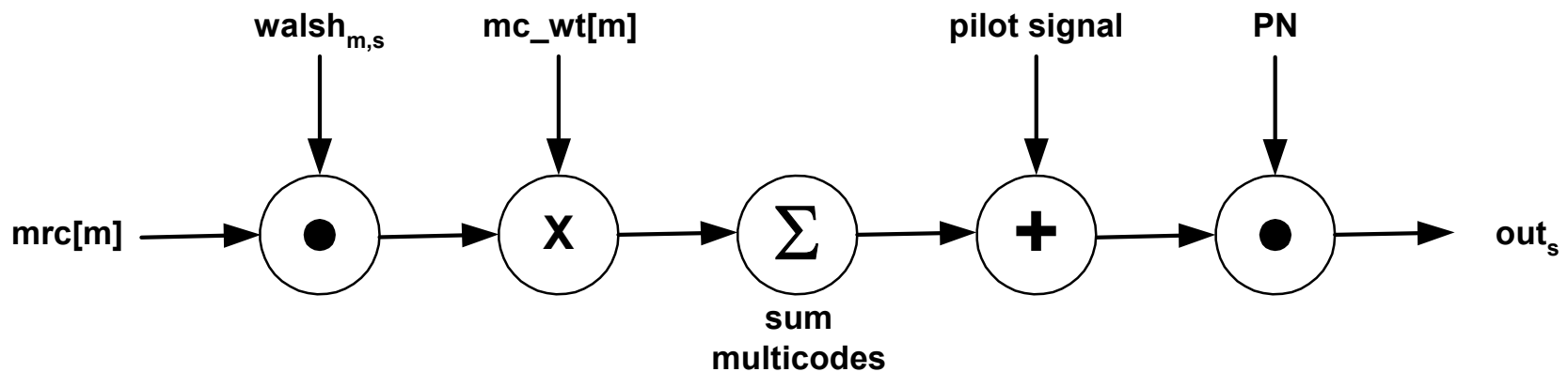
Stage 0: RAKE Data Path

Stages 1 thru n: IC Data Path

# De-spreader Data Path



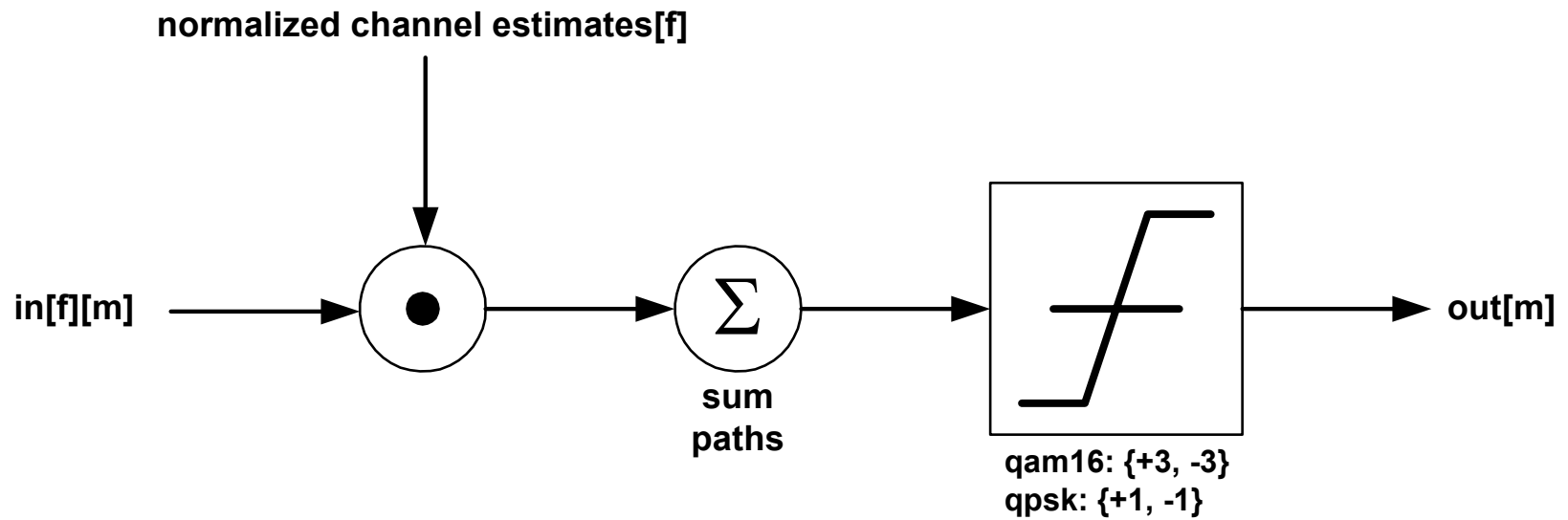
# Re-spreader Data Path



SPREAD\_FACTOR:  $s = 1$  to 16

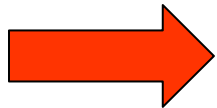


# MRC Data Path



# An Experiment with TTA

- Interference Cancellation Project
- MOVE Tool Set
- Chip Processing Engine
- Design Flow for Generating CPE

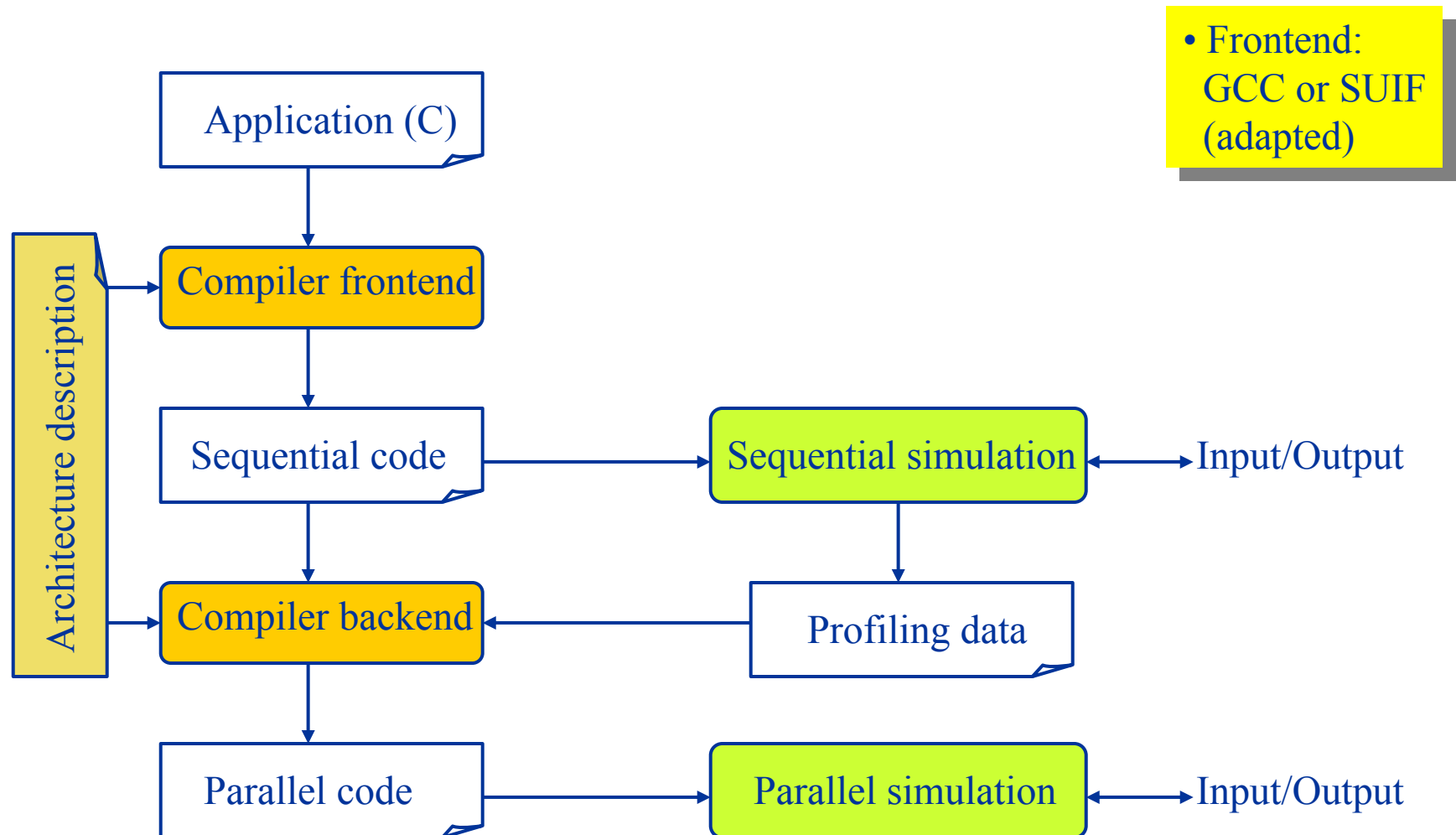


# MOVE Tool Set

- Front-end Compiler
  - Compiles ANSI C description of application into sequential TTA instructions.
- Scheduler (Back-end Compiler)
  - Reschedules TTA instructions to take advantage of additional HW.
- Explorer
  - Uses Simulator & Scheduler to build and simulate different solutions.
  - Builds a “pareto” curve for the different solutions.
- Hardware Generator
  - Generates VHDL description of TTA HW

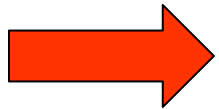


# Code generation trajectory

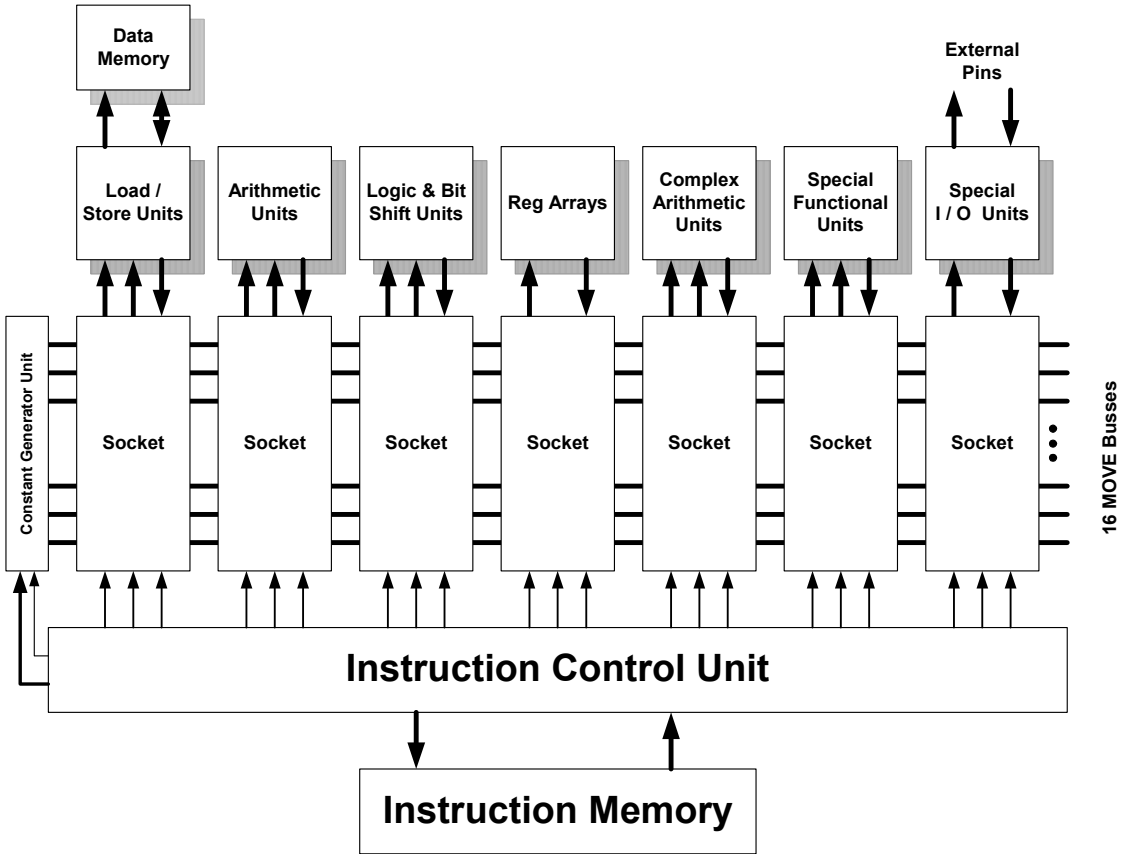


# An Experiment with TTA

- Interference Cancellation Project
- MOVE Tool Set
- Chip Processing Engine
- Design Flow for Generating CPE



# CPE Architecture



# CPE Standard Functional Units

- (16) 32-bit Add / Subtract Units
- (2) 16x16 Multiplier Units
- Arithmetic / Logical Shift Unit
- ALU
- 32x32 bit Register Array Unit
- (16) Load / Store Units (16-bit wide data )



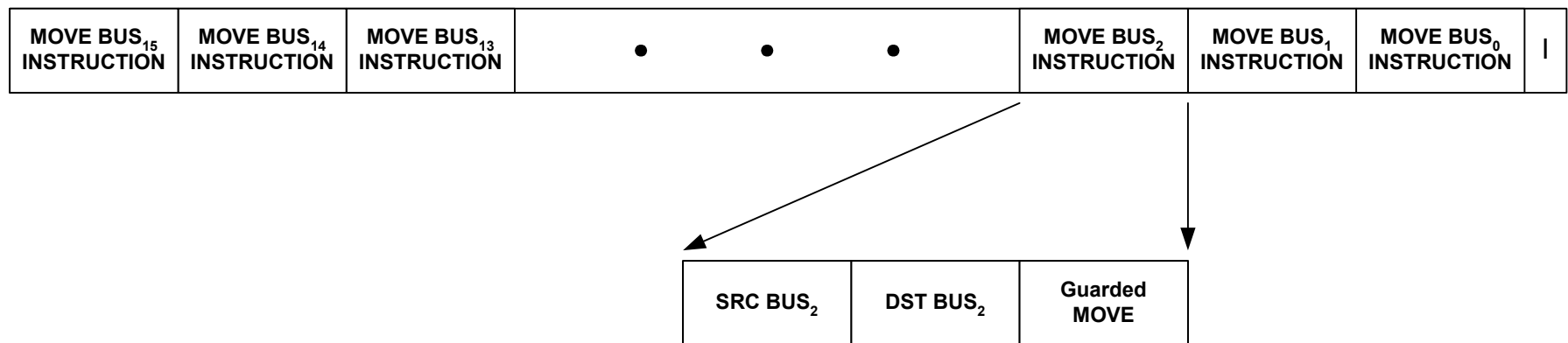
# CPE Special Functional Units

- (16) 16-bit Complex Add / Subtract Units
- (2) 16x16 Complex Multiplier Units
- (4) 16x16 Complex Scaling Units
- PN Generator
- Walsh Code Generator
- 16-bit Clipper / Saturation Unit
- Programming Port Register Unit
- (4) I/O Units (8/16 bit I & Q input data), Symbol Outputs



# CPE Instruction Word

- Uncompressed micro-instruction (241 bits wide)



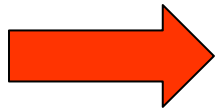
# CPE Remaining work

- More optimization
- Compression Algorithms
- Verify generated VHDL
- Build FPGA for BWA demo board
- Documentation
- Explore other Chip Processing Algorithms
  - Normalize Least Mean Squared Equalization
  - Griffith's Algorithm
  - Cascaded NLMS Algorithm



# An Experiment with TTA

- Interference Cancellation Project
- MOVE Tool Set
- Chip Processing Engine
- Design Flow for Generating CPE

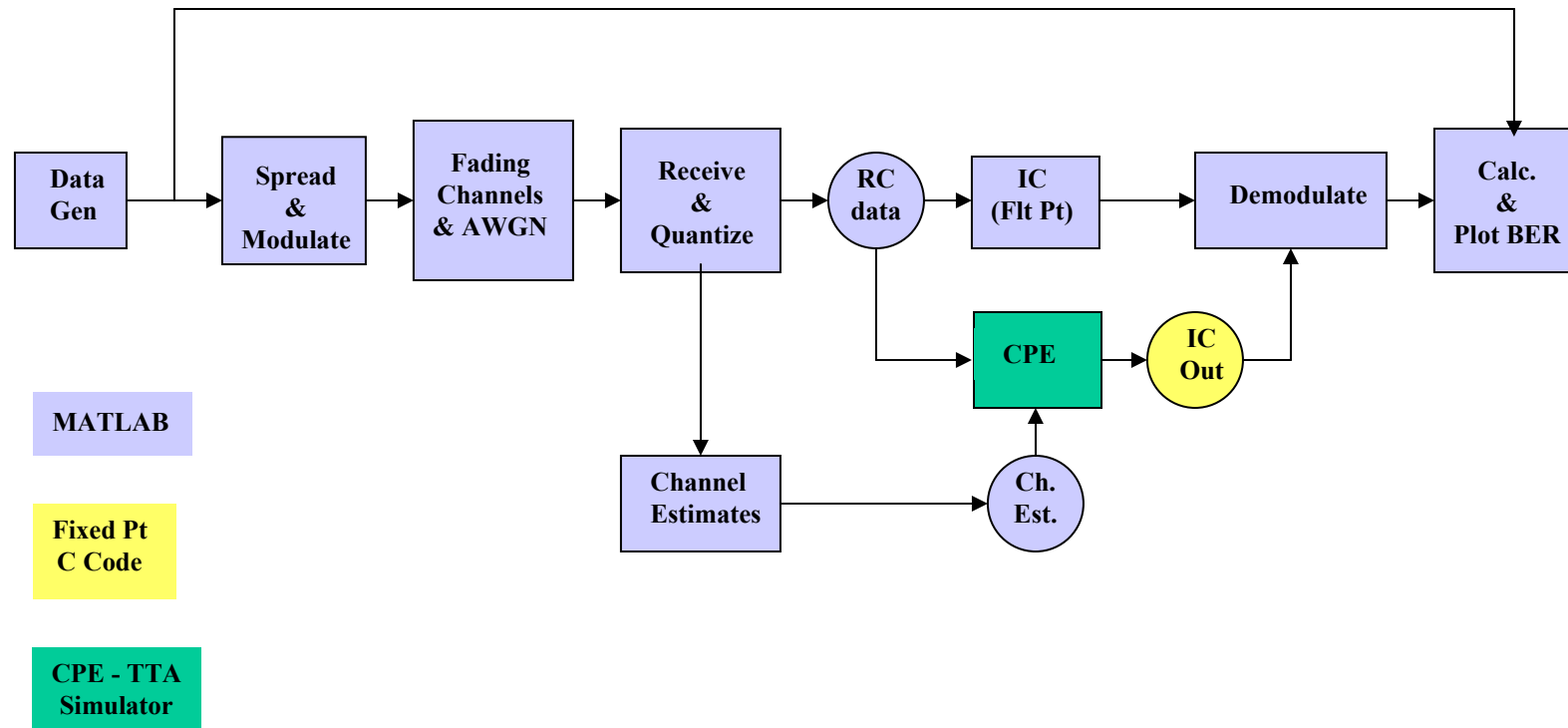


# Design Flow for Generating CPE

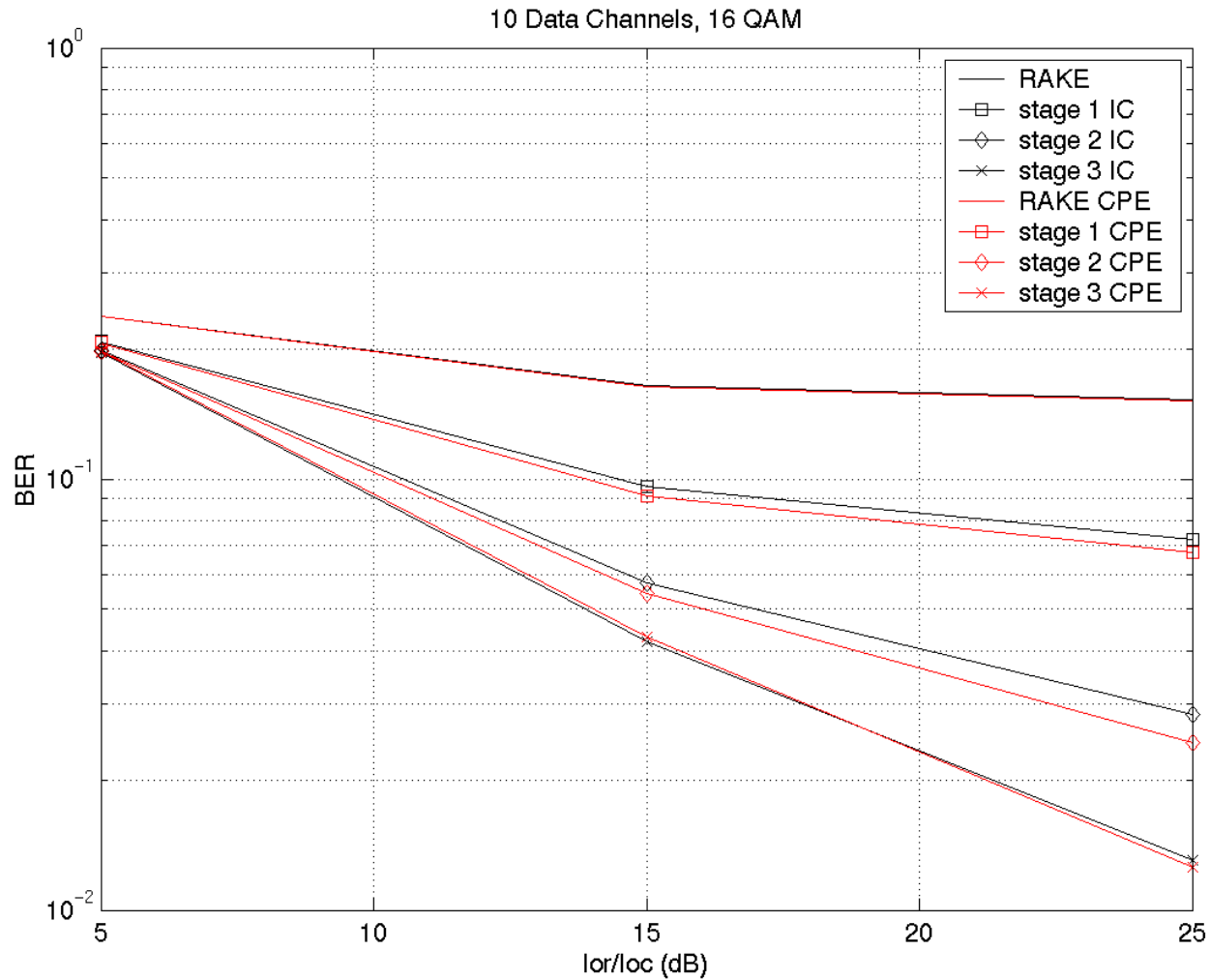
- Fixed Point C Model, TTA Simulation, and MATLAB Testbench
- Compile TTA Code, Simulate and Profile IC Algorithm using Explorer
- Generate HDL for final machine
- Breakdown of Development Effort



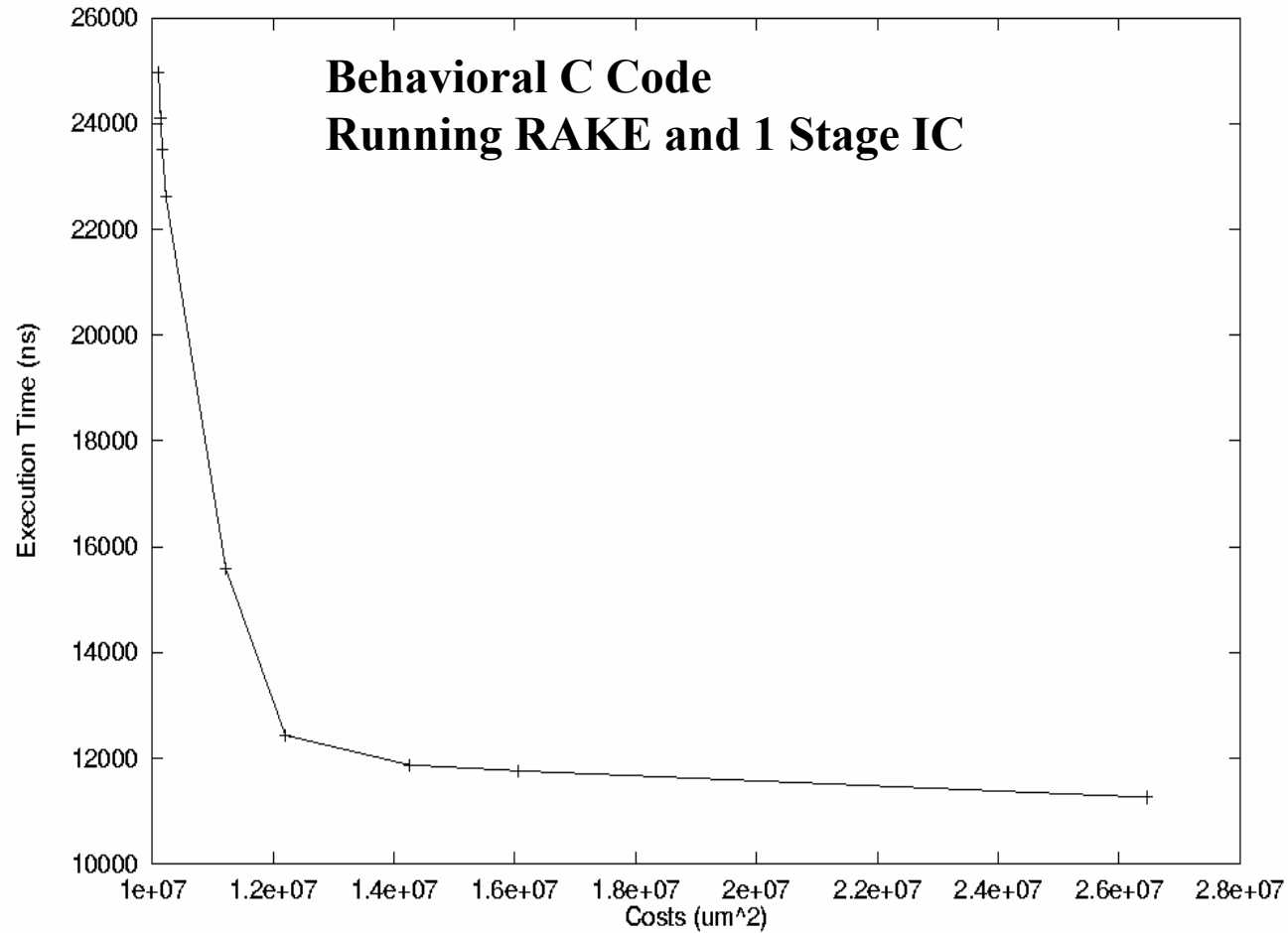
# C Model and MATLAB Test Bench



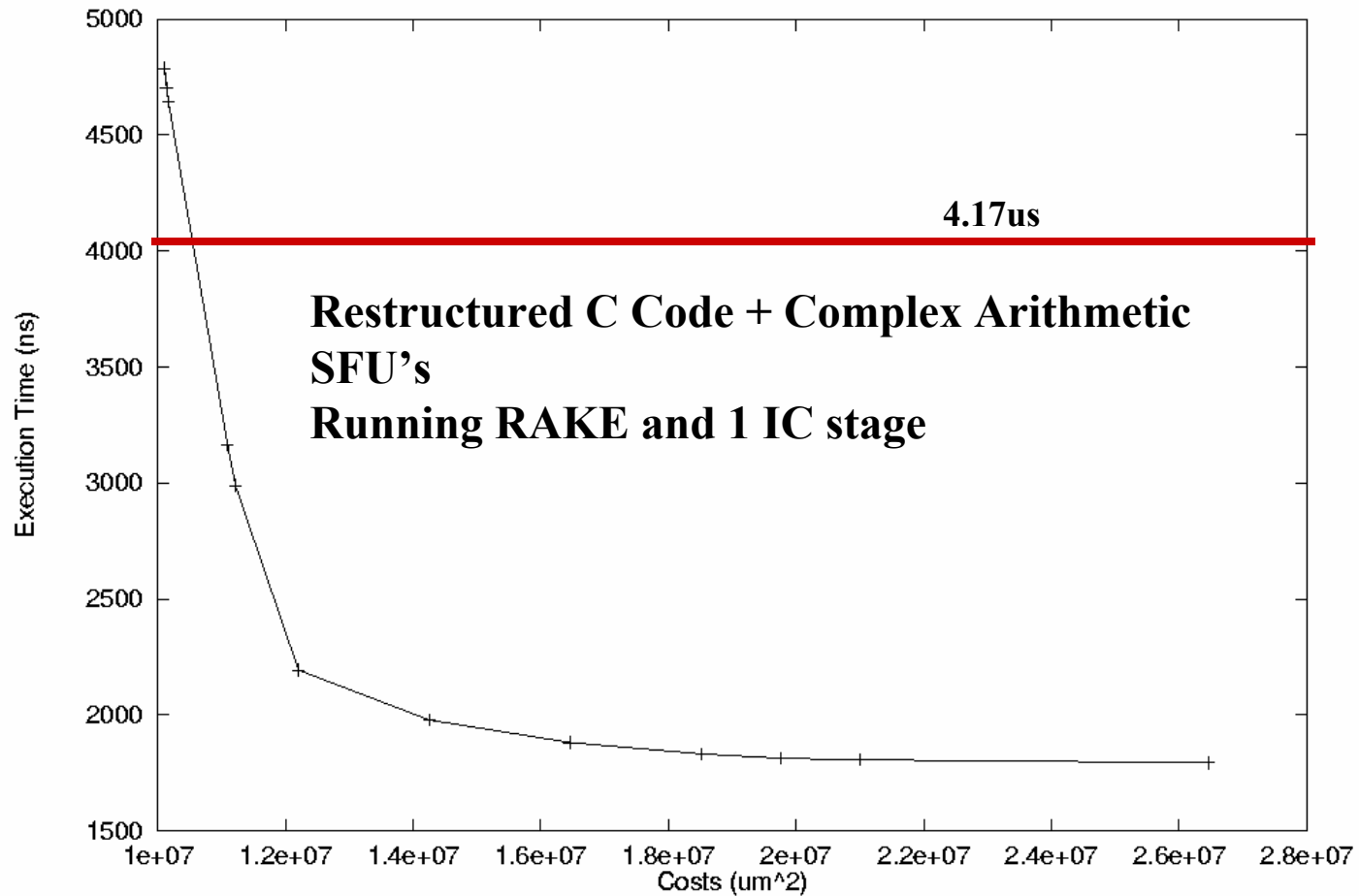
# TTA Simulation Run



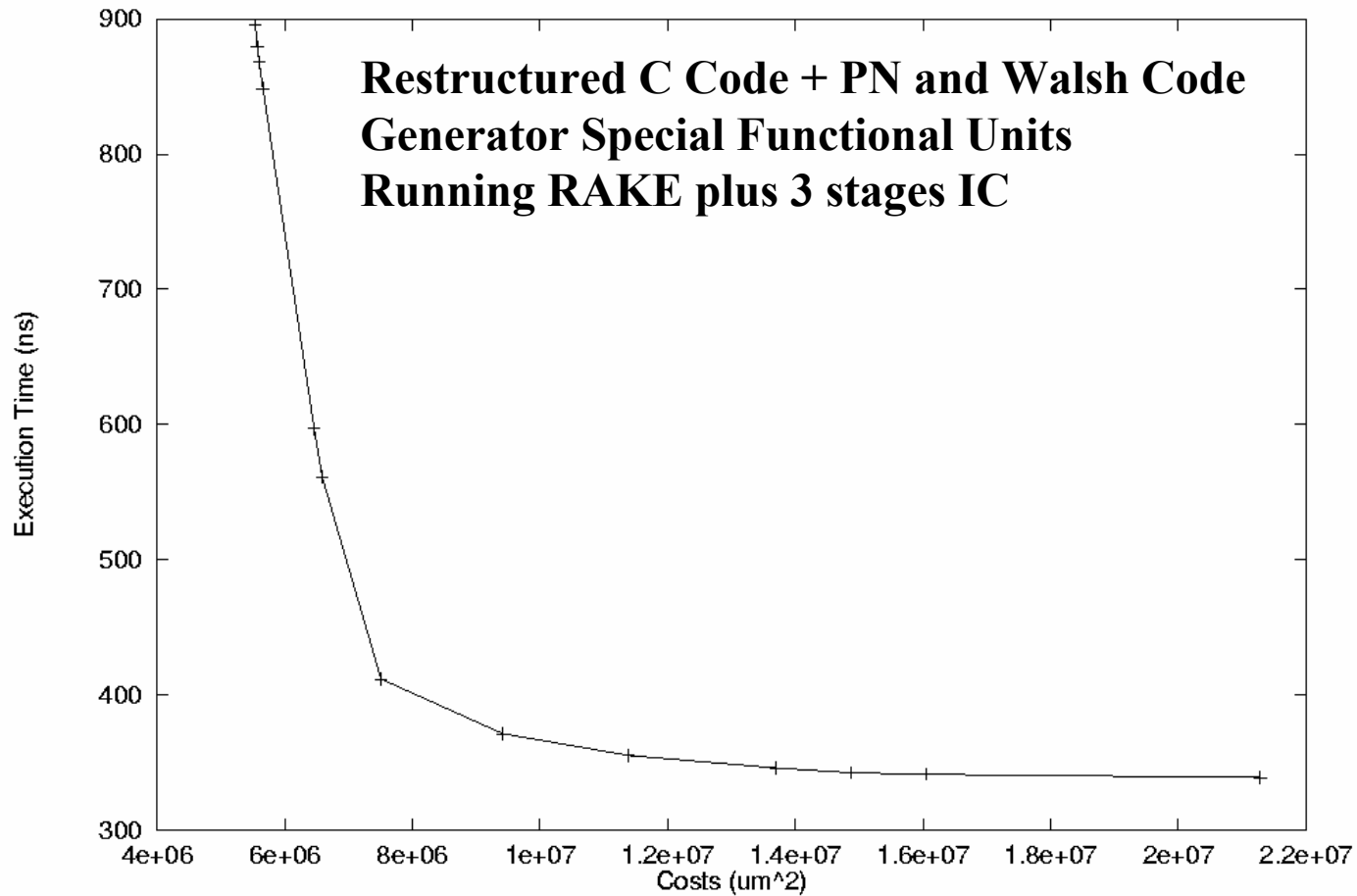
# Explorer Results : Behavioral Code only



# Explorer Results: Complex Arithmetic SFU's added



# Explorer Results: Final Machine



# Gate and Power Estimates

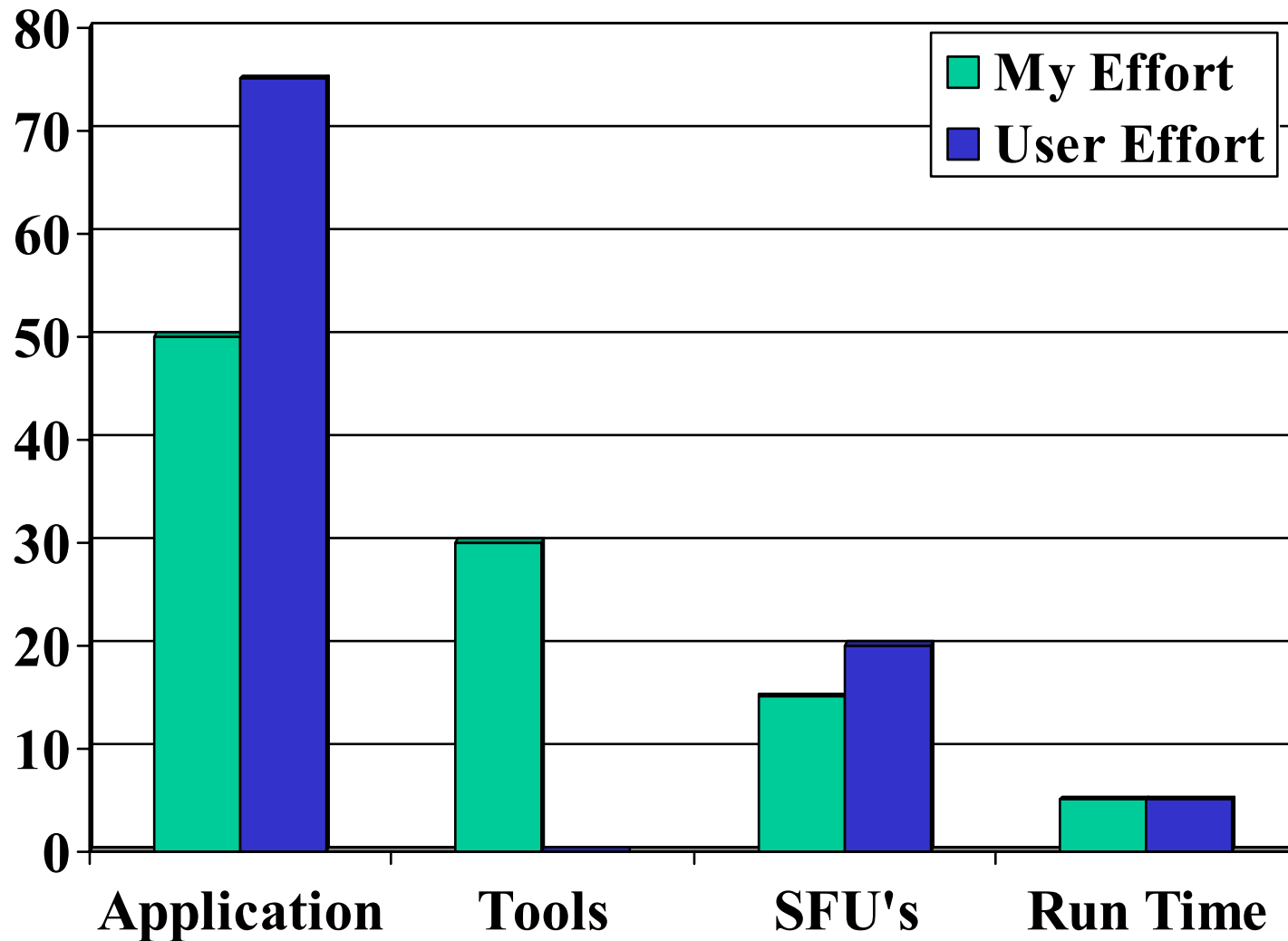
Note: Gate & Power Estimates are only for data path, no control

		ICP @ 122.88 MHz		CPE @ 125 MHz	
		Number of		Number of	
	Max IC	Gates	Power per	Gates	Power per
RAKE	Stages	(1,000s)	Finger (mw)	(1,000s)	Finger (mw)
1	1	83	2.95	73	2.84
1	2	128	3.43	141	3.48
1	3	166	4.92	210	5.97

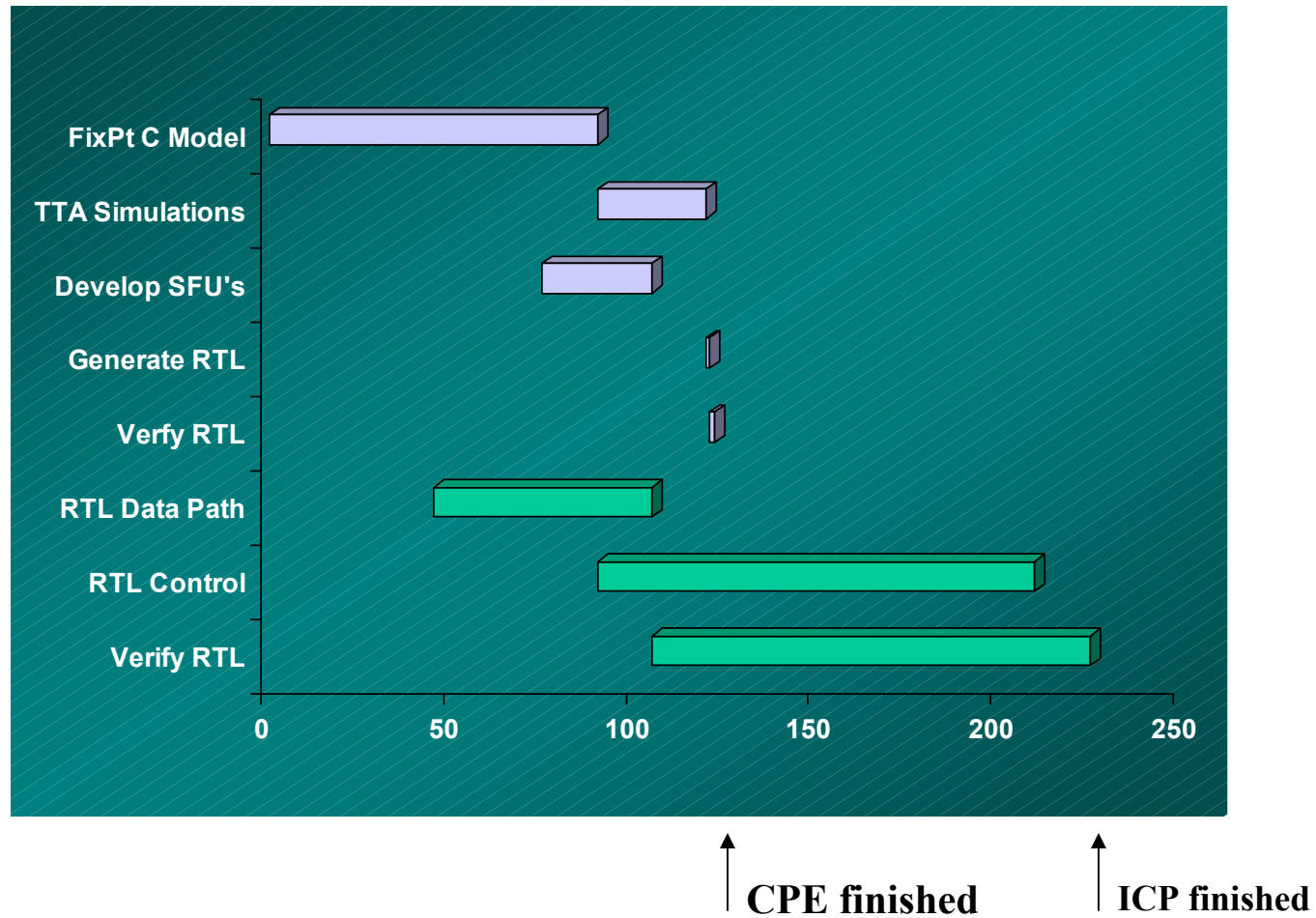
Example: RAKE + 2 IC Stages  
 running 8 fingers:  
 HW size: 128K gates  
 ICP Power:  $8 \times 3.43 \text{ mw} = 27.44 \text{ mw}$   
 CPE Power:  $8 \times 3.48 \text{ mw} = 27.84 \text{ mw}$

Instruction Word Size: 241 bits  
 Number of Instruction Words: 1079

# Development Effort Breakdown (percentage of total effort)



# Impact on Development Time





**TTA Design Methodology reduced design time in half!**

# Requirements for TTA Tools

- Critiquing Move Tool Set
- Improvements to current tool set
- Future Enhancements
- Research Topics for TTA



# Critiquing MOVE Tool Set

- Installing the tool set 
  - **Bloody nightmare!!!**
    - Tool set consists of 2233 separate files
    - 50 pages of README text for manual installation of tools
    - 11 separate make files to be run
    - Hardwired path names in make and script files
- Quality of the tools 
  - **“University quality”**
    - Fixed many bugs
    - Not all tools compiled first time, one still core dumps.
    - VHDL generation not currently coupled in with TTA tool set.
- User Interface
  - **Originally, No Documentation**
    - GUI based tools not currently working
    - Shell commands with arguments (usage message)
    - Poor error recovery (error message, “segment fault, core dumped”)

# Improvements to Current Tool Set

- Tie all tools under common GUI
  - Compiler, Assembler, Linker / Loader, Simulator, Explorer, RTL Generation
- Better Documentation and Help Facilities
  - Document intermediate files
  - Design tutorials for adding SFU's
- Add Interactive Debugging Capability
  - Single Stepping, Break Points, etc
  - Symbolic links back to C source files
- Better Support for Custom Libraries
  - Make it easier for customer to add SFU's
  - Support tools (Wizzard) for building SFU's
  - Support customer ASIC library information for Explorer use.  
Estimations based on cycle counts, gate counts and power consumptions



# Future Enhancements

- Compressed instruction word
- Program Instruction Control vs State Machine Control
- More specialized FU's can be added to the design library (Complex Arithmetic, Matrix Arithmetic, Graphics Support, etc)
- Integrate Synthesis tools into TTA tool Set
- User Guided Explorer



# Research Topics for TTA

- Instruction Compression Algorithms
- Bridging on MOVE busses
- Support Hierarchical Designs

