System-level Synthesis of Dataflow Applications for FPGA-based Distributed Platforms

Hugo A. Andrade, Kaushik Ravindran, Alejandro Asenjo, Casey Weltzin

NI Berkeley, NI Austin

National Instruments Corporation

Tenth Biennial Ptolemy Miniconference

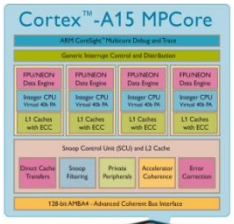
Berkeley, California, November 7th, 2013

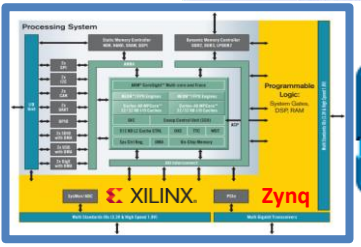
NATIONAL INSTRUMENTS

Modern FPGA-based Distributed Heterogeneous Platforms

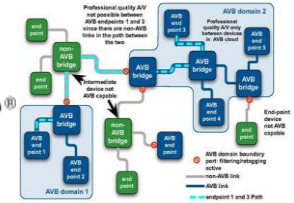
LabVIEW

Desktop, RT, FPGA, uP, DSP Design, IP Builder

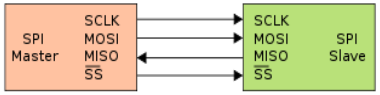




PCI EXPRESS®



I²C BUS



PXI Systems

CompactRIO

Single-Board RIO

High Level Synthesis

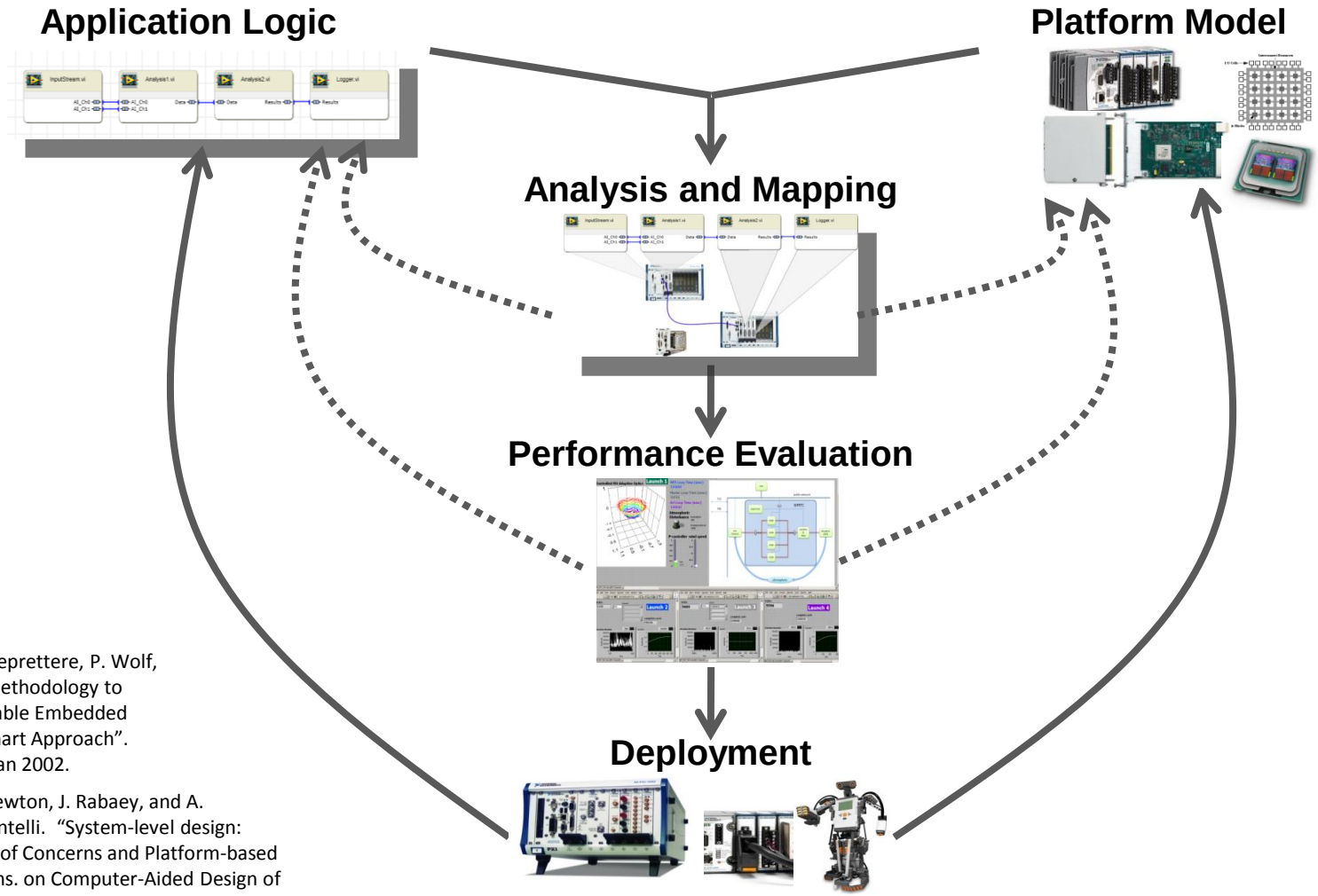
- A method for *increasing* (FPGA or system) *design productivity* while still providing an *efficient implementation*
  - Broadens design productivity to domain experts
    - They can use a high-level language such as ANSI C/C++, Python, MATLAB, LabVIEW
  - Design space exploration: optimizing among performance, footprint, or other constraints
  - Enables targeting more complex platforms from higher level program

HLS Techniques Covered

- Directive-based high-level synthesis
- Synthesis from analyzable domain specific languages
- System-level synthesis to heterogeneous targets

Applicable to textual and graphical hardware programming environments, e.g. VHDL/Verilog or LabVIEW FPGA

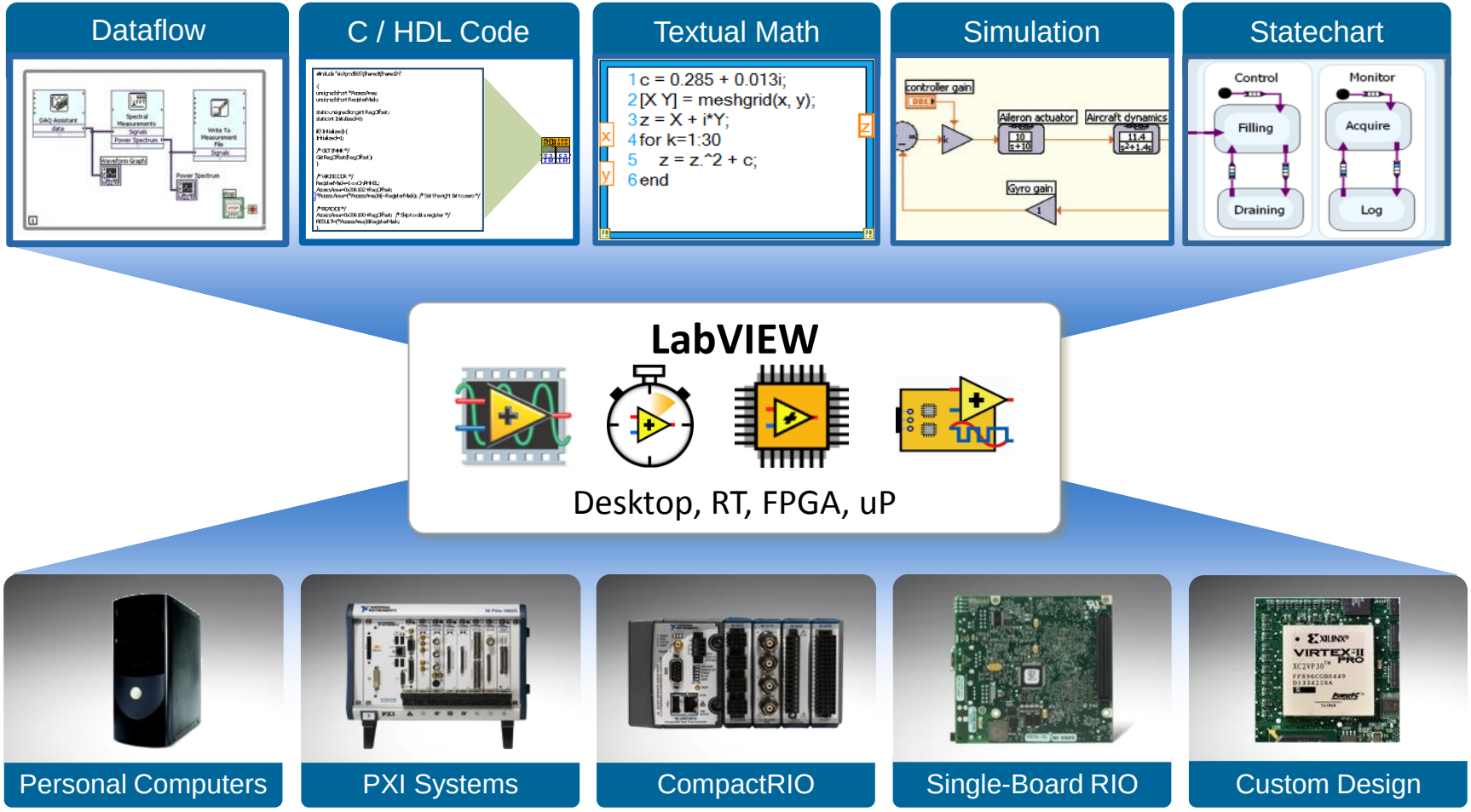
Y-Chart Disciplined Design Methodology



B. Kienhuis, E. F. Deprettere, P. Wolf, K. A. Vissers. "A Methodology to Design Programmable Embedded Systems - The Y-Chart Approach". SAMOS, p.18-37, Jan 2002.

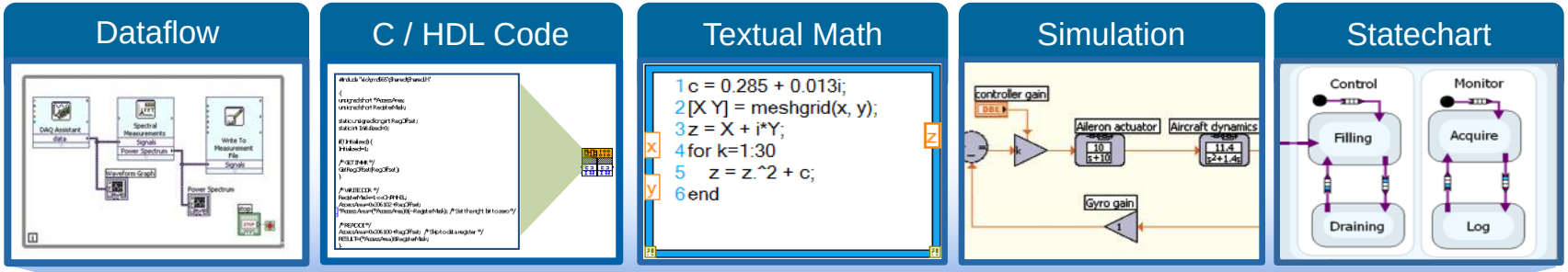
K. Keutzer, A. R. Newton, J. Rabaey, and A. Sangiovanni-Vincentelli. "System-level design: Orthogonalization of Concerns and Platform-based Design". IEEE Trans. on Computer-Aided Design of Integrated Circuits and System, 19(12): p.1523-1543, Dec 2000.

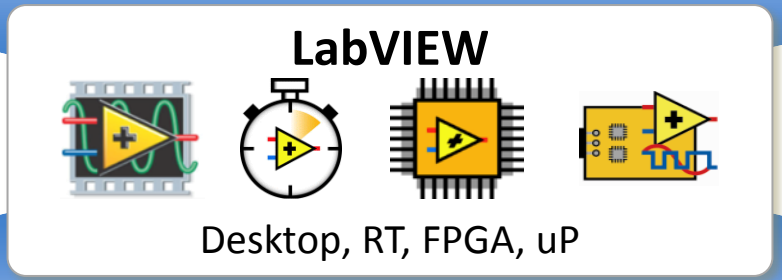
Models of Computation and Platform-Based Design

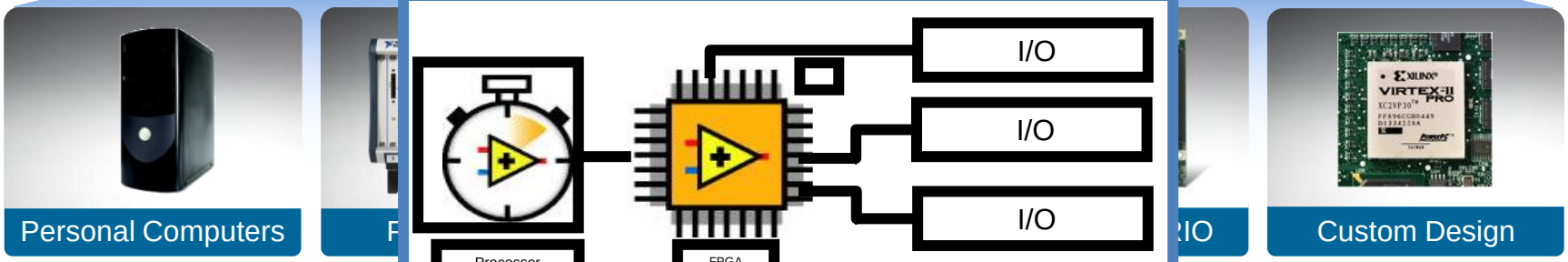


Ref: Models of Computations,. Prof. Edward Lee, Platform Based Design, Prof. Alberto Sangiovanni-Vincentelli

Models of Computation and Platform-Based Design

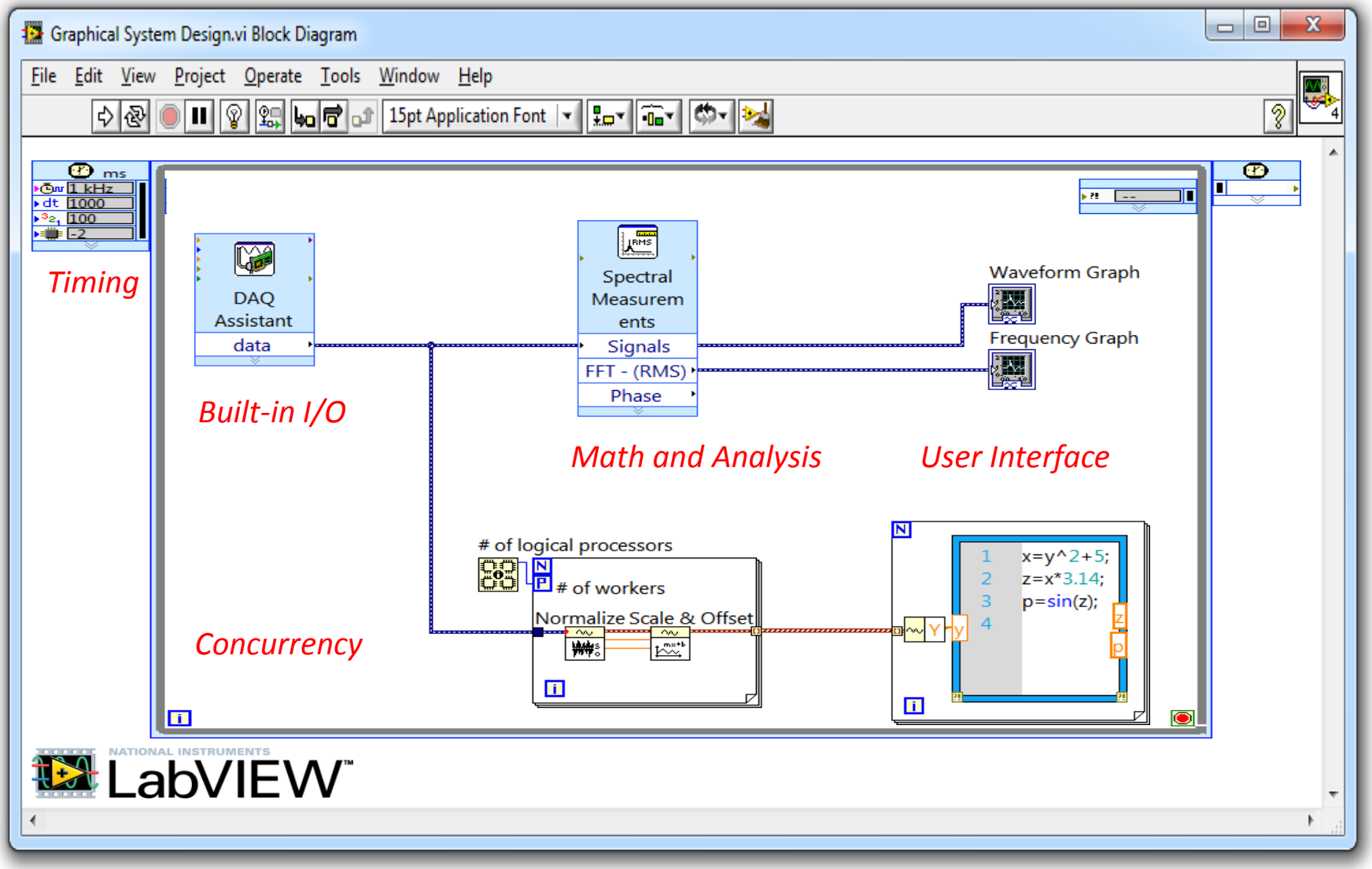




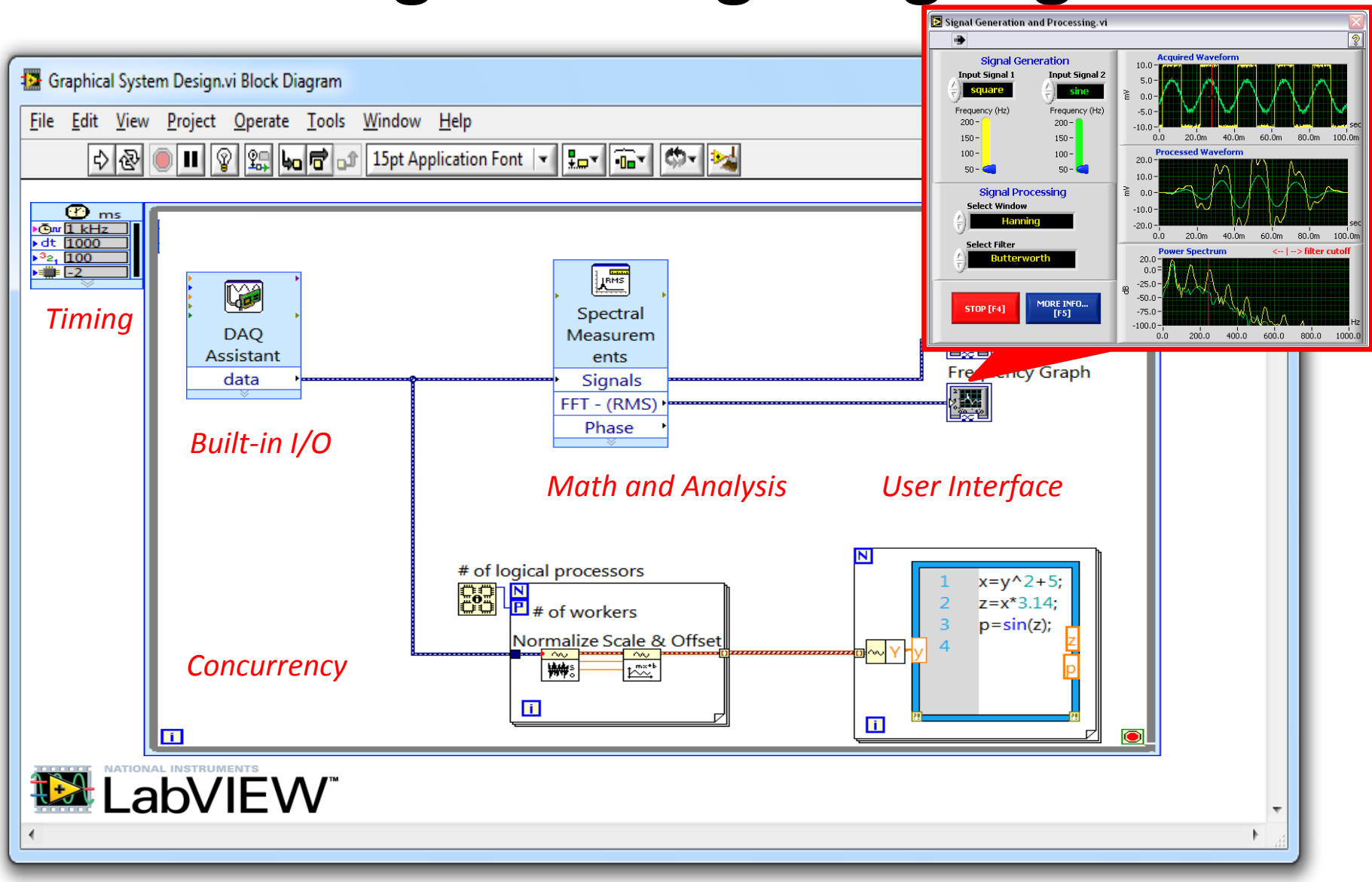


Ref: Models of Computations,. Prof. Edward Lee, Platform Based Design, Prof. Alberto Sangiovanni-Vincentelli

Base Programming Language: G

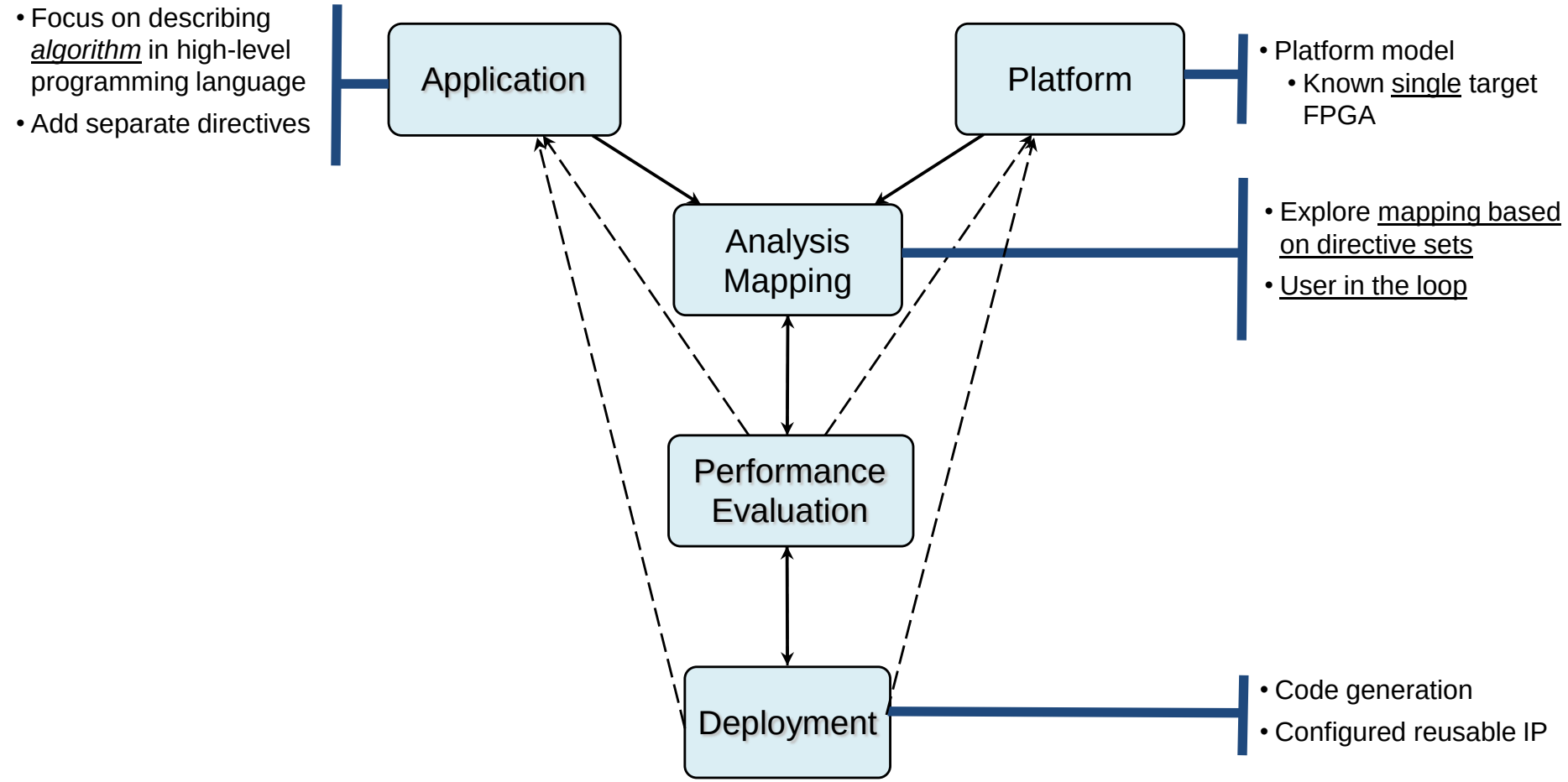


Base Programming Language: G



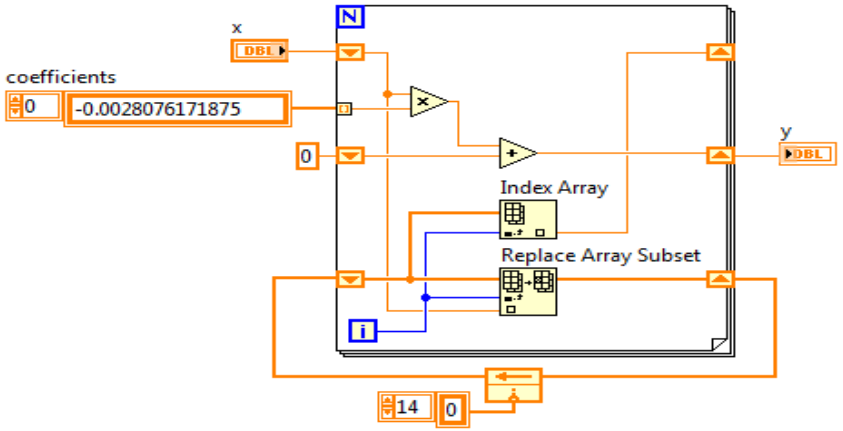
DIRECTIVE-BASED HIGH-LEVEL SYNTHESIS

Approach

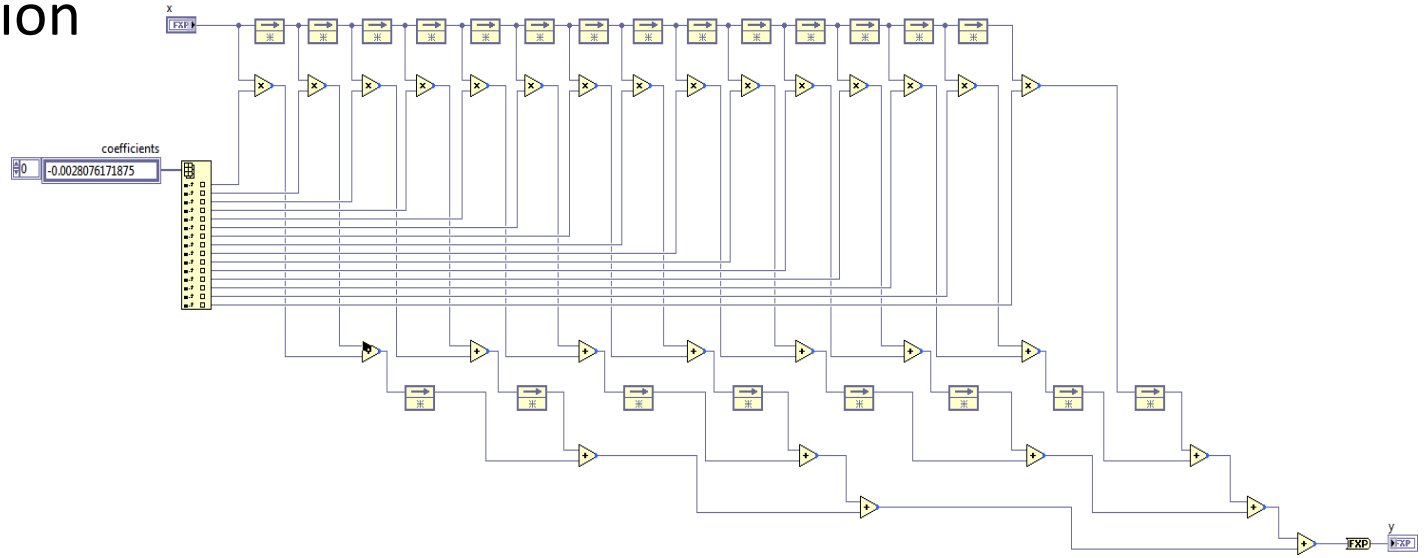

• Focus on describing *algorithm* in high-level programming language
• Add separate directives
Application
Platform
• Platform model
• Known single target FPGA
Analysis Mapping
• Explore mapping based on directive sets
• User in the loop
Performance Evaluation
Deployment
• Code generation
• Configured reusable IP

Current Challenges Programming in G

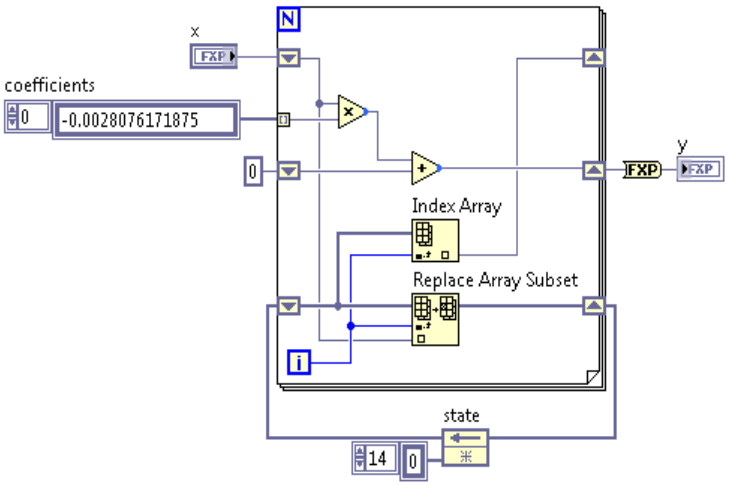
FIR filter in regular LabVIEW:



G implementation after manual unrolling and pipelining:

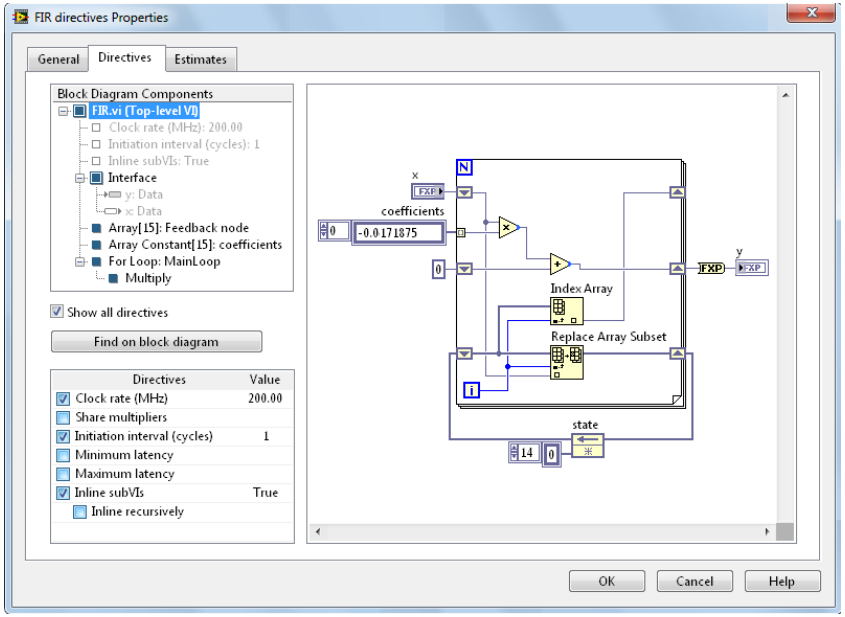


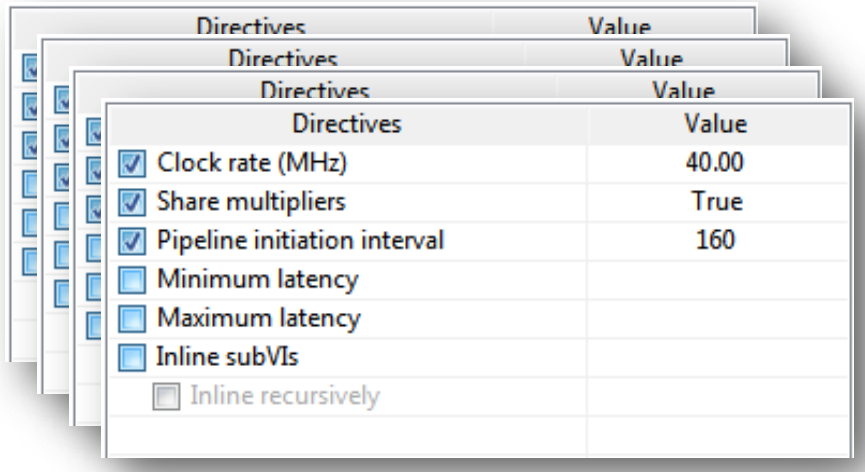
LabVIEW FPGA IP Builder User Flow



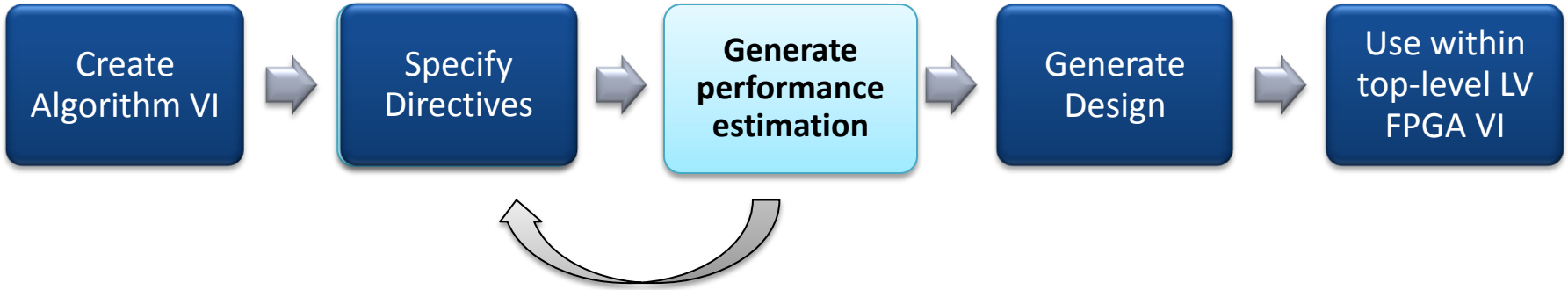
- Use dataflow programming
- Limited functions palette

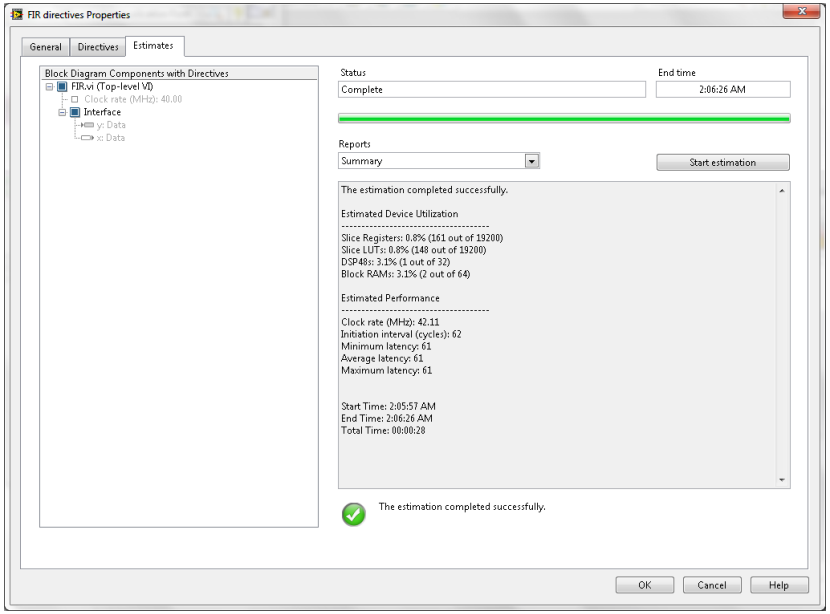
LabVIEW FPGA IP Builder User Flow

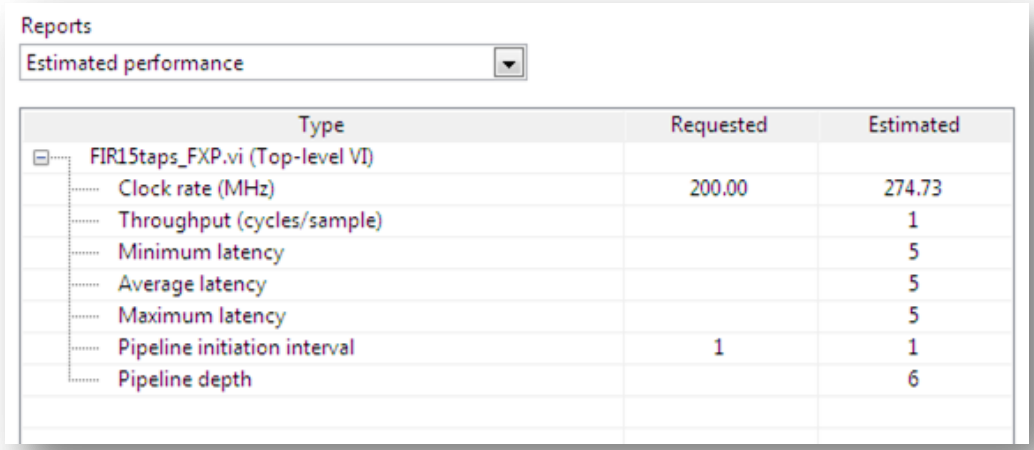




LabVIEW FPGA IP Builder User Flow

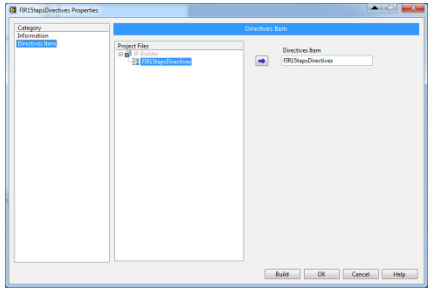






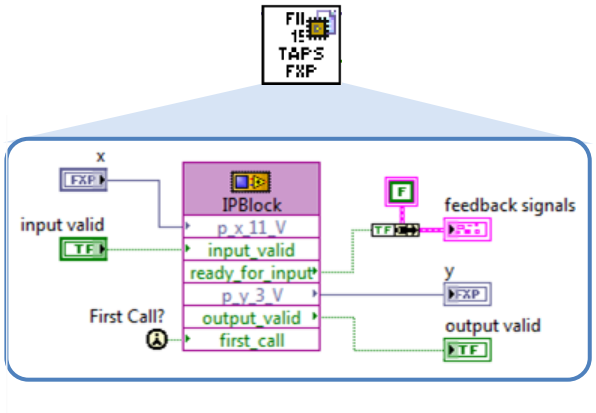
LabVIEW FPGA IP Builder User Flow

Choose VI and Directives

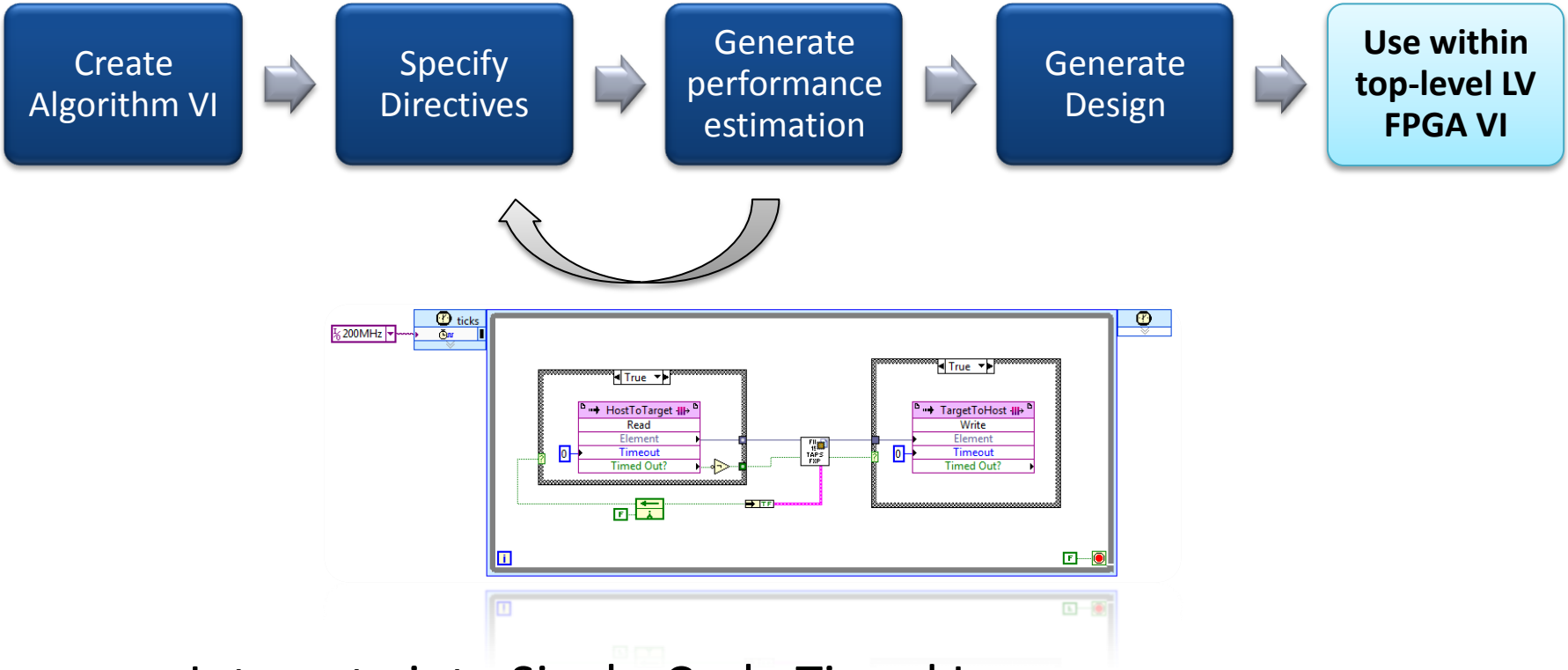


Generate HDL

Create IP VI



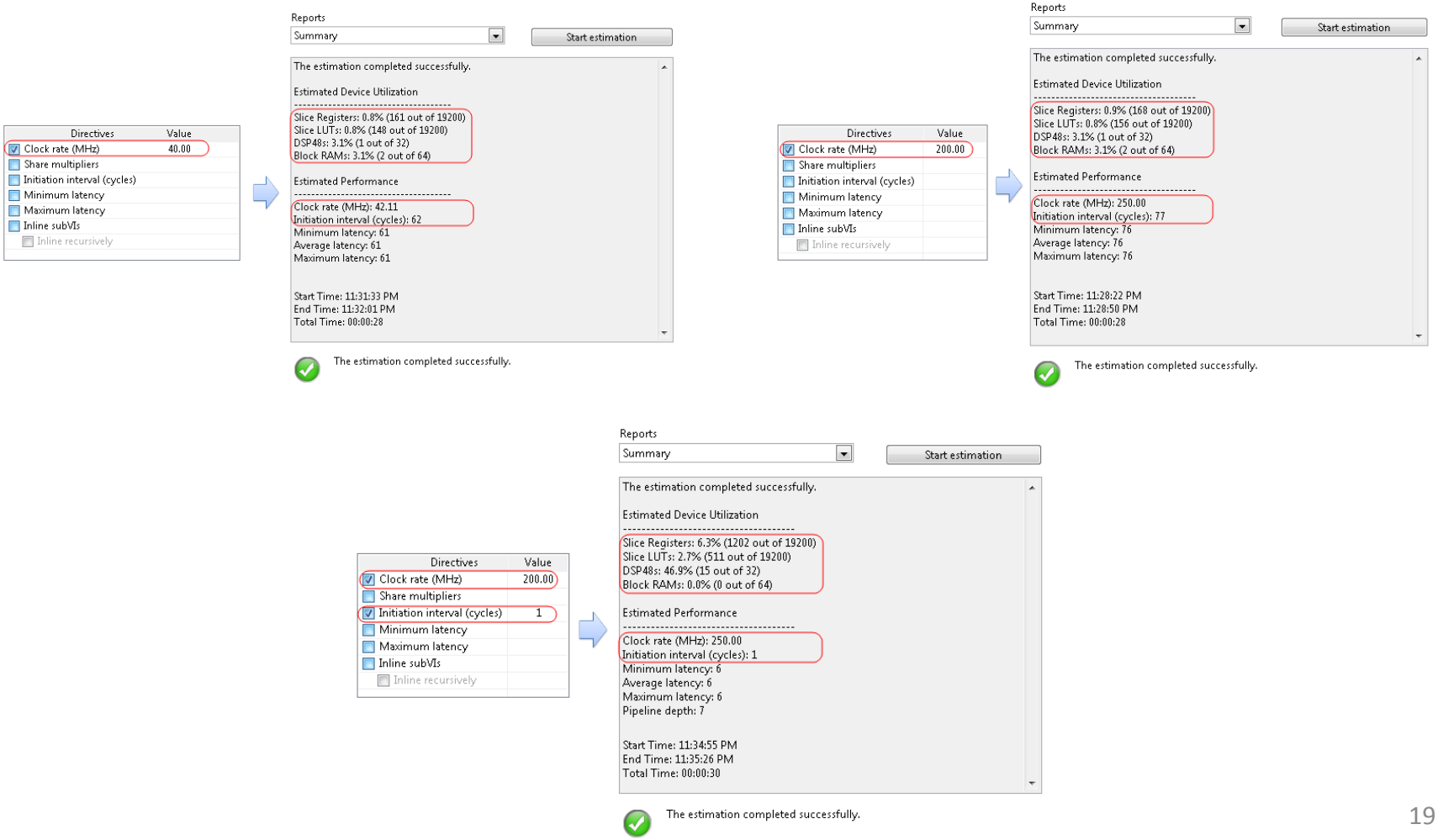
LabVIEW FPGA IP Builder User Flow



- Integrate into Single-Cycle Timed Loop
- Add I/O, DMA FIFOs, Host Communication, etc.

Exploration Example - Directives

$$Throughput = (ClockRate * SamplesPerIteration)/InitiationInterval,$$



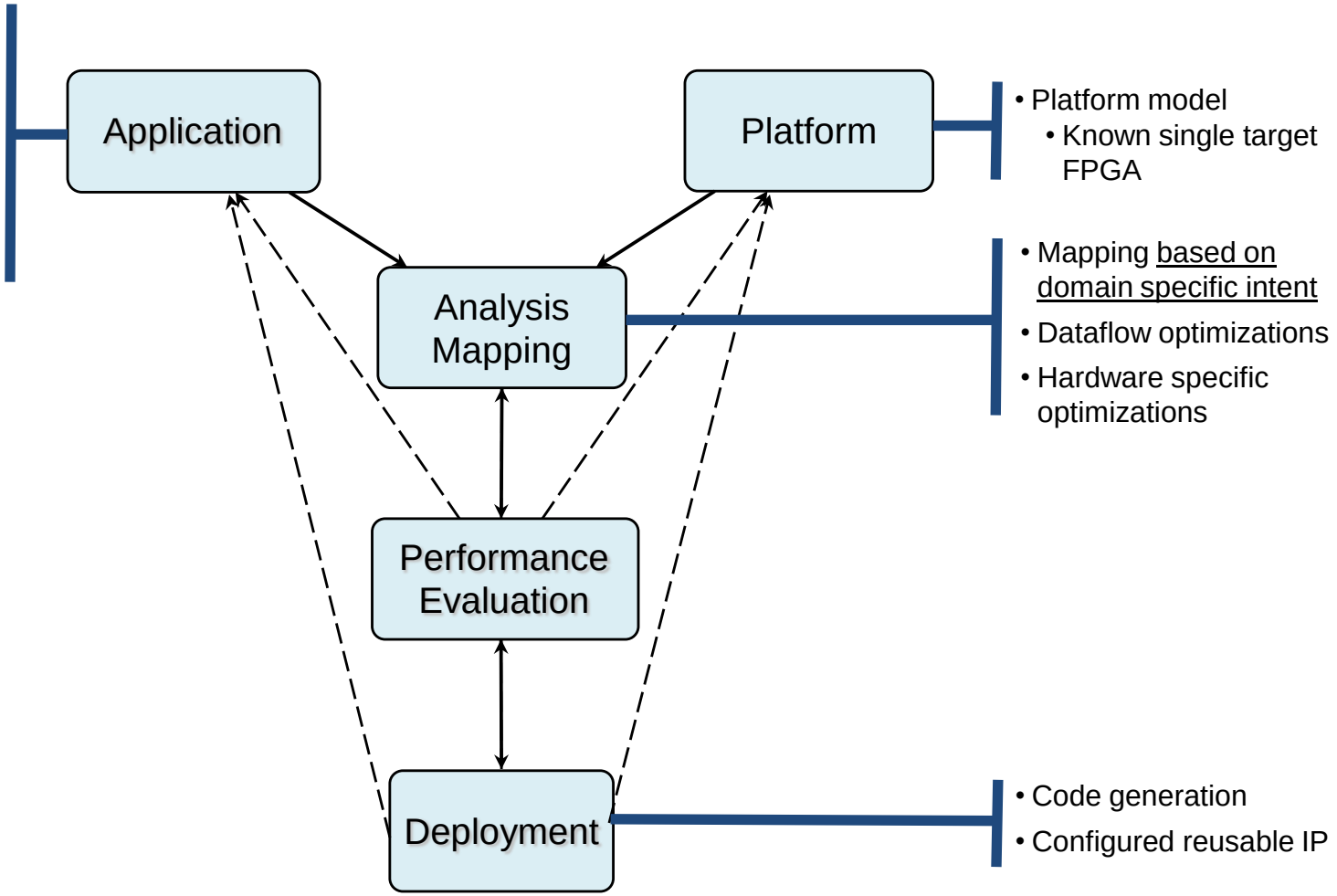
Comparison

Example Set #		1	2	3
Directive Settings	Clock Rate (MHz)	40	200	200
	Initiation Interval (Cycles)	-	-	1
Estimated Performance	Clock Rate (MHz)	42	250	250
	Initiation Interval (Cycles)	62	77	1
	Latency (cycles)	61	76	6
	Throughput (MS/s)	**0.68**	**3.25**	**250**
Estimated Resource Utilization	Registers	**0.80%**	**0.90%**	**6.30%**
	LUTs	**0.80%**	**0.80%**	**2.70%**
	Multipliers	**1**	**1**	**15**
	Block RAMs	2	2	0
Estimate Time (s)		28	28	30

SYNTHESIS FROM ANALYZABLE DOMAIN SPECIFIC LANGUAGES

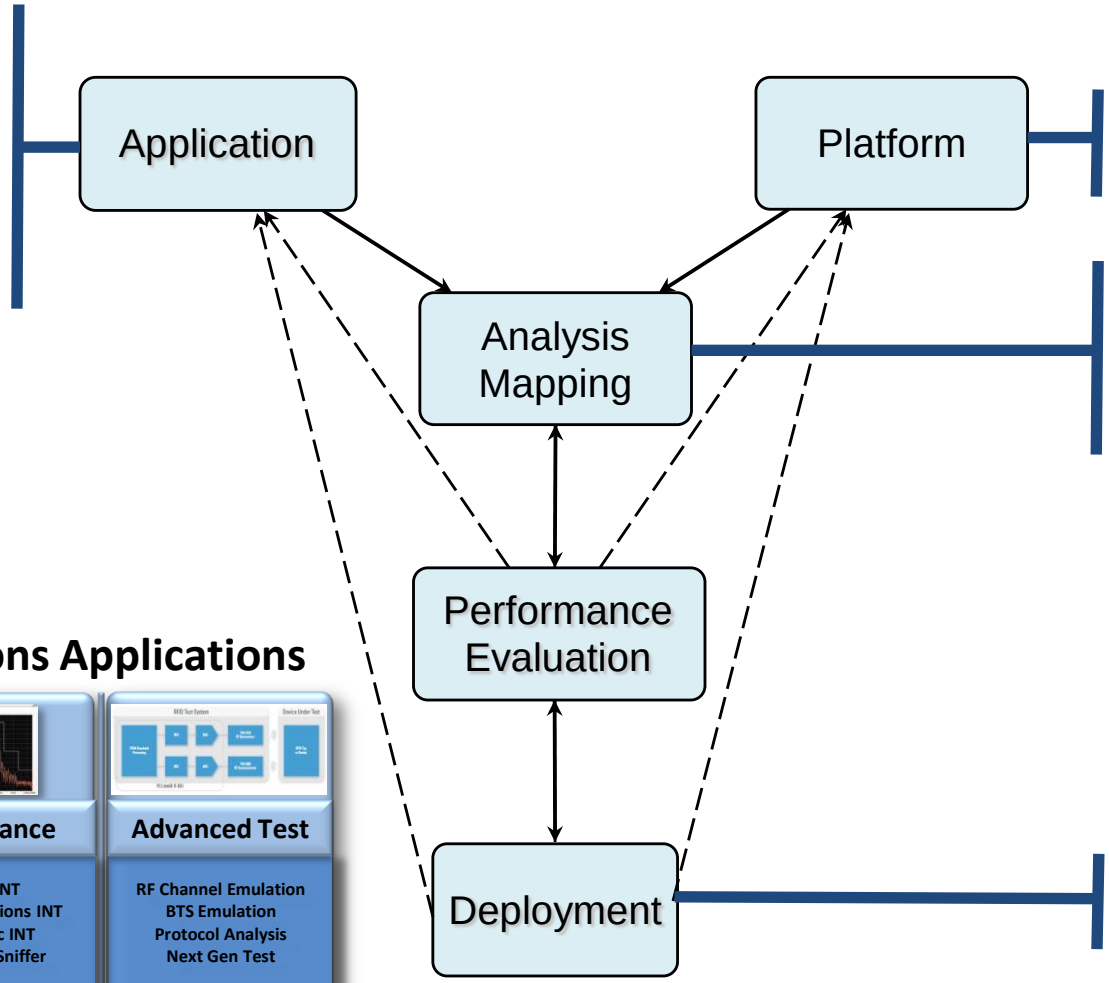
Approach

- *Domain specific application language*
  - E.g. streaming applications in communications domain
- Prequalified IP Component Library



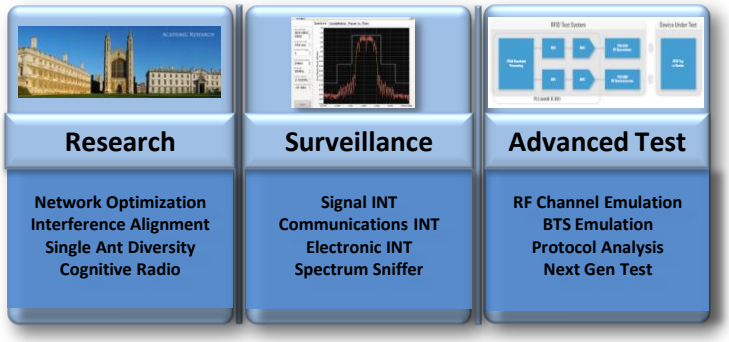
Approach

- *Domain specific application language*
  - E.g. streaming applications in communications domain
- Prequalified IP Component Library

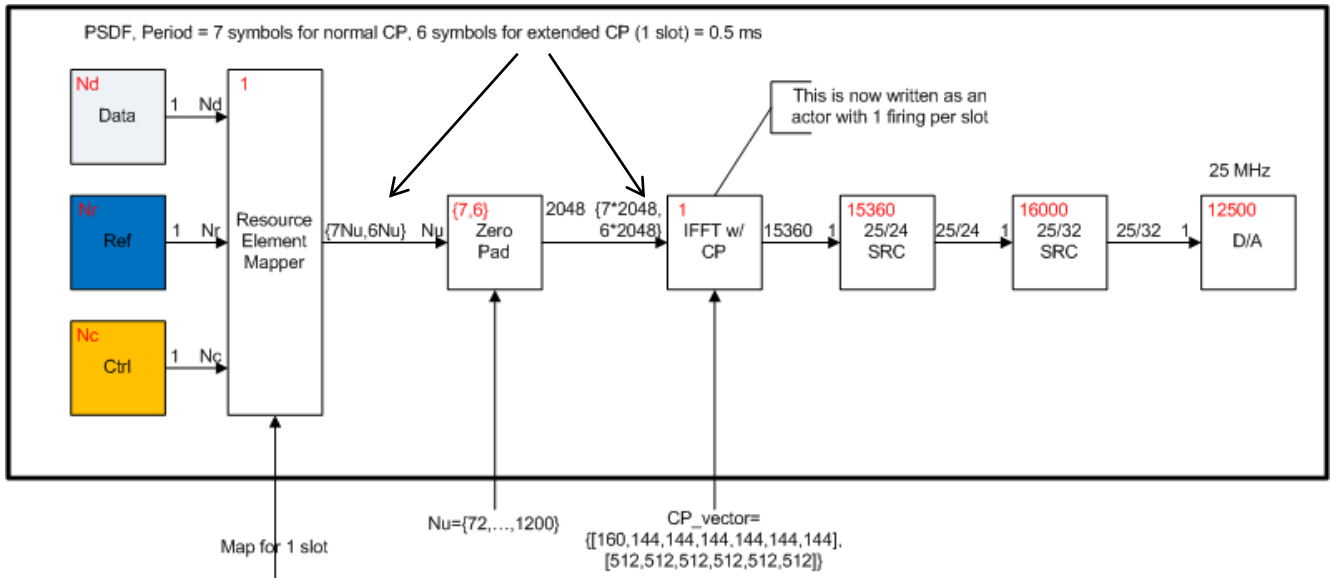


- Platform model
  - Known single target FPGA
- Mapping based on domain specific intent
- Dataflow optimizations
- Hardware specific optimizations
- Code generation
- Configured reusable IP

RF Communications Applications



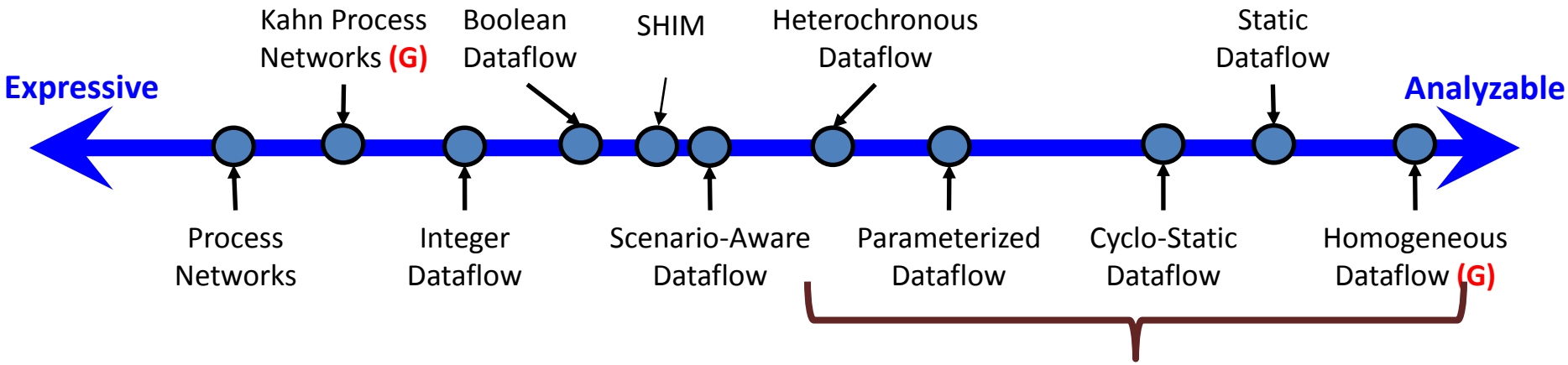
Streaming Model of the OFDM Transmitter

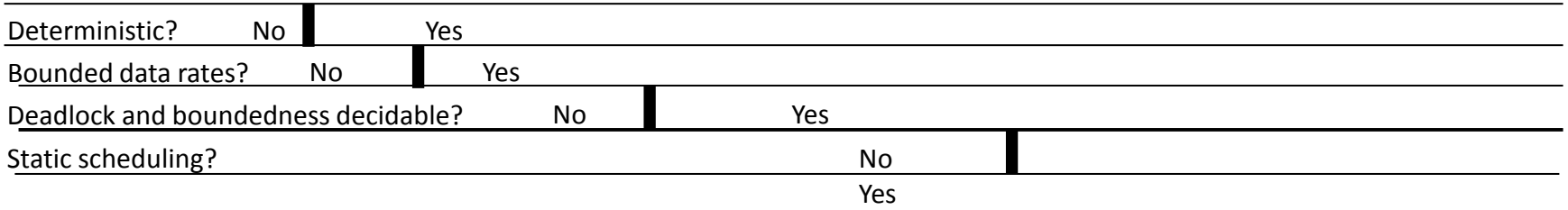


- Nt = {1,2,4}
  - Compile time - # transmitters
- Nu = {72, 180, 300, 600, 900, 1200}
  - Initialization time - Bandwidth
- CP mode = {'Normal', 'Extended'}
  - Run time, To overcome Inter-symbol-interference, Can be applied at symbol boundary
- CP Vector
  - Selection based on CP mode, Elements must be applied at symbol boundary

Challenge: How to express a domain expert's algorithm specification in a model that is viable for analysis and implementation?

MoCs for Streaming Applications

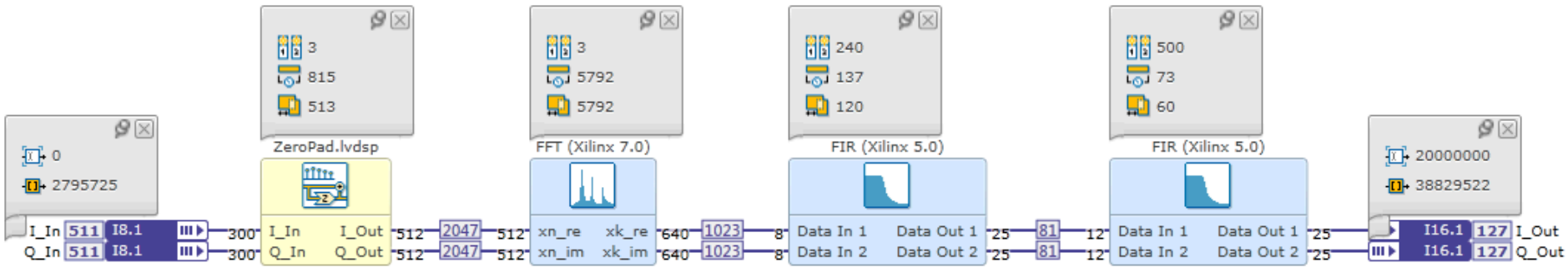




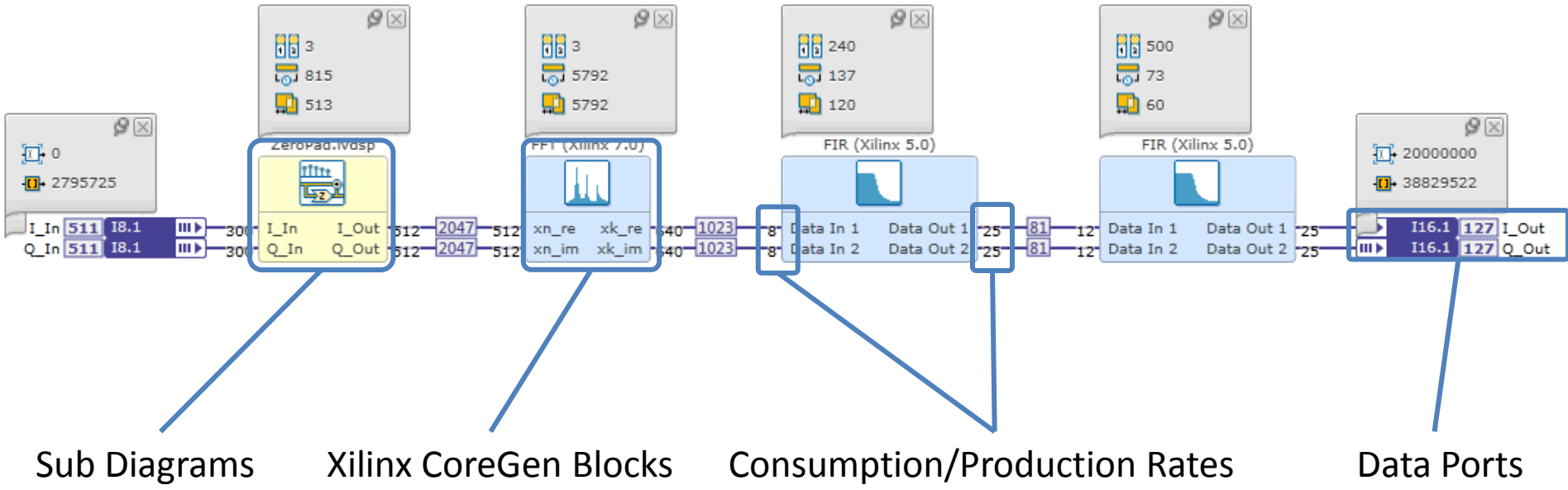
Key trade-off: Analyzability vs. Expressability

[1] Edward A. Lee, "Concurrent Models of Computation for Heterogeneous Software", EECS 290, 2004.
[2] Stephen Edwards, "SHIM: A Deterministic Model for Heterogeneous Embedded Systems", UCB EECS Seminar, 2006.

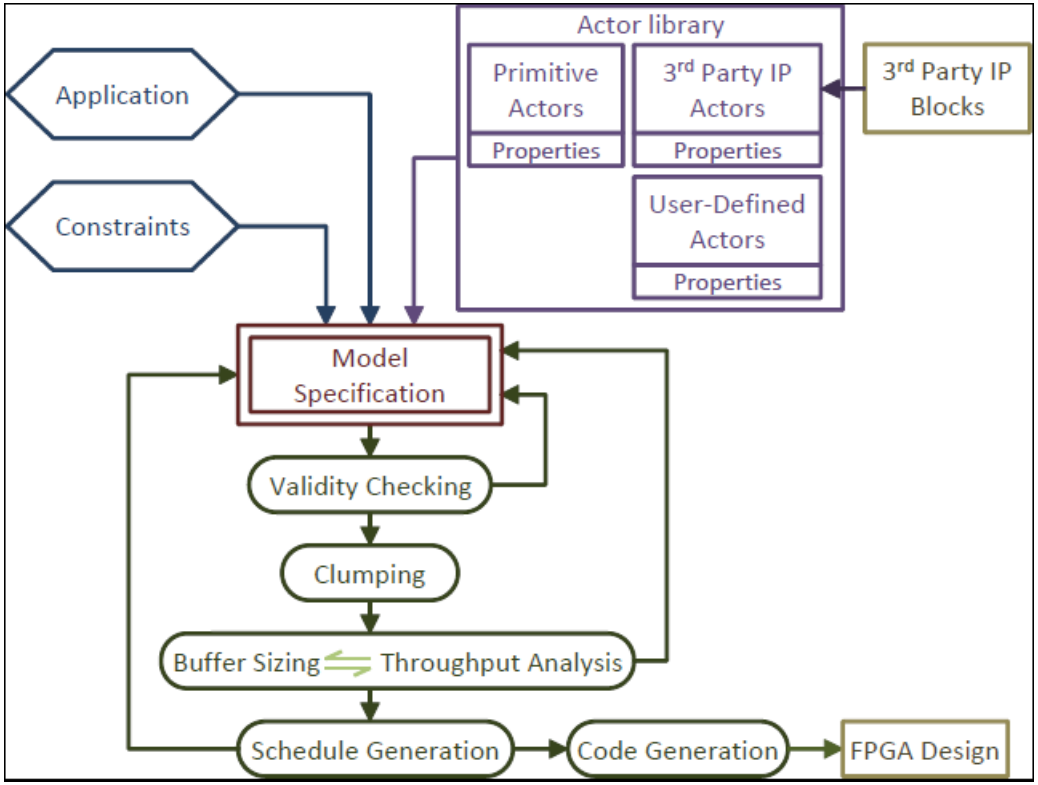
Application Specification in LabVIEW DSP Design Module



Application Specification in LabVIEW DSP Design Module



Tool Flow

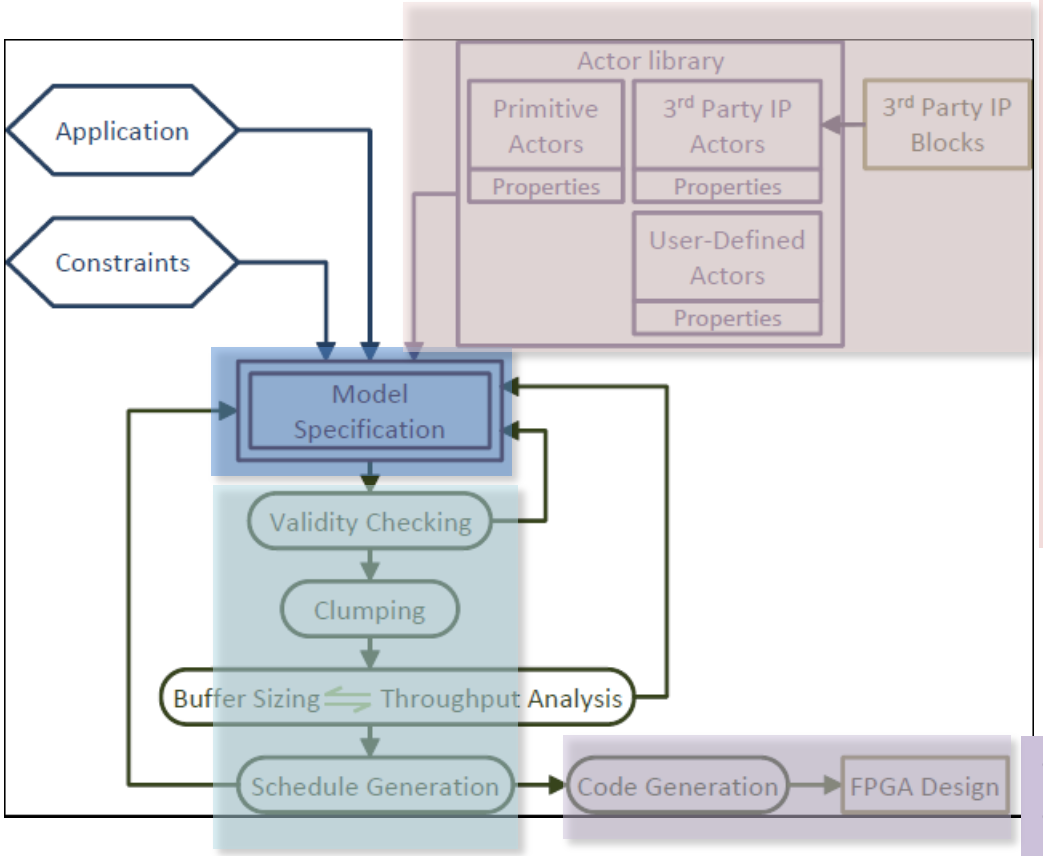

Actor library
Primitive Actors
Properties
3rd Party IP Actors
Properties
User-Defined Actors
Properties
3rd Party IP Blocks
Application
Constraints
Model Specification
Validity Checking
Clumping
Buffer Sizing
Throughput Analysis
Schedule Generation
Code Generation
FPGA Design

Tool Flow

Models of Computation

Analysis and Optimization Back End

Simulation and Verification



- Actor Definition
- Performance Models
- and Timing Library
- IP Modeling and
- Integration

- Code Generation and
- Implementation

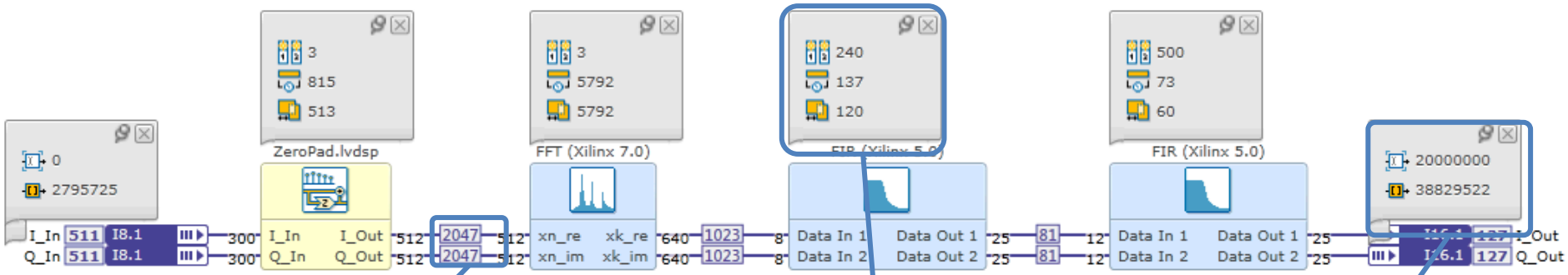
Analysis and Optimization Features

- Core dataflow optimizations
  - Model validation
    - Deadlock detection and boundedness check
  - Throughput and latency computation
  - Buffer size optimization (under throughput constraints)
  - Schedule computation

- Hardware specific optimizations
  - Resource constrained schedule computation
  - Retiming and fusion
  - Rate matching
  - IP interface synthesis

[1] S. S. Bhattacharyya, P. K. Murthy and E. A. Lee, "Software Synthesis from Dataflow Graphs," Kluwer Academic Publishers, Norwell, Mass, 1996.

Analysis and Optimization Results

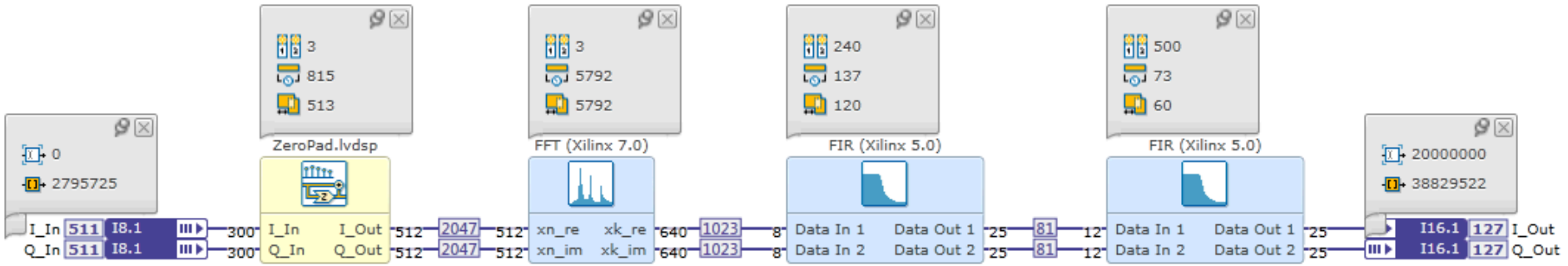


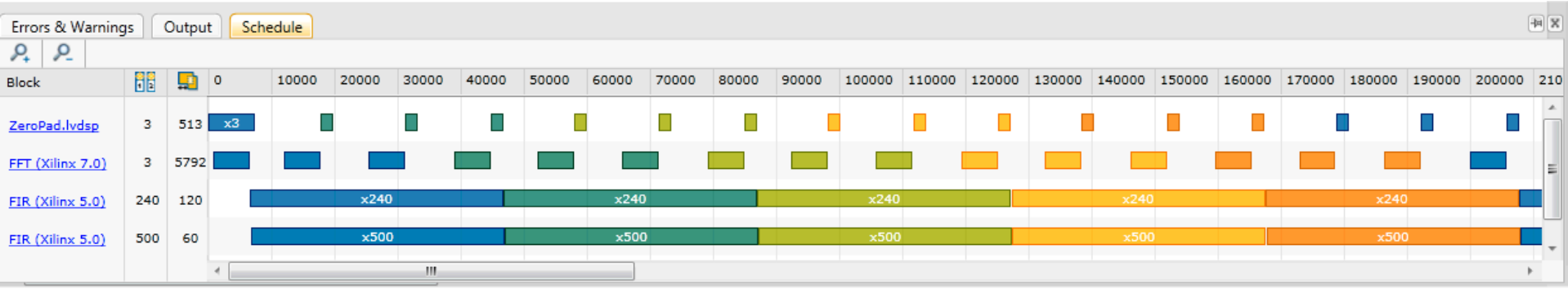
Auto buffer sizing to minimize resources

Calculated firing counts and timing data

Throughput constraints

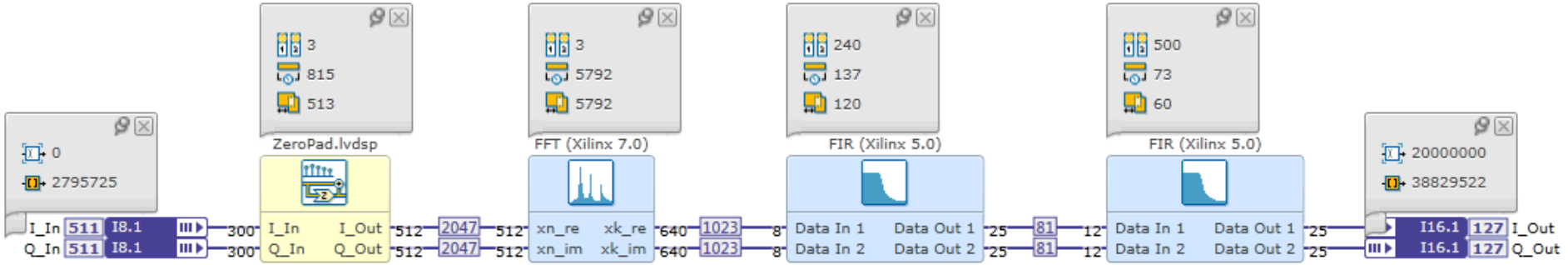
Analysis and Optimization Results

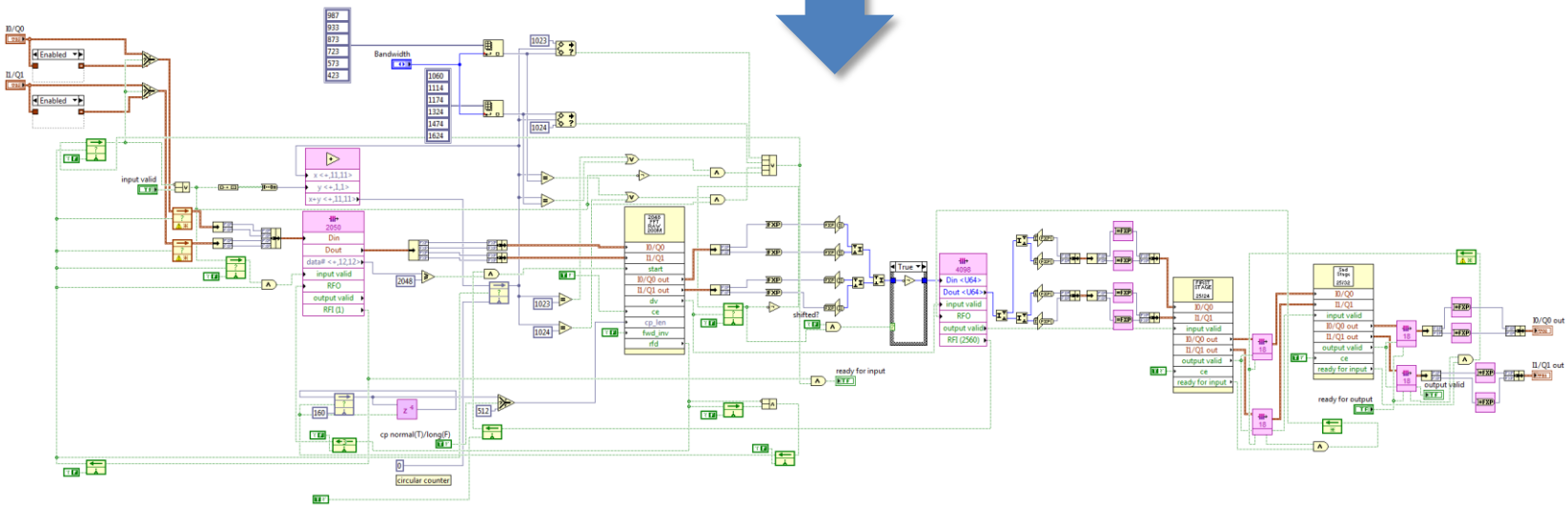




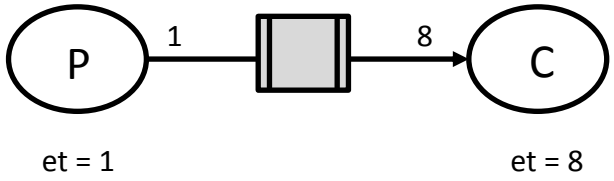
Calculated Schedule View

High-Level Model to FPGA blocks

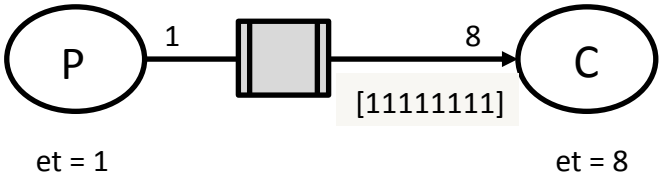




Extending SDF with Access Patterns



SDF
Optimal throughput = 1 sample/cycle
Buffer Size for optimal throughput = 16

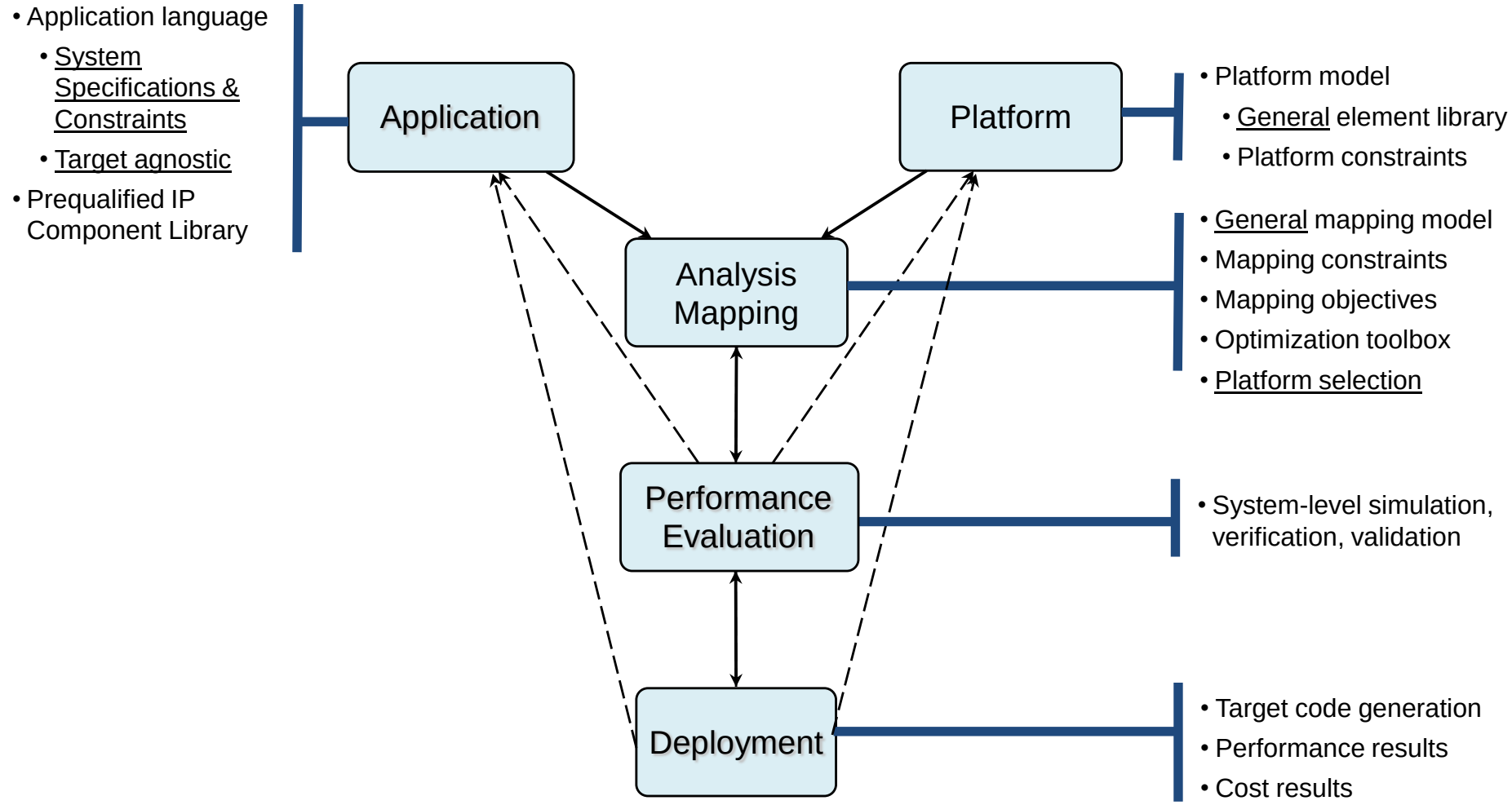


SDF-AP
Optimal throughput = 1 sample/cycle
Buffer Size for optimal throughput = 2

- SDF-AP Model of Computation
  - Access Patterns viewed only by the tool, not the user
  - Formal syntax and operational semantics
  - Definitions of key model properties
    - Executability, boundedness and throughout
  - Case studies to evaluate these algorithms
  - Executability for SDF -AP model is decidable
  - Boundedness for SDF -AP model is decidable

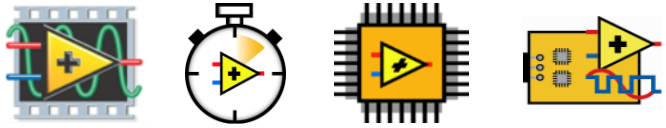
SYSTEM-LEVEL SYNTHESIS TO HETEROGENEOUS TARGETS

Approach

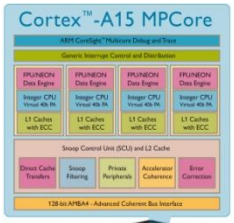

• Application language
• System Specifications & Constraints
• Target agnostic
• Prequalified IP Component Library
Application
Platform
Analysis Mapping
Performance Evaluation
Deployment
• Platform model
• General element library
• Platform constraints
• General mapping model
• Mapping constraints
• Mapping objectives
• Optimization toolbox
• Platform selection
• System-level simulation, verification, validation
• Target code generation
• Performance results
• Cost results

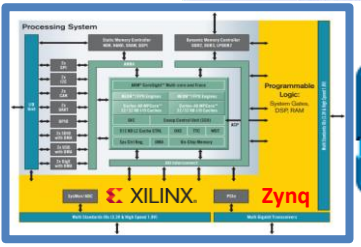
Modern FPGA-based Distributed Heterogeneous Platforms

LabVIEW

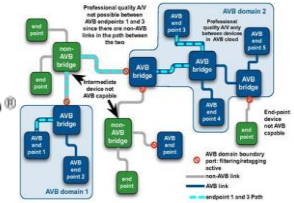


Desktop, RT, FPGA, uP, DSP Design, IP Builder

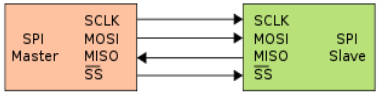




PCI EXPRESS®



I²C BUS



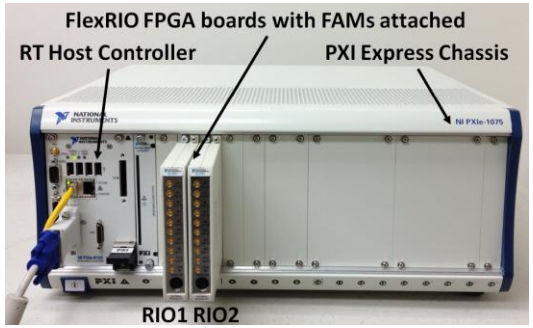
PXI Systems

CompactRIO

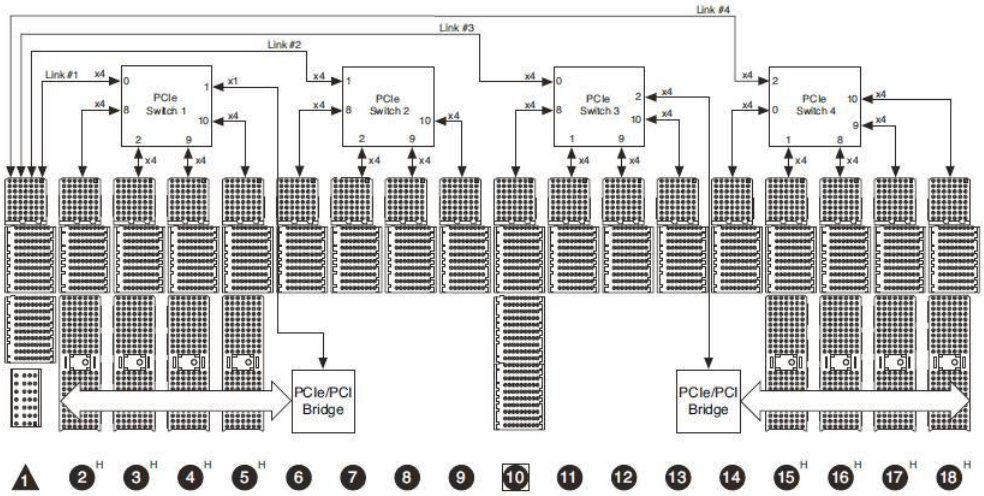
Single-Board RIO

Platform Model

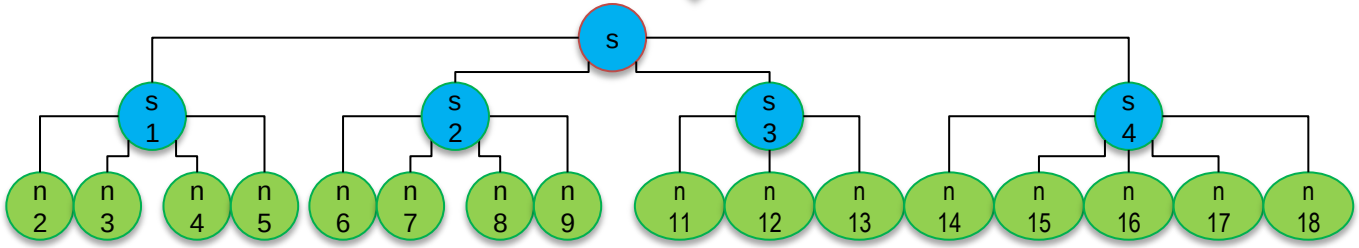
PXIe 1075 Chassis



PXIe 1075 Internal Architecture



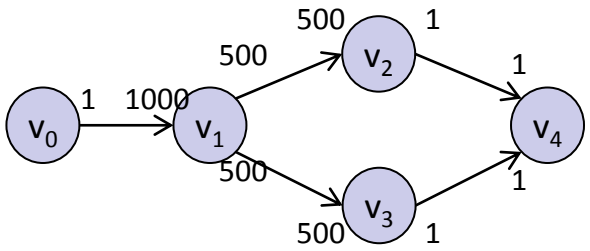
PXIe 1075 Communication Behavior Abstraction

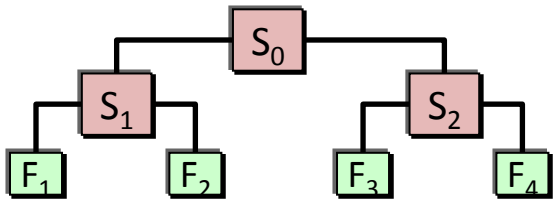


On any edge, max bandwidth supported is 800 MB/s for unidirectional traffic, and 750 for bidirectional traffic for 128 byte packets. The bandwidth can be shared linearly among competing streams

Analysis and Mapping

Problem Inputs





Max BW: 800MB/s

Actor	v_0	v_1	v_2	v_3	v_4
II	1	1000	500	1000	1
Area	100	1000	2000	1500	100

PE	F_1	F_2	F_3	F_4
Area	3000	4000	2000	1000

Throughput: 250 MB/s
Frequency: 100MHz

Optimization problem

Enforce
- Bandwidth constraint per link/switch
- Area constraint per target
- Link bound constraint per link
- Affinity, grouping and exclusion

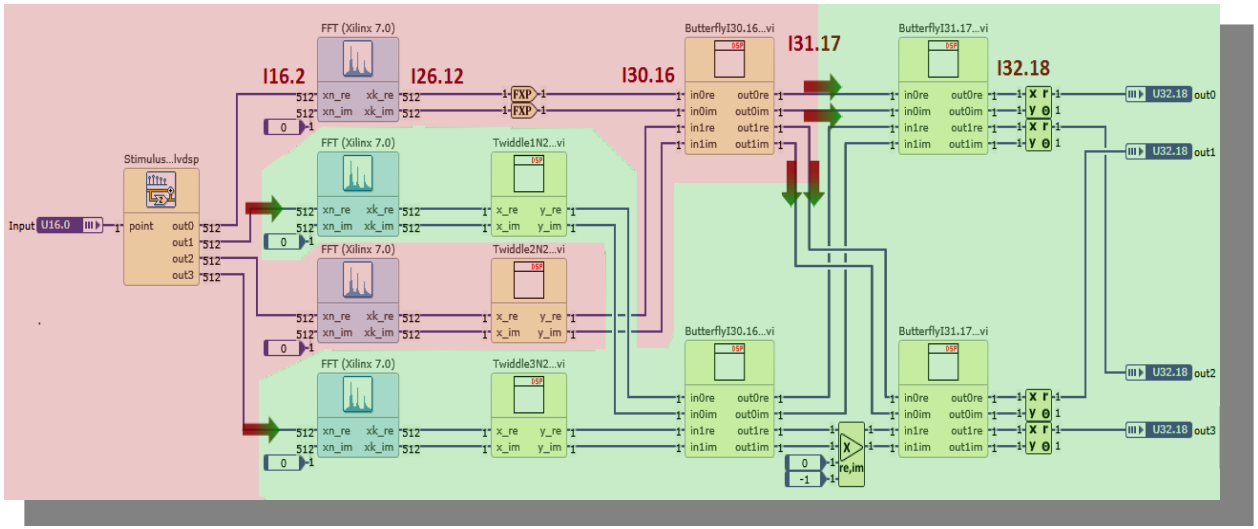
Optimize
- Throughput
- Area
- Latency

Solver and heuristic methods applied, and generate valid mappings

Results:
- Optimal mappings for OCT and OFDM models on 2- and 4- FPGAs in 1-3 mins
- Exact solver methods intractable for problems with over 50 tasks
- Heuristics within 20% of optimal; run-time of 1s for problems over 100 tasks

Experimental Results

- Platform and mapping model enabled experiments
- Static problem formulation (SMT solvers)
- Intuitive optimal results, sometimes even better than manual
- Reasonable scalability (alternate heuristics)



Summary

- Reviewed current trends in HLS
  - Viable method for *increasing* (FPGA or system) *design productivity* while still providing an *efficient implementation*
- Techniques Covered (Y-chart Methodology)
  - Directive-based synthesis
  - Synthesis from analyzable domain specific application models
  - System-level synthesis to heterogeneous targets
- These techniques can be combined hierarchically

Thank you

Questions? Comments?

Acknowledgements

This work was done in collaboration with:

Arkadeb Ghosal, Rhishikesh Limaye, Douglas Kim, Jeff Correll, Jacob Kornerup, Ian Wong, Guoqiang Wang, Guang Yang, Amal Ekbal, Mike Trimborn, Ankita Prasad, Trung N Tran, and Randy Allen.