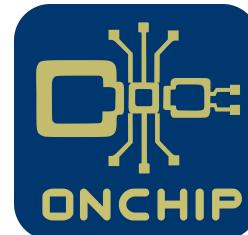


A Verified RISC-V I Based Processor with an External Debugging Capability

Hanssel Morales, Elkim Roa

Integrated Systems Research Group – OnChip
Universidad Industrial de Santander, Bucaramanga - Colombia



Outline

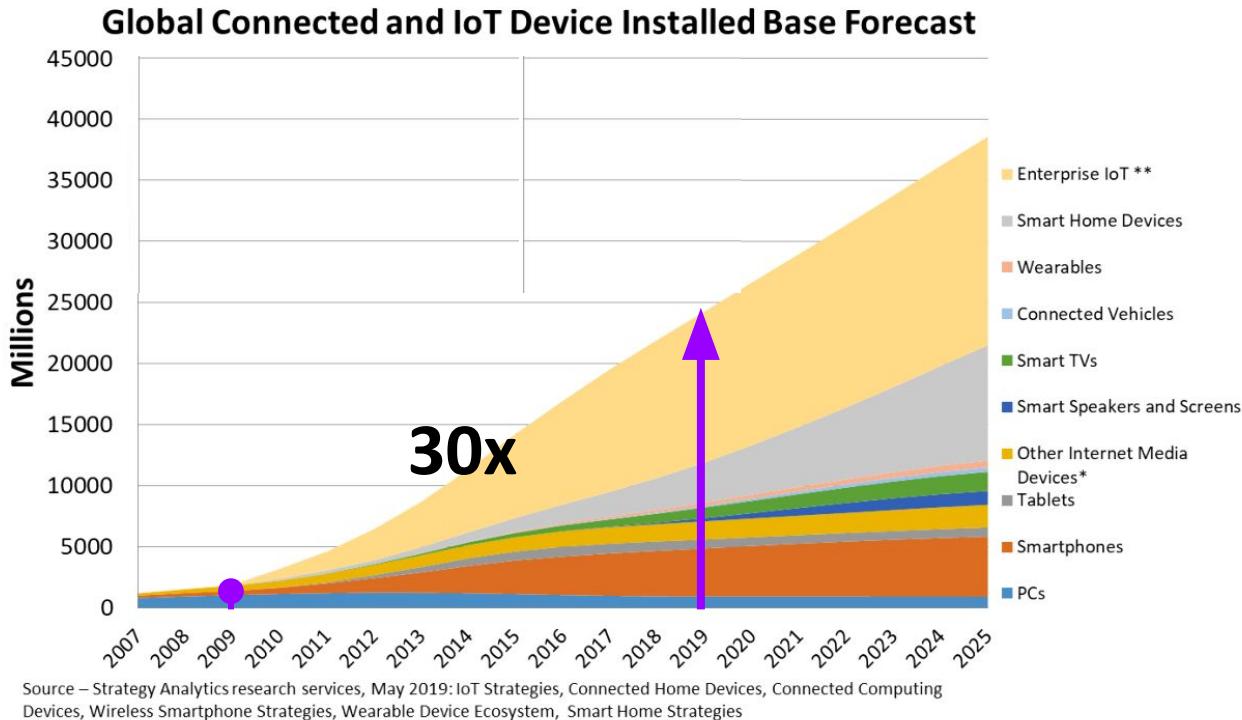
- Motivation
- Core Microarchitecture
- Physical Design
- Verification & FPGA Prototyping
- Further Contributions
- Conclusions

Outline

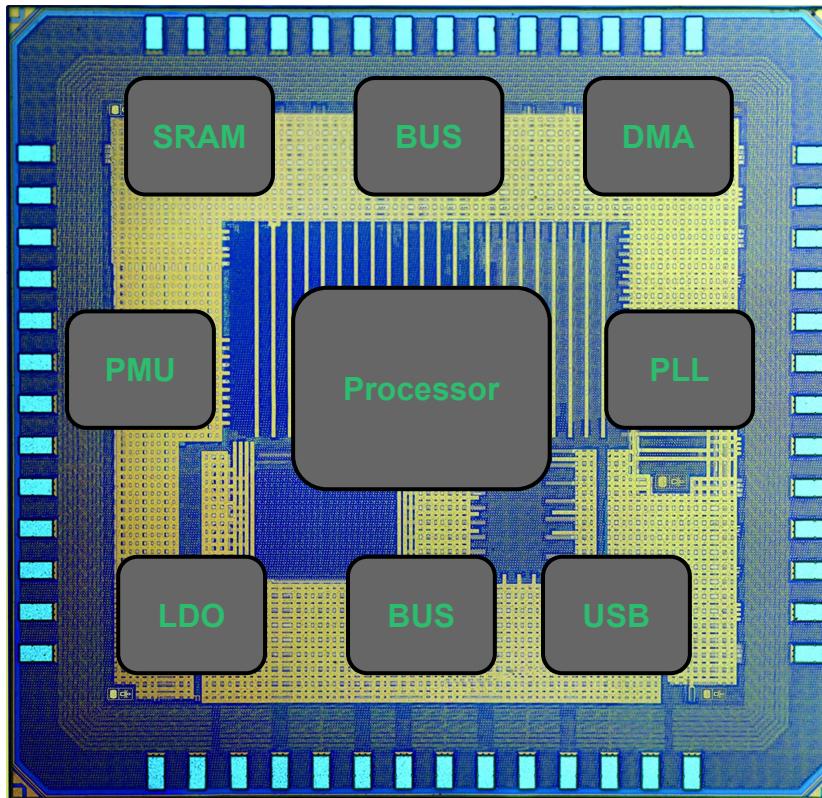
- Motivation
- Core Microarchitecture
- Physical Design
- Verification & FPGA Prototyping
- Further Contributions
- Conclusions

SoC Devices Demand Grows

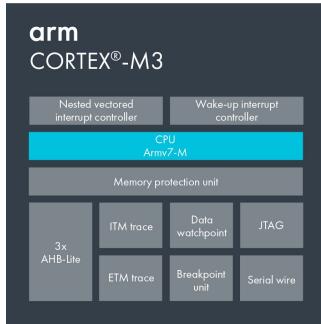
STRATEGY ANALYTICS



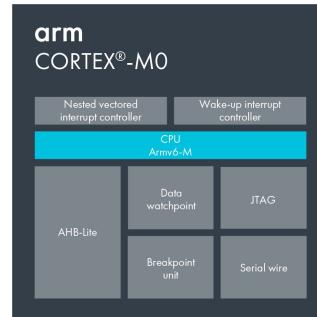
Processor Role in SoC



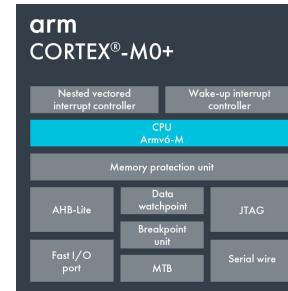
Power Consumption Reduction



141 $\mu\text{W}/\text{MHz}$



66 $\mu\text{W}/\text{MHz}$

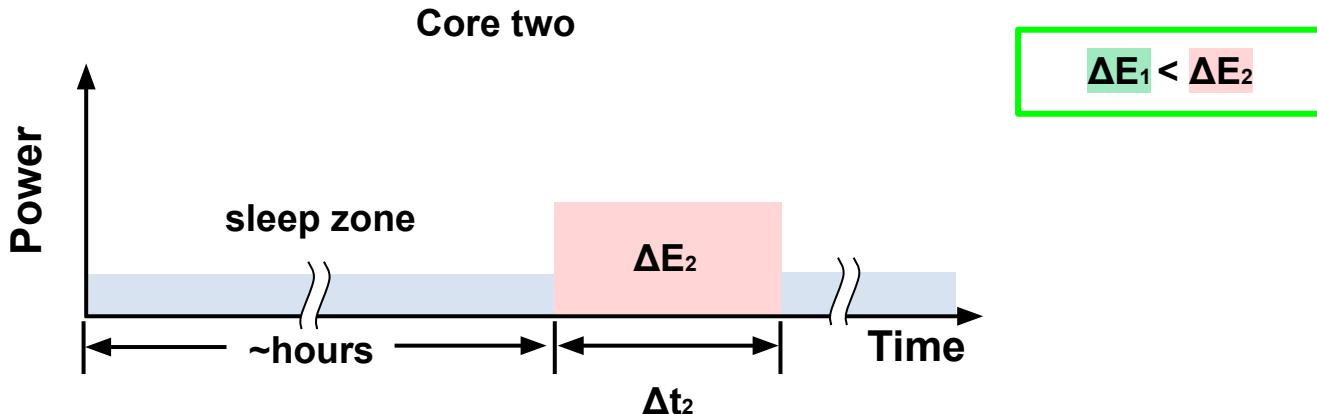
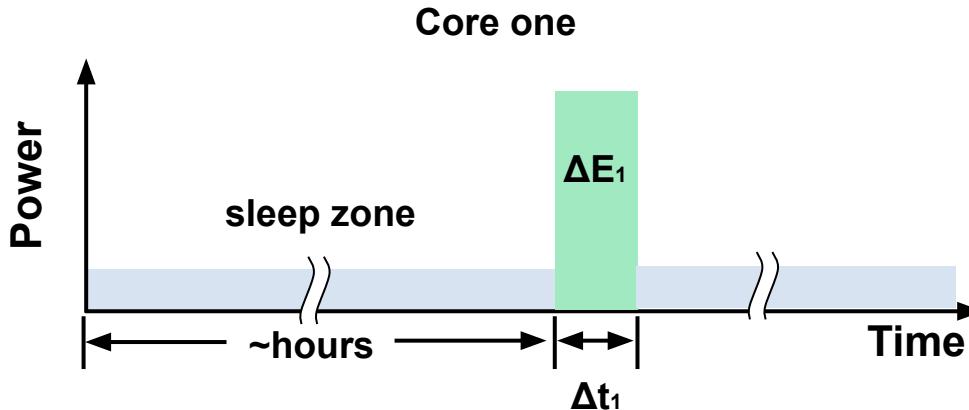


47 $\mu\text{W}/\text{MHz}$

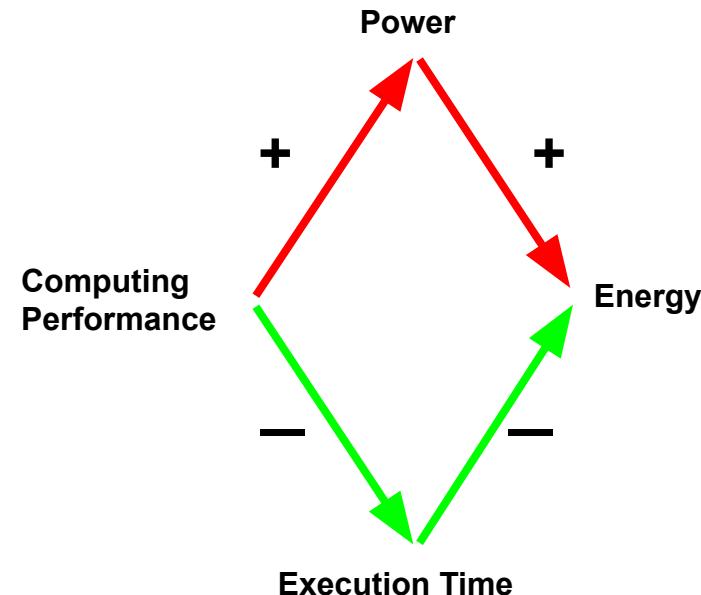
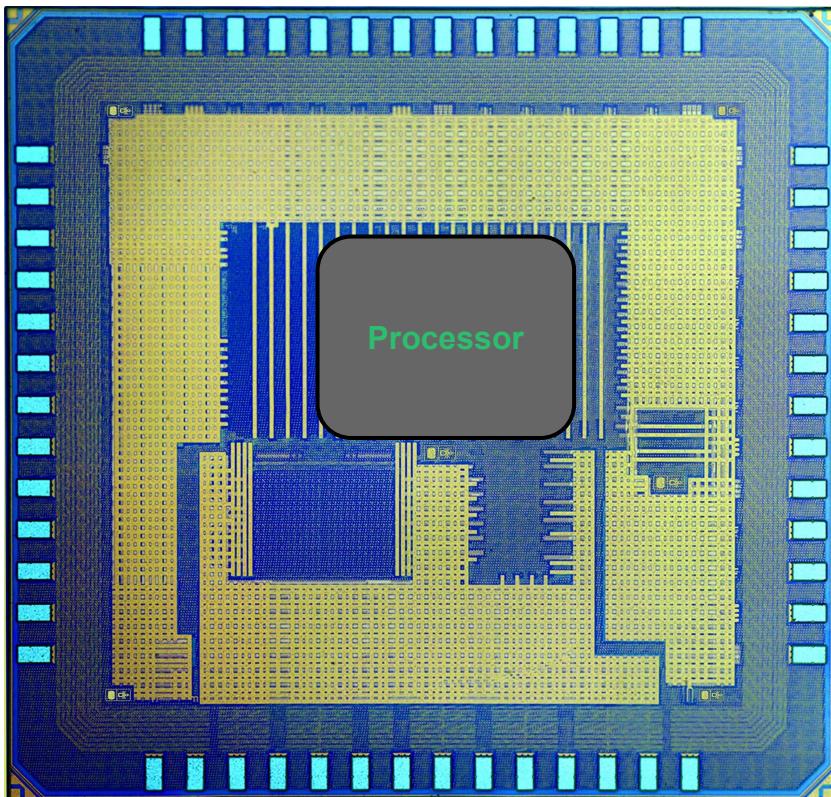
[2] Slow and steady wins the race? a comparison of ultra-low-power risc-v cores for internet-of-things applications 2017(PATMOS)

[3] ARM, "Cortex-m specs," Arm Developer, 2021.

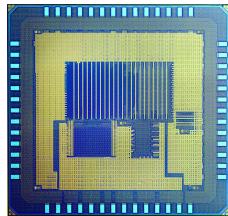
Energy Consumption - IoT Processor



Energy Tradeoff



General Goal



To design and to verify a RISC-V I based processor with
external debug capability

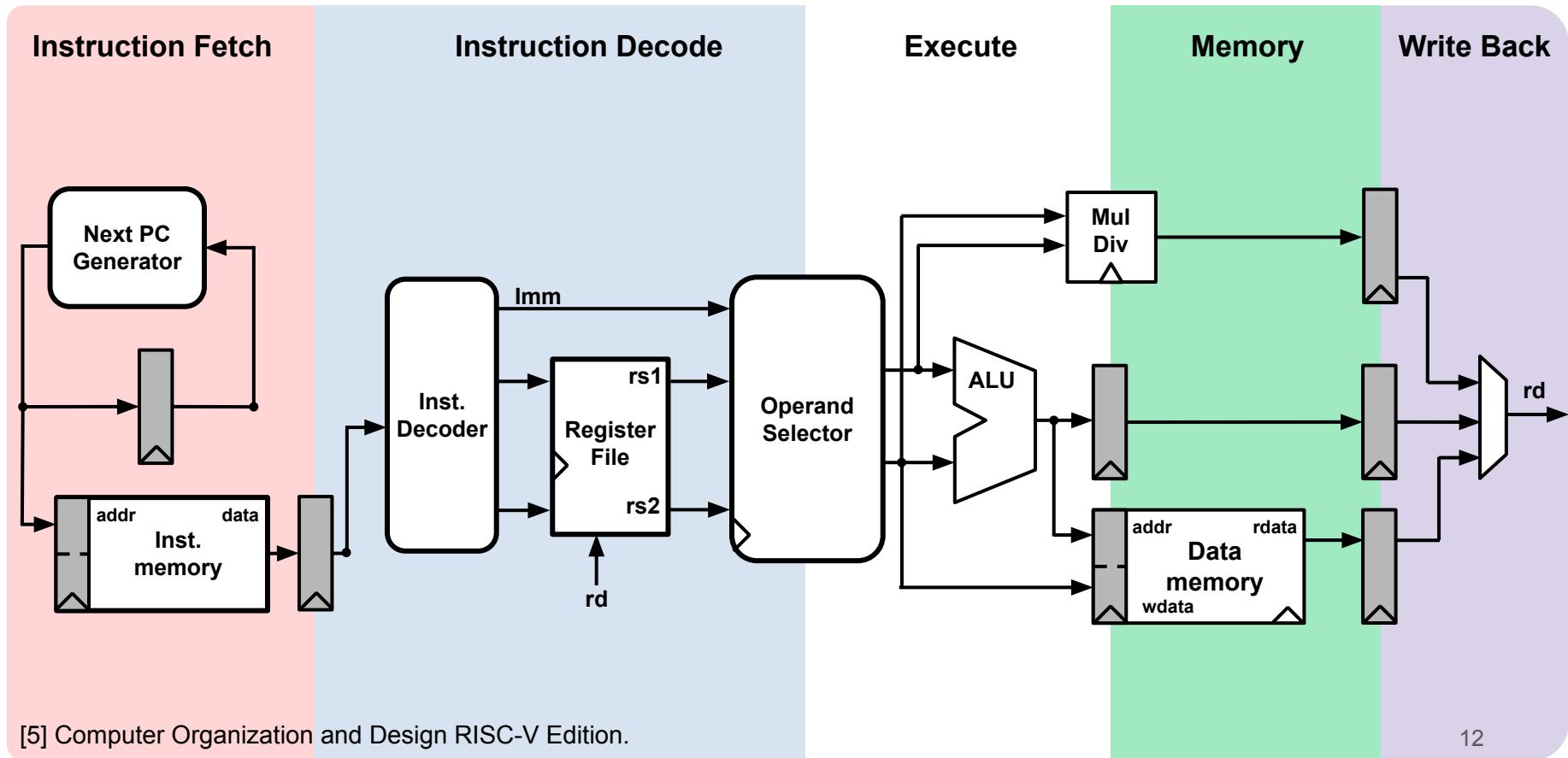
Outline

- Motivation
- Core Microarchitecture
- Physical Design
- Verification & FPGA Prototyping
- Further Contributions
- Conclusions

IoT Market RISC-V ISA Adoption

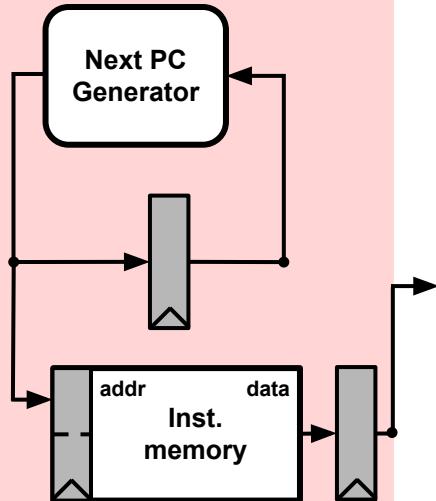


Implemented Microarchitecture

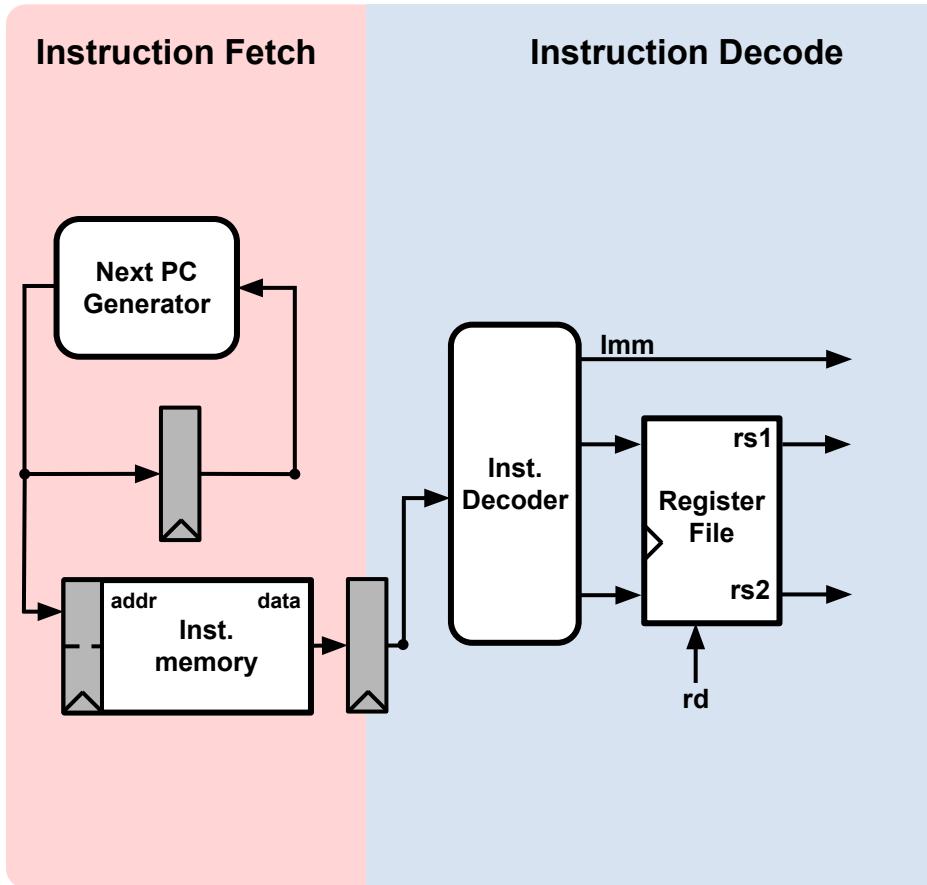


Implemented Microarchitecture

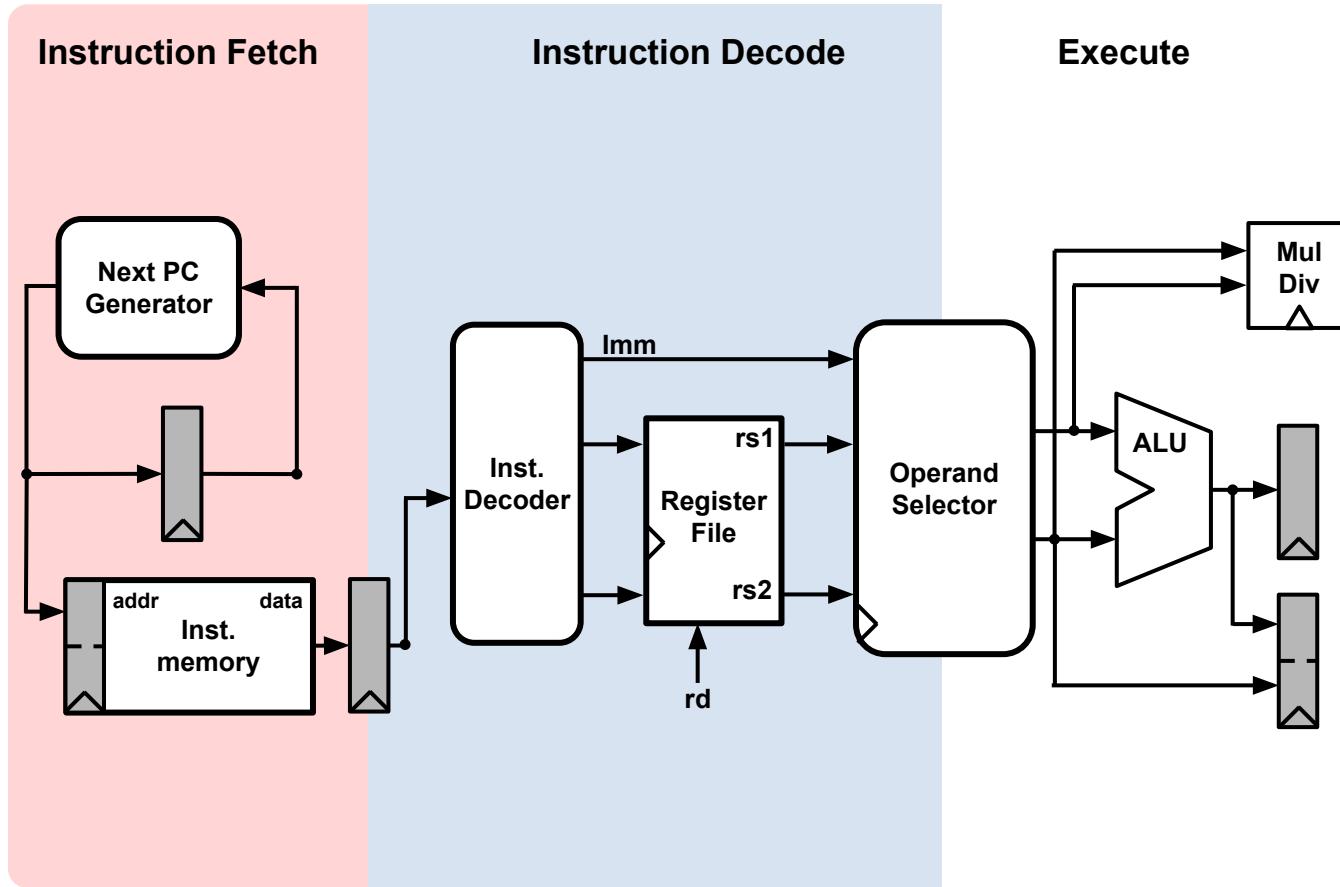
Instruction Fetch



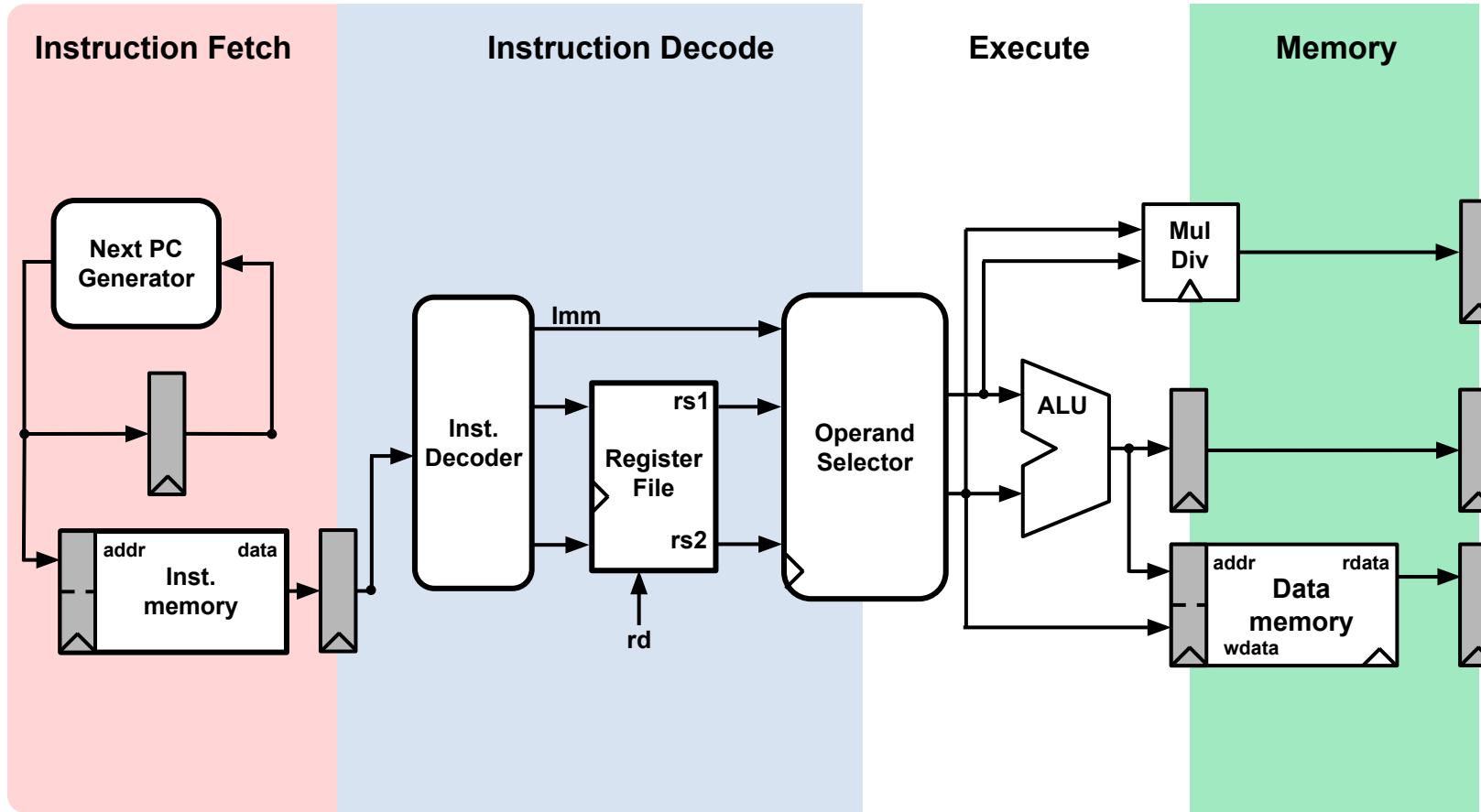
Implemented Microarchitecture



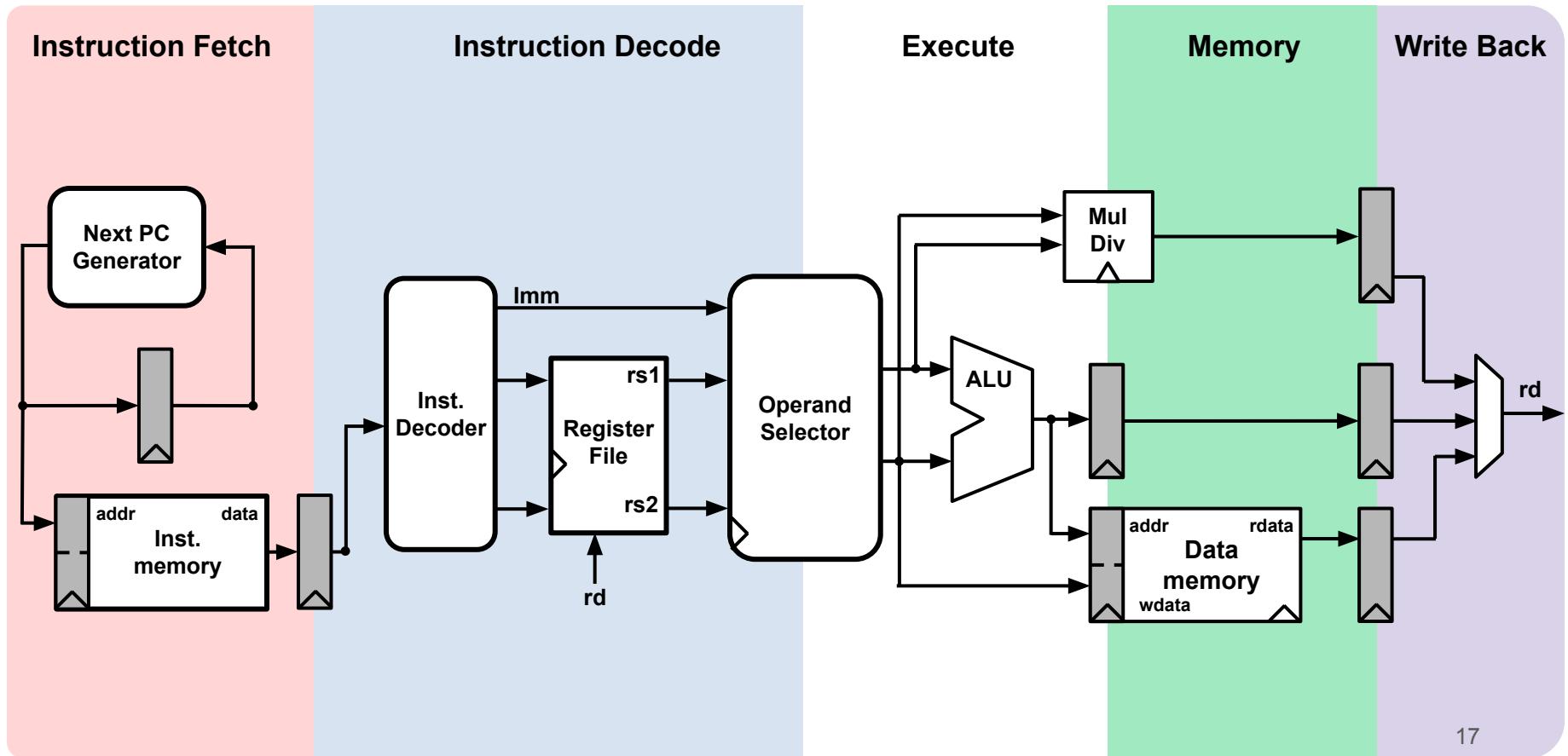
Implemented Microarchitecture



Implemented Microarchitecture



Implemented Microarchitecture



IPC Penalties

Types	Worst Case Cycles Stalled
Dependency between instructions	3
Jumps & Branches	3
Load / Store	It depends on Bus Latency

-Computing Performance



+Energy

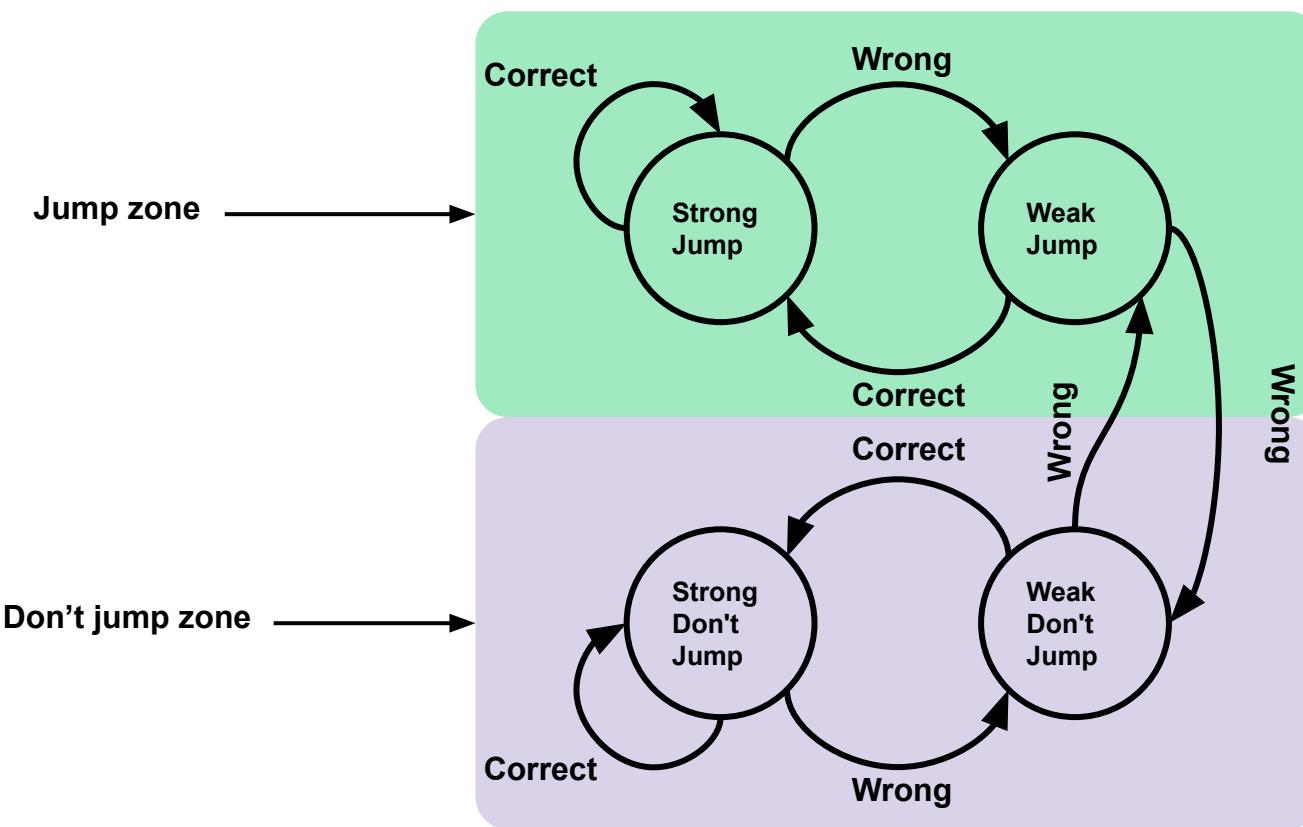
Jump and Branch Penalty

Instruction Fetch	Instruction Decode	Execute	Memory	Write Back
addi t1, t2, 1;	bubble	bubble	bubble	blt t1, t2, 400;
PC=400	PC=?	PC=?	PC=?	PC=4

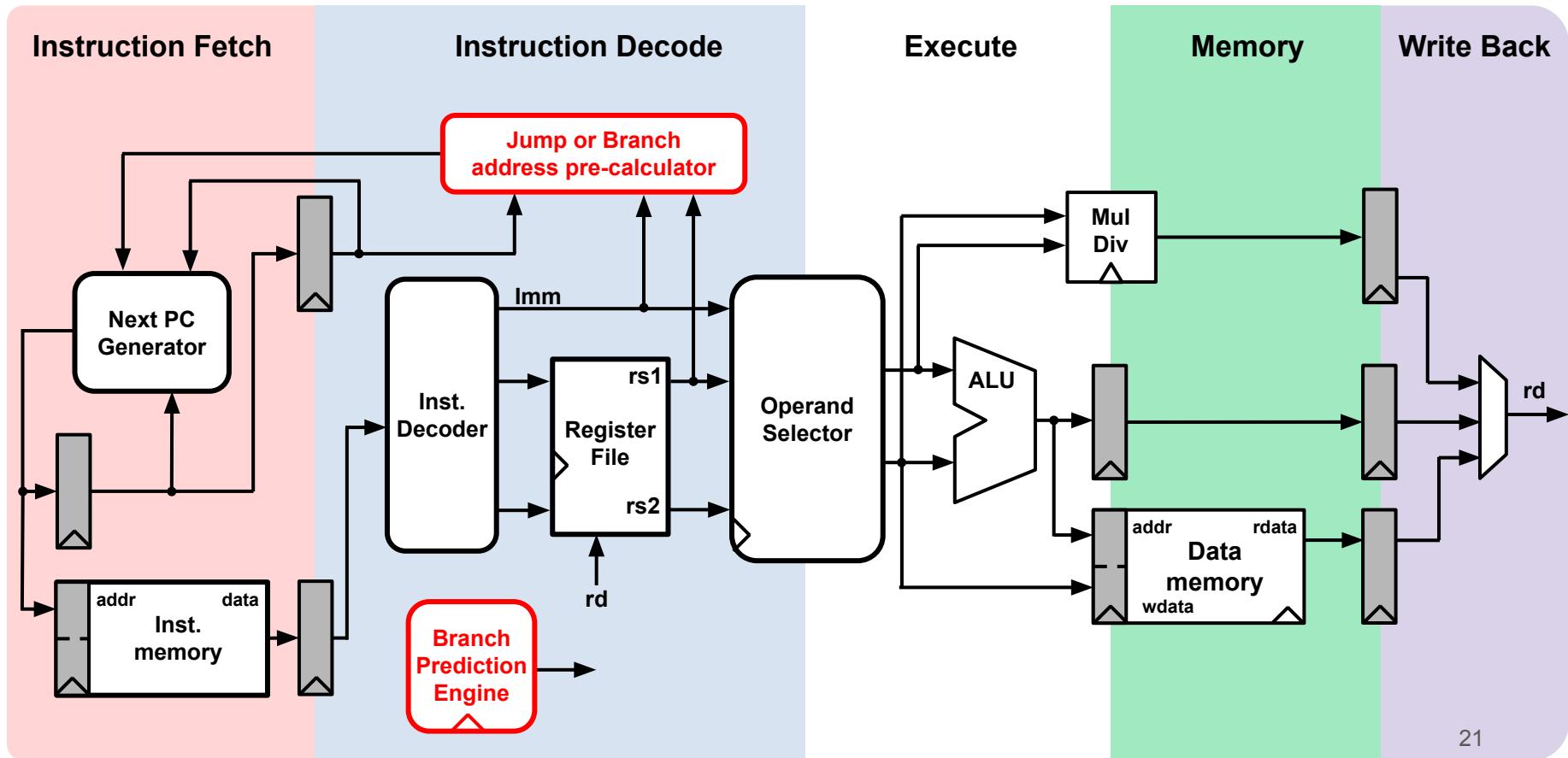
```
if (t1<t2)
{
    NEXT_PC = PC+396
}
else
{
    NEXT_PC = PC +4
}
```

PC=4

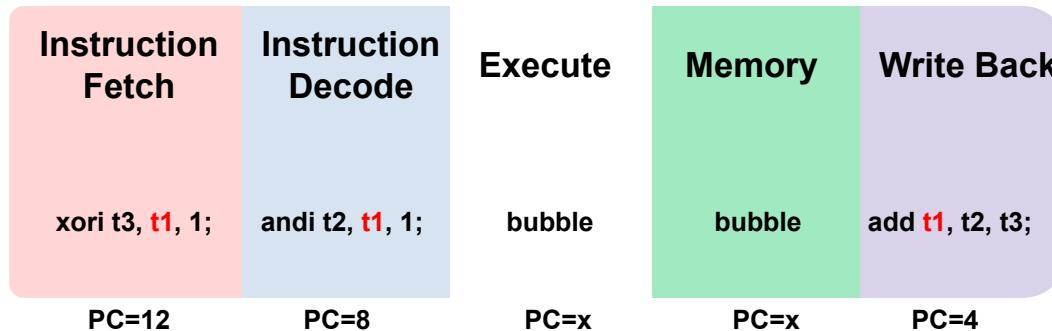
Branch Prediction History FSM



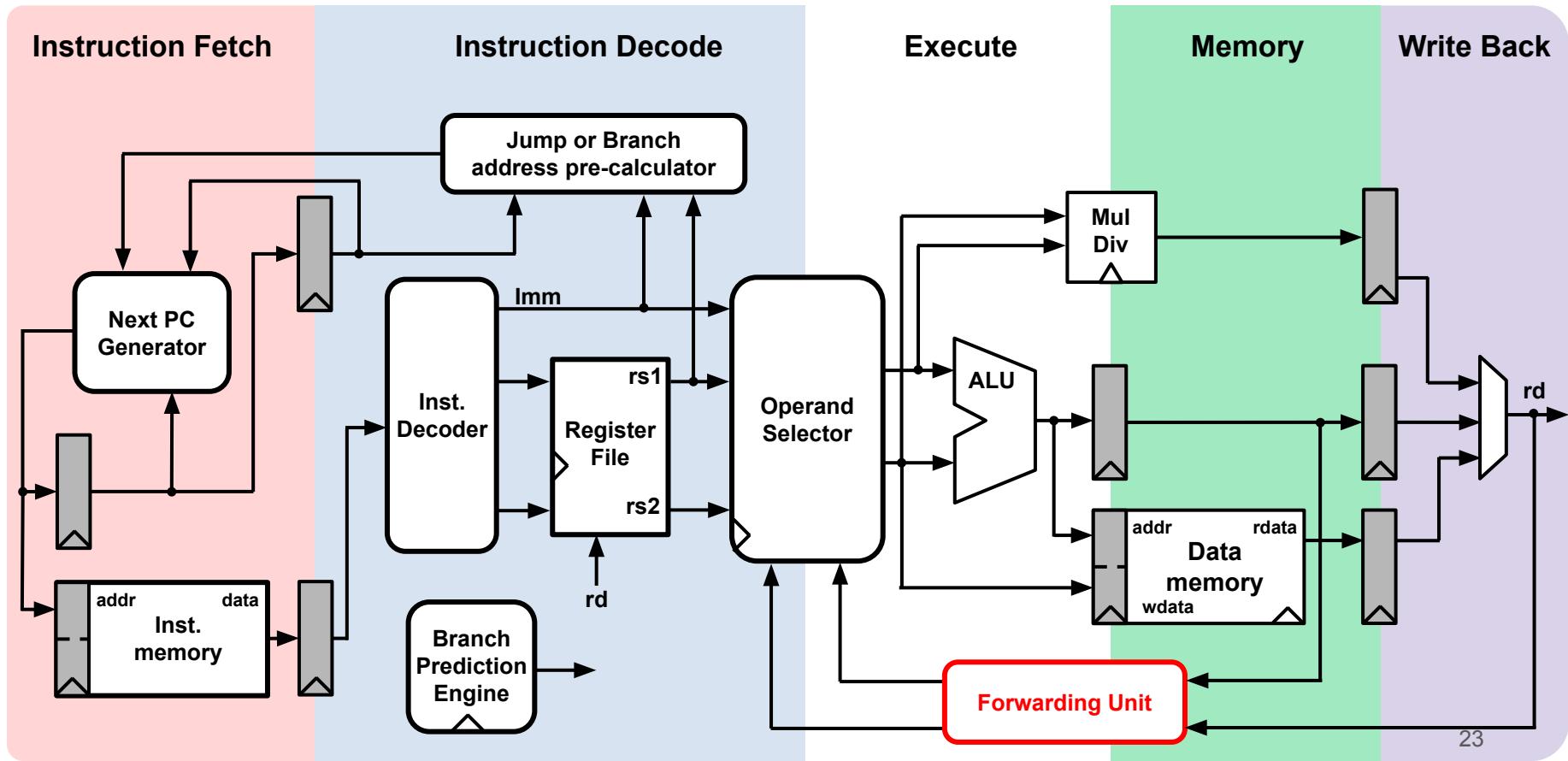
Acceleration Strategies



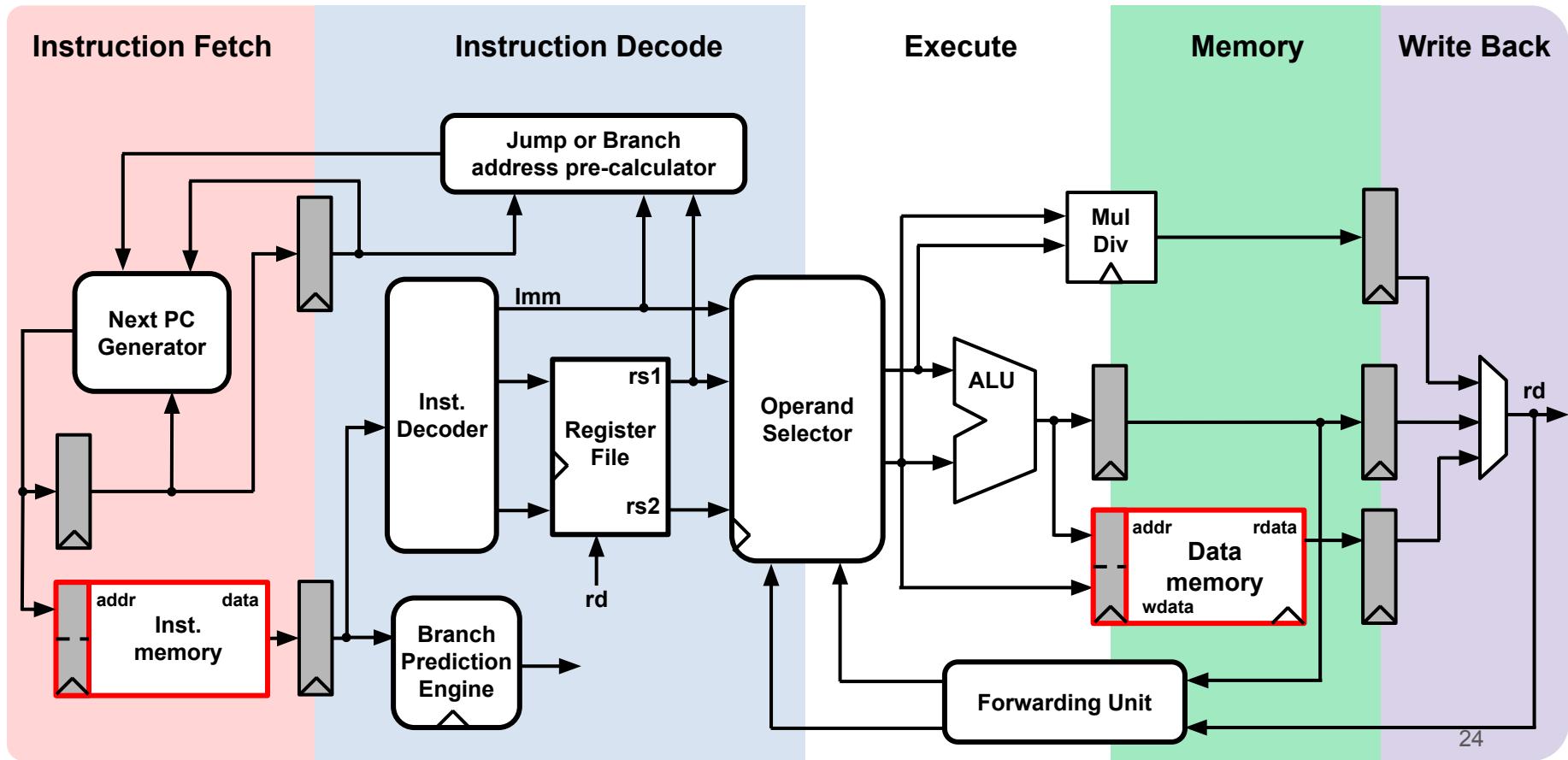
Instruction Dependency



Acceleration Strategies

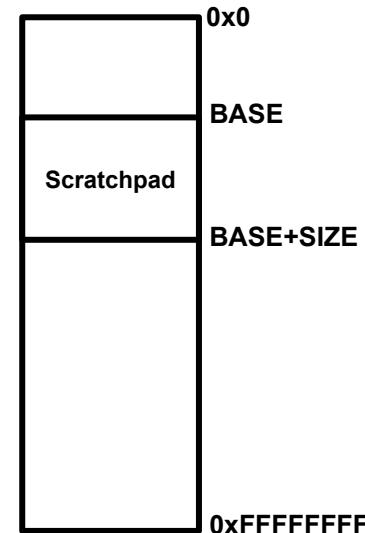
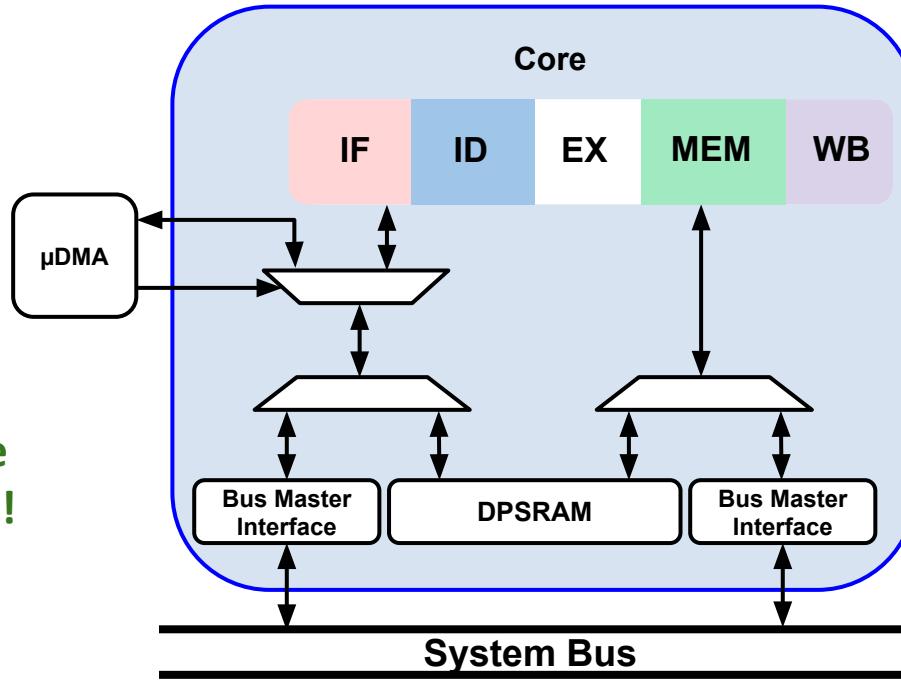


IPC Killer



Scratchpad Block Diagram

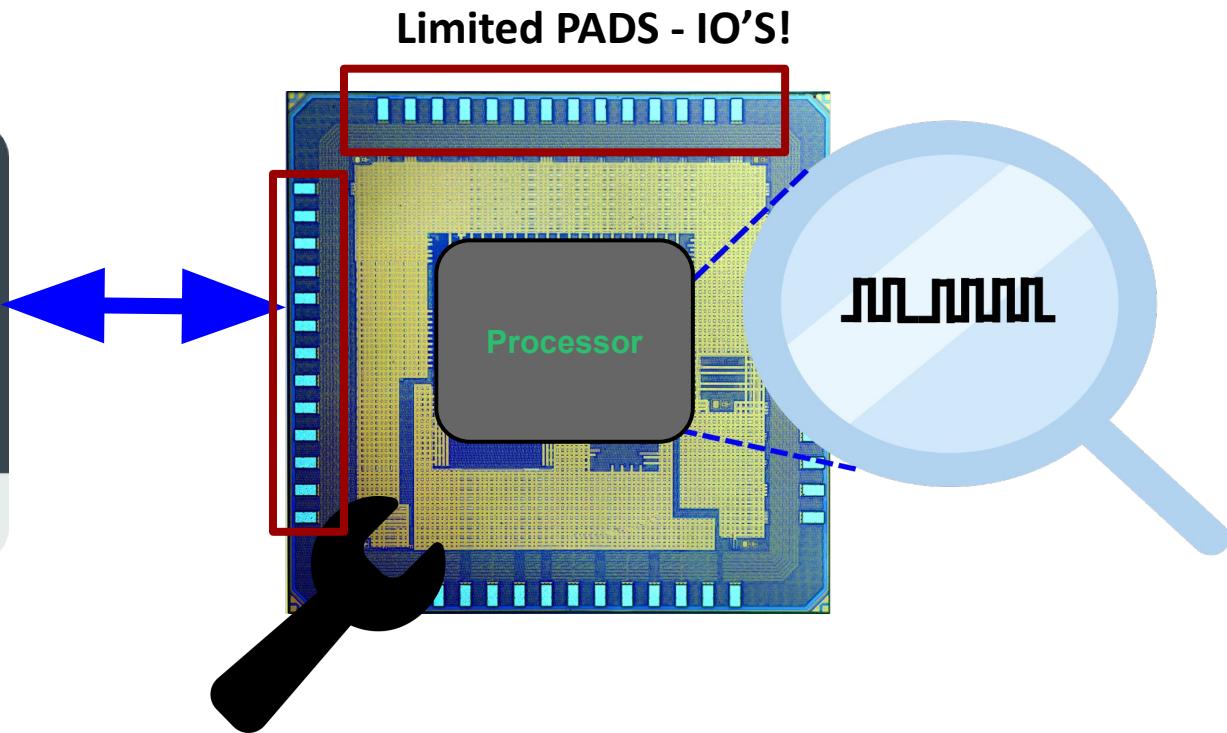
1 cycle
access!



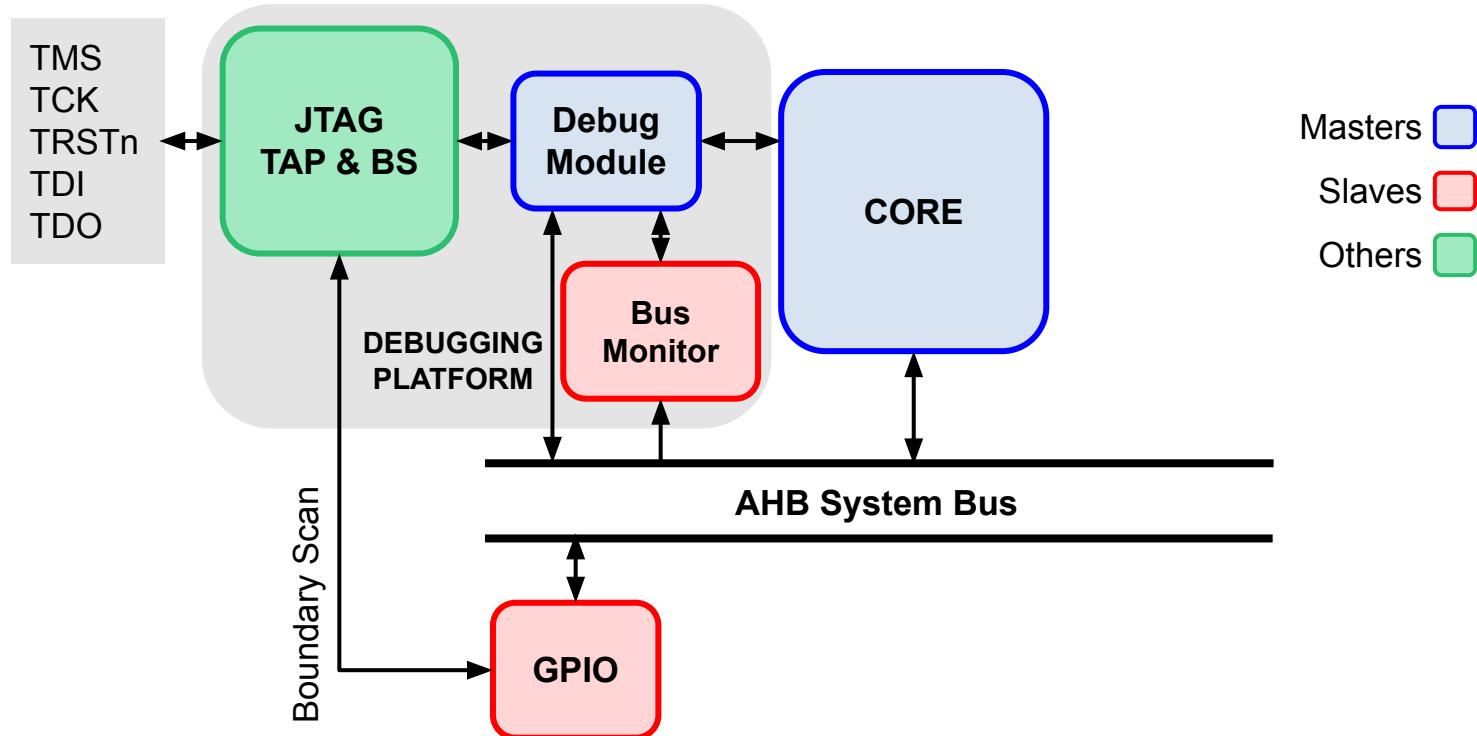
Throughput Penalties

Types	Cycles Stalled	Cycles Reduced
Dependency between instructions	0	3
Branch & Jump	1	2
Load / Store (scratchpad)	1	n
Load / Store (other addresses)	Still depending on Bus Latency	0

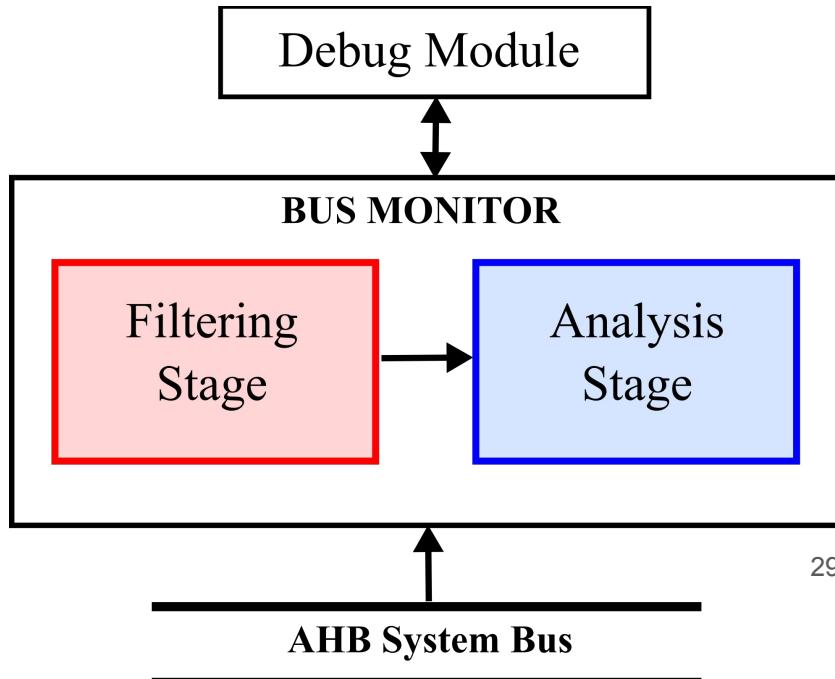
Debugging IC



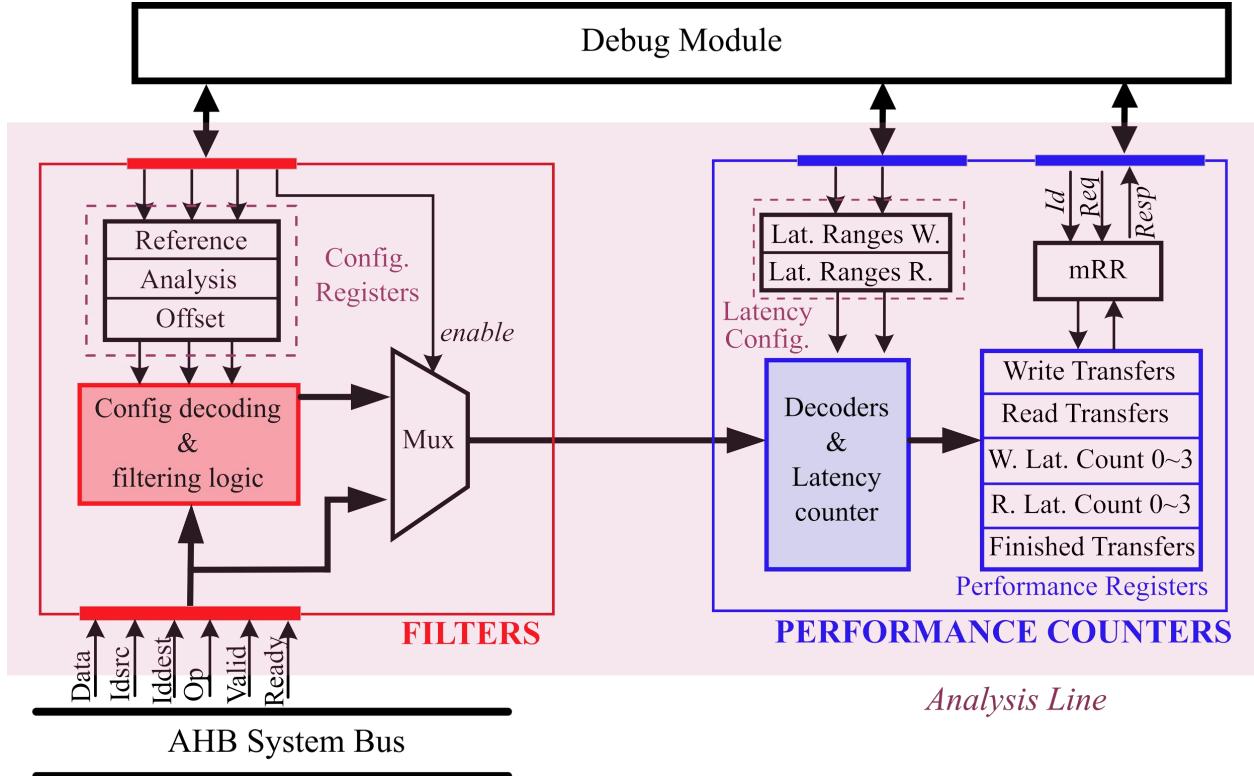
Debug Platform Used



System Bus Monitor

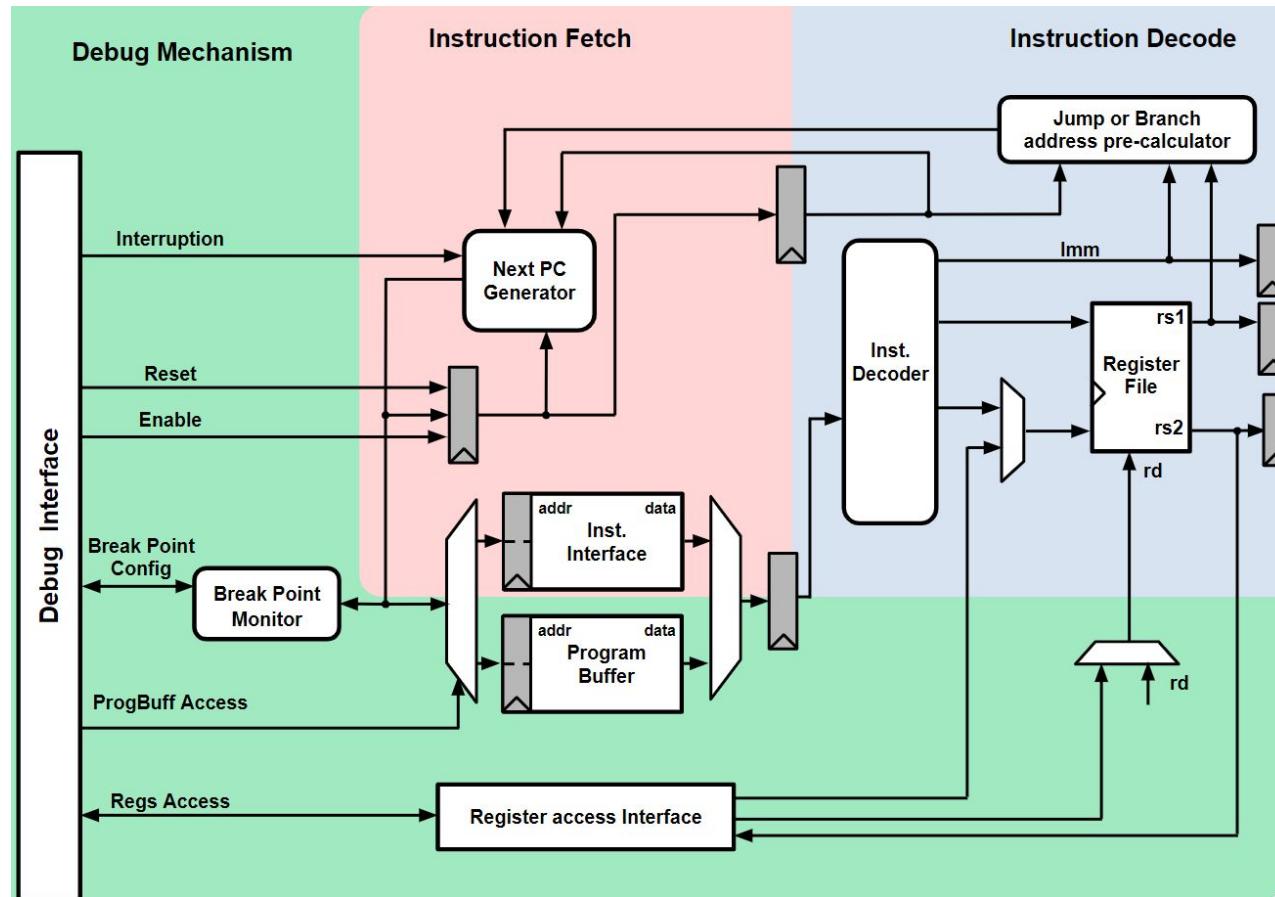


System Bus Monitor



Implemented Debug Interface (RTL)

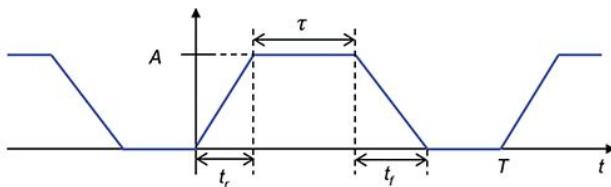
- Halt hart
- Reset hart
- Run Programs
- Access CSR
- Access GPR
- Set Breakpoint



Outline

- Motivation
- Core Microarchitecture
- **Physical Design**
- Verification & FPGA Prototyping
- Further Contributions
- Conclusions

Physical Constraints 200 MHz



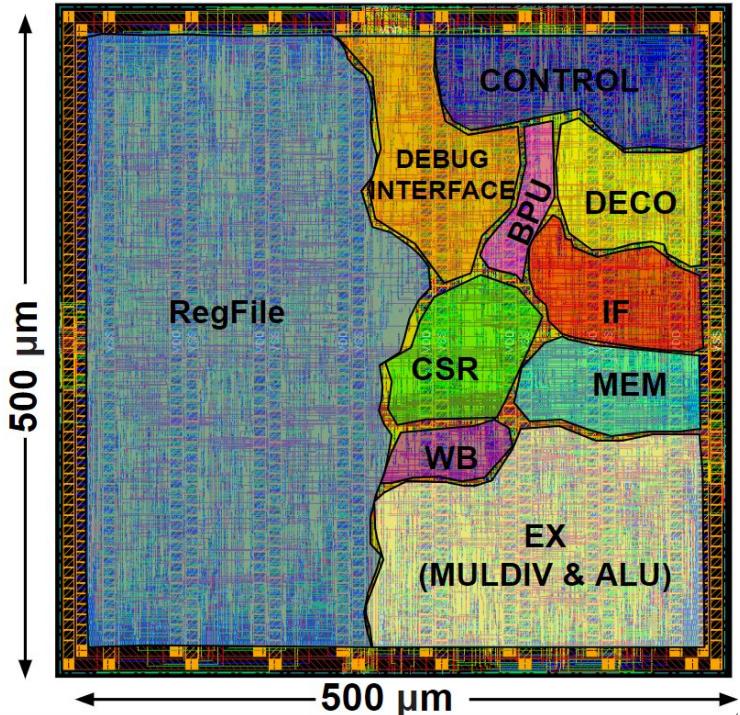
Period	5 ns
Rise and fall times	100 ps
Maximum hold and setup uncertainty	70 ps
Maximum delay from input or output	80 ps
Outputs load	400 fF

Characterization Corners

Corner	Voltage (V)	Temperature (°C)	Process
BC	1.98	0	FF
LT	1.98	-40	FF
ML	1.98	125	FF
TC	1.80	25	TT
WC	1.62	125	SS
WCL	1.62	-40	SS

1.8V LIBRARY CHARACTERIZATION CORNERS

Core Area Reports



Flow	Area
Synth	191880 [μm^2] => 438 μm x 438 μm
PNR	250000 [μm^2] => 500 μm x 500 μm

Number of Cells	7897
Gates Equivalent	19 kGE
Number of Transistors Equivalent	75k

Synthesis Reports

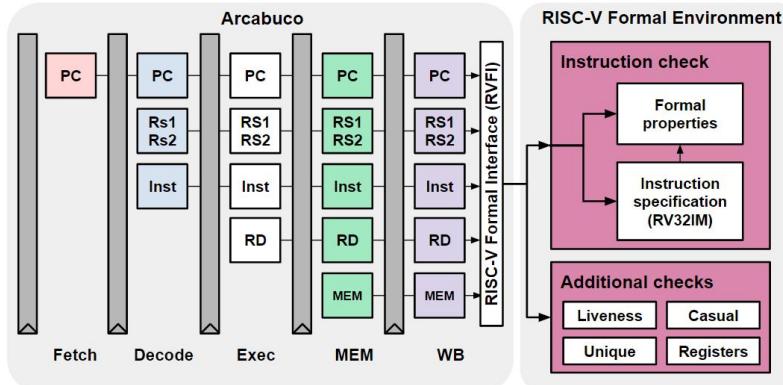
Corner	WC	TC	BC	LT	ML	WCL
Operating Frequency [MHz]	122	200	273	286	231	149
Power Efficiency [$\mu\text{W}/\text{MHz}$]	66.55	84.45	110.51	108.21	117.51	62.14
Leakage Power [μW]	4.426	0.58	1.24	0.32	25.4	0.18

Outline

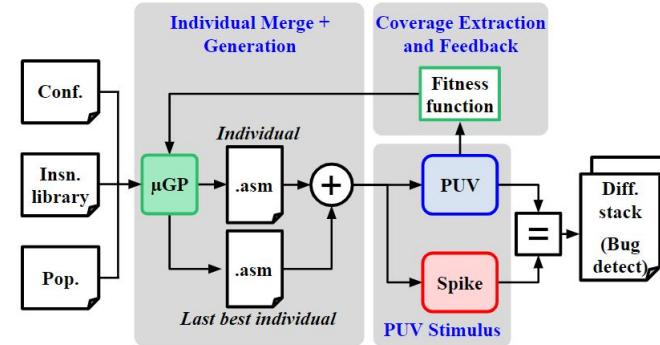
- Motivation
- Core Microarchitecture
- Physical Design
- Verification & FPGA Prototyping
- Further Contributions
- Conclusions

Verification and FPGA Prototyping

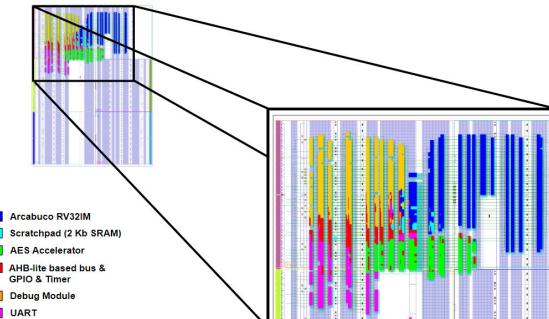
RISC-V Formal Verification



RISC-V Coverage Driven Verification

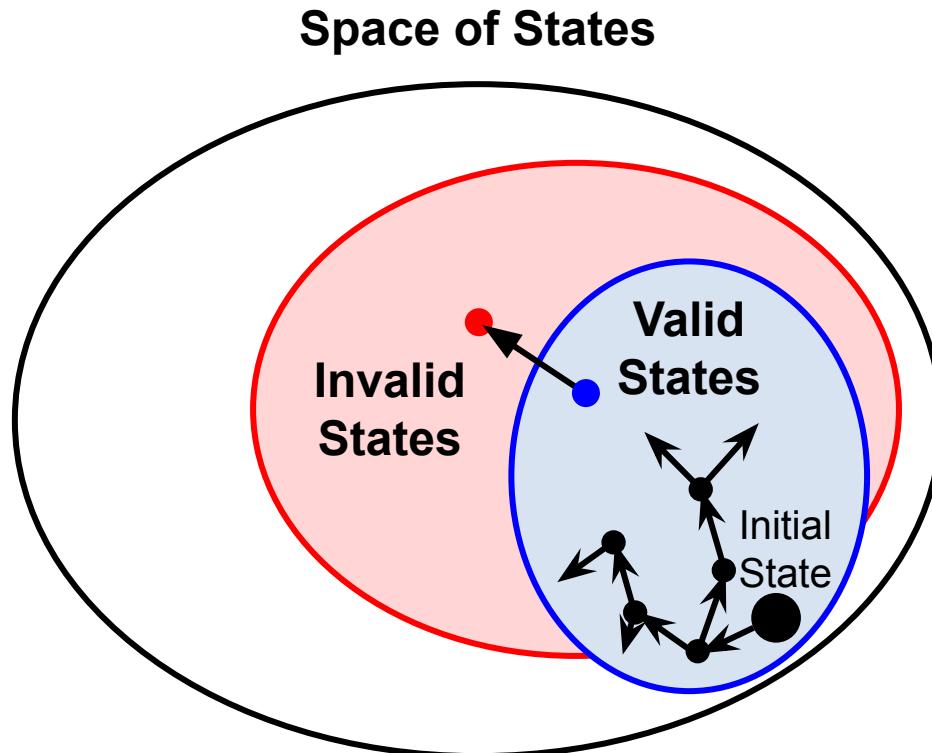


FPGA Place and Route on Arty A7

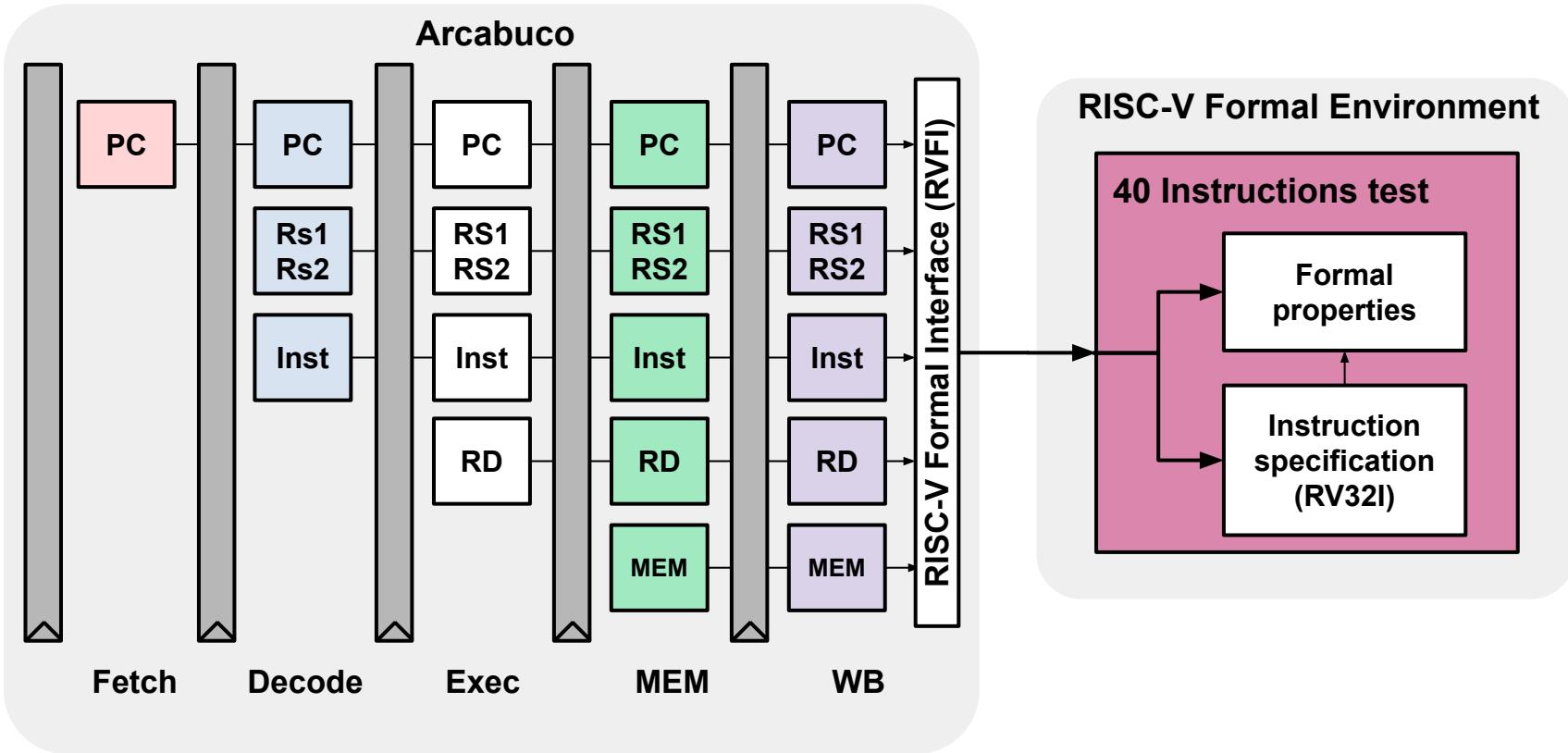


Formal Verification

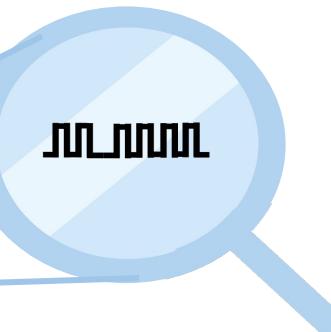
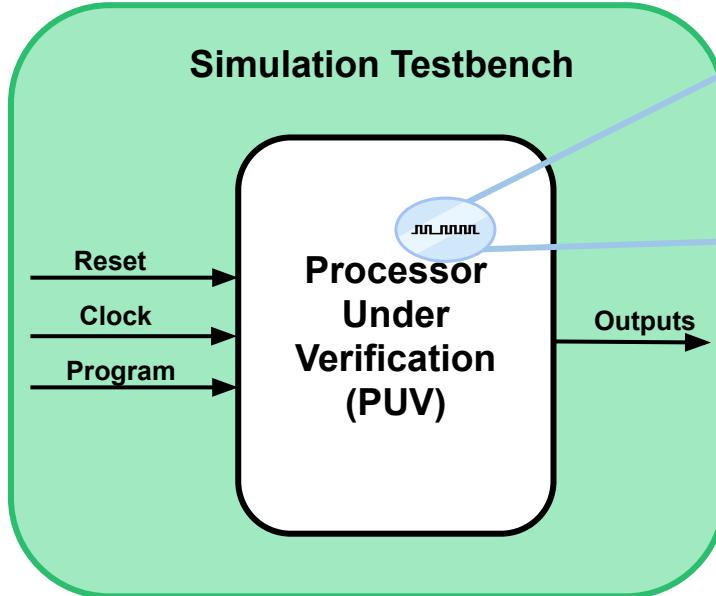
Assertions
&
Assumptions



RISC-V Formal Verification

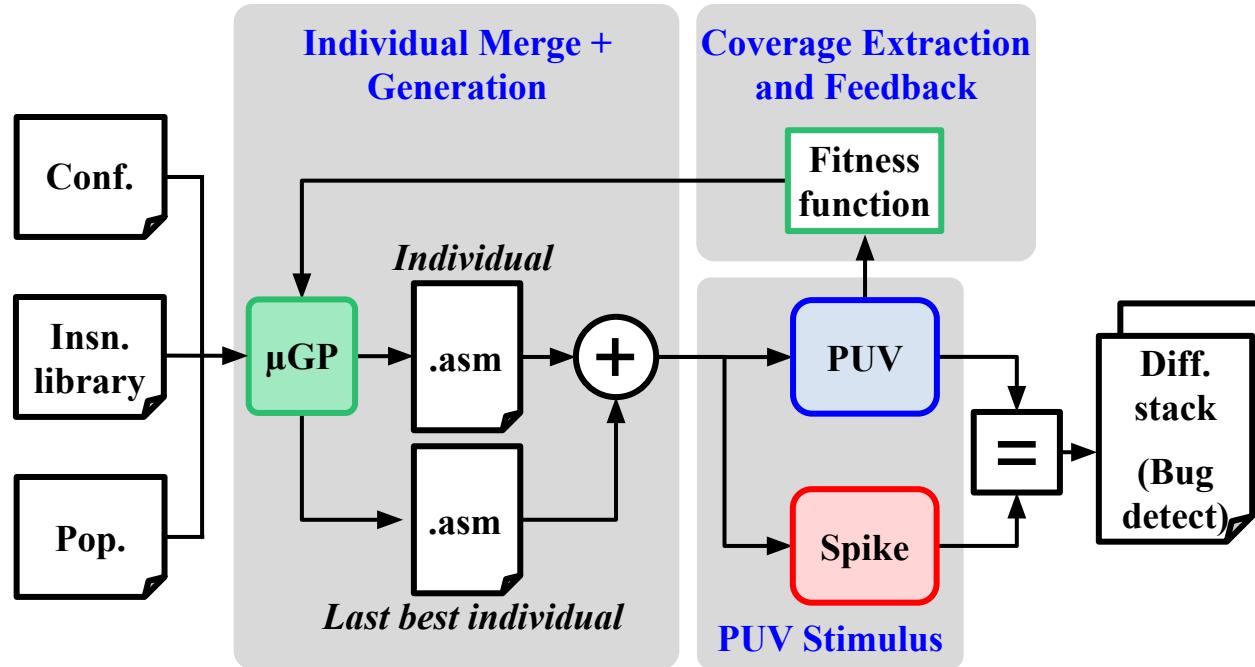


Coverage Extraction

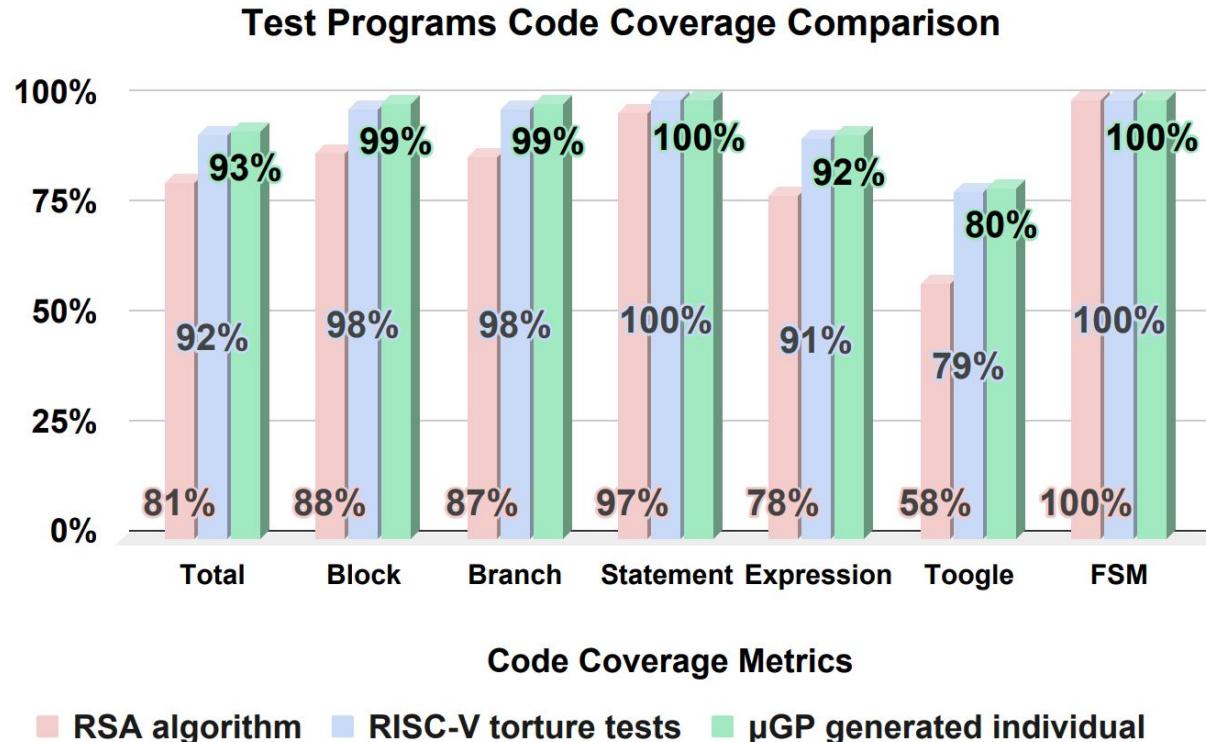


Coverage =>
Percentage of Circuit Excited

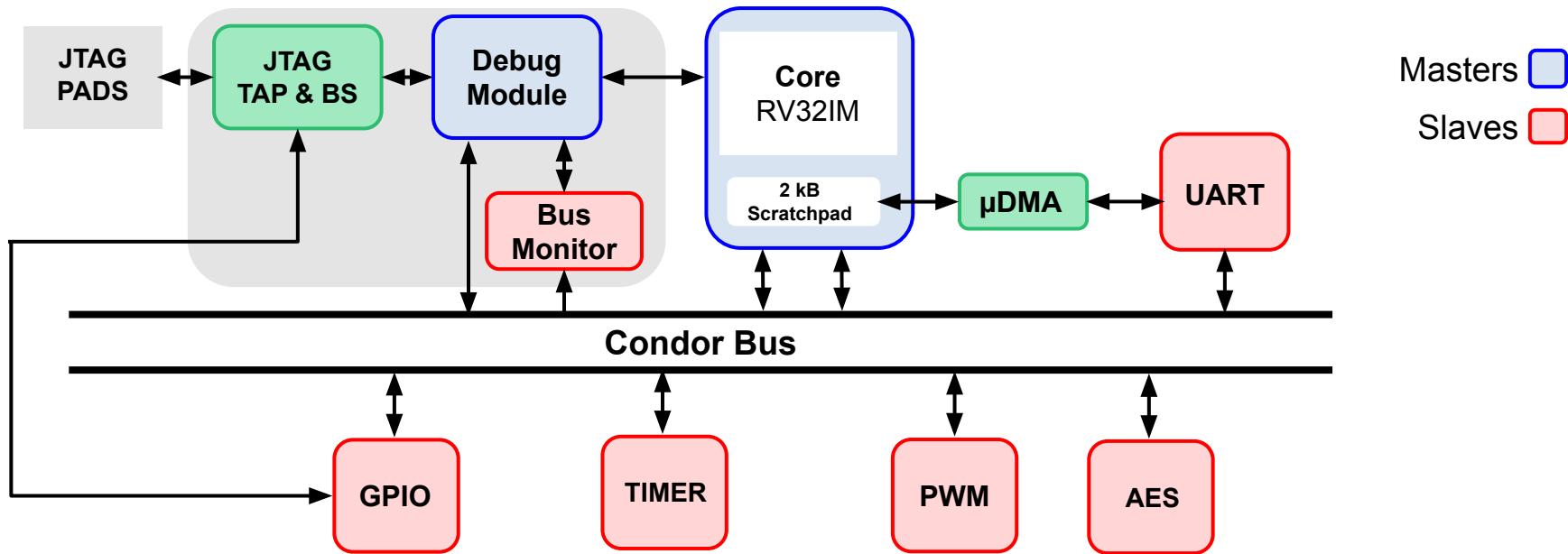
RISC-V Coverage Driven Verification



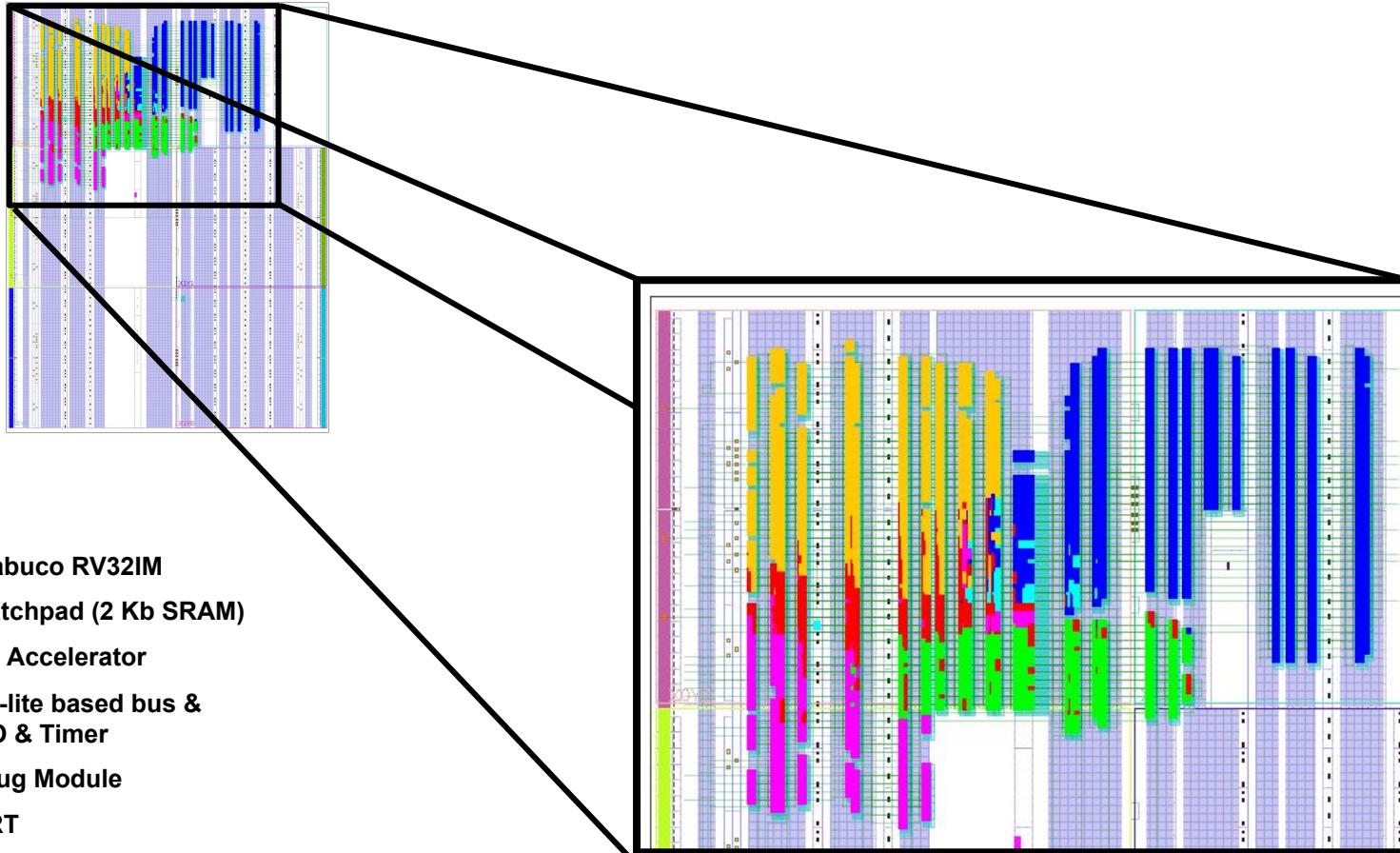
RISC-V Coverage Driven Verification



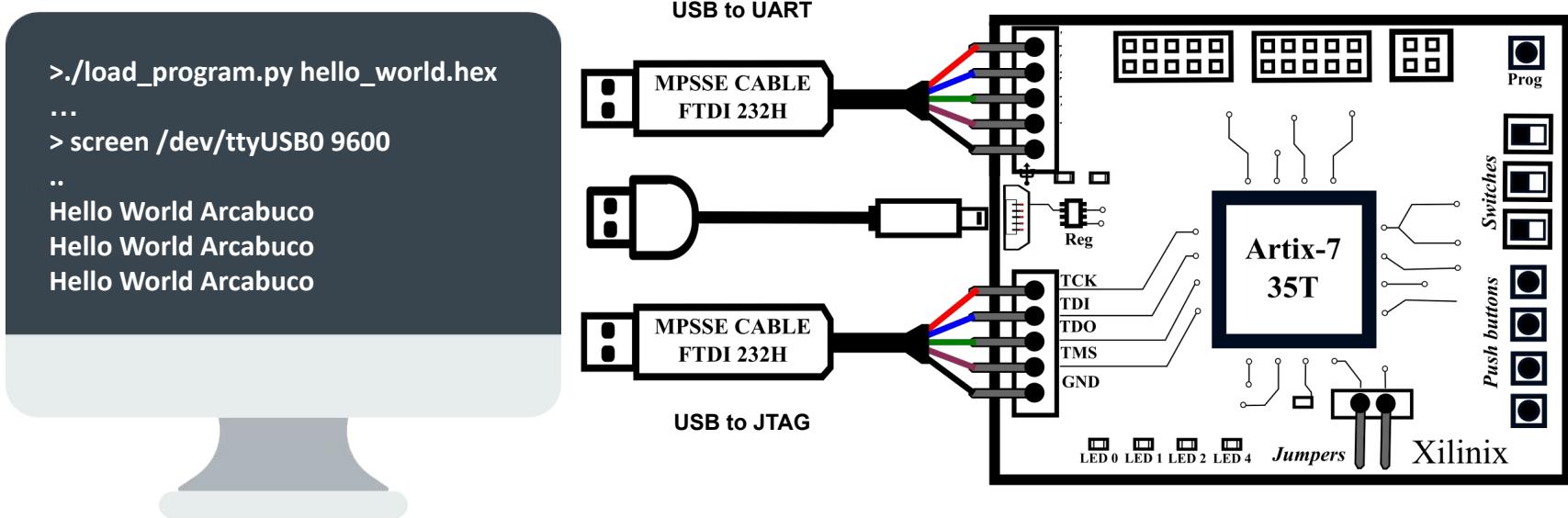
System Prototyping on Nexys 4 & Arty A7



FPGA Place and Route on Arty A7



FPGA Implementation Set-Up



[8] Xilinx, “Artix-7 35T Arty FPGA Evaluation Kit.” [Online].
Available:<https://www.xilinx.com/products/boards-and-kits/arty.html#documentation>

Benchmarking Results

Processor	Drystone (DMIPS/MHz)	CoreMark (CoreMarks/MHz)	Area (μm^2)	Power Efficiency (μW/MHz)
Arm cortex-M3	1.25	3.34	350000@180nm	141
Arcabuco RV32IM	1.13	2.7	250000@180nm	84.45
Arm cortex-M0+	0.95	2.46	98000@180nm	47.4
Arm cortex-M0	0.87	2.33	110000@180nm	66
Arcabuco RV32I	0.87	1.2	200000@180nm	69.1
mRISC-V RV32IM	0.305	-	120776@180nm	97

Benchmarking Results

Processor	Energy per Coremark Iteration (μJ) @1MHz	Energy per Drystone Iteration (μJ) @1MHz
Arm cortex-M3	57.5	112.8
Arcabuco RV32IM	31.2	74.7
Arm cortex-M0+	19.2	50
Arm cortex-M0	28.3	75.8
Arcabuco RV32I	57.5	57.5
mRISC-V RV32IM	-	318

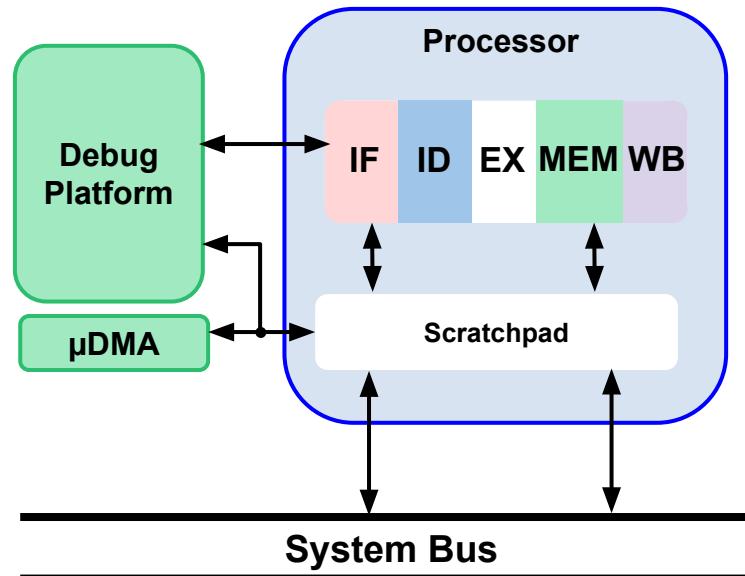
Outline

- Motivation
- Core Microarchitecture
- Physical Design
- Verification & FPGA Prototyping
- Further Contributions
- Conclusions

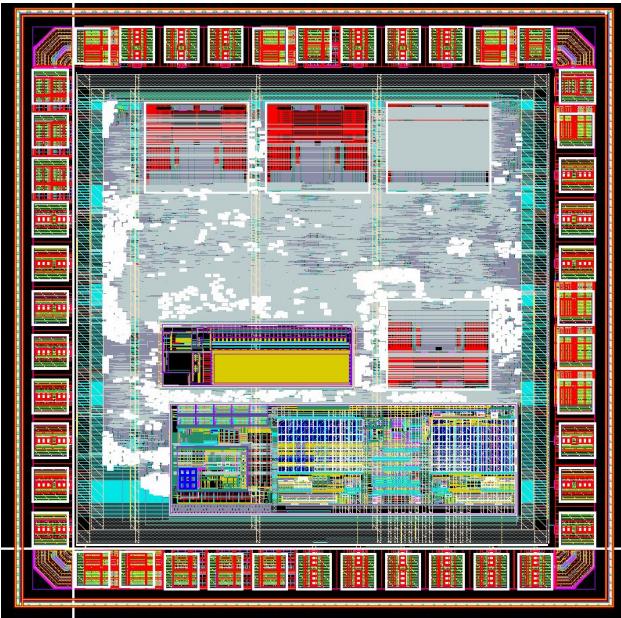
Further Contributions

Core has been used

- In two master thesis, related to core verification and buses.
- In an undergrad thesis related to encryption acceleration.
- Computer architecture class as example core (avg grade 4,1)

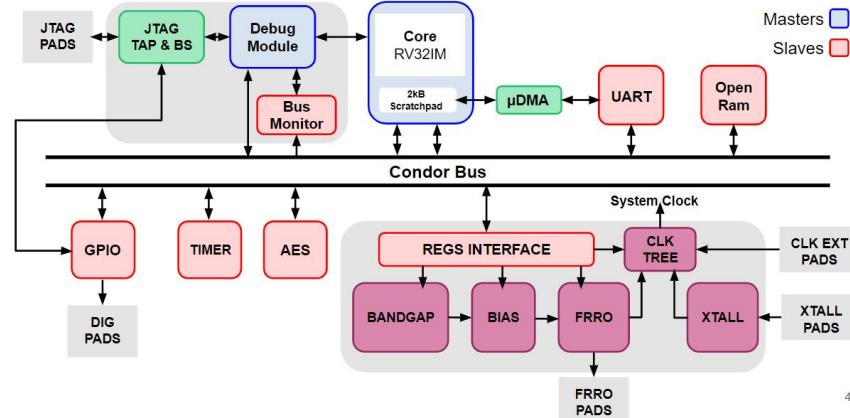


Apicalis Chip integration



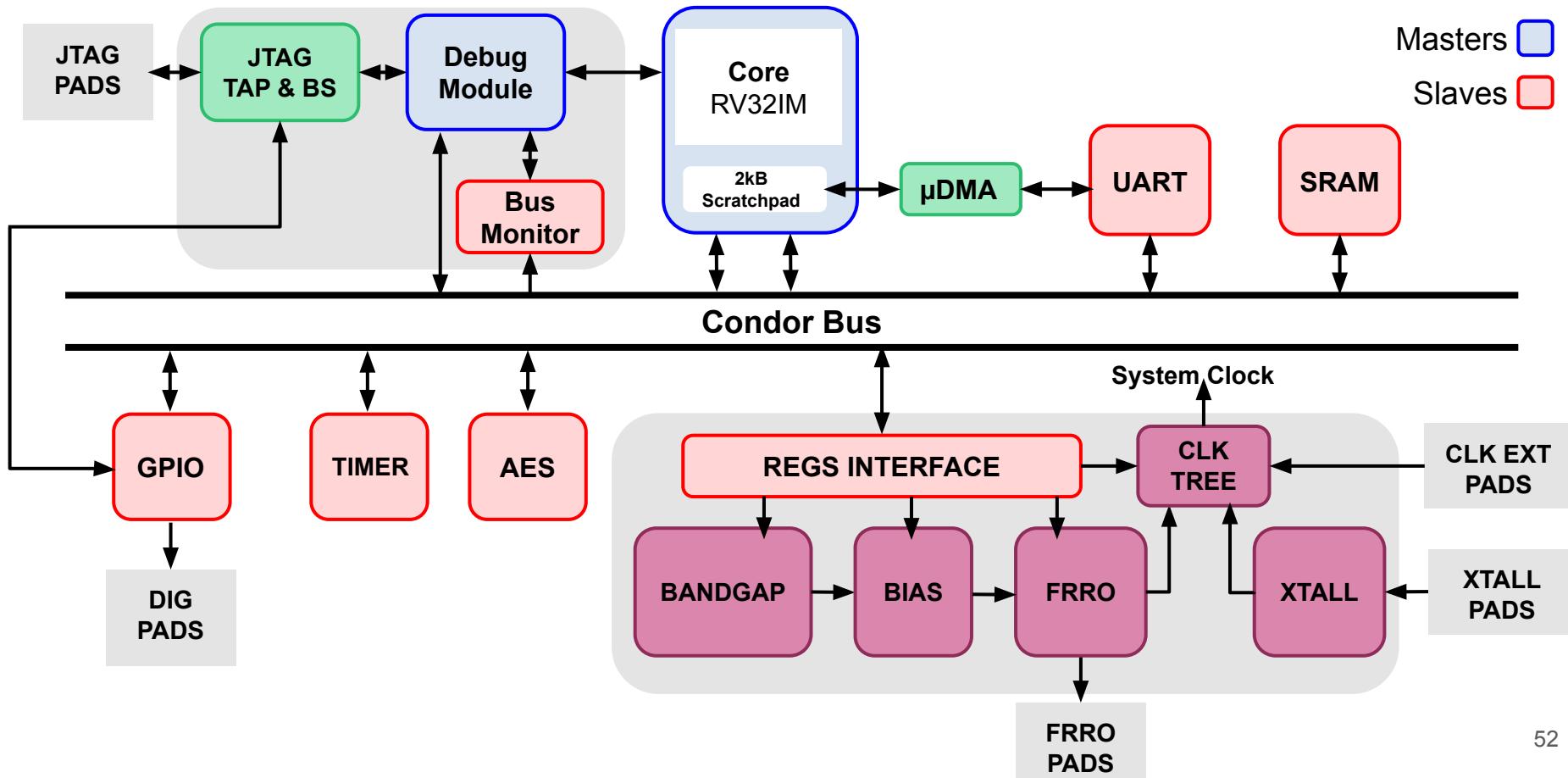
Call for R9 EDS/IEEE Student ASIC Design Fabrication

EDS approved funding of ASIC MPW fabrication run including up-to 9 designs for EDS student members in Region 9 (Latin America and Caribe), for a total of US\$ 25,000.00



4

This is Apicalis (Microarchitecture)



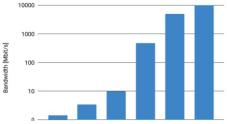
1 IEEE Paper Authored, 4 Co-Authored

2019 IEEE 10th Latin American Symposium on Circuits & Systems (LASCAS)

A Low-Area Direct Memory Access Controller Architecture for a RISC-V Based Low-Power Microcontroller

Hanssel Morales, Christian Duran, and Elkim Roa
Integrated Systems Research Group - OnChip, Universidad Industrial de Santander, Bucaramanga - Colombia
 {hanssel.morales, christian.duran}@correo UIS.edu.co

Abstract—In this work, we present a low area DMA controller that enables low-cost SoCs where subsystems need constant memory access. Small interfaces and a unique FIFO handling of read/write transactions are fundamental blocks in this design. The proof of concept, the testing system also includes a RISC-V TS212 processor, a USB 1.1/2.0 PHY and a QSPI interface. The DMA controller is a whole system implemented using TSMC 0.18/ μ m technology nodes, where the DMA occupies 4.2% of the total area. The controller shows a total DMA area of 1997 gates using 4 information channels, which is 75.3% smaller area in comparison with recent low-area DMA's.



IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: REGULAR PAPERS

On the Cross-Correlation Based Loop Gain Adaptation for Bang-Bang CDRs

Javier Ardila[✉], Student Member, IEEE, Hanssel Morales, Student Member, IEEE, and Elkim Roa, Member, IEEE

Abstract—This article describes the cross-correlation function as an alternative method to adapt loop gain in digital CDMA. Cross-correlation function is compared with the cross-correlation spectral density based controlling, the adaptive loop gain technique, XCALG, for clock and data recovery systems that implement bang-bang phase detector. Considering the modulation of the loop gain due to jitter noise, previous works exploit the autocorrelation function at the output of the bang-bang phase detector as a proper estimator to track the correlation between the reference and the signal. In contrast, we propose the XCALG as a new alternative to perform loop gain adaptation featuring better observability, less impact from jitter sources while keeping a safe phase margin. Theors behind this idea is presented and the XCALG is demonstrated through bit-error rate system simulation. In addition, preliminary implementation costs of the cross-correlation function are discussed using a 65nm CMOS technology node.

Index Terms—CDR, loop gain adaptation, cross-correlation, clock and data recovery correlation autocorrelation XCALG.

A Low-Cost Bug Hunting Verification Methodology for RISC-V-based Processors

Camillo Rojas, Hanssel Morales and Elkin Roa
Integrated Systems Research Group - OnChip, Universidad Industrial de Santander Bucaramanga - Colombia
Email: camilo.rojas2@correo UIS.edu.co, erfroa@uis.edu.co

Abstract—Agile hardware design strategies have shown a fast adoption in academia and industry by bringing ideas from the software development side. However, adopted design methodologies exhibit traditional verification scenarios based on handmade testbenches. Here we describe a instruction stream generators use an open-loop approach, therefore only give guidance to target specific instruction types and values, without the processor under verification (PLV) modules.

verification methodology for RISC-V-based processors with human-independent testbenches creation, employing high-effort verification methods throughout all processor design cycle. We demonstrated the methodology by performing verification tests in single-issue in-order (SII) 32-bit RISC-V ISA based processor described in Chisel. In contrast to standard verification methods, the proposed methodology can detect bugs hard to isolate even after final FPGA implementations in-field. The generated test programs show higher coverage metrics, and $\times 30$ fewer instructions compared to official RISC-V torture unit tests.

struction stream generators use an open-loop approach, therefore only give guidance to target specific instruction types and values, without the processor under verification (PUV) modules.

In this paper, we describe a verification methodology for RISC-V-based processors, that eliminates the need for time-consuming human dependent testbenches by employing complementary and reliable verification methods. We demonstrated the methodology by finding non-trivial bugs in a 32-bit RISC-V ISA based processor. These bugs were identified despite performing several field-programmable gate array (FPGA) C program demos. In contrast, the proposed methodology detected bugs by the generation of coverage test programs. By conducting code coverage extractions, we demonstrate the increase of code

I. INTRODUCTION

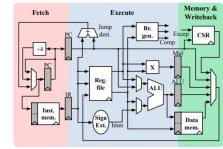
Simulation and Formal: The Best of Both Domains for Instruction Set Verification of RISC-V Based Processors

Ckristian Duran*, Hanssel Morales*, Camilo Rojas*, Annachiara Ruospo[†], Ernesto Sanchez[†] and Elkim Roa*

* Integrated Systems Research Group - OnChip, Universidad Industrial de Santander - Colombia

[†]Politecnico di Torino - Italy, e-mail: ckristian.duran@correo.uis.edu.co

Abstract—The instruction set architecture (ISA) specifies a contract between hardware and software; it covers all possible operations that have to be performed by a processor. Verifying the instruction against a golden execution model following the ISA is becoming a common practice to verify processors. Despite many potential applications, practice verification frameworks require an extensive test set to cover most of the processor states. In this paper, we suggest a verification scheme combining two different approaches: a formal verification methodology and a simulation methodology, for exclusive error detection. The first approach drives automatic program generation using genetic algorithms to maximize coverage of the test and the contrast against an instruction set simulator. The second is a formal verification



An Energy-Efficient RISC-V RV32IMAC Microcontroller for Periodical-Driven Sensing Applications

Kristian Duran¹, Megan Wachs², Luis E. Rueda G.¹, Albert Huntington², Javier Ardila¹, Jack Kang², Andres Amaya¹, Hector Gomez¹, Juan Romero¹, Laude Fernandez¹, Felipe Flechas¹, Rolando Torres¹, Juan Moya¹, Wilmer Ramirez¹ Julian Arenas¹, Juan Gomez¹, Hanssle Morales¹, Camilo Rojas¹, Alex Mantilla¹, Elkin Roa¹ and Krste Asanovic^{2,3}

¹OnChip - Universidad Industrial de Santander, Colombia, ²SiFive Inc., San Mateo, CA, ³U. California at Berkeley, CA

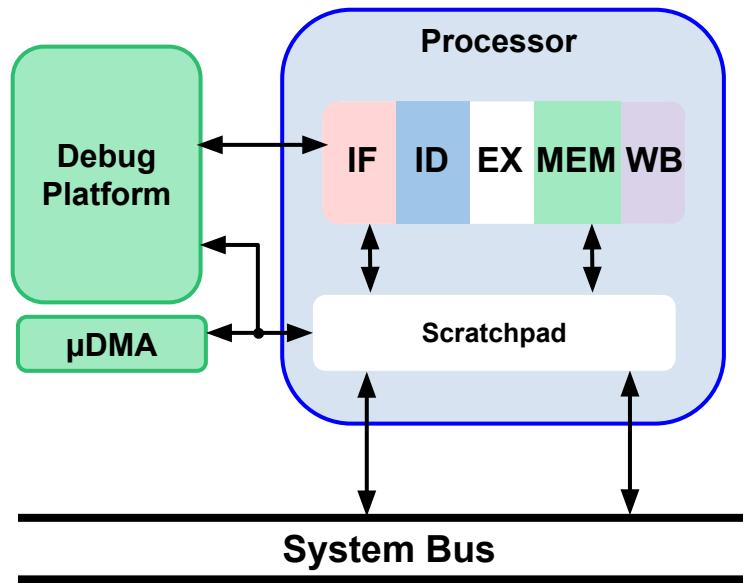
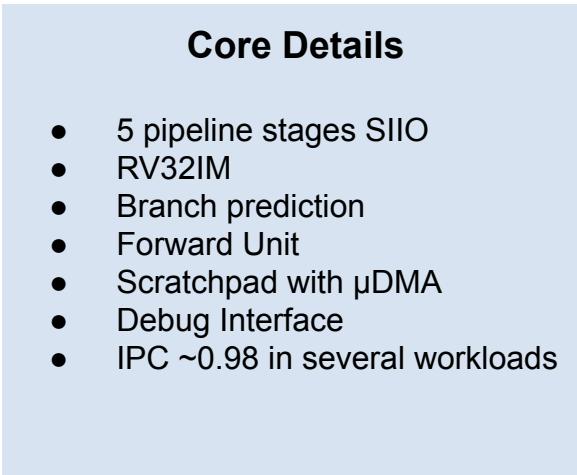
Unicamp - Universidad Industrial de Santander, Colombia, SHINE, Inc., San Mateo, CA, U. California at Berkeley, CA

Outline

- Motivation
- Core Microarchitecture
- Physical Design
- Verification & FPGA Prototyping
- Further Contributions
- Conclusions

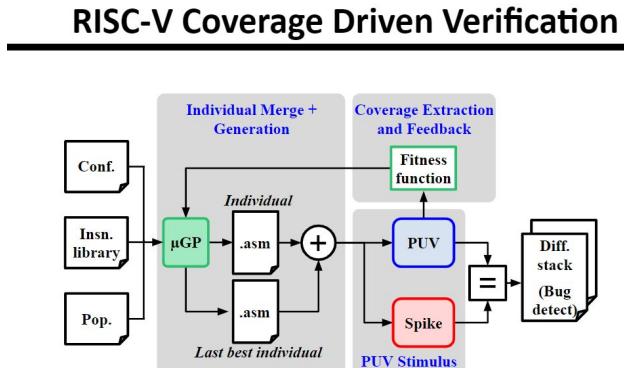
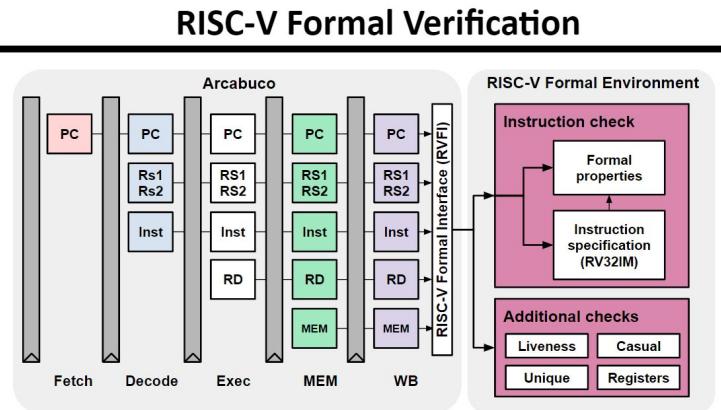
Conclusions

1. An RTL description of a RISC-V I based processor was implemented



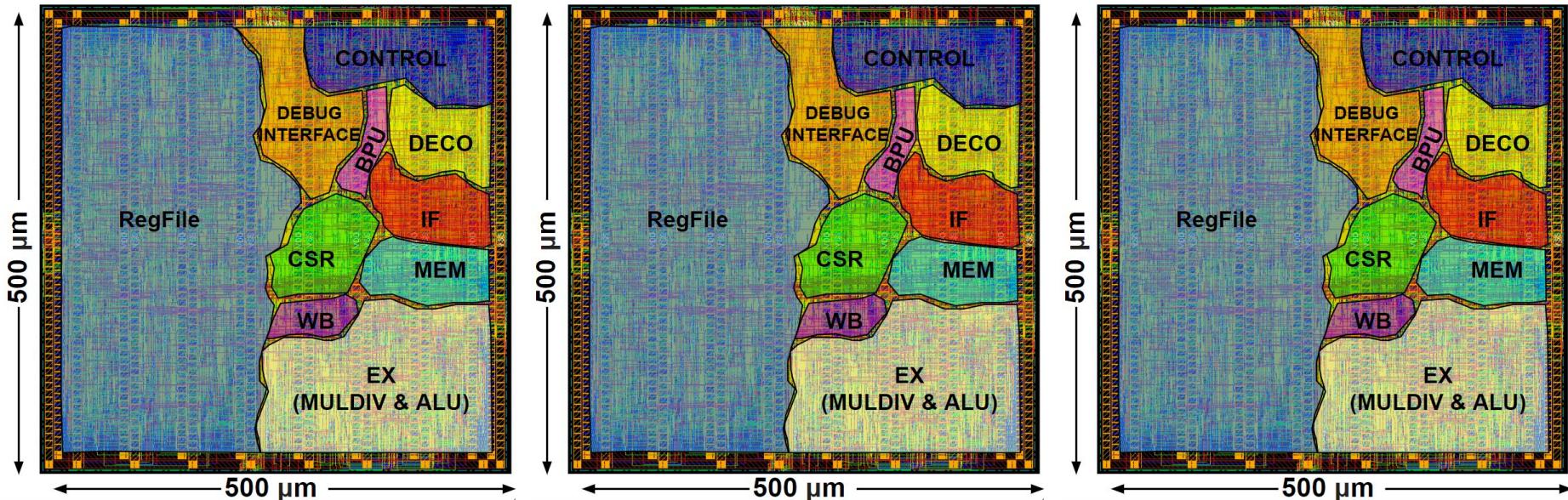
Conclusions

2. The processor RISC-V I specification accomplishment was verified with two different verification frameworks.



Conclusions

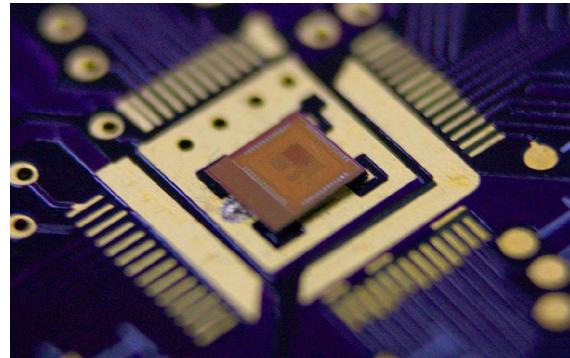
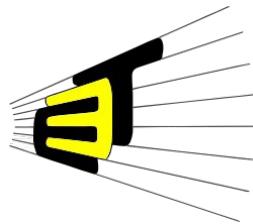
3. The RTL was synthesized in TSMC 180nm technology node and post-synthesis simulations were performed.



References

- [1] Strategy Analytics Global Connected and IoT Device Forecast Update 2019
- [2] Slow and steady wins the race? a comparison of ultra-low-power risc-v cores for internet-of-things applications 2017(PATMOS)
- [3] ARM, “Cortex-m specs,” Arm Developer, 2021.
- [4] J. L. Hennessy and D. A. Patterson, Communications of the ACM, 2019
- [5] Computer Organization and Design RISC-V Edition.
- [6] W. Ramirez, et.al “A flexible debugger for a risc-v based 32-bit system-on-chip,” (LASCAS) , 2020, pp. 1–4.
- [7] C. Rojas, H. Morales and E. Roa, "A Low-Cost Bug Hunting Verification Methodology for RISC-V-based Processors," (ISCAS), 2021
- [8] Xilinx, “Artix-7 35T Arty FPGA Evaluation Kit.” [Online].

Thanks! Questions?

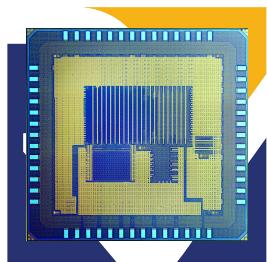


hanssel.morales@correo.uis.edu.co
onchip@uis.edu.co



[@onchipUIS](https://twitter.com/onchipUIS)

General Goal



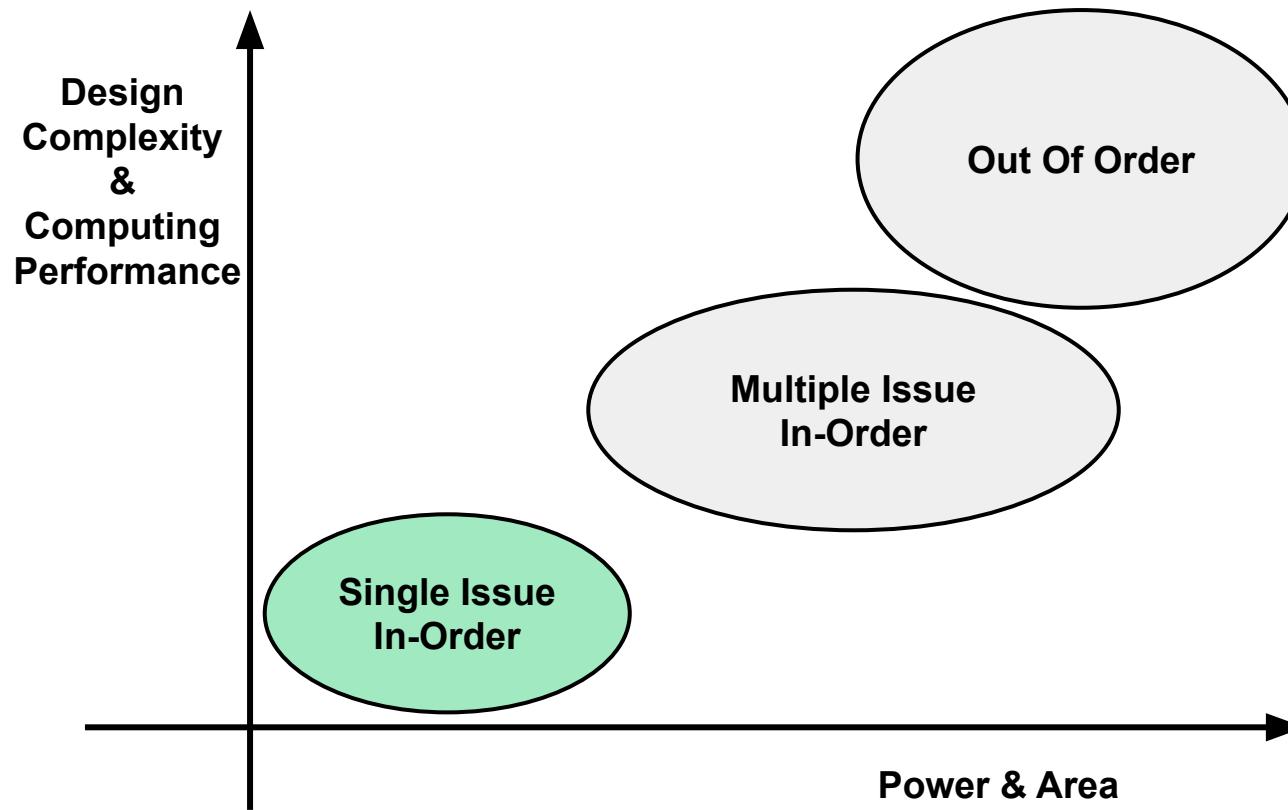
RISC-V®

To design and to verify a RISC-V I based processor with
external debug capability

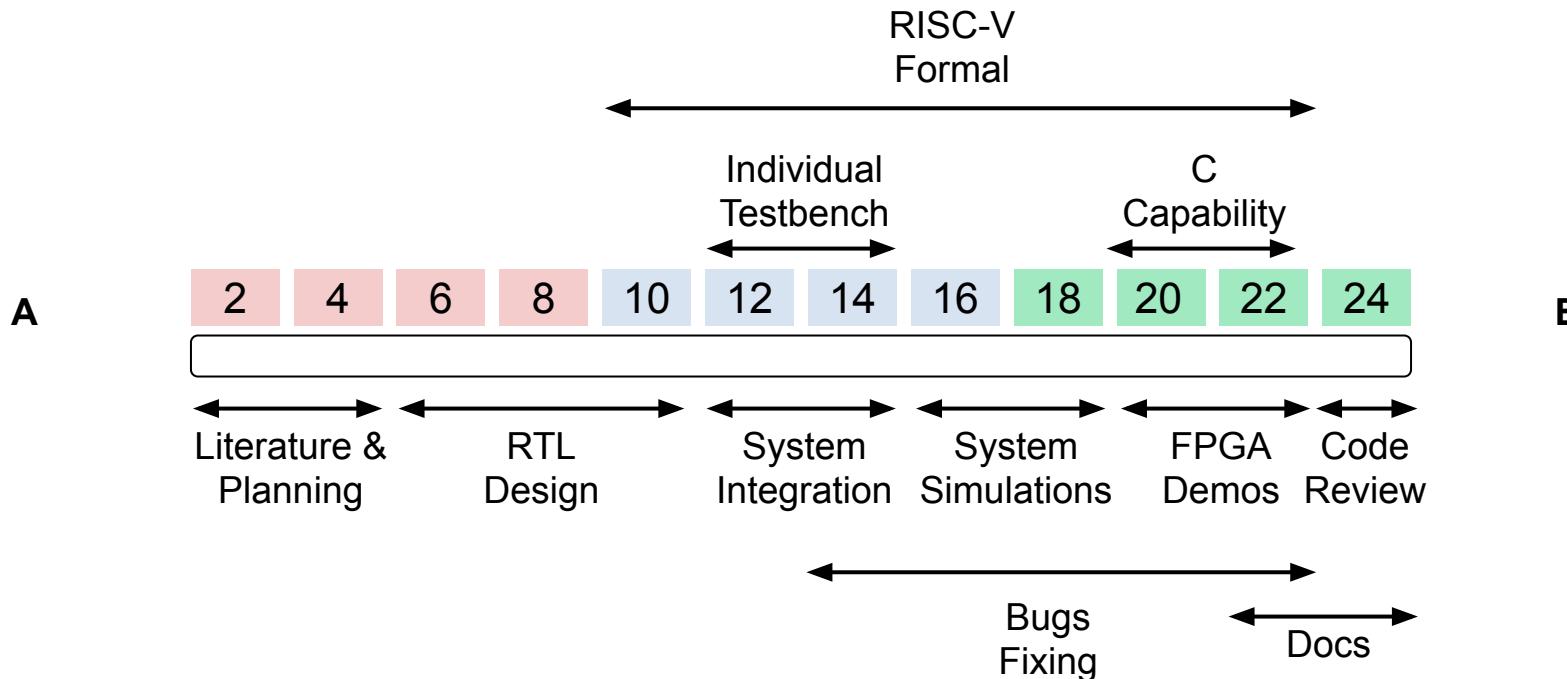
Specific Objectives

1. To implement an RTL description of a RISC-V I based processor in Chisel.
2. To synthesize the RTL in TSMC 180nm technology node and perform post-synthesis simulations to achieve netlist validation.
3. To design interfaces circuitry for supporting an external debug platform able to monitor and control the processor.
4. To validate the processor RISC-V I specification accomplishment using an instruction set compliance verification framework.

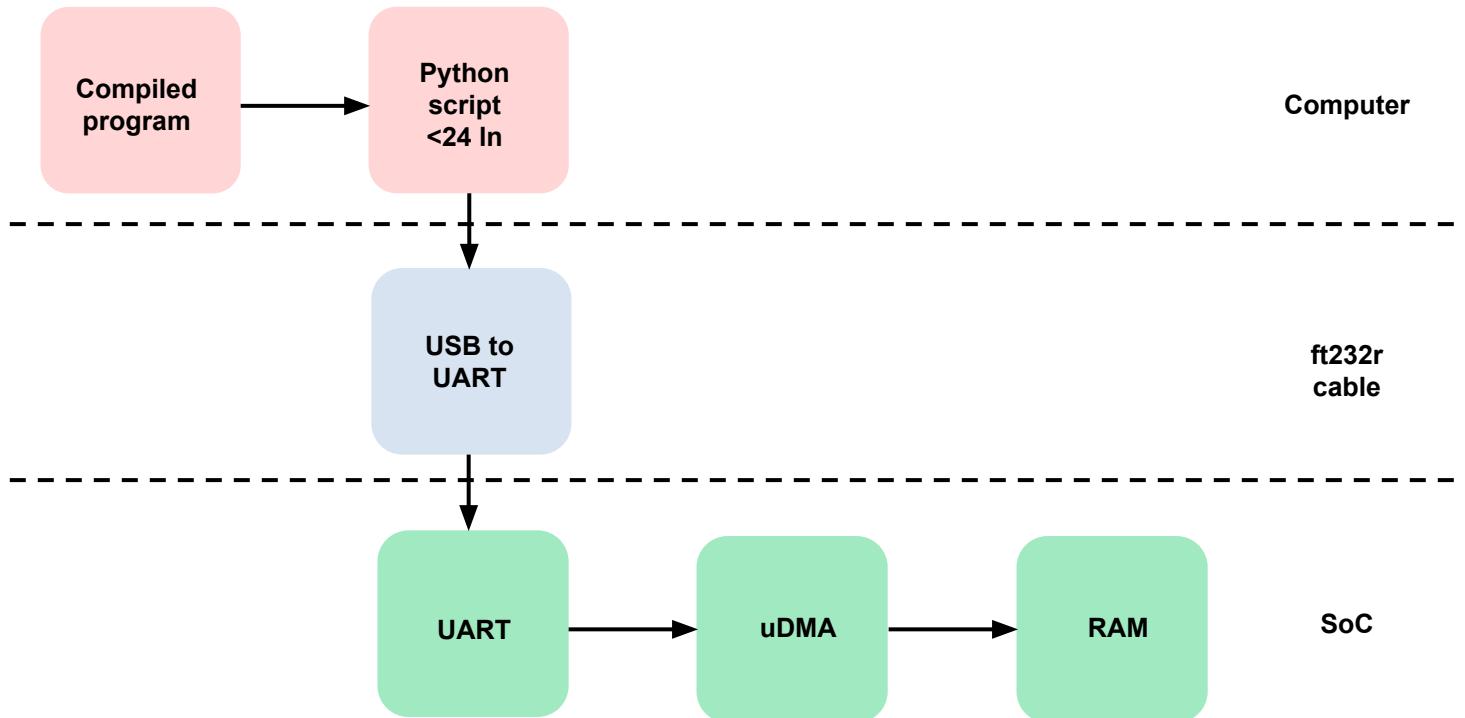
Microarchitecture Target



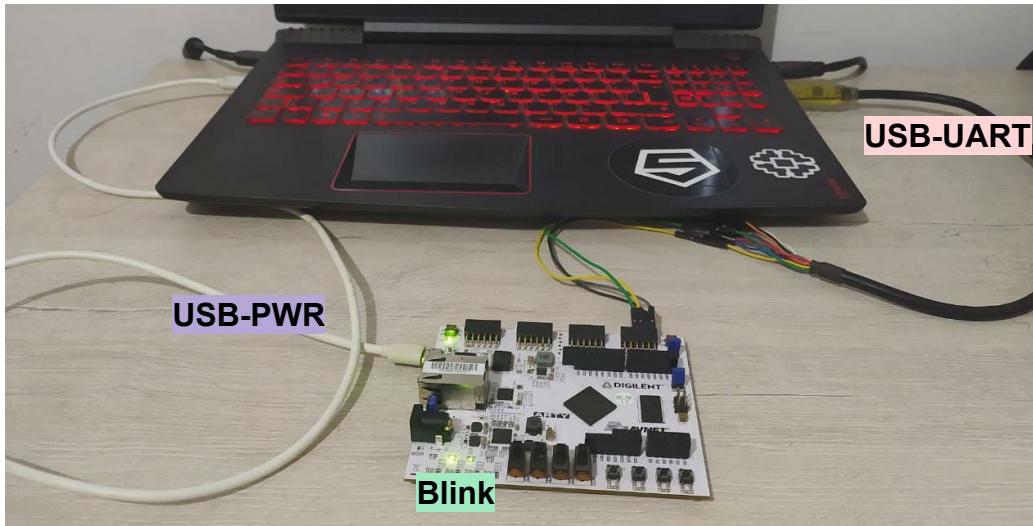
Design and Timing Verification



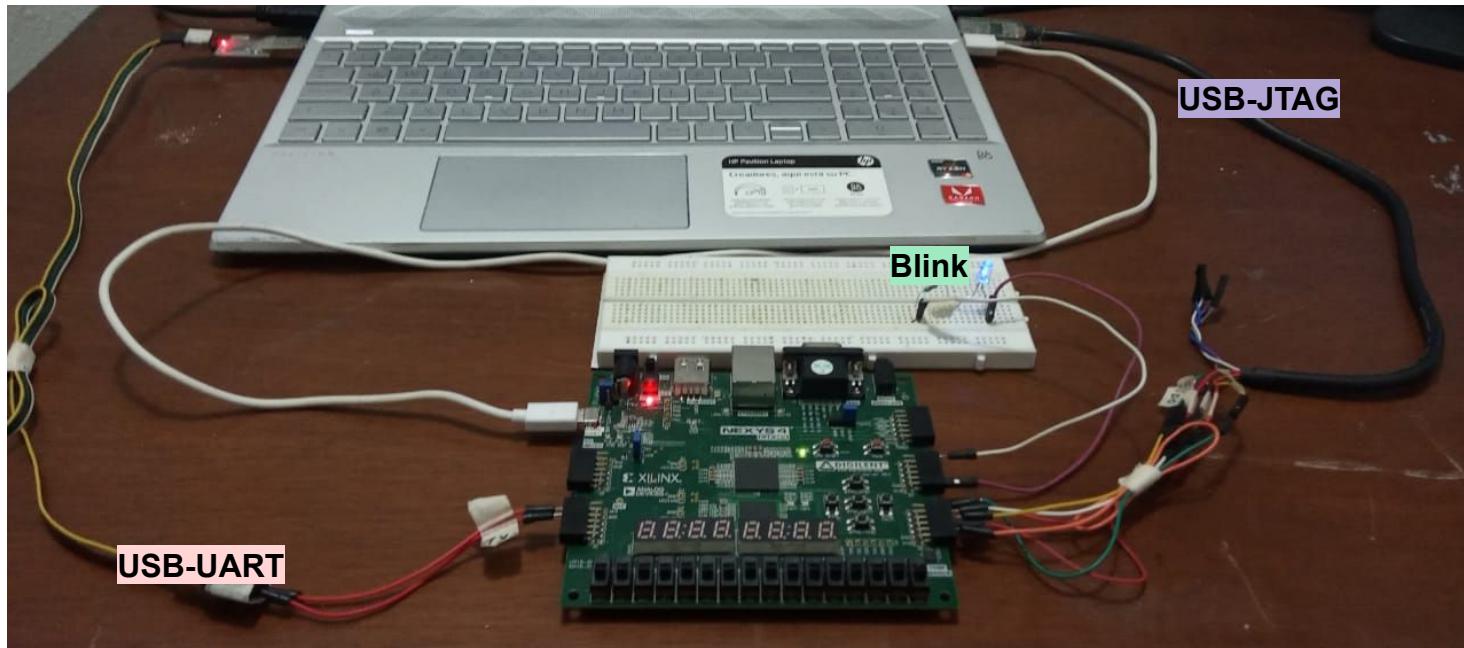
Programming Flow



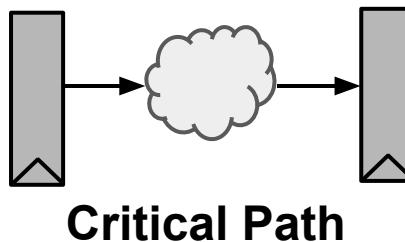
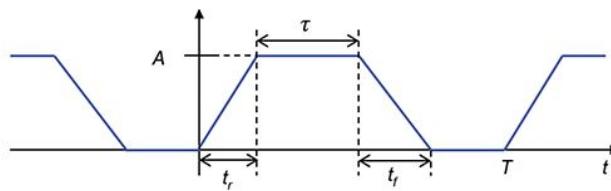
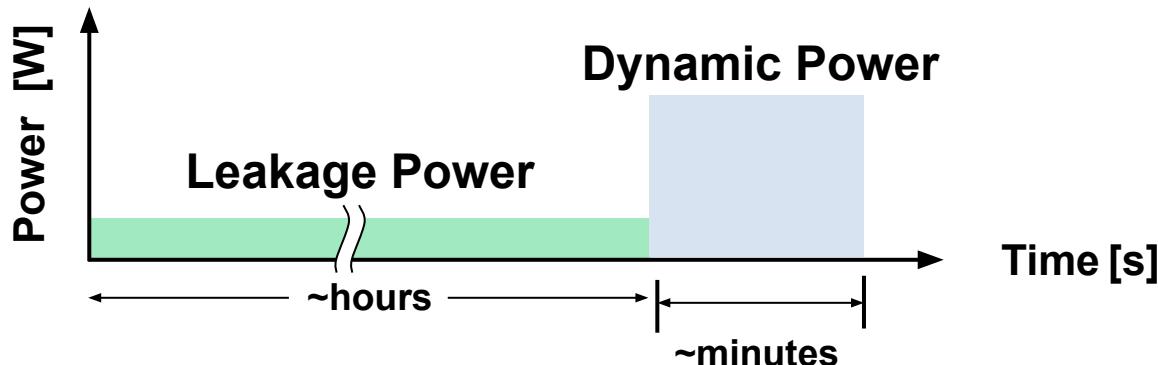
FPGA Implementation



FPGA Implementation



Synthesis Reports



A Low-Area Direct Memory Access Controller Architecture for a RISC-V Based Low-Power Microcontroller

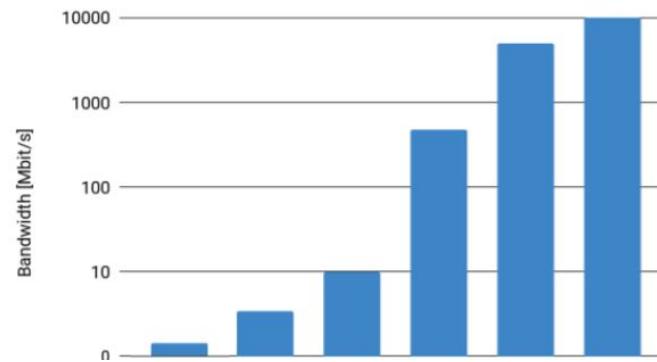
Hanssel Morales, Ckristian Duran, and Elkim Roa

Integrated Systems Research Group - OnChip, Universidad Industrial de Santander, Bucaramanga - Colombia

{hanssel.morales, ckristian.duran}@correo.uis.edu.co, efroa@uis.edu.co

Abstract—In this work, we present a low area DMA controller that enables low-cost SoCs where subsystems need constant memory access. Small interfaces and a unique FIFO handling read/write transactions are fundamental blocks in this design. As proof of concept, the testing system also includes a RISC-V RV32IM processor, a USB 1.1/2.0 PHY and a QSPI interface. We implemented a whole microcontroller using a TSMC 0.18 μ m technology node, where the DMA occupies 4.2% of the total area. The results show a total DMA area of 1997 gates using 4 information channels, which is 75.3% smaller area in comparison with recent low-area DMAs.

Index Terms—Direct Memory Access, DMA, RISC-V, Microcontroller, IoT



Simulation and Formal: The Best of Both Domains for Instruction Set Verification of RISC-V Based Processors

Ckristian Duran*, Hanssel Morales*, Camilo Rojas*, Annachiara Ruospo†, Ernesto Sanchez† and Elkim Roa*

* Integrated Systems Research Group - OnChip, Universidad Industrial de Santander - Colombia

†Politecnico di Torino - Italy. e-mail: ckristian.duran@correo.uis.edu.co

Abstract—The instruction set architecture (ISA) specifies a contract between hardware and software; it covers all possible operations that have to be performed by a processor, regardless of the implemented architecture. Verifying the instruction execution against a golden execution model following the ISA is becoming a common practice to verify processors. Despite many potential applications, existing verification frameworks require an extensive test set to cover most of the processor states. In this paper, we suggest a verification scheme combining two different domains, simulation- and formal-verification, establishing a methodology for exclusive error detection. The first approach drives automatic program generation using genetic algorithms to maximize coverage of the test and the contrast against an instruction set simulator. The second is a formal verification approach, where an interface carries specific processor states according to the ISA specification. By combining these two, we present a reliable way to perform more accurate instruction verification by increasing processor state coverage and formal assertions to detect different kinds of errors. Compared to extensive torture test sets, this approach reaches a more

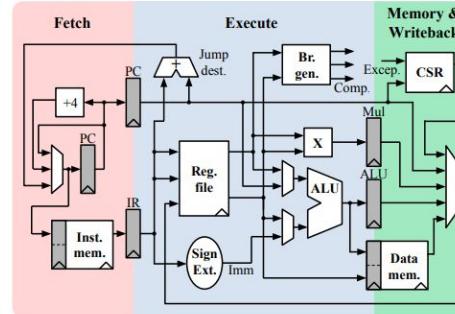


Fig. 1. Microarchitecture of the processor under verification (PUV). PUV comprise 3-stage single issue in order pipeline.

to simulation models implementing an evolutionary framework

A Low-Cost Bug Hunting Verification Methodology for RISC-V-based Processors

Camilo Rojas, Hanssel Morales and Elkim Roa

Integrated Systems Research Group - OnChip, Universidad Industrial de Santander Bucaramanga - Colombia

Email: camilo.rojas2@correo.uis.edu.co, efroa@uis.edu.co

Abstract—Agile hardware design strategies have shown a fast adoption in academia and industry by bringing ideas from the software development side. However, adopted design methodologies exhibit traditional verification scenarios based on handmade testbenches. Here we describe a verification methodology for RISC-V-based processors with human-independent testbenches creation, employing high-effort verification methods throughout all processor design cycle. We demonstrated the methodology by performing verification tests in a single-issue in-order (SISO) 32-bit RISC-V ISA based processor described in Chisel. In contrast to standard verification methods, the proposed methodology can detect bugs hard to isolate even after final FPGA implementations in-field. The generated test programs show higher coverage metrics, and $\times 30$ fewer instructions compared to official RISC-V torture unit tests.

I. INTRODUCTION

instruction stream generators use an open-loop approach, therefore only give guidance to target specific instruction types and values, without the processor under verification (PUV) modules.

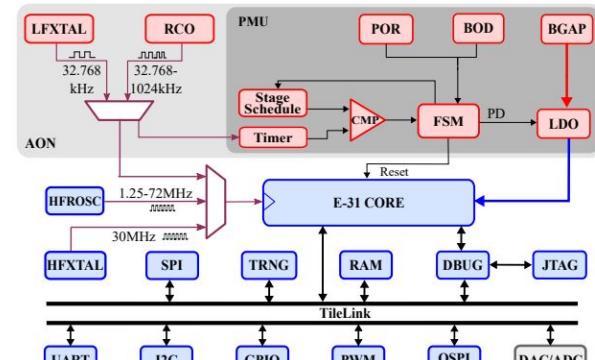
In this paper, we describe a verification methodology for RISC-V-based processors, that eliminates the need for time-consuming human dependent testbenches by employing complementary and reliable verification methods. We demonstrated the methodology by finding non-trivial bugs in a 32-bit RISC-V ISA based processor. These bugs were not identified despite performing several field-programmable gate array (FPGA) C program demos. In contrast, the proposed methodology detected bugs by the generation of high coverage test programs. By conducting code coverage metrics extractions, we demonstrate the increase of code

An Energy-Efficient RISC-V RV32IMAC Microcontroller for Periodical-Driven Sensing Applications

Ckristian Duran¹, Megan Wachs², Luis E. Rueda G.¹, Albert Huntington², Javier Ardila¹, Jack Kang², Andres Amaya¹, Hector Gomez¹, Juan Romero¹, Laude Fernandez¹, Felipe Flechas¹, Rolando Torres¹, Juan Moya¹, Wilmer Ramirez¹, Julian Arenas¹, Juan Gomez¹, Hanssel Morales¹, Camilo Rojas¹, Alex Mantilla¹, Elkim Roa¹ and Krste Asanovic^{2,3}

¹OnChip - Universidad Industrial de Santander, Colombia, ²SiFive, Inc. - San Mateo, CA, ³U. California at Berkeley, CA

Abstract—Reported work on minimum-energy (ME) computing for low-power applications has focused entirely on tracking solely the microprocessor ME voltage supply. However, the use of low-power systems requires accounting for regulator losses, voltage monitors, biasing, peripheral, clock sources, and start-up energies to adapt the correct ME supply to different operation modes. Here we demonstrate a 32-bit RISC-V IMAC based microcontroller (MCU) in 180nm CMOS technology featuring a low-energy always-on (AON) subsystem extending on ME adaption by including peripherals. AON peripherals enable the MCU for low-duty-cycle sensor node applications. Low-energy clock sources and voltage monitors enable 32.768kHz to 55MHz operation and power-gate the MCU into three power states adjusted to work at the ME supply operation. Measured start-up energies using integrated RC-based oscillators show restarting



On the Cross-Correlation Based Loop Gain Adaptation for Bang-Bang CDRs

Javier Ardila[✉], *Student Member, IEEE*, Hanssel Morales, *Student Member, IEEE*, and Elkim Roa, *Member, IEEE*

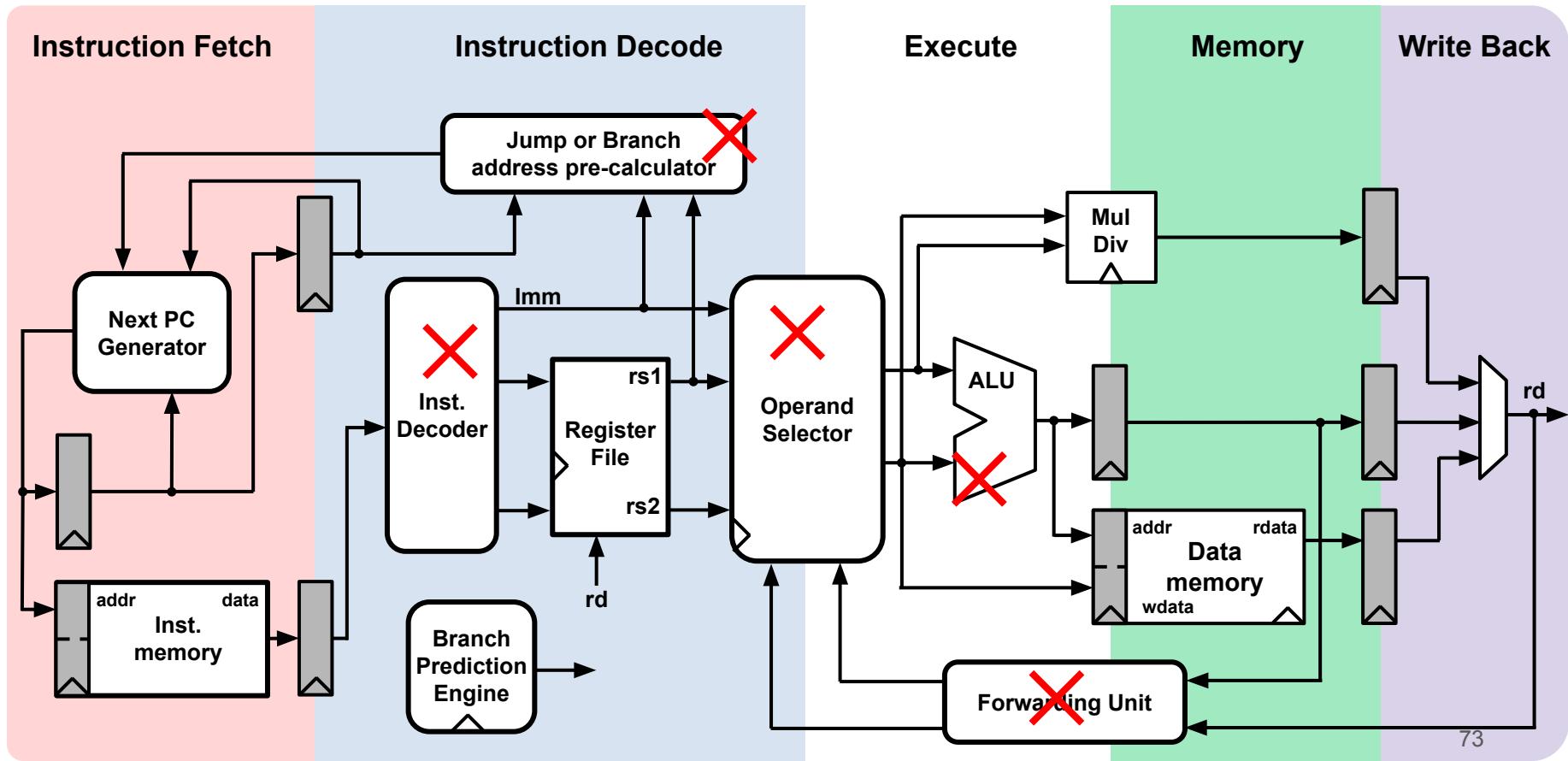
Abstract—This article describes the cross-correlation function as an alternative method to adapt loop gain in digital CDRs. Cross-correlation function inherent properties and their link with the cross-power spectral density help consolidating the adaptive loop gain technique, XCALG, for clock and data recovery systems that implement bang-bang phase detector. Considering the modulation of the loop gain due to jitter noise, previous works exploit the autocorrelation function at the output of the bang-bang phase detector as a proper indicator to track the system dynamics and perform loop gain adaptation. In contrast, we propose the XCALG as a new alternative to perform loop gain adaptation featuring better observability, less impact from jitter sources while keeping a safe phase margin. Theory behind the idea is enhanced, and the XCALG is demonstrated through behavioral system simulations. In addition, preliminary implementation costs of the cross-correlation function are discussed using a 65nm CMOS technology node.

Index Terms—CDR, loop gain adaptation, cross-correlation, clock and data recovery, correlation, autocorrelation, XCALG.

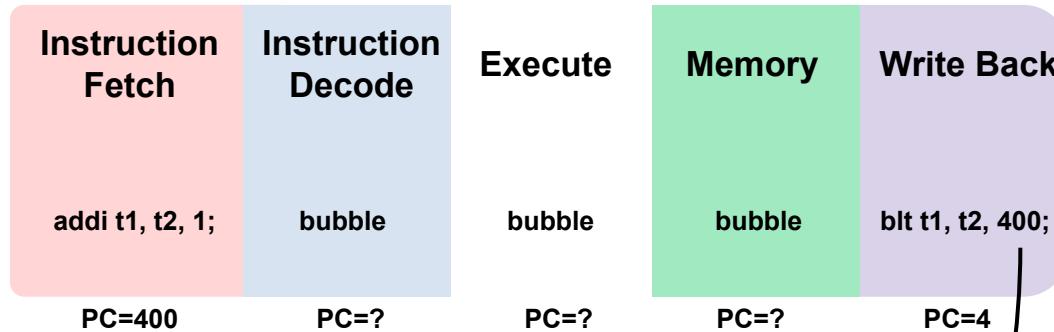
introduce an algorithm to perform gain optimization using autocorrelation with the mean-squared-error (MSE) criterion. They demonstrate a criterion for CDR lock in terms of the power spectral density (PSD) by looking for the sign of the autocorrelation in BBPD output (hereafter $R_X(n)$) at the $D + 1$ point, where D is the loop delay. The point $n = D + 1$ falls close to the first zero-crossing point in $R_X(n)$, and even small variations in D could generate different signs in the $R_X(D + 1)$ evaluation, making this criterion sensitive to small variations in loop latency. In other words, the actual zero-crossing point will be different from $D + 1$ even for small latency variations. Nonetheless, the major concern is not about the difference between $D + 1$ and the actual zero-crossing point of $R_X(n)$, but the fact that at this point $R_X(D + 1) \approx 0$ even for a system with poor phase margin (PM).

[✉] Author to whom all correspondence should be addressed.

Bugs Detected



Branch Execution Without Prediction



```
if (t1 < t2)
{
    NEXT_PC = PC + 396
}
else
{
    NEXT_PC = PC + 4
}
```

Iterative Loops Structure

C code

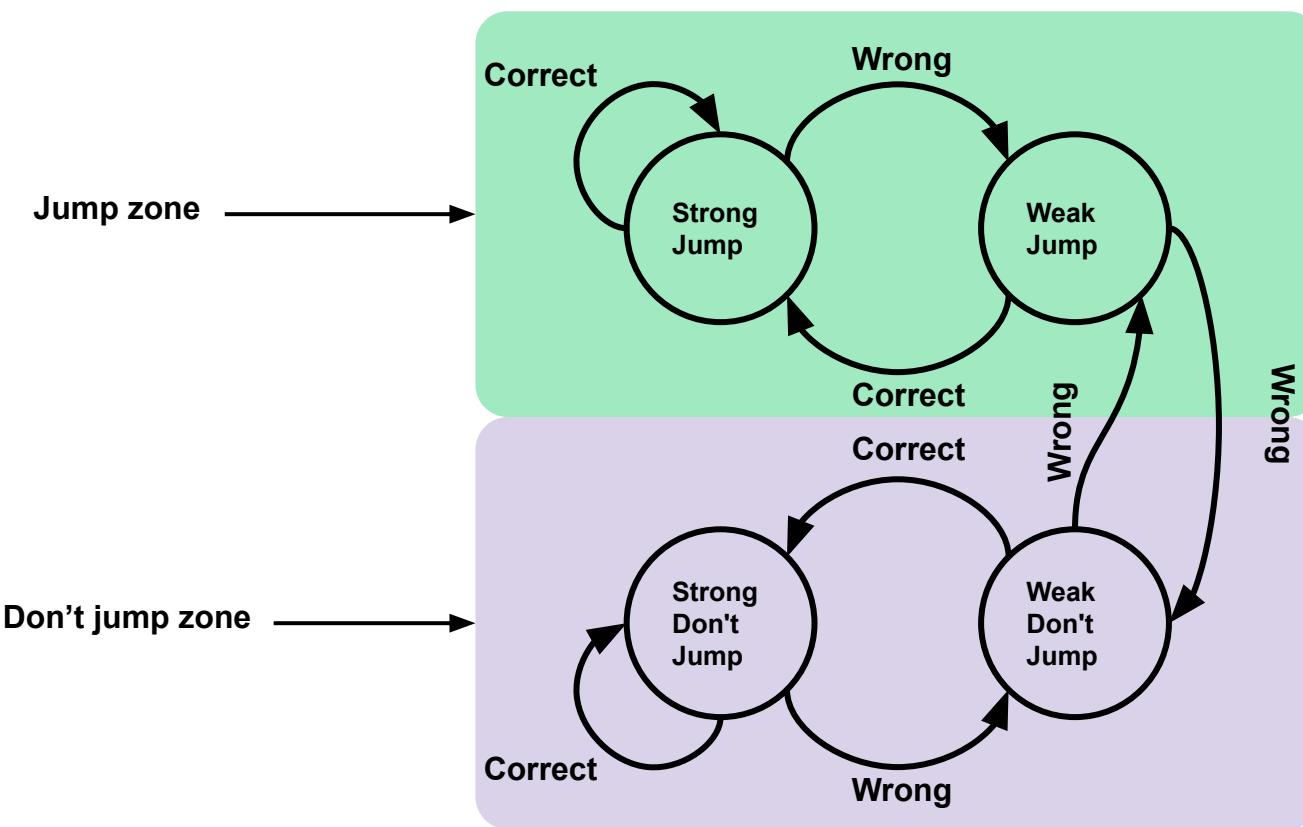
```
for(i=0;i<1000; i++)  
{  
    //some operation  
}
```

Equivalent risc-v assembler

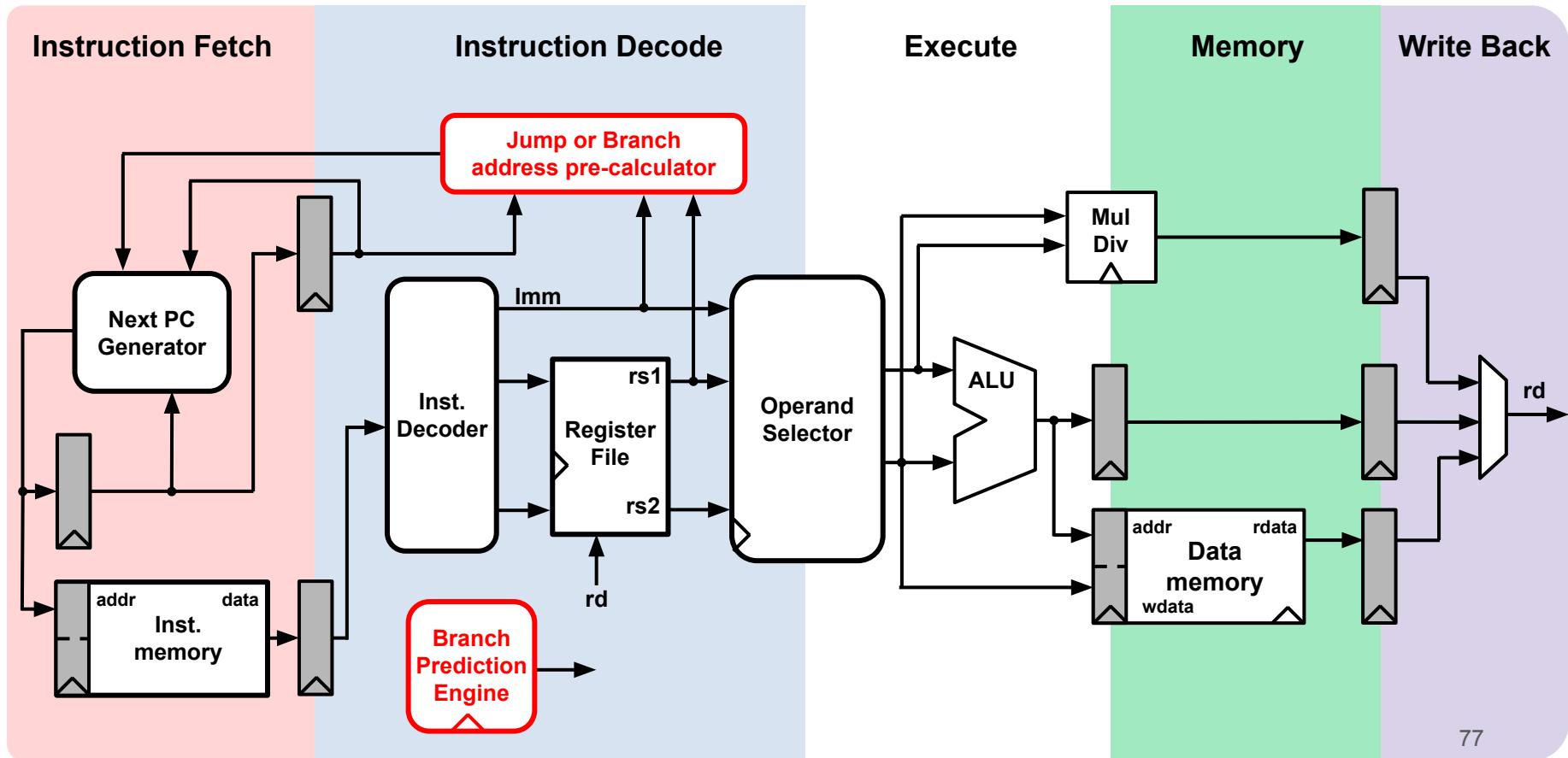
```
li t1, 0  
li t2, 1000  
for:  
    blt t1, t2, 400  
    //some operation  
    addi t1, t1, 1  
j for
```

1000 times the same
branch decision!

Branch Prediction History FSM



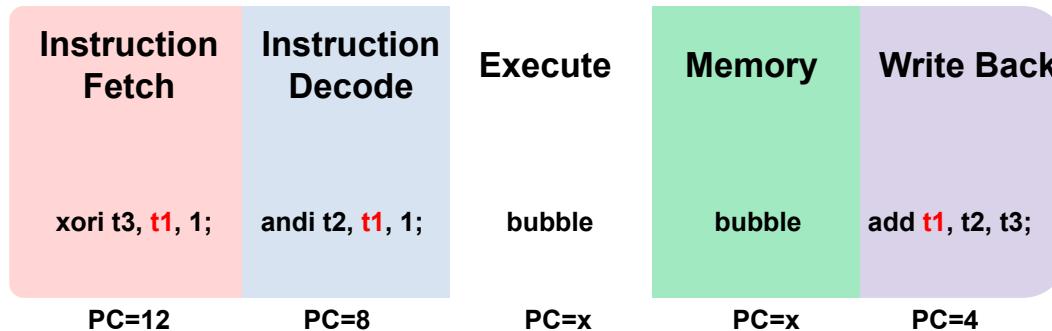
Acceleration Strategies



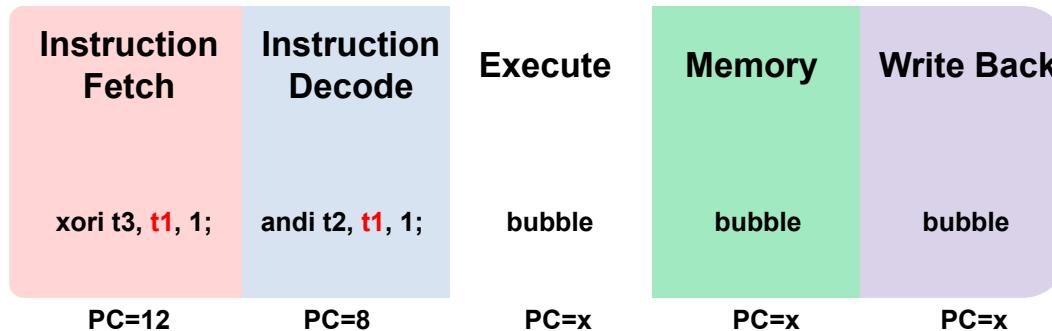
IPC Penalties

Types	Worst Case Cycles Stalled	Cycles Reduced
Jumps	1	2
Branches	1	2

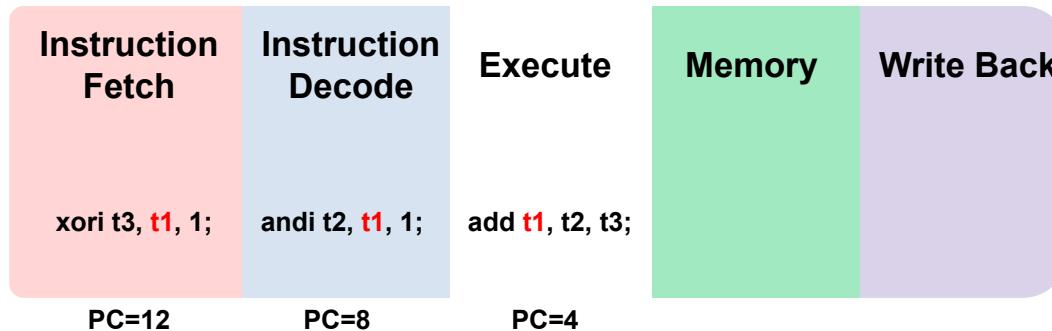
Instruction Dependency



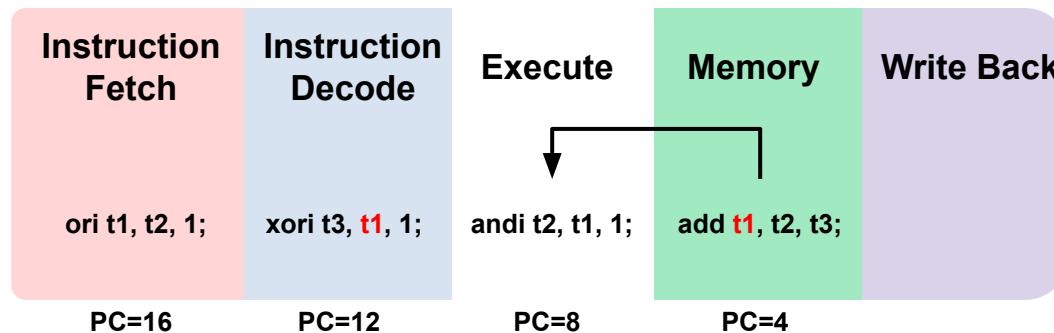
Instruction Dependency



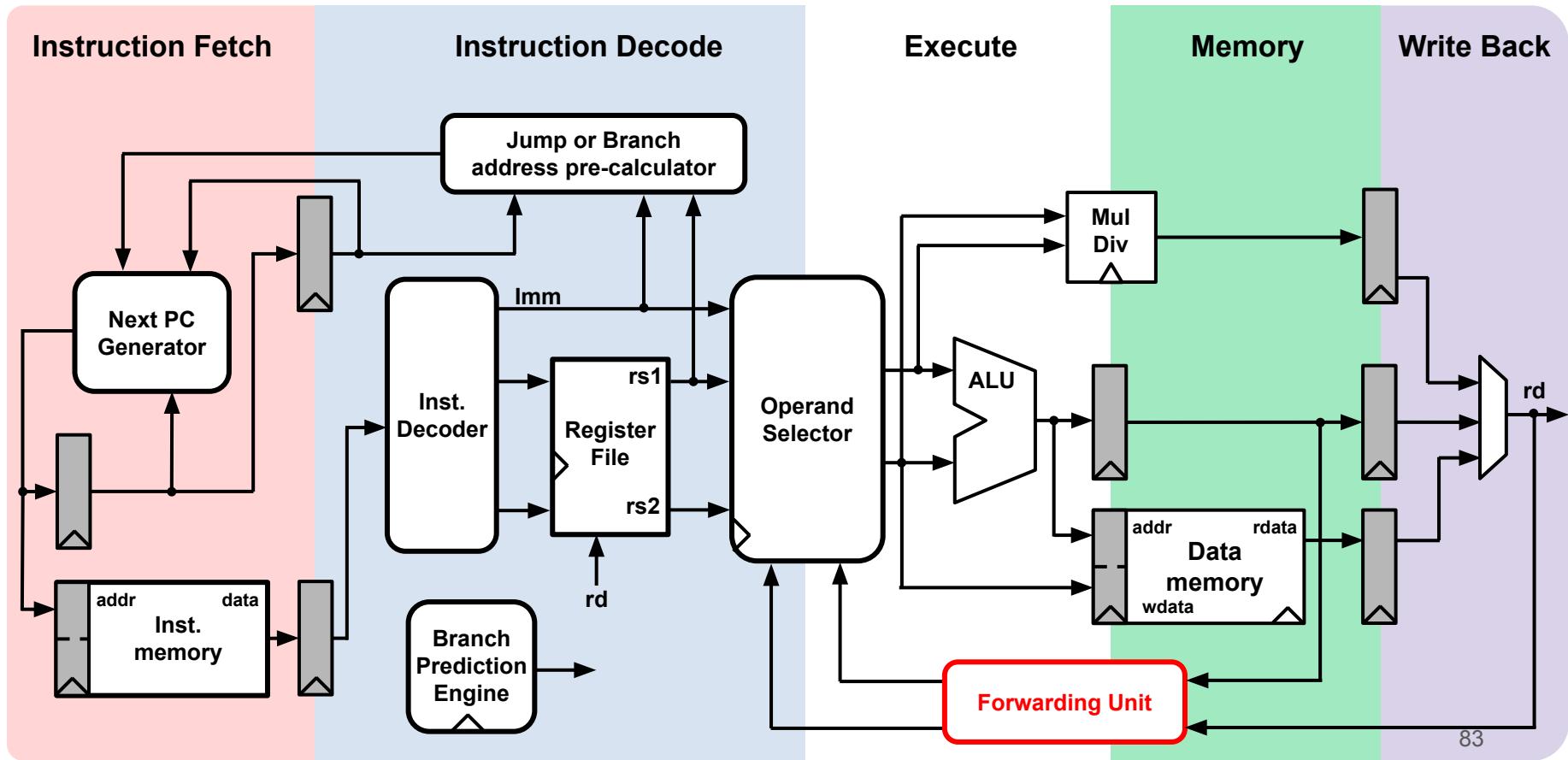
Instruction Dependency



Forward Data



Acceleration Strategies



Energy Calculation

$$\text{Coremarks/MHz} = \frac{\text{Iterations}}{\text{Time}_{\text{seconds}} * F_{\text{MHz}}}$$

$$(\text{Coremarks/MHz}) * F_{\text{MHz}} = \frac{\text{Iterations}}{\text{Time}_{\text{seconds}}}$$

Defining $\Rightarrow F_{\text{MHz}} = 1$

$$\frac{\text{Time}_{\text{seconds}}}{\text{Iterations}} = \frac{1}{(\text{Coremarks/MHz})}$$

$$\frac{\text{Time}_{\text{seconds}} * \text{Power}_{\text{watts}}}{\text{Iterations}} = \frac{\text{Power}_{\text{watts}}}{(\text{Coremarks/MHz})}$$

$$\frac{\text{Energy}_{\text{Joules}}}{\text{Iterations}} = \frac{\text{Power}_{\text{watts}}}{(\text{Coremarks/MHz})}$$