DESIGN, AUTOMATION & TEST IN EUROPE
14 – 15 March 2022 · on-site event
16 – 23 March 2022 · online event
The European Event for Electronic System Design & Test

Design and Evaluation Frameworks for Advanced RISC-based Ternary Processor

D. Kam, J. G. Min, J. Yoon, S. Kim, S. Kang and Y. Lee

Department of Electrical Engineering

Pohang University of Science and Technology (POSTECH), South Korea

Value Creating University POSTECH EPIC LAB

Biography

Email address
rkaehddbs@postech.ac.kr
Personal information:
https://sites.google.com/view/epiclab/member/dykam?authuser=0

Dongyun Kam

- Ph.D. student
  - Department of Electrical Engineering
  - Pohang University of Science and Technology (POSTECH)
  - South Korea

- Interesting topics
  - 5G/B5G wireless communication systems
  - Error correcting codes
  - VLSI circuit designs
  - Ternary processors
  - RISC-V processors

Outline

- Introduction
- Ternary number systems and operators
- Proposed frameworks for ternary processors
- Proposed ART-9 core design
- Simulation results
- Conclusion

Outline

- **Introduction**
- Ternary number systems and operators
- Proposed frameworks for ternary processors
- Proposed ART-9 core design
- Simulation results
- Conclusion

Introduction

- **Downscaling digital circuits met limitation**
  - **Limitation caused by increased interconnecting/routing overhead**

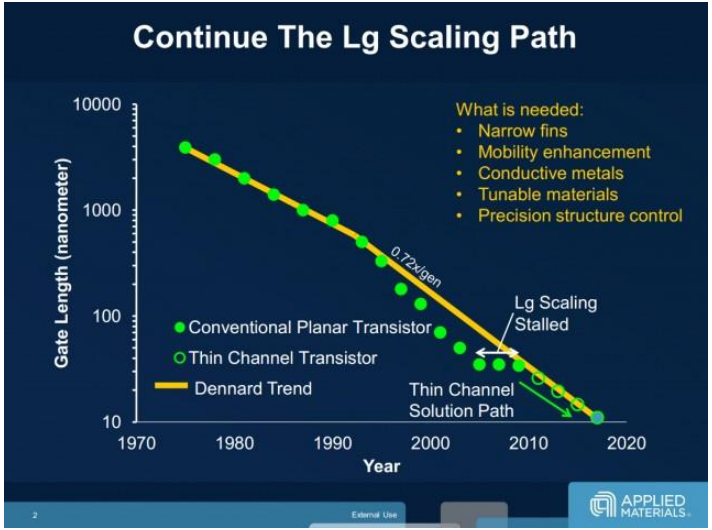


Fig.1. Down-scaling of CMOS technology [R1]

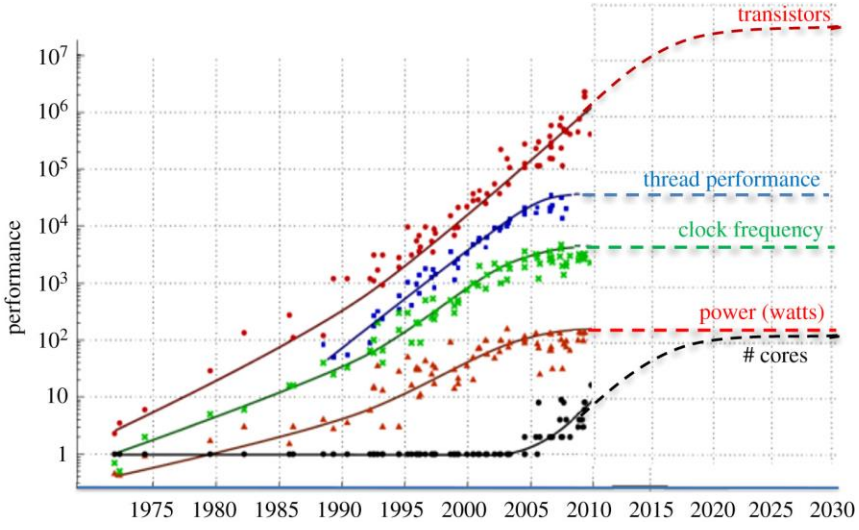


Fig.2. Change of the CPU perfoamcnes [R2]

[R1] https://www.extremetech.com/computing/162376-7nm-5nm-3nm-the-new-materials-and-transistors-that-will-take-us-to-the-limits-of-moores-law
[R2] https://royalsocietypublishing.org/doi/10.1098/rsta.2019.0061

Introduction

- **Multi-valued logic (MVL) circuits & Ternary processor**
  - **MVL based circuits can be the solution of addressing limitation**

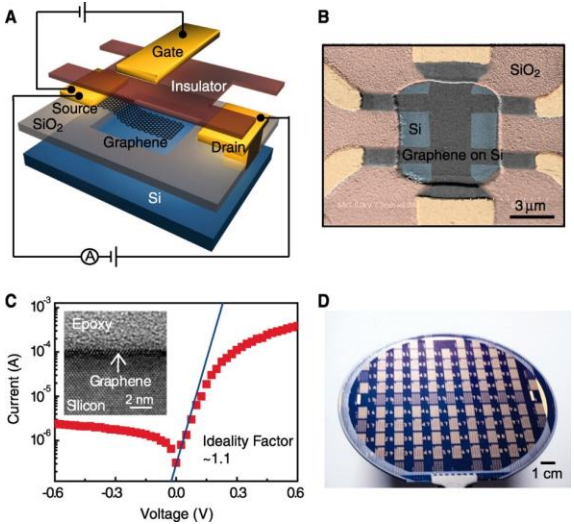


Fig.3. Graphene barristors [R1]

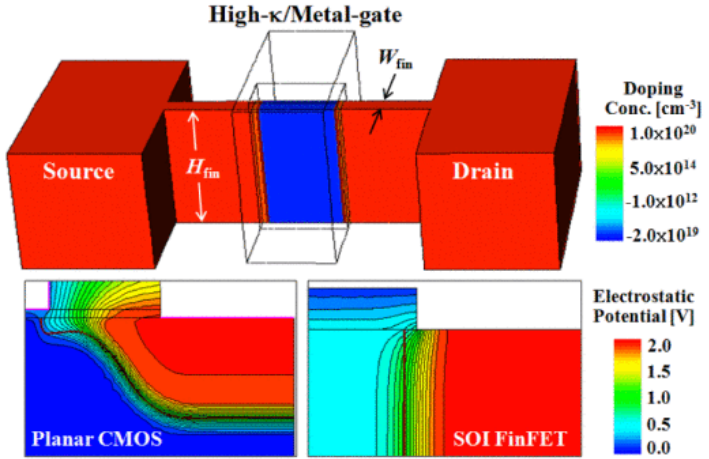


Fig.4. Carbon nanotube field-effect transistor (CNTFET) [R2]

[R1] H. Yang et al., *Science*, 2012
[R2] S. Lin et al., *IEEE MWSCAS*, 2009.

Outline

- Introduction
- **Ternary number systems and operations**
- Proposed frameworks for ternary processors
- Proposed ART-9 core design
- Simulation results
- Conclusion

Ternary Number Systems

- 1 trit : three voltage levels (GND, VDD/2, VDD)
- Fixed-point number systems
  - n-trit number $X = (x_{n-1}, x_{n-2}, \dots, x_0)_3$

$$Y_{integer} = \sum_{k=0}^{n-1} x_k * 3^k$$

  - Unsigned number : $x_i \in \{0, 1, 2\}$
  - Signed (balanced) number : $x_i \in \{-1, 0, 1\}$

Ternary Operators

- **Ternary-based logical operations**
  - **Based on balanced number system**

2-input AND

Input 2 \ Input 1	-1	0	1
-1	-1	-1	-1
0	-1	0	0
1	-1	0	1

2-input OR

Input 2 \ Input 1	-1	0	1
-1	-1	0	1
0	0	0	1
1	1	1	1

2-input XOR

Input 2 \ Input 1	-1	0	1
-1	-1	0	1
0	0	0	0
1	1	0	-1

1-input STI

Input	-1	0	1
Output	1	0	-1

1-input NTI

Input	-1	0	1
Output	1	-1	-1

1-input PTI

Input	-1	0	1
Output	1	1	-1

Fig. 1. Truth tables of ternary logic operations

Ternary Operators

- **Ternary-based arithmetic operations**
  - **Based on RISC-based binary processors**
  - **The existing arithmetic operators**
    - **Ternary adder [R1]**
    - **Ternary comparator [R2]**
    - **Ternary divider [R2]**
    - **Ternary multiplier [R3]**

[R1] S. Heo et al. *IEEE Electron Device Lett.*, 2018.
[R2] B. Parhami et al., *ACSSC*, 2013.
[R3] Y. Kang et al, *IEEE ISMVL*, 2017.

Previous Works

- **Existing ternary processors**
  - **CISC-based 4-trit processor (Fig. 5)**
  - **RISC-based 9-trit processor (Fig. 6)**

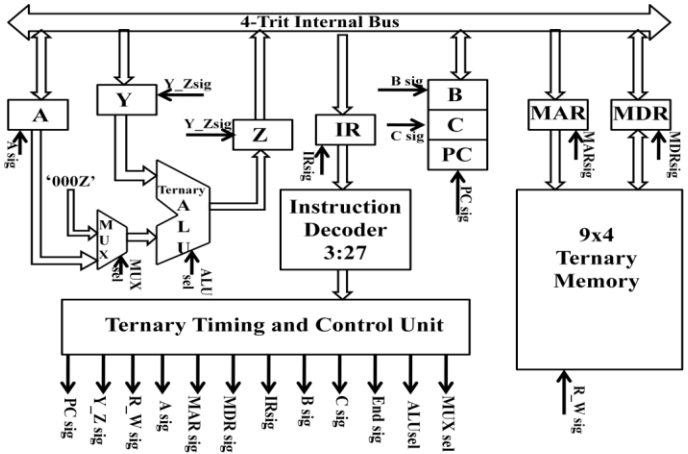


Fig. 5. CISC-based 4-trit processor [R1]

The Trillium Architecture

Part of http://homepage.cs.uiowa.edu/~dwjones/ternary/
by Douglas W. Jones
THE UNIVERSITY OF IOWA Department of Computer Science

Abstract

The Trillium instruction set is designed for a small ternary computer with a 9-trit word and a 19,68 suitable for a microcontroller or programmable interface processor in modern terms.

Warning: The material presented here is very preliminary. First posting on the web, Nov. 21, 2017.

- Data Representations
- Instruction Format
- Addressing Modes
- Registers
- Processor Status Word
- Instructions
  - Store
  - Load
  - Load Immediate
  - Add
  - Subtract
  - Add With Carry
  - Subtract With Borrow
  - Shift Left
  - Shift Right Unsigned
  - Shift Right Signed

Fig.6. RISC-based 9-trit ternary processor [R2]

[R1] S. Narkhede et al., *IJCA*, 2013
[R2] https://homepage.divms.uiowa.edu/~jones/ternary/trillium.shtm

Limitation of Previous Works

- **Lack of open-literature**
  - **Only few papers for ternary processors.**
- **Not considering the technology-level design**
  - **Previous ternary processors are just designed for functionality.**
- **Inefficient ternary processor**
  - **CISC-based ternary processor**
  - **Inefficient instructions and architecture**

What we propose

- **Frameworks to design and evaluate ternary processors**
  - **Software-level compiling framework**
  - **Hardware-level evaluation frameworks**

- **Advanced RISC-based ternary 9-trit processor (ART-9)**
  - **Optimized instruction set architecture (ISA)**
  - **5-stage pipelined core architecture**

Outline

- Introduction
- Ternary number systems and operators
- **Proposed frameworks for ternary processors**
- Proposed ART-9 core design
- Simulation results
- Conclusion

Proposed Frameworks

- **Software-level compiling framework**
  - **Given ternary-based ISA**
  - **Converting C-level codes into ternary assembly**
  - **Adopting RV-32I ISA and the existing GCC [R1]**

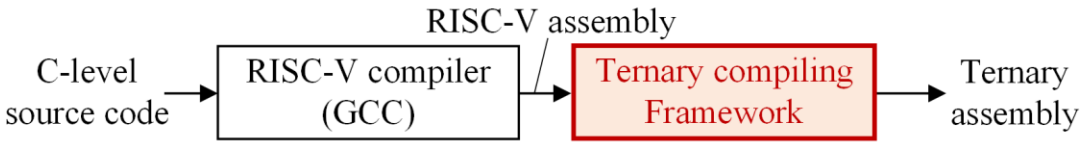


Fig.7. Overall procedure of the proposed softwre-level framework

Proposed Frameworks

- **Software-level compiling framework**
  - **Three compiling steps**
    - **Instruction mapping, operand conversion, redundancy check**

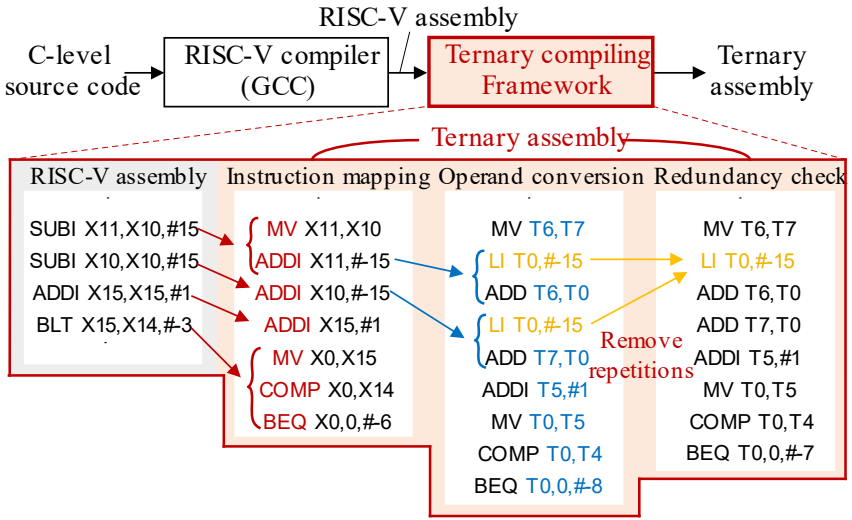


Fig.8. Proposed software-level compiling framework

Proposed Frameworks

- **Software-level compiling framework**
  - **Instruction mapping**

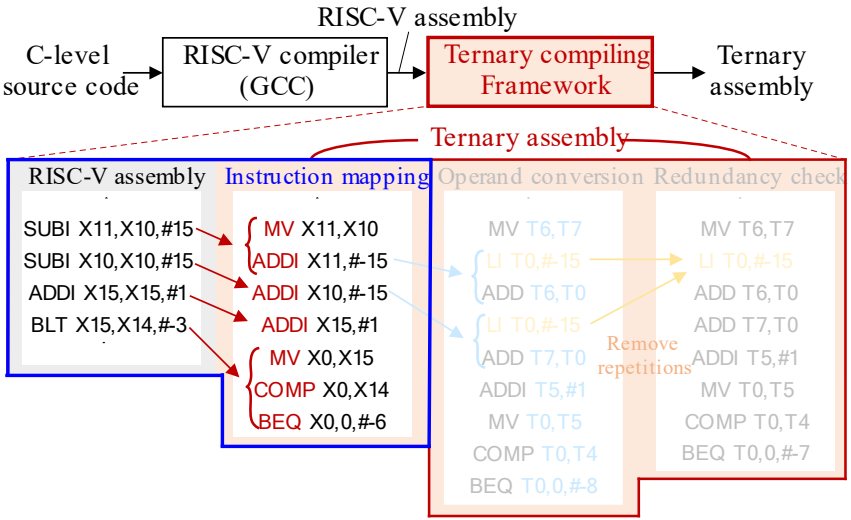


Fig.9. Instruction mapping step in the proposed framework

Proposed Frameworks

- **Software-level compiling framework**
  - **Operand conversion**

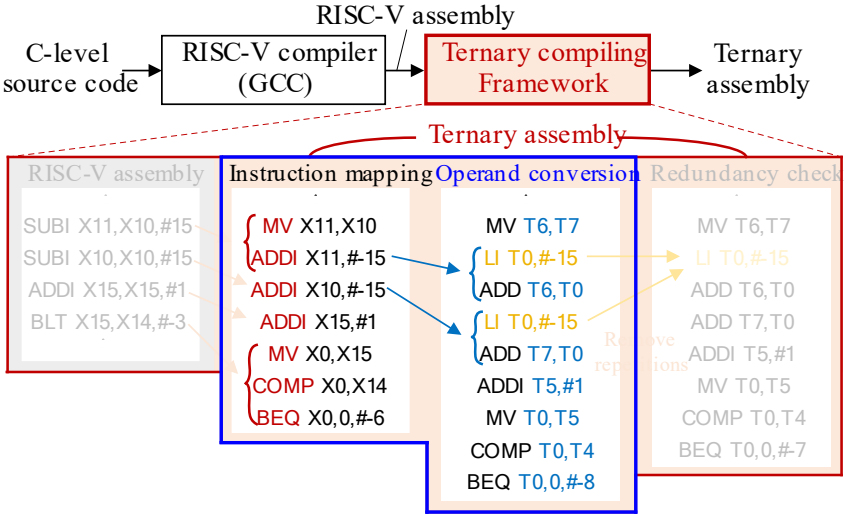


Fig.10. Operand conversion step in the proposed framework

Proposed Frameworks

- **Software-level compiling framework**
  - **Redundancy check**

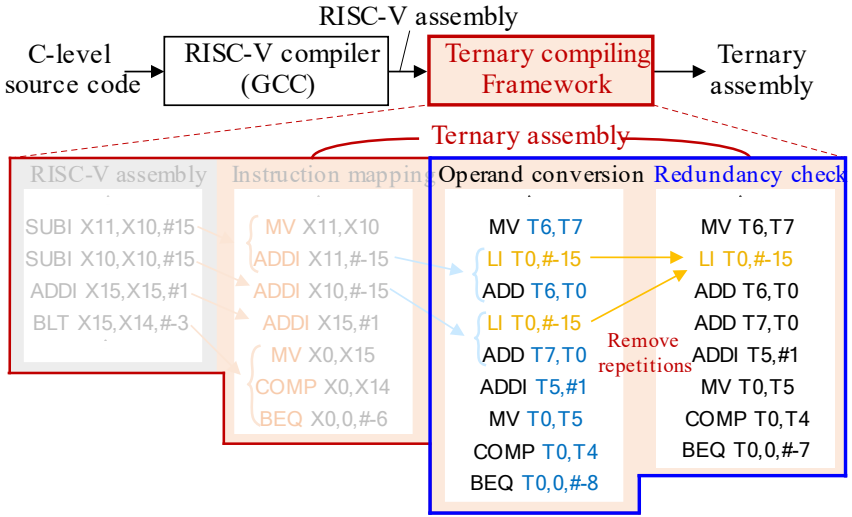


Fig.11. Redundancy check in the proposed framework

Proposed Frameworks

- **Hardware-level evaluation framework**
  - **Considering the practical implementation**

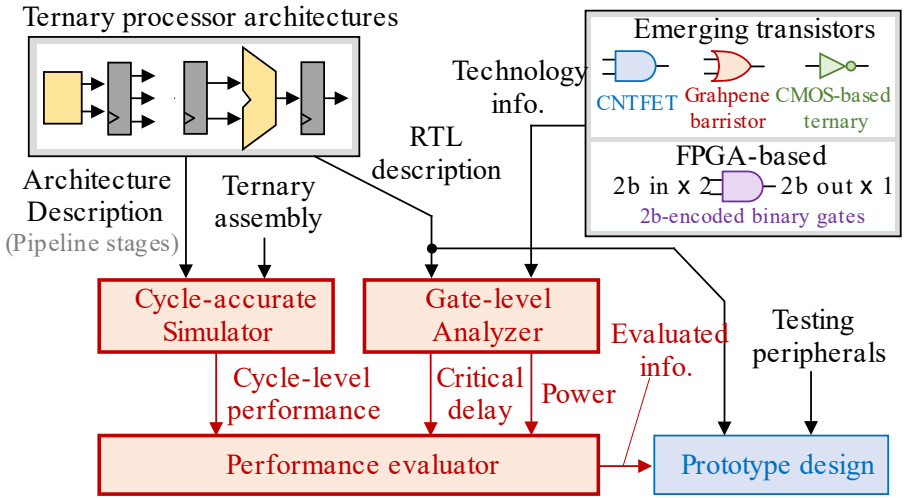


Fig.12. Proposed hardware-level evaluation framework

Proposed Frameworks

- **Hardware-level evaluation framework**
  - **Cycle-accurate simulator**

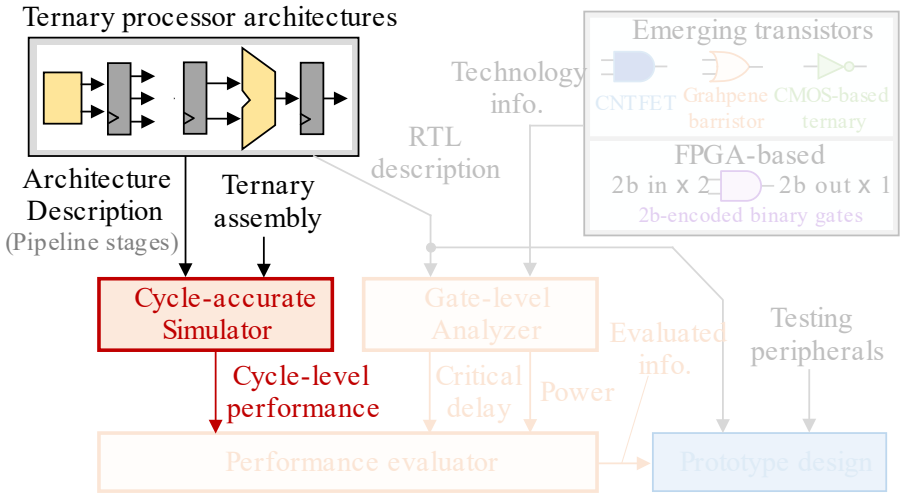


Fig.13. Proposed cycle-accurate simulator

Proposed Frameworks

- **Hardware-level evaluation framework**
  - **Gate-level analyzer**

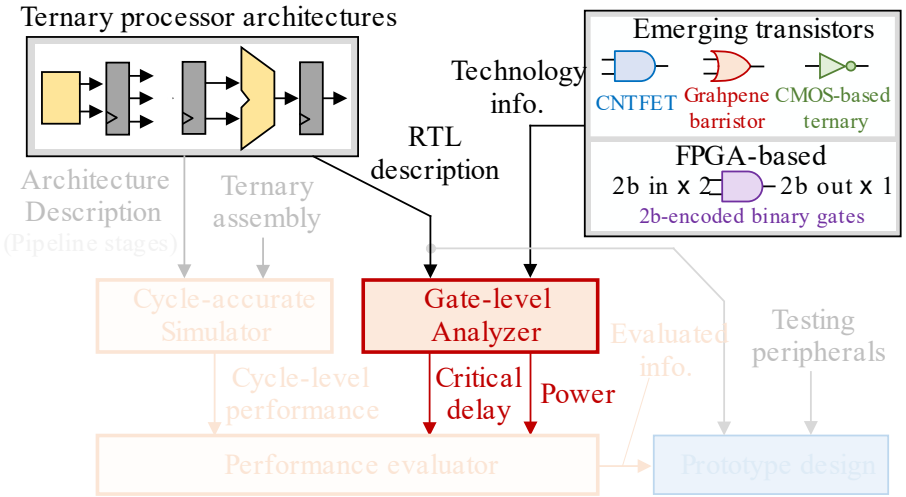


Fig.14. Proposed gate-level analyzer using technology information

Proposed Frameworks

- **Hardware-level evaluation framework**
  - **Total performance evaluator**

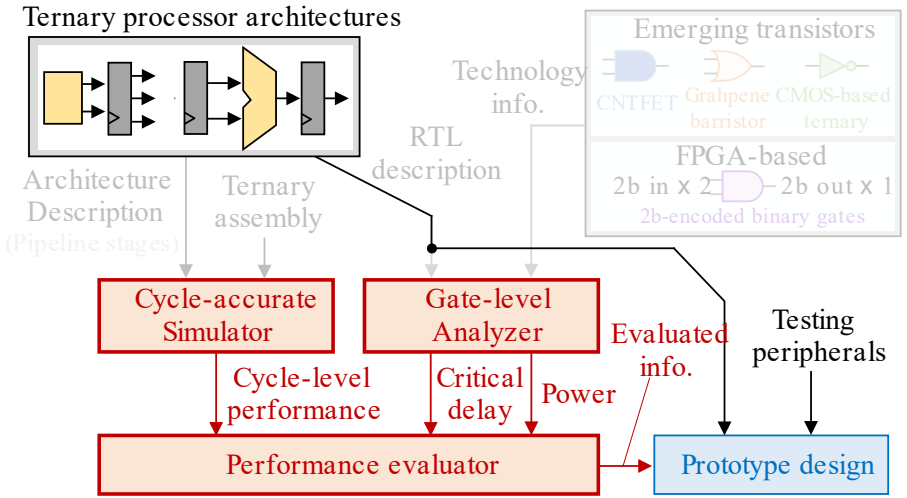


Fig.15. Prototype design using performance evaluator

Outline

- Introduction
- Ternary number systems and operators
- Proposed frameworks for ternary processors
- **Proposed ART-9 core design**
- Simulation results
- Conclusion

Proposed Processor Design

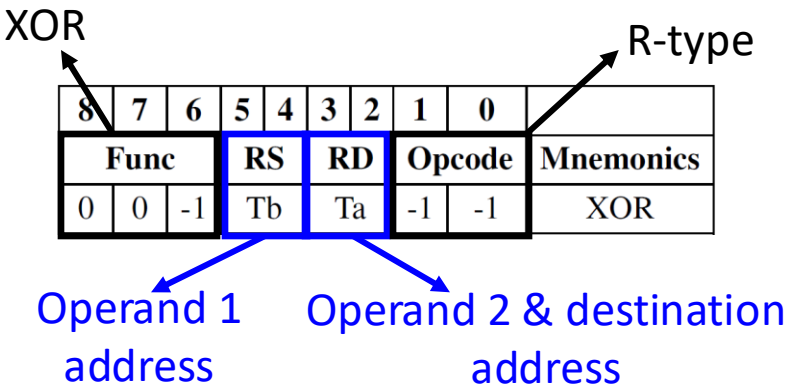
- **Advanced RISC-based ternary 9-trit processor (ART-9)**
  - **Properties of contempoaray RISC-type binary processors.**
  - **ART-9 instruction set architecture (ISA)**
  - **Pipelined core architecture**

Proposed Processor Design

- **ART-9 ISA**
  - **9-trit length instructions**
    - 24 instructions
    - 2-trit opcode
    - R-type, I-type, B-type, M-type
    - Ternary instruction memory (TIM)
  - **9-trit length data**
    - Ternary data memory (TDM)
  - **Nine 9-trit length registers**
    - General-purpose ternary register file (TRF)
    - Program counter (PC) register

Proposed Processor Design

- **ART-9 ISA**
  - **9-trit length instructions**
    - **R-type (12 logical/arithmetic operations)**
    - **Register addressing**
    - **Two operands**



Type	9-trit instructions	Operation
R	MV Ta,Tb	TRF[Ta] = TRF[Tb]
	PTI Ta,Tb	TRF[Ta] = PTI(TRF[Tb])
	NTI Ta,Tb	TRF[Ta] = NTI(TRF[Tb])
	STI Ta,Tb	TRF[Ta] = STI(TRF[Tb])
	AND Ta,Tb	TRF[Ta] = TRF[Ta] & TRF[Tb]
	OR Ta,Tb	TRF[Ta] = TRF[Ta] | TRF[Tb]
	XOR Ta,Tb	TRF[Ta] = TRF[Ta] $\oplus$ TRF[Tb]
	ADD Ta,Tb	TRF[Ta] = TRF[Ta] + TRF[Tb]
	SUB Ta,Tb	TRF[Ta] = TRF[Ta] − TRF[Tb]
	SR Ta,Tb	TRF[Ta] = TRF[Ta] $\gg$ TRF[Tb][1:0]
	SL Ta,Tb	TRF[Ta] = TRF[Ta] $\ll$ TRF[Tb][1:0]
	COMP Ta,Tb	TRF[Ta] = *compare*(TRF[Ta],TRF[Tb])

Proposed Processor Design

- **ART-9 ISA**
  - **9-trit length instructions**
    - **I-type (6 immediate-based processing)**
    - **Supporting 2~5-trit immediate values**

8	7	6	5	4	3	2	1	0	
Func/imm			**imm**		**RD**		**Opcode**		**Mnemonics**
1	0	imm[2]	imm[1:0]		Ta		-1	-1	ADDI
⋮									
imm[8:6]			imm[5]	0	Ta		-1	0	LUI
imm[2:0]			imm[4:3]		Ta		1	-1	LI

Various imm length I-type

Type	9-trit instructions	Operation
I	ANDI Ta,imm	TRF[Ta] = TRF[Ta] & imm[2:0]
	ADDI Ta,imm	TRF[Ta] = TRF[Ta] + imm[2:0]
	SRI Ta,imm	TRF[Ta] = TRF[Ta] ≫ imm[1:0]
	SLI Ta,imm	TRF[Ta] = TRF[Ta] ≪ imm[1:0]
	LUI Ta,imm	TRF[Ta] = {imm[3:0],00000}
	LI Ta,imm	TRF[Ta] = {TRF[Ta][8:5],imm[4:0]}

Proposed Processor Design

- **ART-9 ISA**
  - **9-trit length instructions**
    - **B-type (4 jump/branch instructions)**
    - **Calculating next PC values with an immediate**

8	7	6	5	4	3	2	1	0	
Func/imm			**imm/RS**		**RD**		**Opcode**		**Mnemonics**
imm[2:0]			B	imm[3]	Ta		0	-1	BEQ
imm[2:0]			B	imm[3]	Ta		0	0	BNE
imm[2:0]			imm[4:3]		Ta		1	0	JAL
imm[2:0]			Tb		Ta		1	1	JALR

Various imm length B-type

Type	9-trit instructions	Operation
B	BEQ Ta,B,imm	PC = PC + imm[3:0] if TRF[Ta][0] == B
	BNE Ta,B,imm	PC = PC + imm[3:0] if TRF[Ta][0] != B
	JAL Ta,imm	TRF[Ta] = PC+1, PC = PC + imm[4:0]
	JALR Ta,Tb,imm	TRF[Ta] = PC+1, PC = TRF[Tb]+imm[2:0]

Proposed Processor Design

- **ART-9 ISA**
  - **9-trit length instructions**
    - **M-type (2 memory access instructions)**
    - **Harvard architecture (separate two memory; TIM/TDM)**

8	7	6	5	4	3	2	1	0	
Func/imm			**RS**		**RD**		**Opcode**		**Mnemonics**
imm[2:0]			Tb		Ta		-1	1	LOAD
imm[2:0]			Tb		Ta		0	1	STORE

TRF address

Type	9-trit instructions	Operation
M	LOAD Ta,Tb,imm	TRF[Ta] = TDM[TRF[Tb]+imm[2:0]
	STORE Ta,Tb,imm	TDM[TRF[Tb]+imm[2:0]] = TRF[Ta]

Proposed Processor Design

- **ART-9 core architecture**
  - **Similar to binary RISC-V architecture**
  - **5-stage pipelining**
    - **IF, ID, EX, MEM, WB**

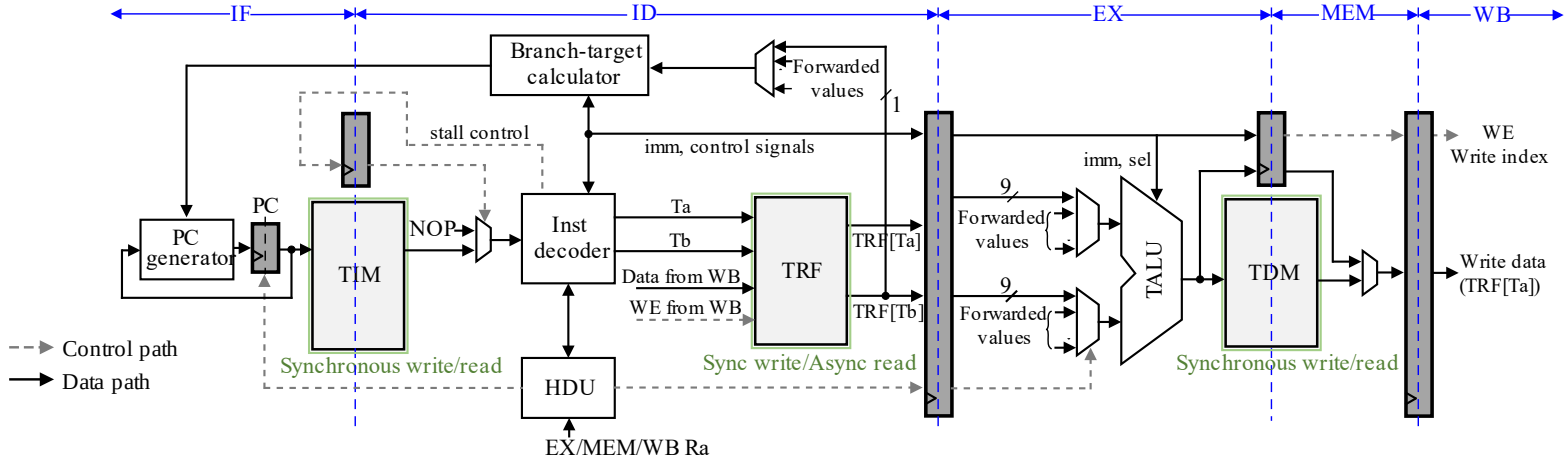


Fig.16. 5-stage pipelined ART-9 core architecture

Proposed Processor Design

- **ART-9 core architecture**
  - **TIM/TDM :Synchronous write/read**
  - **TRF : Synchronous write / Asynchronous read**

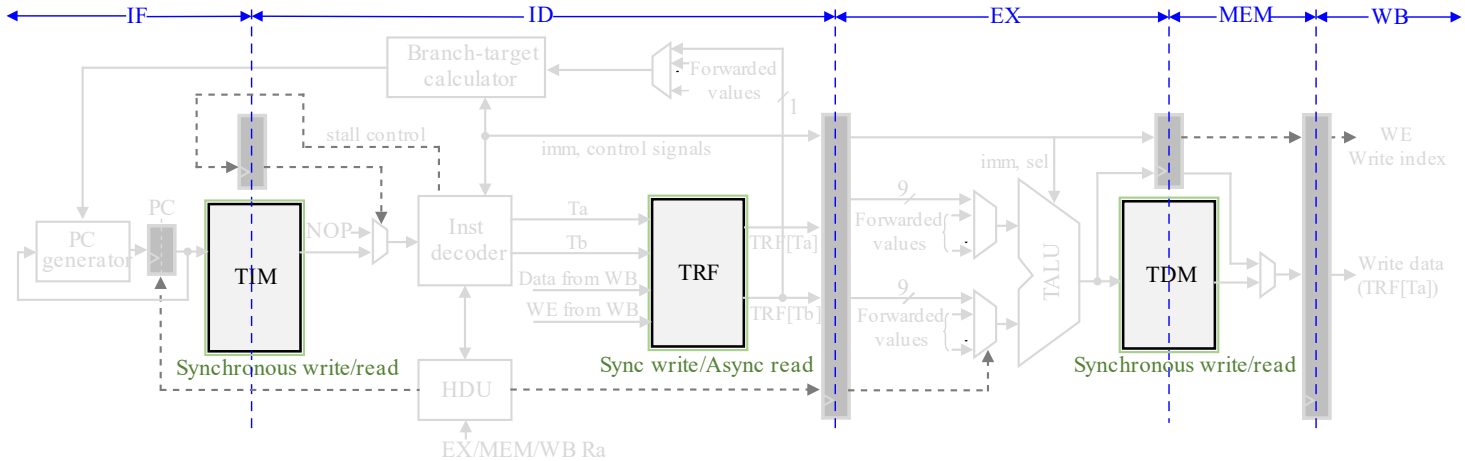


Fig.17. TIM, TRF and TDM for the proposed ART-9 processor

Proposed Processor Design

- **ART-9 core architecture**
  - **Ternary ALU**
    - **Supporting ART-9 ISA**

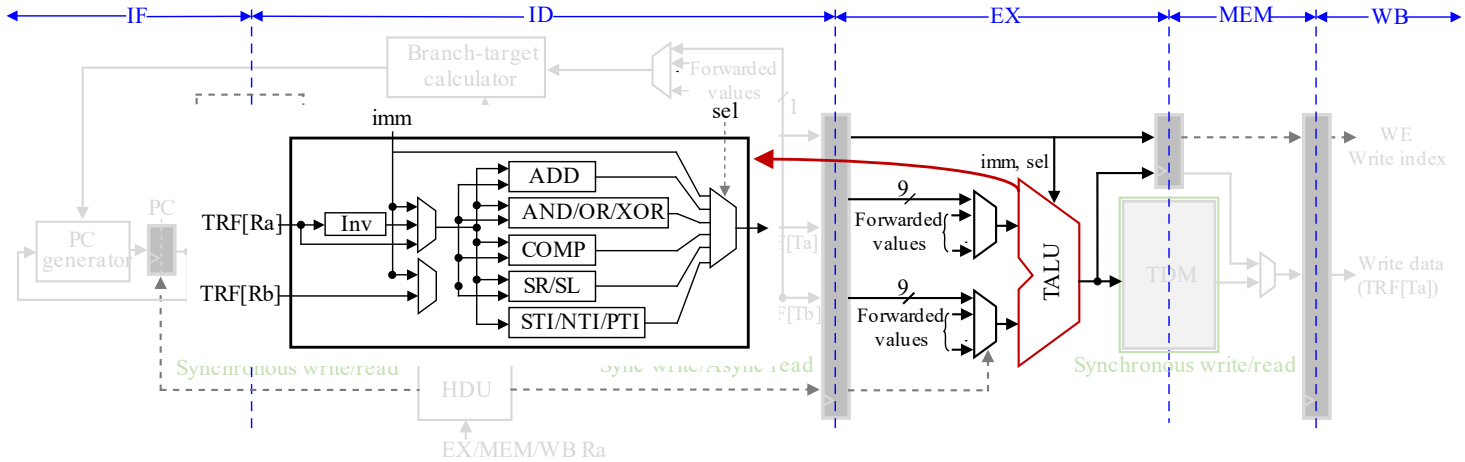


Fig.18. TALU of the proposed ART-9 processor

Proposed Processor Design

- **ART-9 core architecture**
  - **Minimizing stall instructions**
    - **Handling data/control hazards**

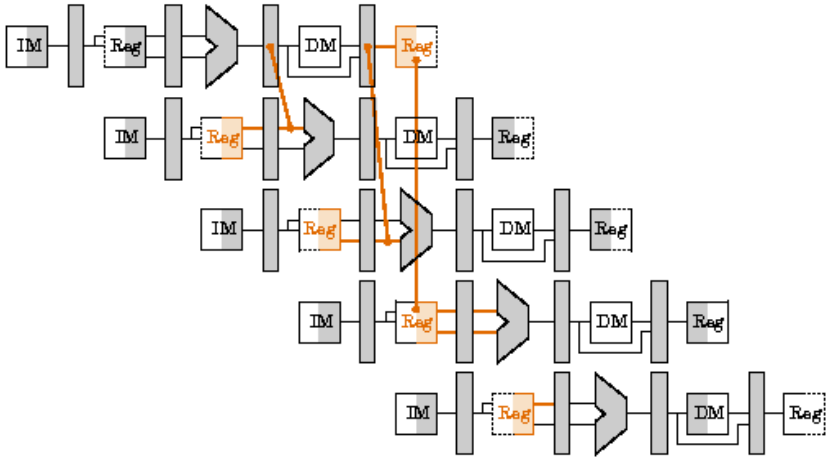


Fig.19. Pipelined processing of binary RISC-V processor [R1]

[R1] https://www.cs.fsu.edu/~hawkes/cda3101lects/chap6/index.html?$$$F6.37.html$$$

Proposed Processor Design

- **ART-9 core architecture**
  - **Hazard control**
    - **Data-forwarding, Load-use/branch hazard (NOP Instruction)**

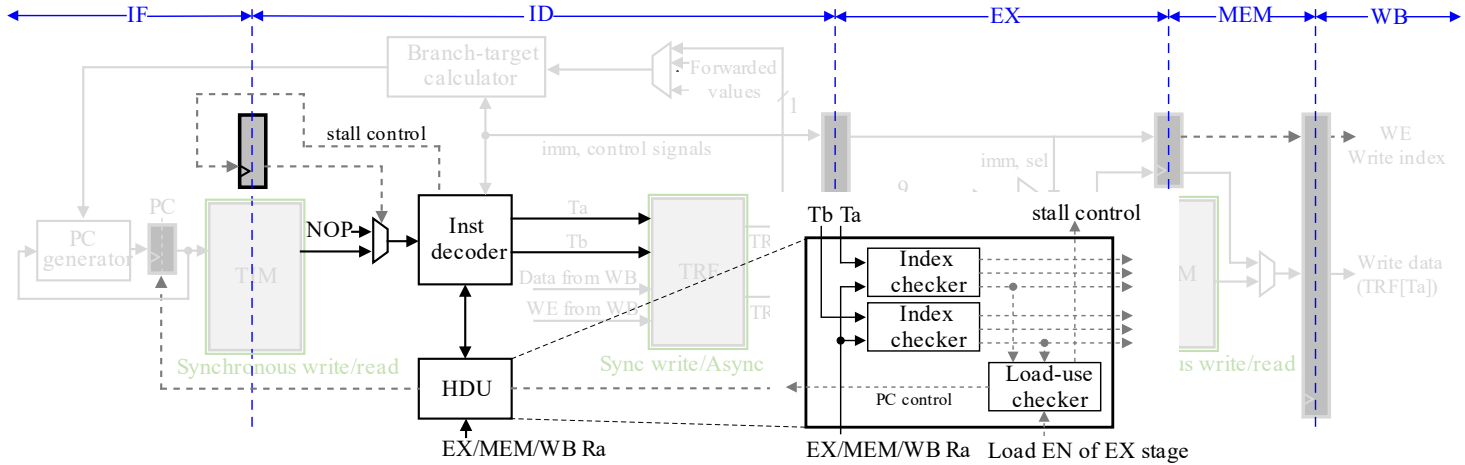


Fig.20. Hazard control for the minimum number of stall instructions

Outline

- Introduction
- Ternary number systems and operators
- Proposed frameworks for ternary processors
- Proposed ART-9 core design
- **Simulation results**
- Conclusion

Simulation Results

- **Benchmark evaluations**
  - **Utilizing the proposed comiling framework**
  - **RV-32I (32b ISA), ARMv6M (16b ISA)**

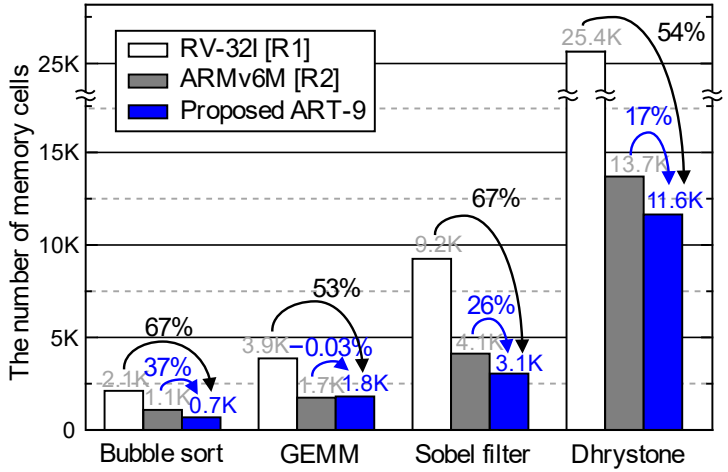


Fig.21. Required memory size for four benchmark programs

[R1] https://riscv.org/specifications/
[R2] J. Yiu et al., "The Definitive Guide to ARM Cortex-M0 and Cortex-M0+ Processors", 2015

Simulation Results

- **Benchmark evaluations**
  - **In terms of processing cycles,**
  - **Comparison with Light-weight processors using RV-32IM [25]**
  - **Fast processing**

	Bubble sort	GEMM	Sobel filter	Dhrystone
This work	2,432	10,748	7,822	134,200
PicoRV32 [R1]	9,227	11,290	18,250	186,607

[R1] https://github.com/cliffordwolf/picorv32

Simulation Results

- **Benchmark evaluations**
  - **Dhrystone benchmark**
  - **Comparison with Light-weight processors using RV-32I(M) [24], [25]**

	This work	VexRiscv [R1]	PicoRV32 [R2]
ISA Architecture	ART-9 ISA	RV-32I	RV-32IM
# of instructions	24	40	48
Pipelined stages	5	5	1
Multiplier	X	O	O
DMIPS/MHz	0.42	0.65	0.31
# of memory-cells	11.6K trits	25.4K bits	23.7K bits

[R1] https://github.com/SpinalHDL/VexRiscv
[R2] https://github.com/cliffordwolf/picorv32

Simulation Results

- **Hardware-level evaluation**
  - **32nm CNTFET ternary models**
    - **Gate-level hardware cost**
    - **Even superior to the neat-threshold ARM Cortex-M3 [R1]**

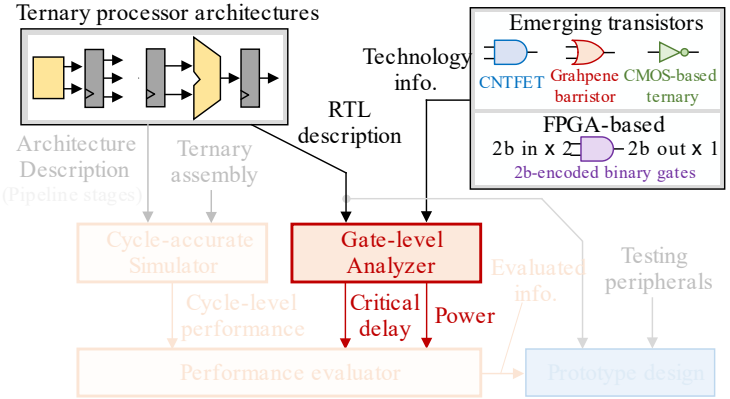


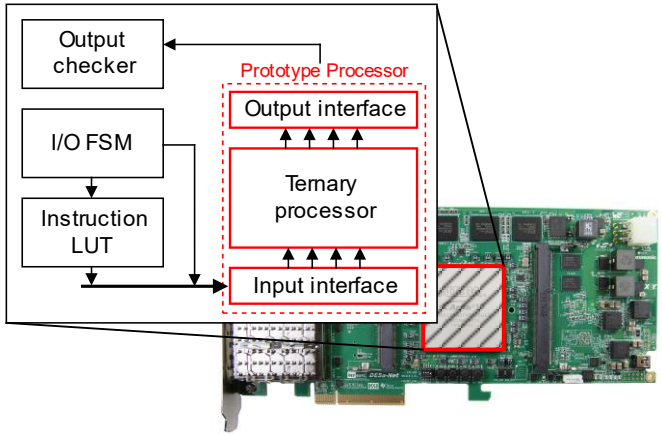
Fig.22. Proposed gate-level analyzer

Voltage	Total gates	Power	DMIPS/W
0.9V	652	42.7 μW	3.06×10^6

[R1] R. G. Dreslinski et al., *IEEE Micro*, 2013

Simulation Results

- **Hardware-level evaluation**
  - **FPGA-based platfrom**
    - **2b-encoded ternary gates**



Voltage	Frequency	ALMs	Registers	RAM	Power
0.9V	150MHz	803	339	9,216 bits	1.09W

Fig.23. FPGA-based evaluation platform

Outline

- Introduction
- Ternary number systems and operators
- Proposed frameworks for ternary processors
- Proposed ART-9 core design
- Simulation results
- **Conclusion**

Conclusion

- There are many transistor-level works for ternary circuits.
- However, only few works have studied ternary processors.
- In this paper,
  - We firstly proposed software-and hardware-level frameworks to design ternary processors.
  - We also proposed advanced RISC-based ternary 9-trit (ART-9) processor, which is designed with ART-9 ISA and pipelined architecture.
  - We finally evaluate the proposed ART-9 core by using the proposed frameworks.

DATE 22
DESIGN, AUTOMATION & TEST IN EUROPE
14 – 15 March 2022 · on-site event
16 – 23 March 2022 · online event
The European Event for Electronic System Design & Test

Design and Evaluation Frameworks for Advanced RISC-based Ternary Processor

Thank you