NC STATE UNIVERSITY

FabScalar RISC-V

Rangeen Basu Roy Chowdhury

Anil Kumar Kannepalli

Eric Rotenberg

FabScalar

- Generates synthesizable RTL (Verilog) for arbitrary superscalar cores within a canonical superscalar template
- Vision
  - Accelerate development of single-ISA heterogeneous multi-core processors comprised of many microarchitecturally-diverse core types
  - Superscalar technology accessible to everyone (not just few elite teams at Goliath processor companies)
  - Research framework
    - High-fidelity cycle time, power, and area estimation of whole cores
    - Proof-of-concept of new microarchitectures
    - Technology-driven computer architecture research
    - FPGA and ASIC prototyping

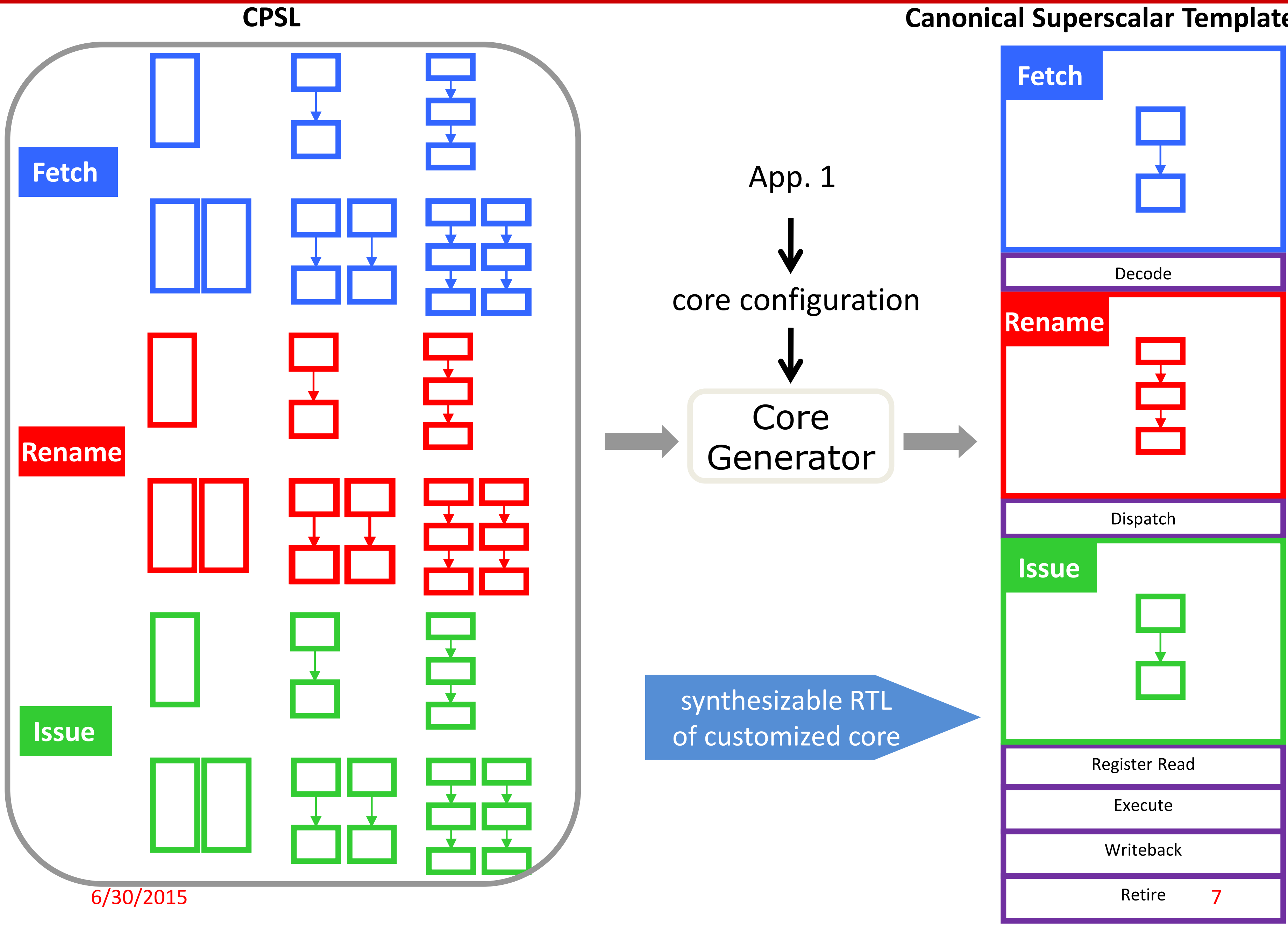
[1] FabScalar: Composing Synthesizable RTL Designs of Arbitrary Cores within a Canonical Superscalar Template , ISCA 2011

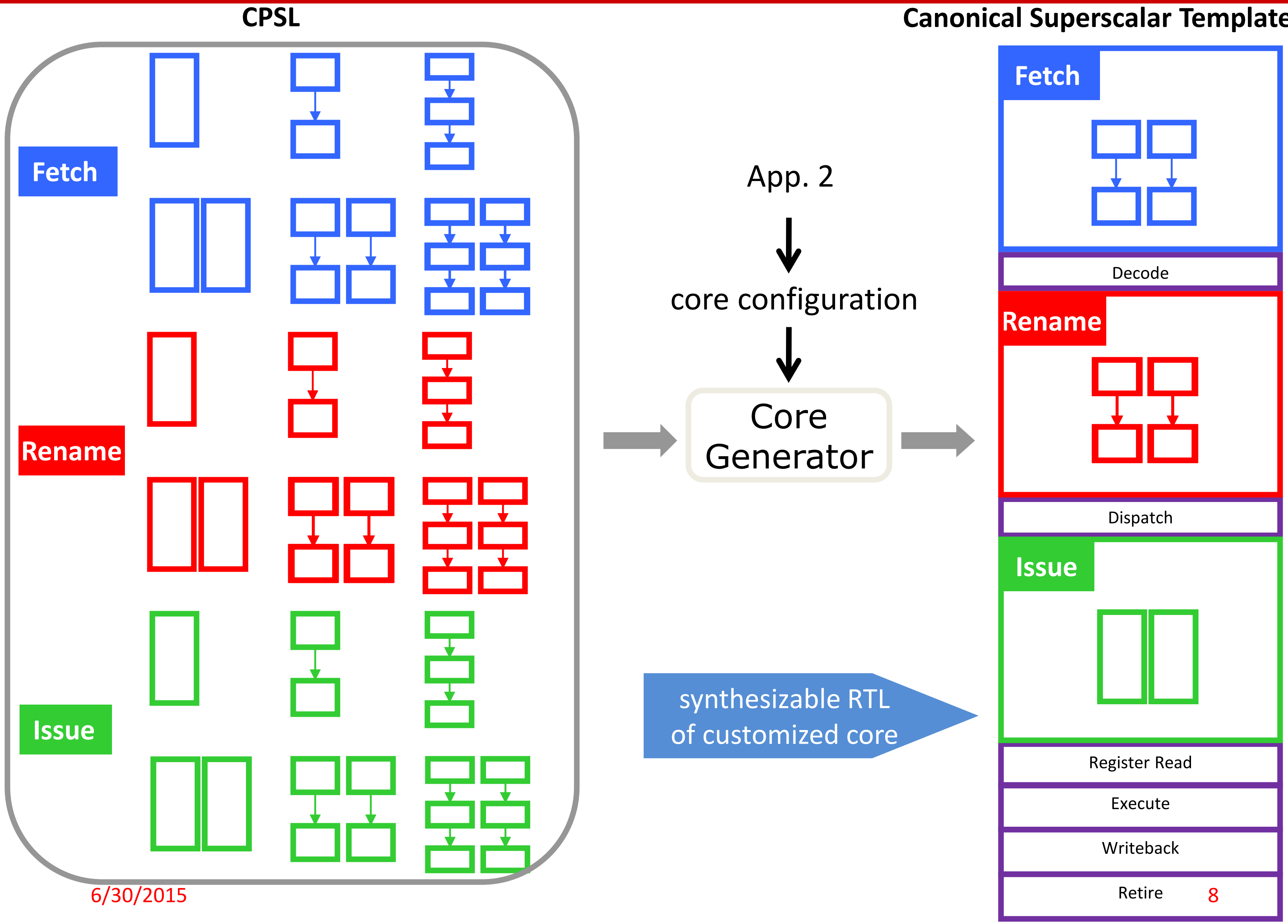
Outline

- FabScalar Toolset
  - Approach
  - Other Tools
- FabScalar Outreach
  - User data
- FabScalar Based Chips
- FabScalar Evolution
- FabScalar RISC-V
  - Microarchitecture
  - Performance

FabScalar Approach

- Canonical Superscalar Template
  - Defines canonical pipeline stages and their interfaces
- Canonical Pipeline Stage Library (CPSL)
  - Provides many different designs for each canonical pipeline stage
  - Diversity is focused along three key dimensions:
    - *Superscalar Complexity*: Superscalar width, Sizes of stage-specific structures for extracting instruction-level parallelism (ILP)
    - *Sub-pipelining*: Pipeline depth of a canonical stage
    - *Stage-specific design choices*: e.g., different speculation alternatives, recovery alternatives, *etc*.
- Core Generator
  - References CPSL and Template to compose a core of desired configuration


CPSL
Canonical Superscalar Template
Fetch
Rename
Issue
App. 1
core configuration
Core Generator
synthesizable RTL of customized core
Fetch
Decode
Rename
Dispatch
Issue
Register Read
Execute
Writeback
Retire


CPSL
Fetch
Rename
Issue
App. 2
core configuration
Core Generator
synthesizable RTL of customized core
Canonical Superscalar Template
Fetch
Decode
Rename
Dispatch
Issue
Register Read
Execute
Writeback
Retire

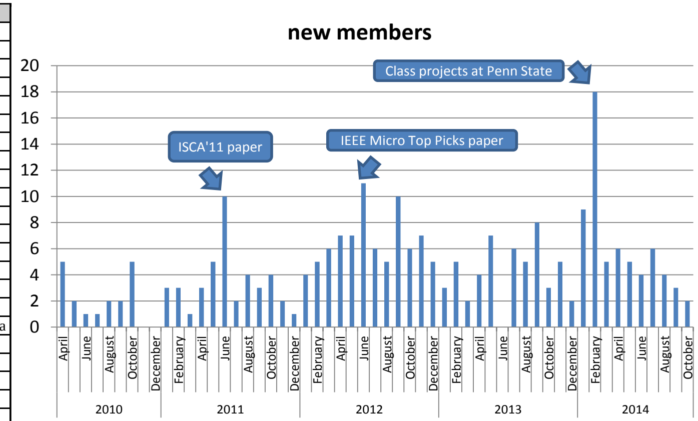
Tools Offered by FabScalar

- FabScalar
  - Template, CPSL, and Core Generator (just described)
- FabMem
  - Support for highly-ported RAMs and CAMs
    - Estimation tool
    - Memory compiler (auto-generate layouts that pass LVS and DRC)
  - Targets FreePDK 45nm
- FabFPGA
  - A version of FabScalar for FPGA prototyping

FabScalar Outreach

U.S. Universities	Int'l Universities	Industry Labs	Countries
UC Santa Cruz (CA)	Ghent University (Belgium)	Global Foundries	Australia
UC San Diego (CA)	Simon Fraser University (Canada)	Intel Labs (2 sites)	Belgium
Northwestern University (IL)	Tsinghua University (China)	Synopsis	Brazil
UIUC (IL)	TU Darmstadt (Germany)	Calxeda	Canada
Harvard University (MA)	Alexander Tech. Educ. Institute of Thessaloniki (Greece)	IBM	China
NCSU (NC)	IIT Delhi (India)		Denmark
Cornell University (NY)	IIT Madras (India)		France
Univ. of Rochester (NY)	Politecnico di Milano (Italy)		Germany
Drexel University (PA)	Mei University (Japan)		Greece
UT Austin (TX)	National University of Singapore (Singapore)		India
UT Dallas (TX)	KAIST (South Korea)		Iran
Univ. of Virginia (VA)	Barcelona Supercomputing Center (Spain)		Israel
Virginia Tech (VA)	Cambridge University (UK)		Italy
UW Madison (WI)	ABV-IIITM (India)		Japan
SUNY Binghamton (NY)	Bilkent University (Turkey)		Norway
Utah State University (UT)	DA-IICT (India)		Singapore
Columbia University (NY)	Karlsruhe Institute of Technology (Germany)		South Korea
Stanford University (CA)	Wuhan University (China)		Spain
Univ. of Maine (ME)	Chalmers University (Sweden)		Sweden
USC (CA)	SouthEast University (China)		Turkey
UC Riverside (CA)	Univ. of Tehran (Iran)		UK
CMU (PA)	Tel Aviv University (Israel)		USA
Georgia Tech (GA)	Chinese Academy of Sciences (China)		
UC Irvine (CA)	Yonsei University (South Korea)		
Univ. of Michigan (MI)	University of Augsburg (Germany)		
Duke University (NC)	Federal University of Mato Grosso do Sul (Brazil)		
Arizona State University (AZ)	Hunan University (China)		
NYU Polytechnic (NY)	State Key Laboratory of High Perf. Computing (China)		
Univ. of Central Florida (FL)	Zhejiang University (China)		
Univ. of Chicago (IL)	Univ. of British Columbia (Canada)		
Penn State University (PA)	IIT Bombay (India)		
Univ. of Minnesota (MN)	IIIT (India)		
Stony Brook University (NY)	Univ. of Waterloo (Canada)		
	Univ. of Victoria (Canada)		
	Univ. of Campinas (Brazil)		
	NTNU - Norwegian Univ. of Science & Technology (Norway)		
	Federal University of Santa Catarina (Brazil)		
	University of Tokyo (Japan)		
	ENS Rennes / IRISA (France)		
	Nagoya University (Japan)		
	Politecnico di Torino (Italy)		
	Islamic Azad University (Iran)		
	Technical University of Denmark (Denmark)		
	The University of New South Wales (Australia)		
	Pontifícia Universidade Católica do Rio grande do Sul / PUCRS (Brazil)		

(a) Affiliations.



(b) New members over time.

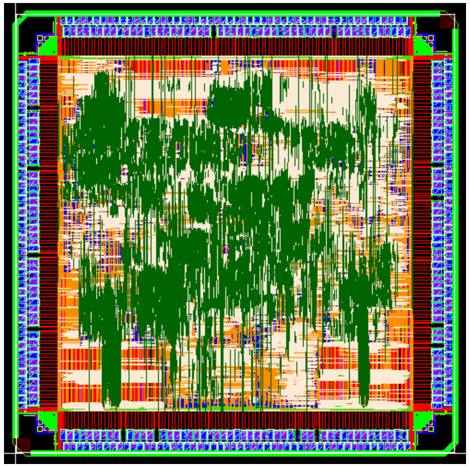
# topics	98
# posts to topics	412
average posts/topic	4.2
# views of topics	2,983
average views/topic	30

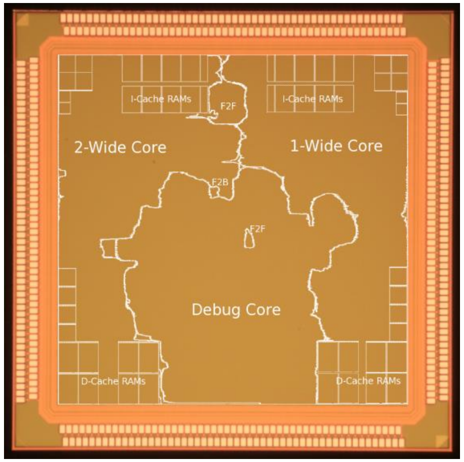
(c) Google group activity.

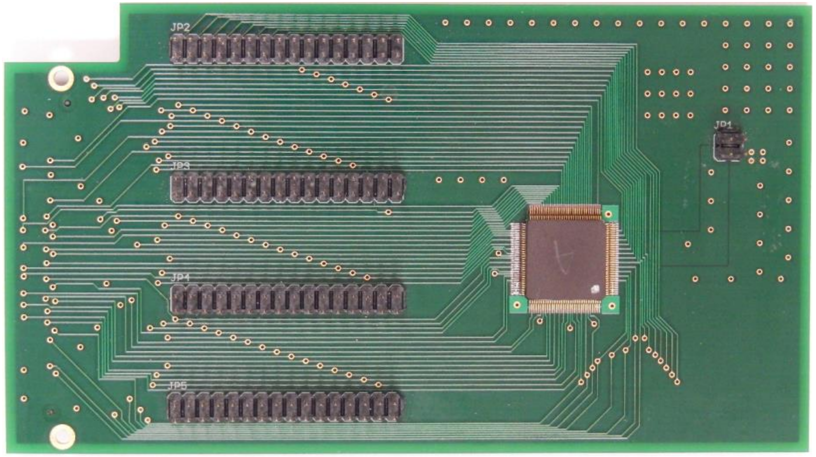
User data through October 2014.

FabScalar Based Chips at NC State

- H3 (“Heterogeneity in 3D”)
    - Two cores with different microarchitectures
    - Hardware support for fast thread migration





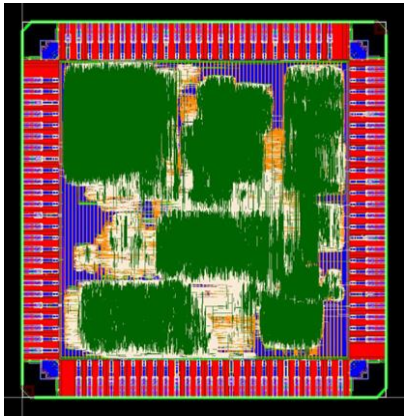


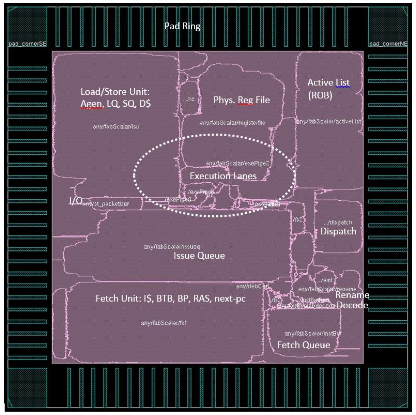
[5] Rationale for a 3D Heterogeneous Multi-core Processor, ICCD 2013. (post-tapeout, pre-silicon)

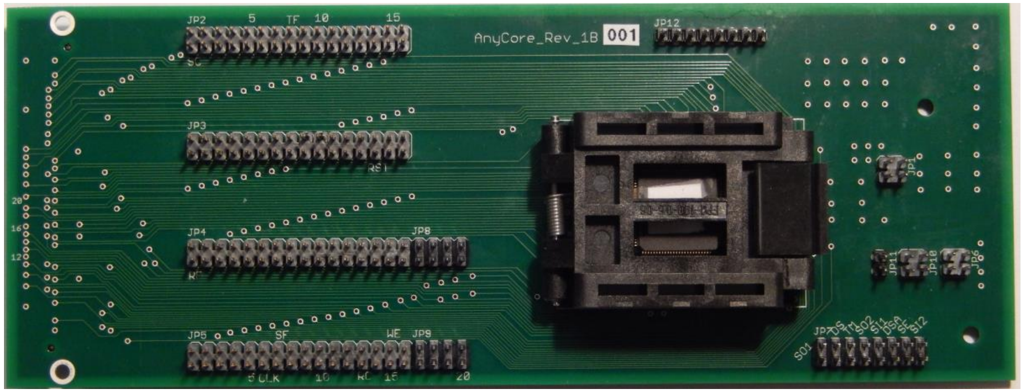
[6] Experiences With Two FabScalar-based Chips, WARP 2015. (post-silicon)

FabScalar Based Chips at NC State

- AnyCore
  - One core with reconfigurable microarchitecture
  - Adapts to workload to improve efficiency



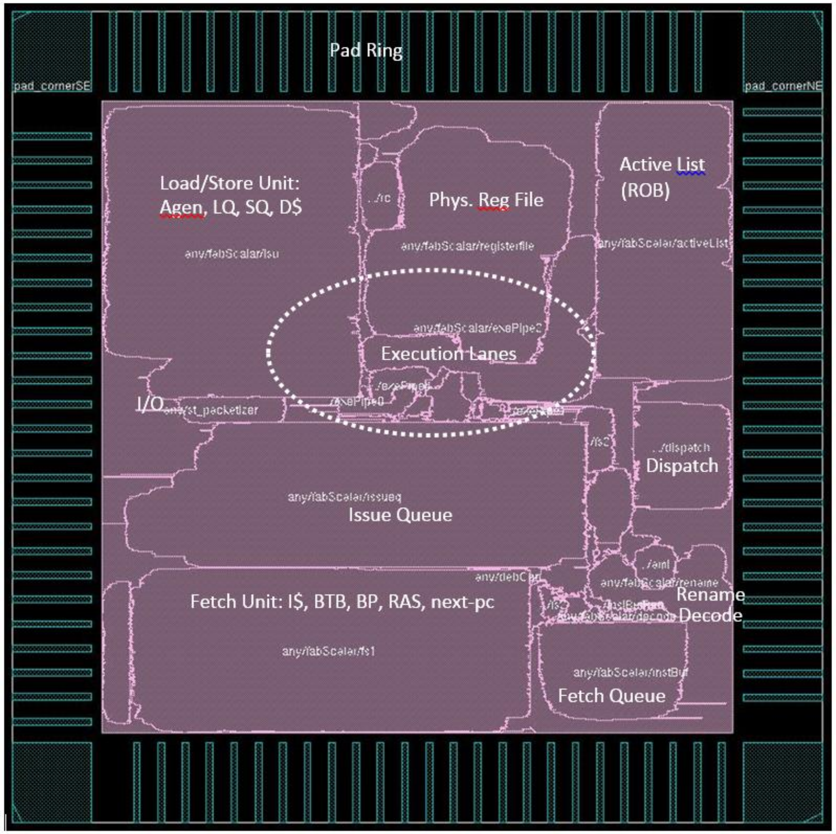




[6] Experiences With Two FabScalar-based Chips, WARP 2015.

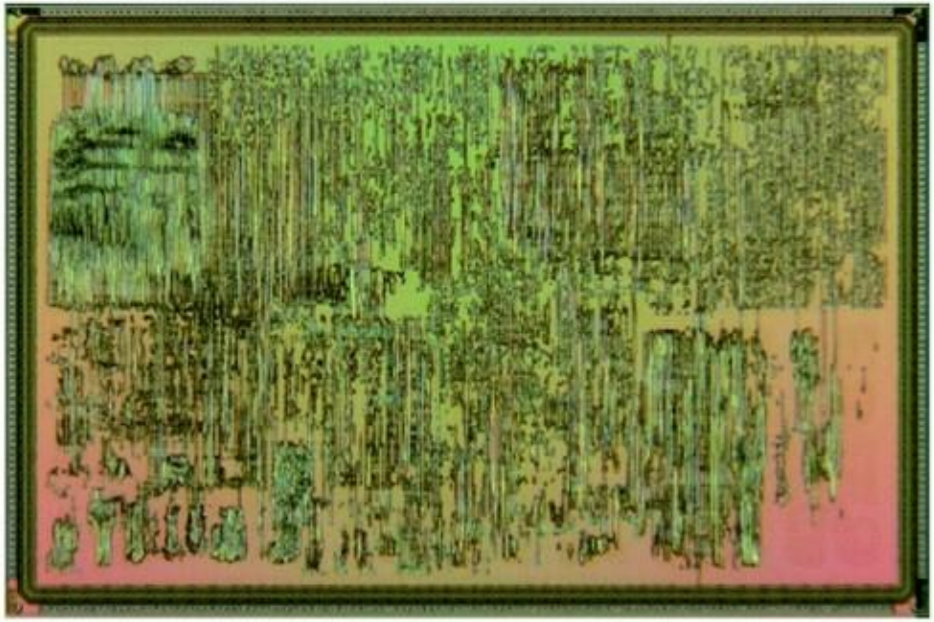
AnyCore Zoomed-in

Adaptive microarchitecture feature	Configurations
fetch/dispatch width (instructions/cycle)	1, 2, 3, 4
issue width (instructions/cycle)	3, 4, 5
physical register file & active list	64, 96, 128
load and store queues (each)	16, 32
issue queue	16, 32, 48, 64



Non-NCSU FabScalar Based Chips

- Mei University, Japan fabricated a FabScalar MIPS32 based chip
  - Coprocessor 0
  - L1 Caches
  - AMBA based system bus

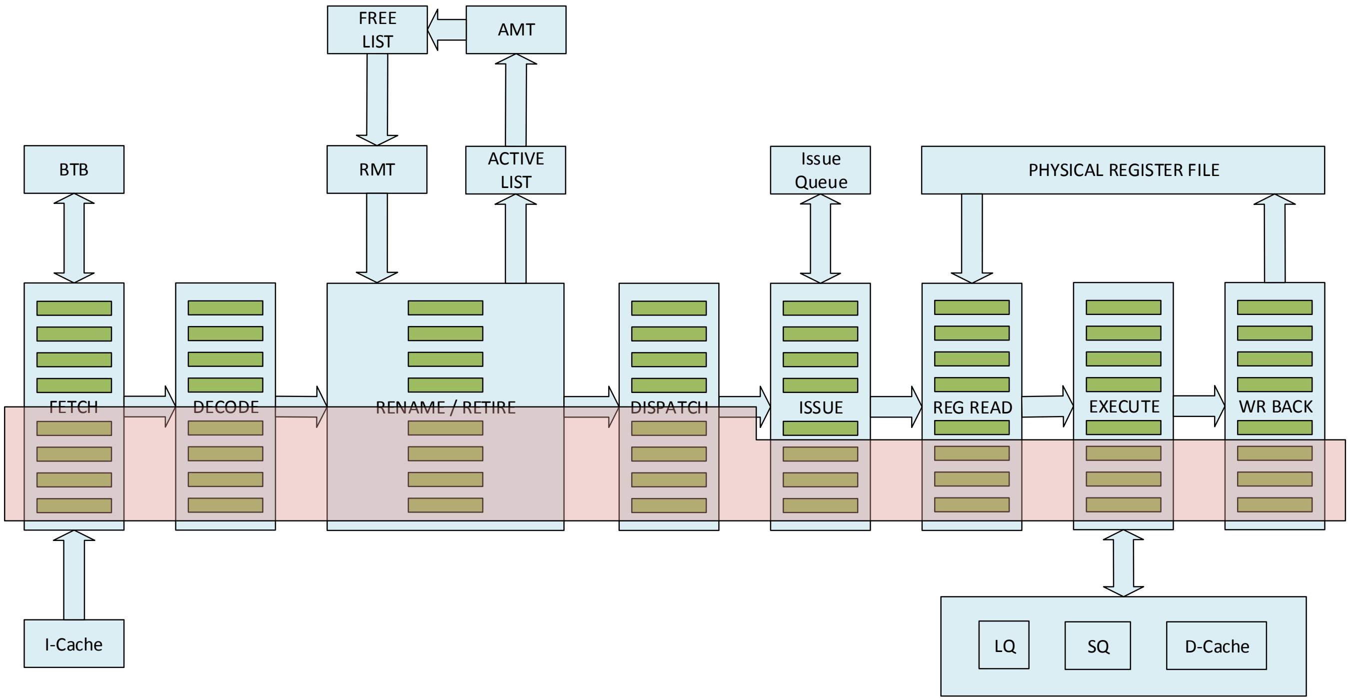


FabScalar Evolution

Problem	Solution
CPSL approach requires making changes in each stage variant, or modifying scripts that generate CPSL.	**Superset Core**: A single parameterized System Verilog description. - Structure sizes already parameterized - Parameterized widths and sub-pipelining
No multi-core / SoC support	**FabCache, FabBus:** Prof. T. Sasaki @ Mei Univ. - Generate diverse cache hierarchies [7] - Generate buses for multi-core and accelerator support [8] (AMBA protocol)
PISA (SimpleScalar) ISA: - No privileged ISA. - No software ecosystem (old gcc, no linux)	**FabScalar-MIPS ports:** - FabScalar-MIPS32 + Co-processor 0 (MMU) + Linux (Prof. T. Sasaki @ Mei Univ.) - FabScalar-MIPS64 + Co-processor 1 (FPU)
MIPS ISA: - Proprietary ISA: Concerned about releasing FabScalar-MIPS Superset Core - OOO compatibility: Has frustrating ISA features (delay slots, conditional moves)	**FabScalar-RISC-V:** - Open ISA - No frustrating features w.r.t. OOO implementation - Privileged ISA - Software ecosystem

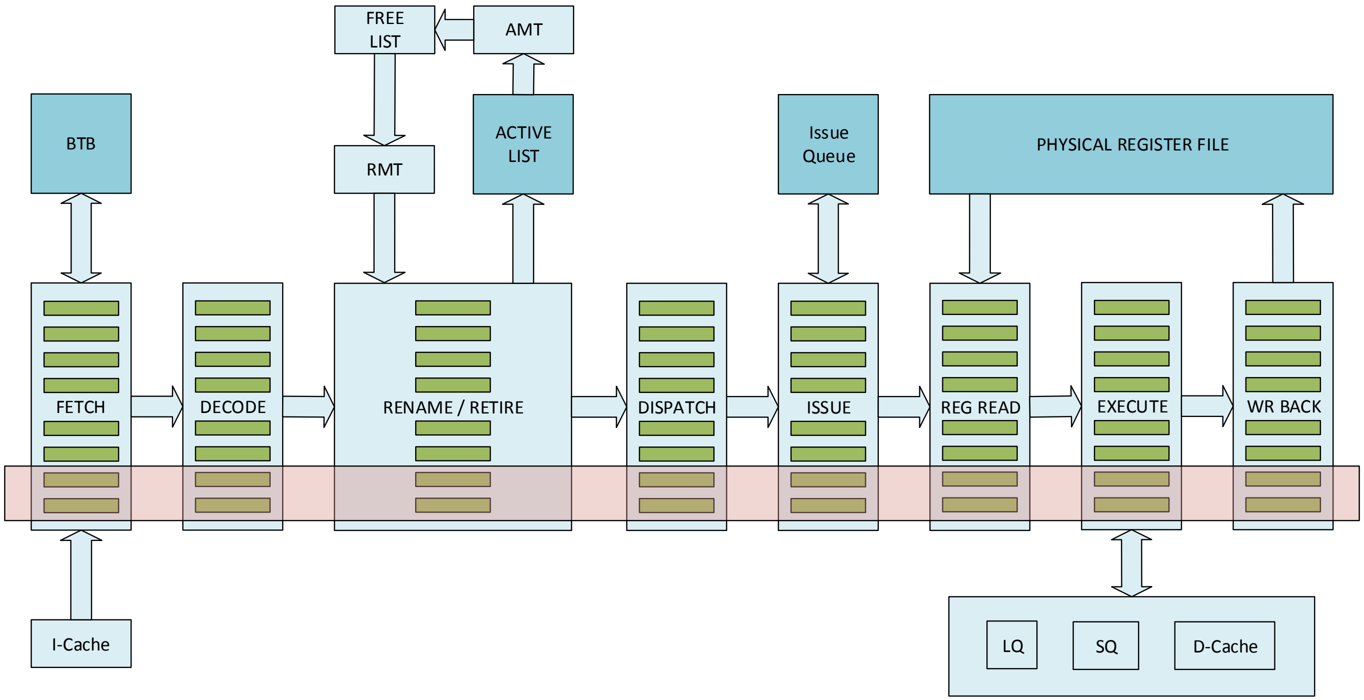
FabScalar Superset Core

`define FETCH_FOUR_WIDE	`define ISSUE_TWO_DEEP
`define ISSUE_THREE_WIDE	`define RR_TWO_DEEP



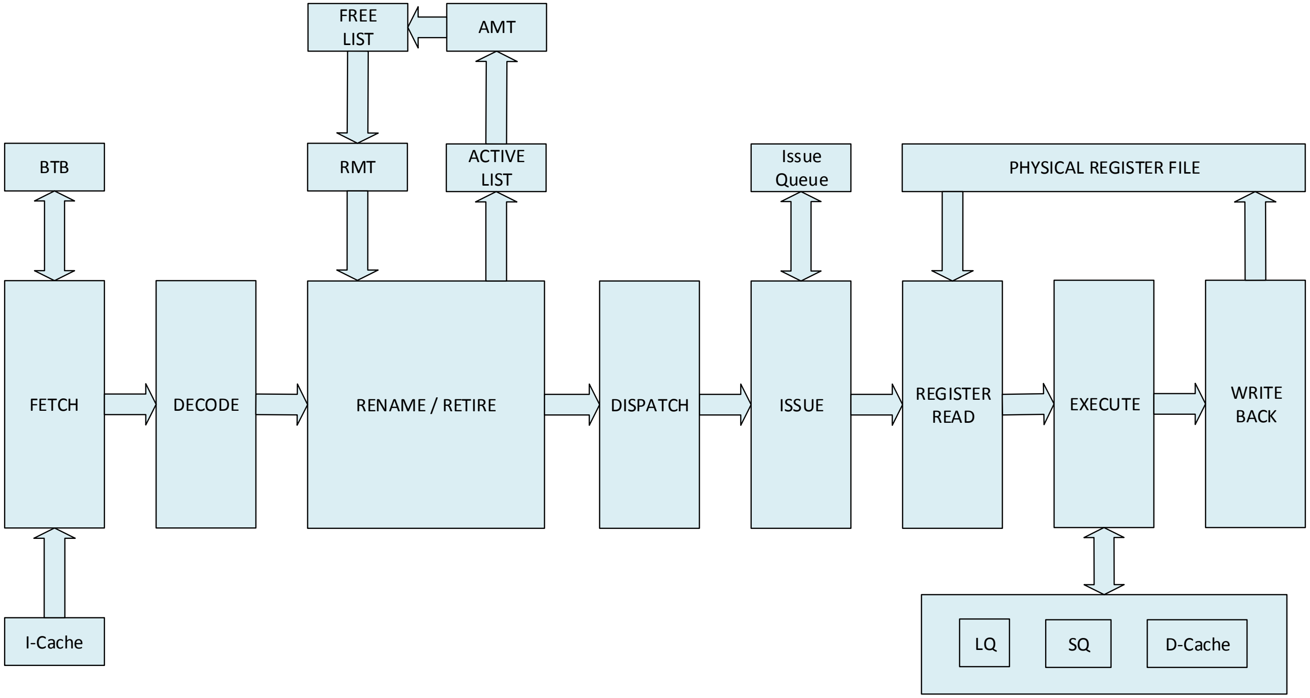
FabScalar Superset Core

```
`define FETCH_TWO_WIDE
`define ISSUE_TWO_WIDE
`define SIZE_PRF  128
`define SIZE_BTB  2048
`define SIZE_ACTIVE_LIST  128
`define SIZE_IQ   64
```



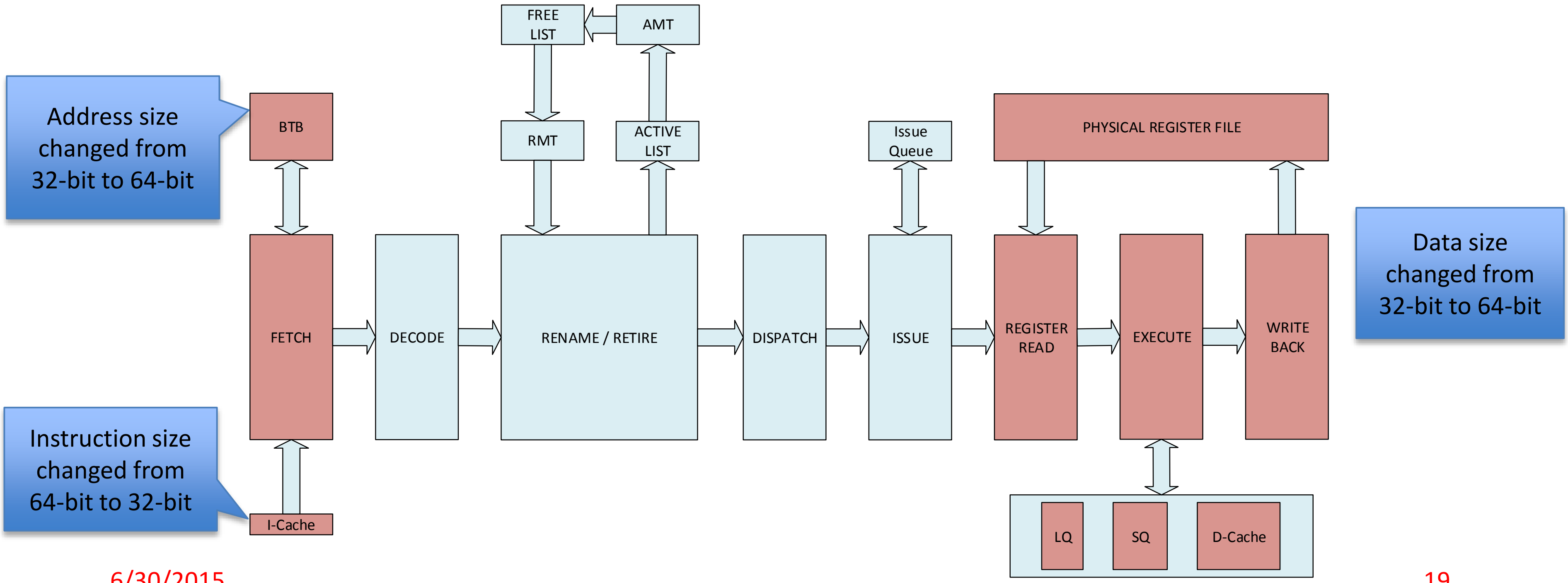
Changes for RISC-V port

- Starting point was PISA Superset Core (64-bit instructions, 32-bit address and data)
  - RISC-V 64-bit has 32-bit instructions and 64-bit data



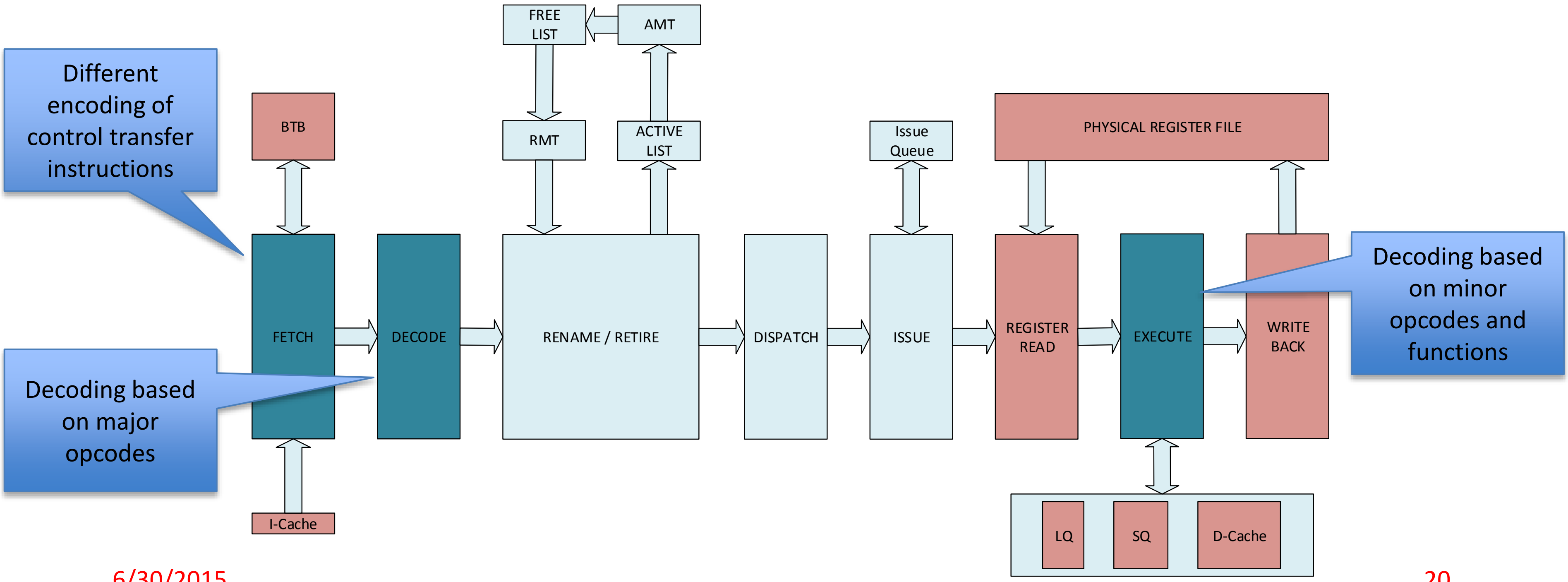
Changes for RISC-V port

- Starting point was PISA Superset Core (64-bit instructions, 32-bit address and data)
  - RISC-V 64-bit has 32-bit instructions and 64-bit data



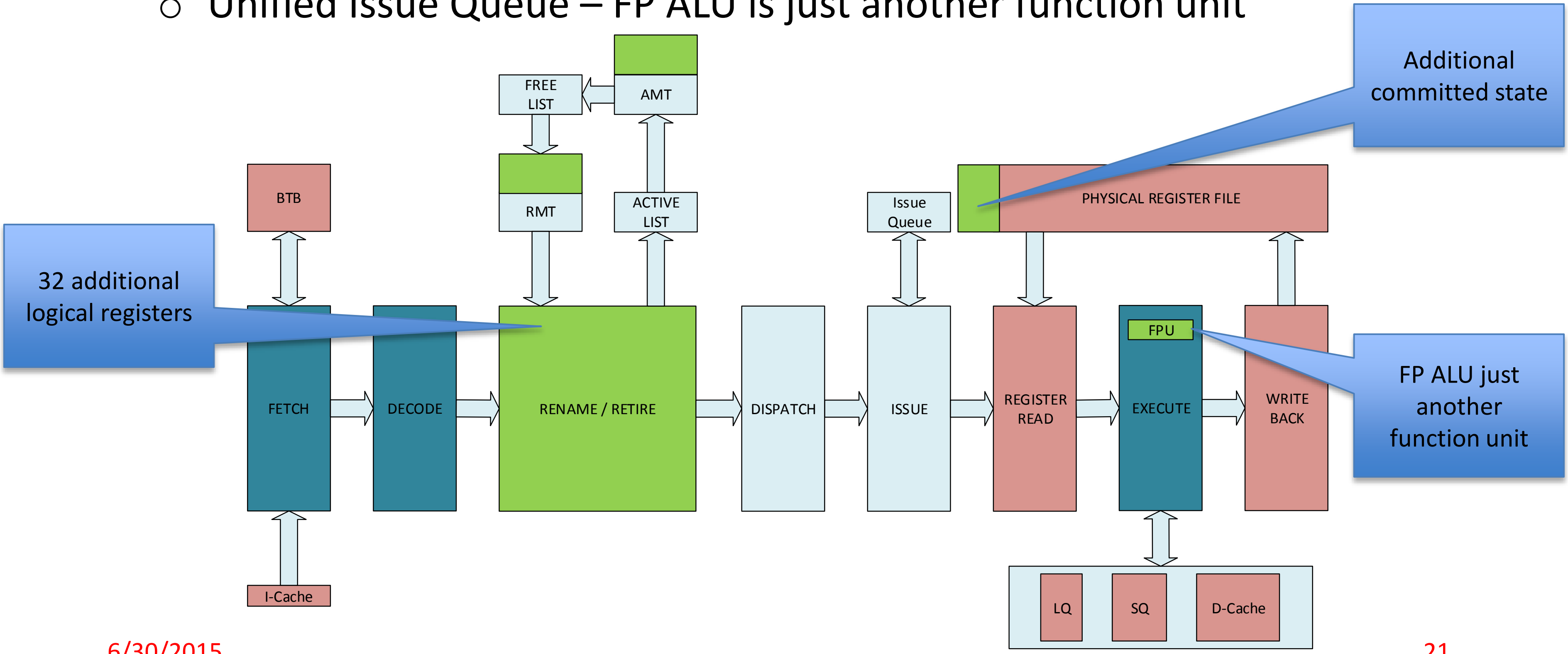
Changes for RISC-V port

- RISC-V very similar to PISA (no delay slots, no conditional moves, *etc.*)
  - RISC-V specific changes mostly in Fetch, Decode, and Execute



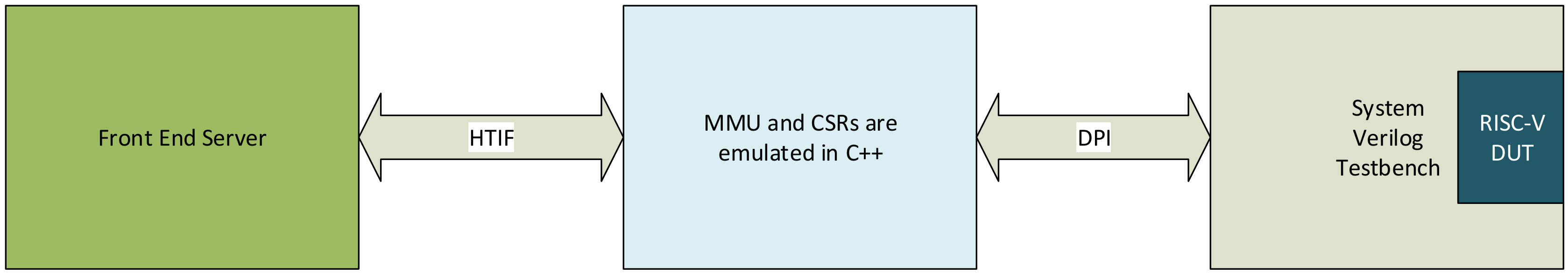
Changes for RISC-V port

- 64-bit for both INT and FP makes adding FP straightforward
  - Unified Physical Register File
  - Unified Issue Queue – FP ALU is just another function unit



FabScalar RISC-V Test Harness

- MMU and CSRs are currently implemented in C++
  - Accessed through System Verilog DPI
  - Will be replaced with RTL implementations
- The C++ part communicates with the Front End Server through HTIF



Basic Performance Evaluation, 4-wide Superscalar Configuration

Array Reduction

```
for(i=0;i<20000;i++){
  temp = a[i];
  sum = sum + 3;
  sum = sum + 4;
  sum = sum + 5;
  sum1 = sum1 + temp;
  sum2 = sum2 + temp;
 }
```

Assembly

```
1016c: lw  a5,0(a6)
10170: addi a2,a2,3
10174: addi a2,a2,4
10178: addi a2,a2,5
1017c: addw a3,a3,a5
10180: addw a4,a4,a5
10184: addi a6,a6,4
10188:  bne a6,a1,1016c <main+0x34>
```

IPC = 3.7

FabScalar RISC-V Offerings

- FabScalar RISC-V: An open-source tool
  - Parameterized OOO superscalar implementation of RV64G
  - Complete with uncore components
  - Verification infrastructure
- CAD flow for easy synthesis and place-and-route
- A C++ timing simulator for performance studies
- FabScalar RISC-V will be available on GitHub in Fall
  - Users can commit improvements
  - Users can "cherry-pick" specific changes and bug fixes

Future Work

- Implement privileged ISA to boot Linux on FabScalar cores
- Untether FabScalar cores (Do not use HTIF)
- Add testcases to stress different design features
- Port FabFPGA to RISC-V

References

1. N. K. Choudhary, S. V. Wadhavkar, T. A. Shah, H. Mayukh, J. Gandhi, B. H. Dwiel, S. Navada, H. H. Najaf-abadi, and E. Rotenberg. FabScalar: Composing Synthesizable RTL Designs of Arbitrary Cores within a Canonical Superscalar Template. *38th IEEE/ACM International Symposium on Computer Architecture*, pp. 11-22, June 2011.
2. N. K. Choudhary, S. V. Wadhavkar, T. A. Shah, H. Mayukh, J. Gandhi, B. H. Dwiel, S. Navada, H. H. Najaf-abadi, and E. Rotenberg. FabScalar: Automating Superscalar Core Design. *IEEE Micro, Special Issue: Micro's Top Picks from the Computer Architecture Conferences*, 32(3):48-59, May-June 2012.
3. Niket K. Choudhary et al. "FabScalar", in the *Workshop on Architecture Research Prototyping (WARP)*, in conjunction with ISCA-36, 2009.
4. B. H. Dwiel, N. K. Choudhary, and E. Rotenberg. FPGA Modeling of Diverse Superscalar Processors. *2012 IEEE International Symposium on Performance Analysis of Systems and Software*, pp. 188-199, April 2012.
5. E. Rotenberg, B. H. Dwiel, E. Forbes, Z. Zhang, R. Widialaksono, R. Basu Roy Chowdhury, N. Tshibangu, S. Lipa, W. R. Davis, and P. D. Franzon. Rationale for a 3D Heterogeneous Multi-core Processor. Proceedings of the *31st IEEE International Conference on Computer Design (ICCD-31)*, pp. 154-168, October 2013.
6. E. Forbes, R. Basu Roy Chowdhury, B. Dwiel, A. Kannepalli, V. Srinivasan, Z. Zhang, R. Widialaksono, T. Belanger, S. Lipa, E. Rotenberg, W. R. Davis, and P. D. Franzon. Experiences with Two FabScalar-based Chips. *6th Workshop on Architectural Research Prototyping (WARP-6)*, June 14, 2015.
7. T. Okamoto, T. Nakabayashi, T. Sasaki, T. Kondo. FabCache: Cache Design Automation for Heterogeneous Multi-core Processors. *First International Symposium on Computing and Networking (CANDAR),* Dec. 2013
8. Takaki Okamoto, Tomoyuki Nakabayashi, Takahiro Sasaki, Toshio Kondo. Detail Design and Evaluation of Fab Cache. *2014 Second International Symposium on Computing and Networking (CANDAR)*
9. Y. Seto, T. Nakabayashi, T. Sasaki, and T. Kondo. FabBus: A Bus Framework for Heterogeneous Multi-core processor. *28th International Technical Conferench on Circuits/Systems, Computers and Communications (ITC-CSCC2013)*, July 2013.