LOTR
Lord-Of-The-Ring

AMICHAI BEN-DAVID

Introduction

- ~20 minutes presentation.
- ~1-2 minutes per Title.

Part 1

- Architectural idea.

Part 2

- Workflow & Execution.

RISCV

- **Open ISA**
  - freely available to academia and industry.
- **Avoids "Over-Architecting" for specific micro-arch/technology**
  - microcode, in-order, decoupled, Out-Of-Order etc.
  - ASIC, FPGA, full-custom
- **Small "base" integer ISA**
  - RV32I is usable by itself.
- **Customizable:**
  - Optional Standard Extensions.
  - May add any Non-Standard Extensions.

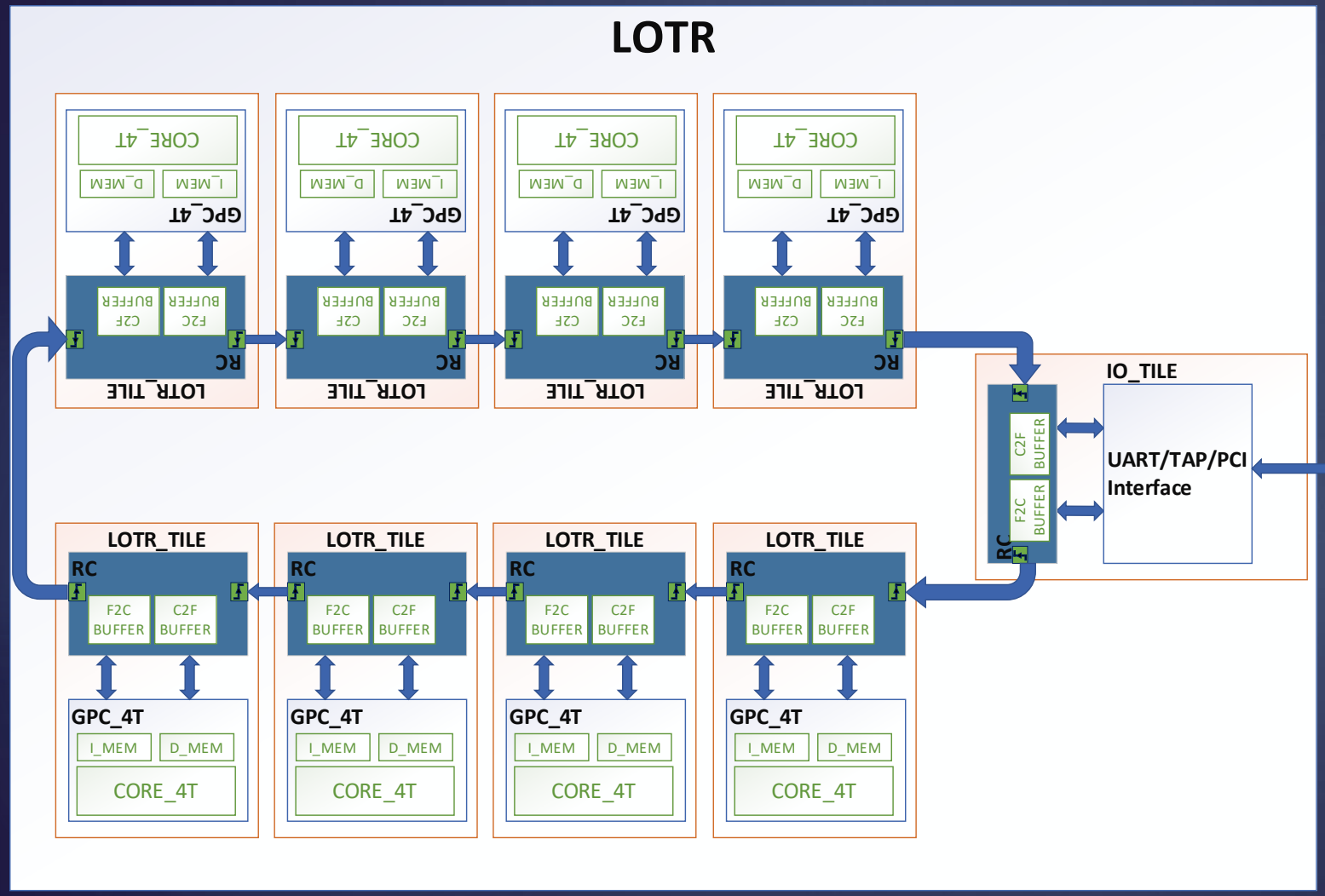
RISC-V®

RISC-V: The Free and Open RISC Instruction Set Architecture

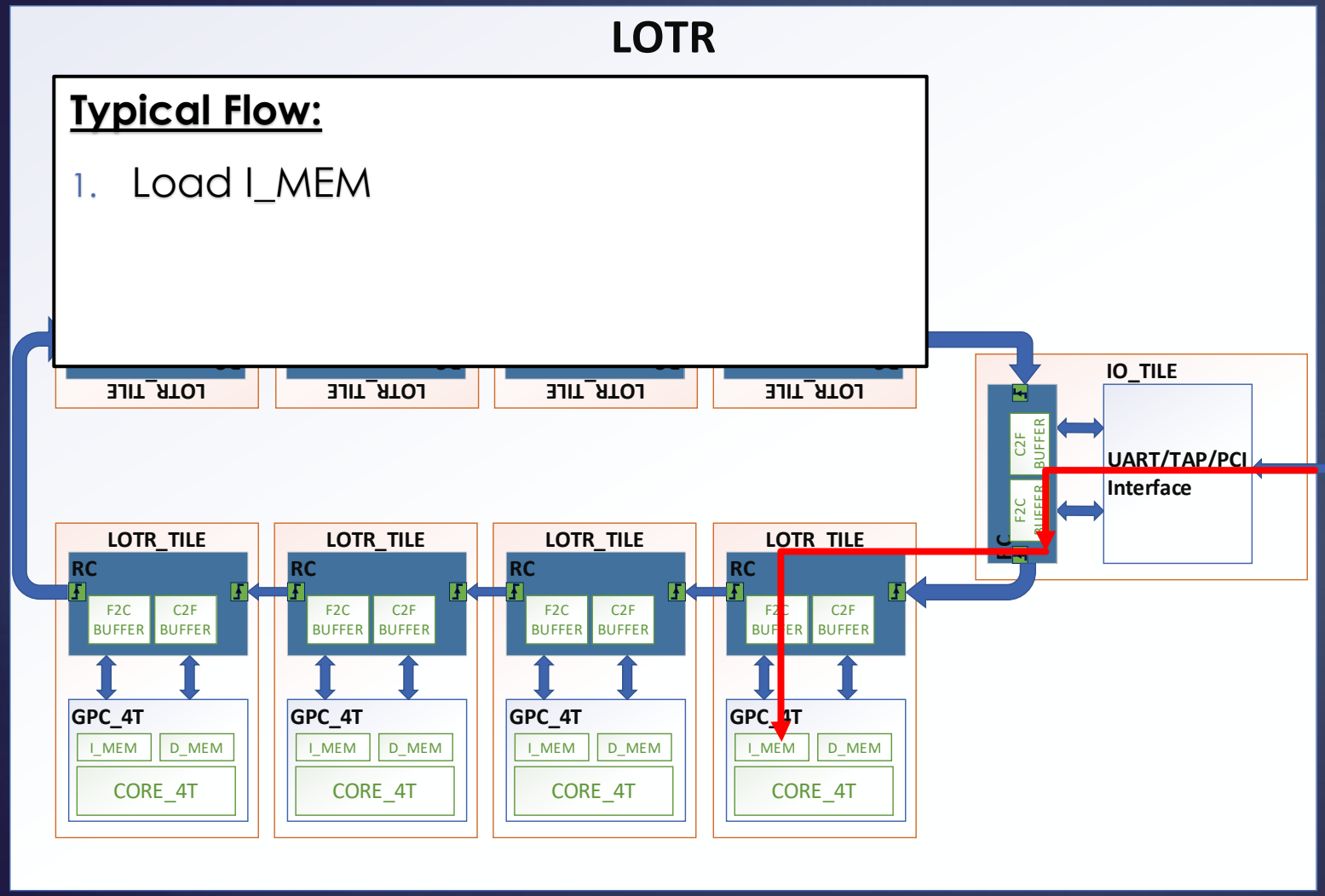
Base	Version	Status
RVWMO	2.0	**Ratified**
RV32I	**2.1**	**Ratified**
RV64I	**2.1**	**Ratified**
RV32E	*1.9*	*Draft*
RV128I	*1.7*	*Draft*
Extension	**Version**	**Status**
M	**2.0**	**Ratified**
A	**2.1**	**Ratified**
F	**2.2**	**Ratified**
D	**2.2**	**Ratified**
Q	**2.2**	**Ratified**
C	**2.0**	**Ratified**
Counters	*2.0*	*Draft*
L	*0.0*	*Draft*
B	*0.0*	*Draft*
J	*0.0*	*Draft*
T	*0.0*	*Draft*
P	*0.2*	*Draft*
V	*0.7*	*Draft*
Zicsr	**2.0**	**Ratified**
Zifencei	**2.0**	**Ratified**
Zam	*0.1*	*Draft*
Ztso	*0.1*	*Frozen*

The Architectural idea

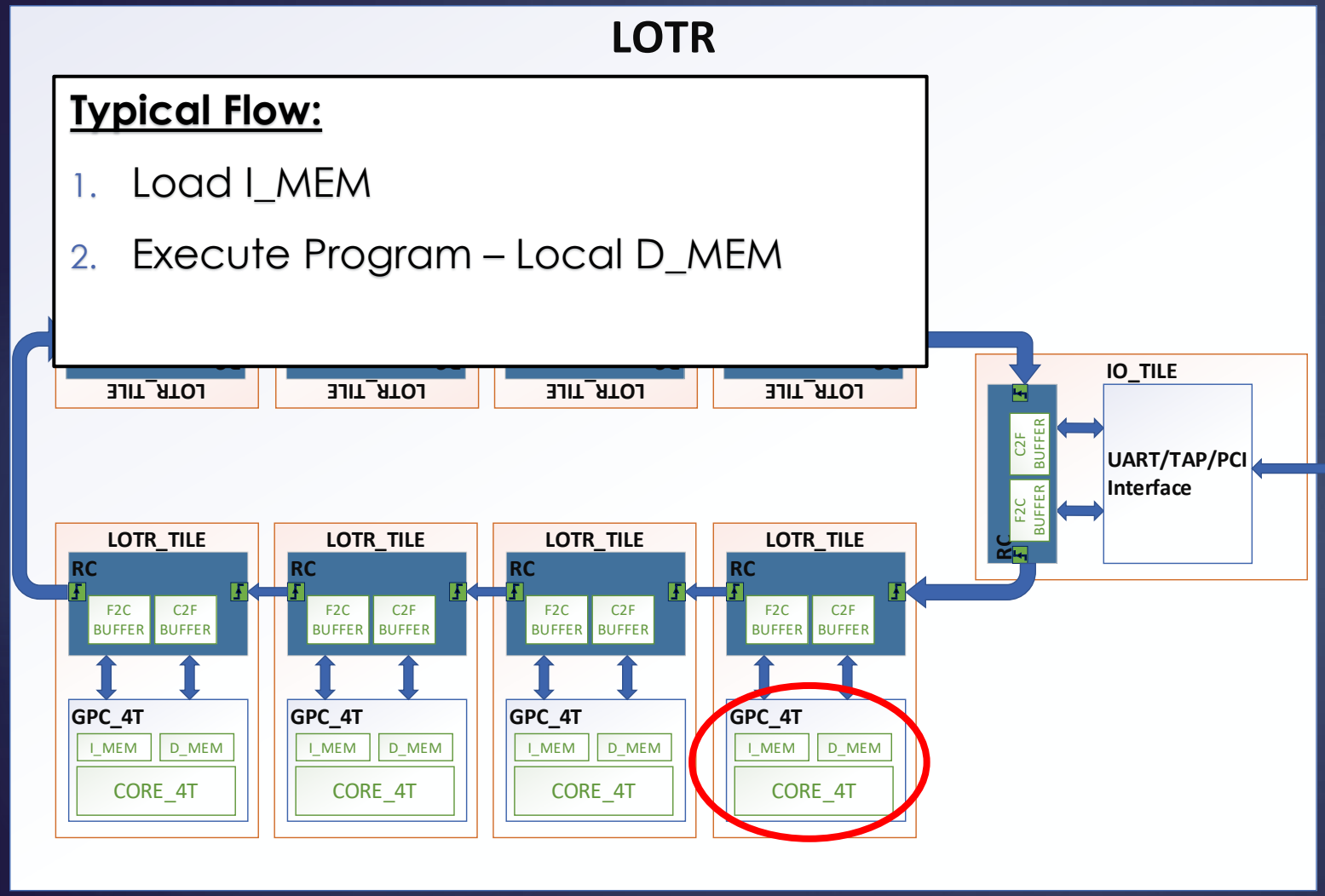
Lord-Of-The-Ring



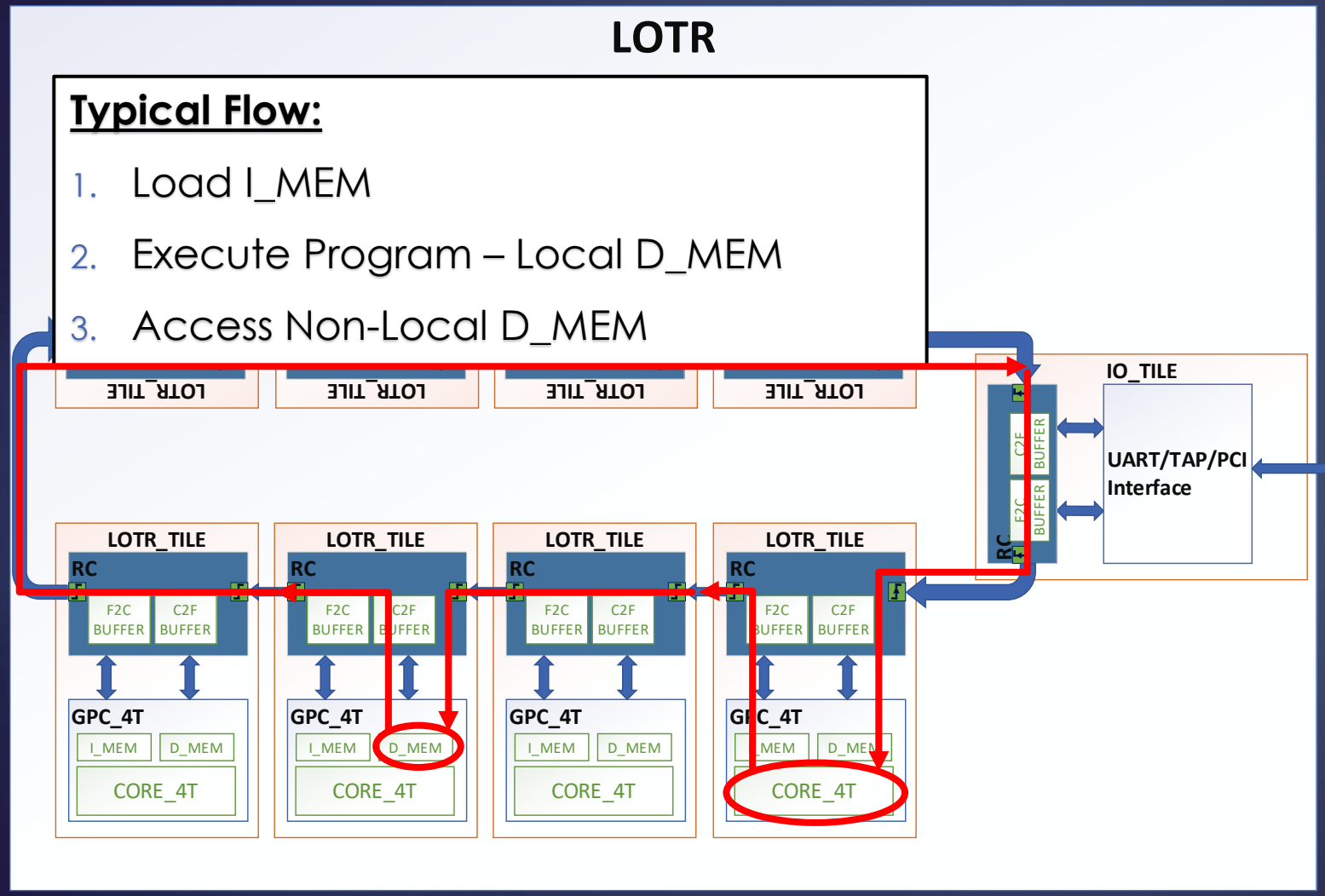
Lord-Of-The-Ring



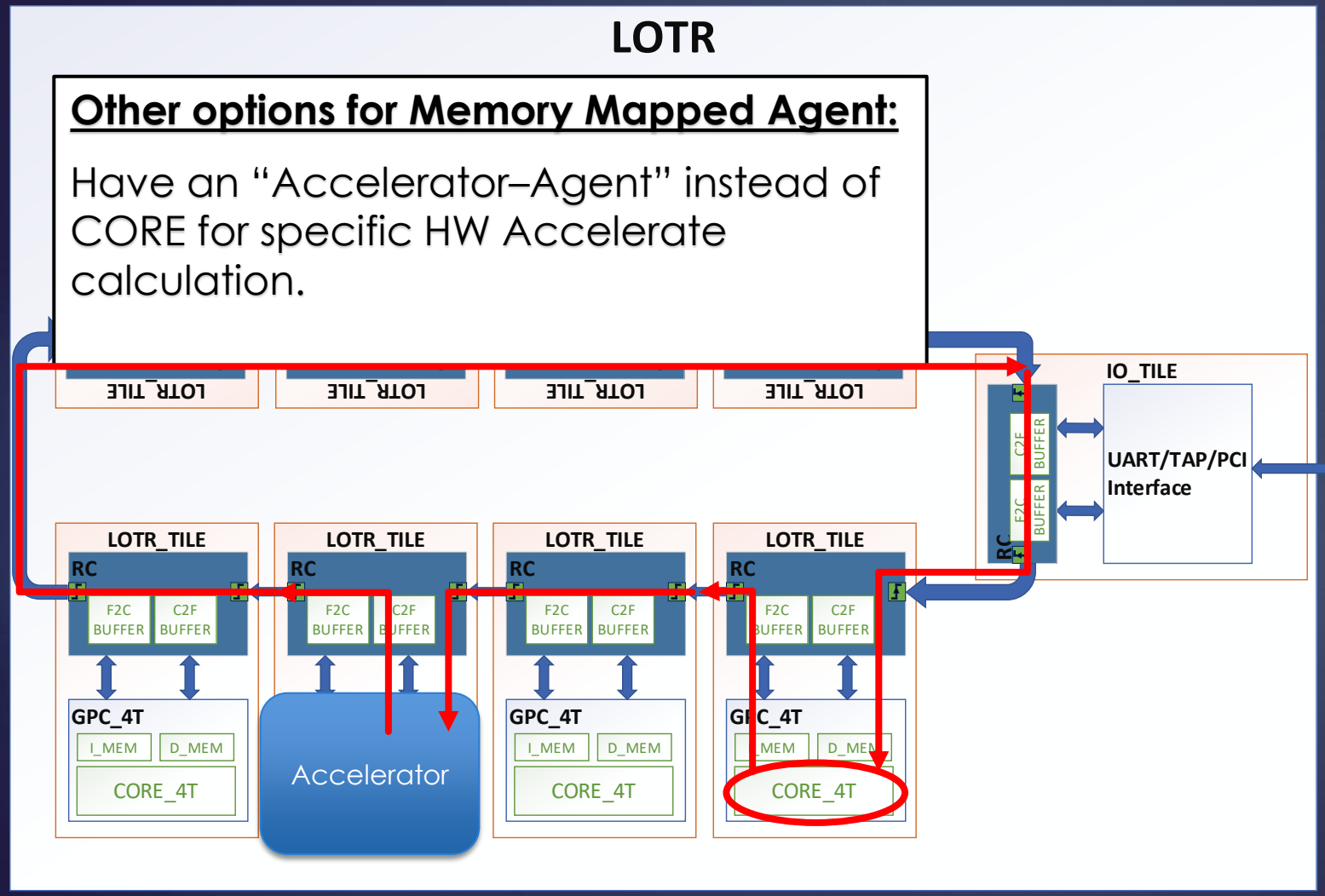
Lord-Of-The-Ring



Lord-Of-The-Ring

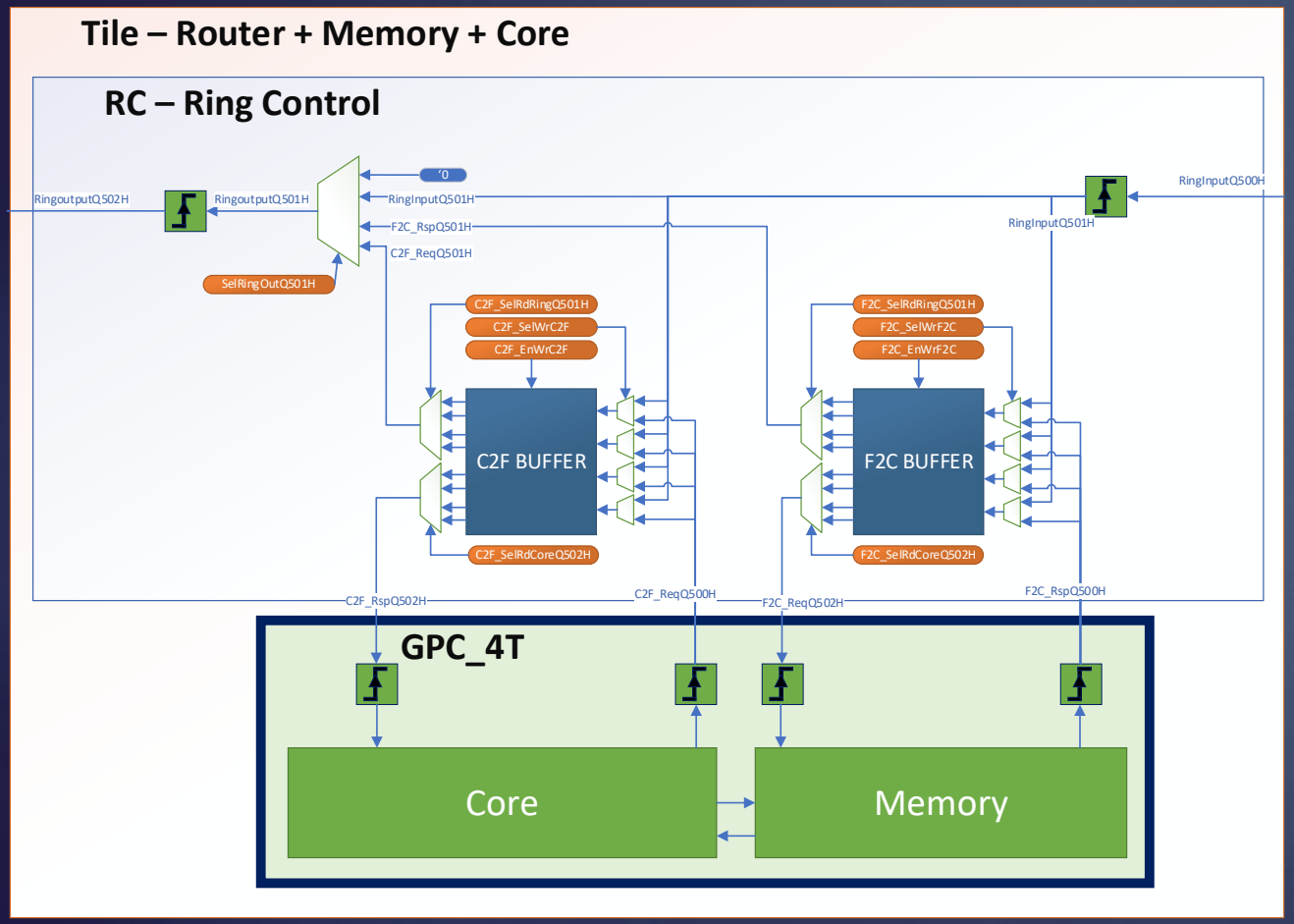


Lord-Of-The-Ring

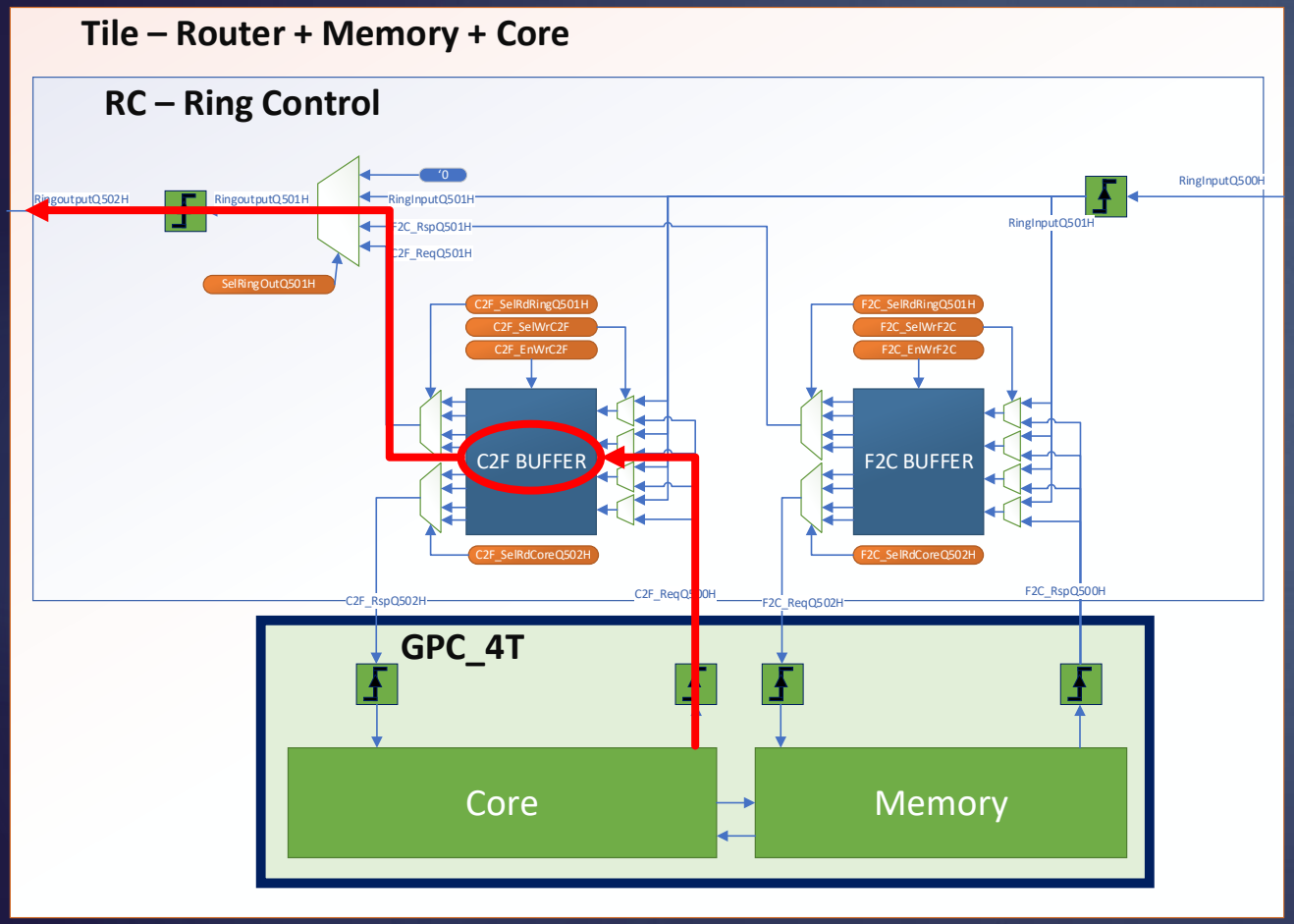


Lord-Of-The-Ring

Ring-Controller



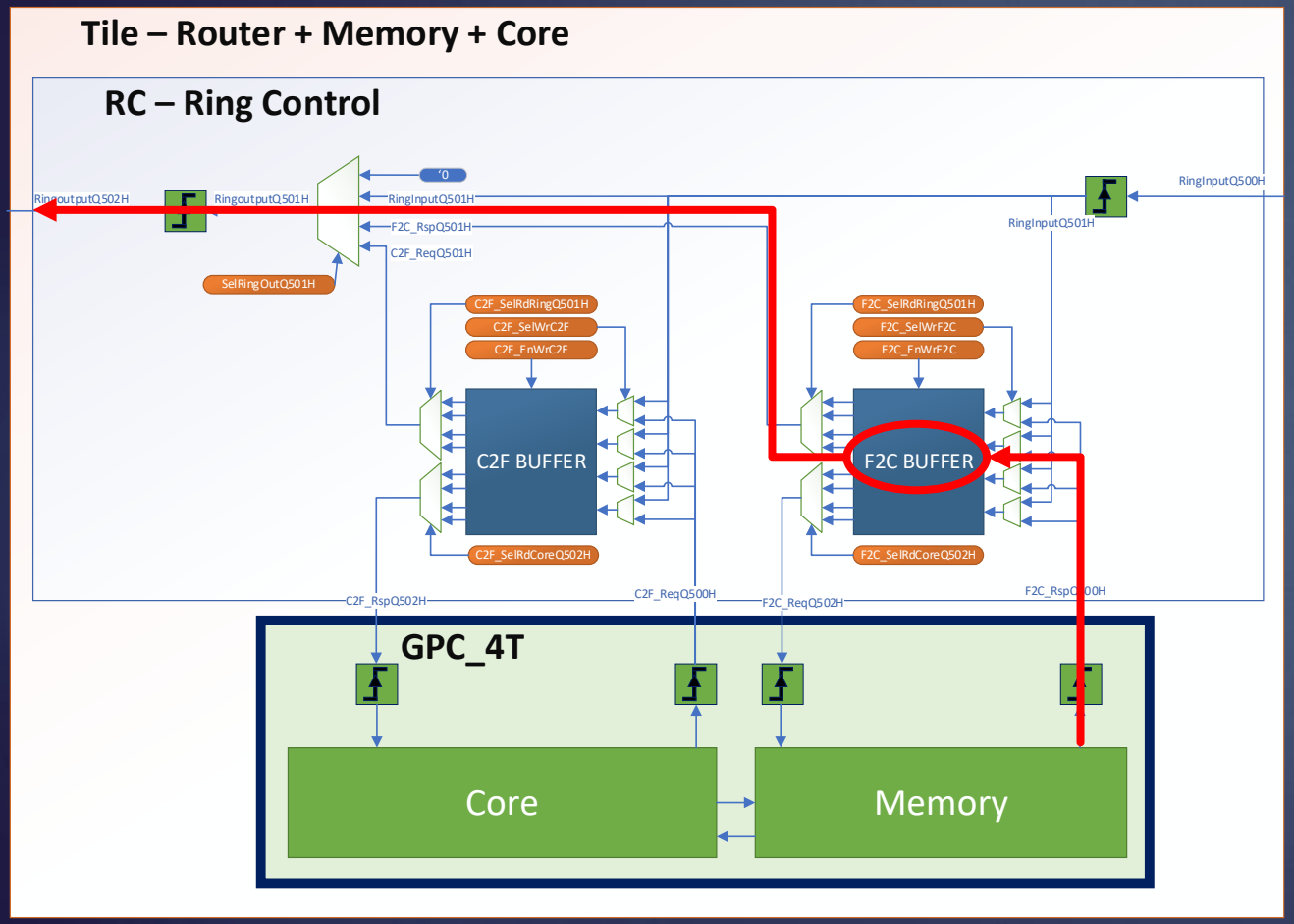
Ring-Controller



Data Path:

1. Core Request

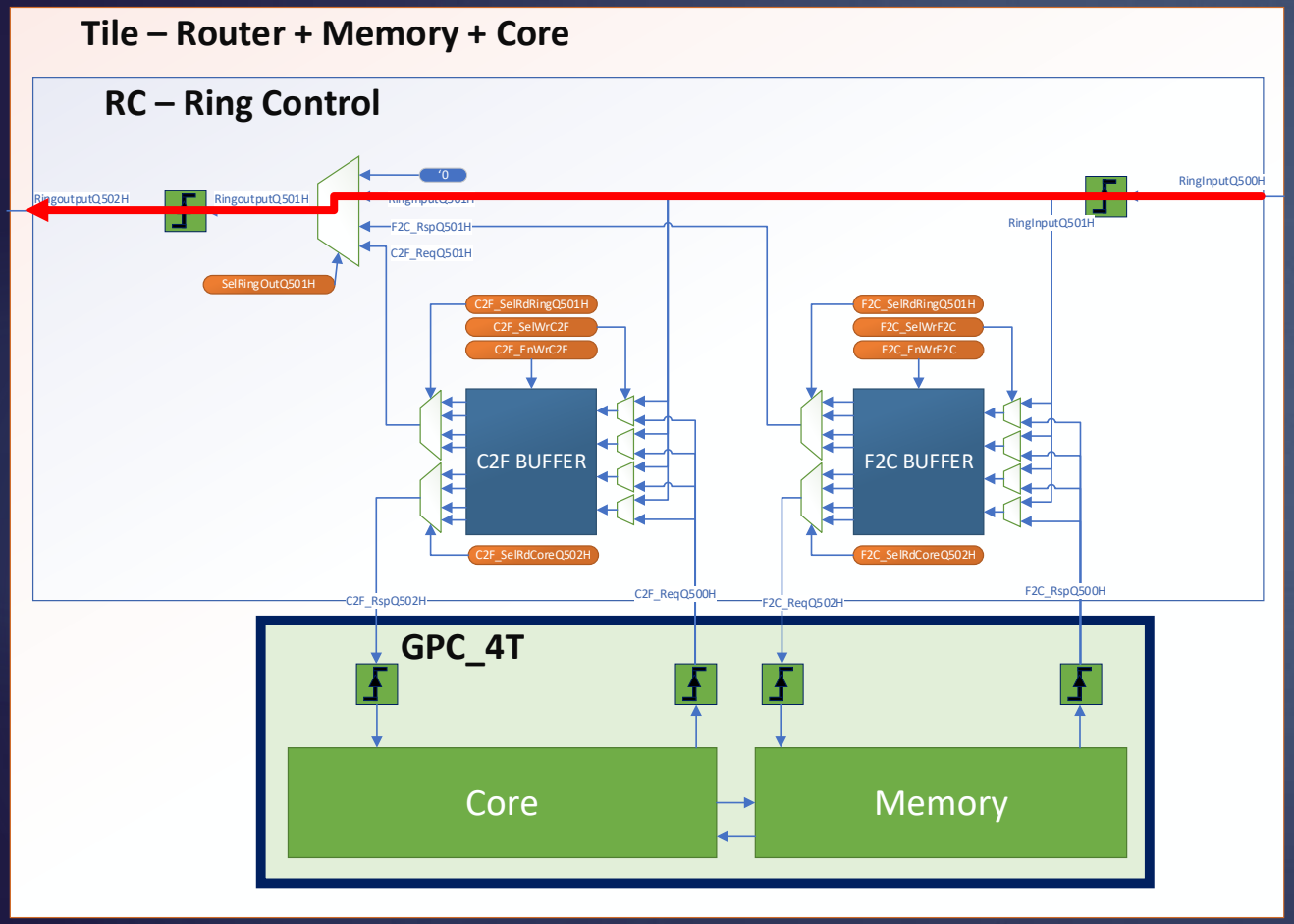
Ring-Controller



Data Path:

1. Core Request
2. Memory Response

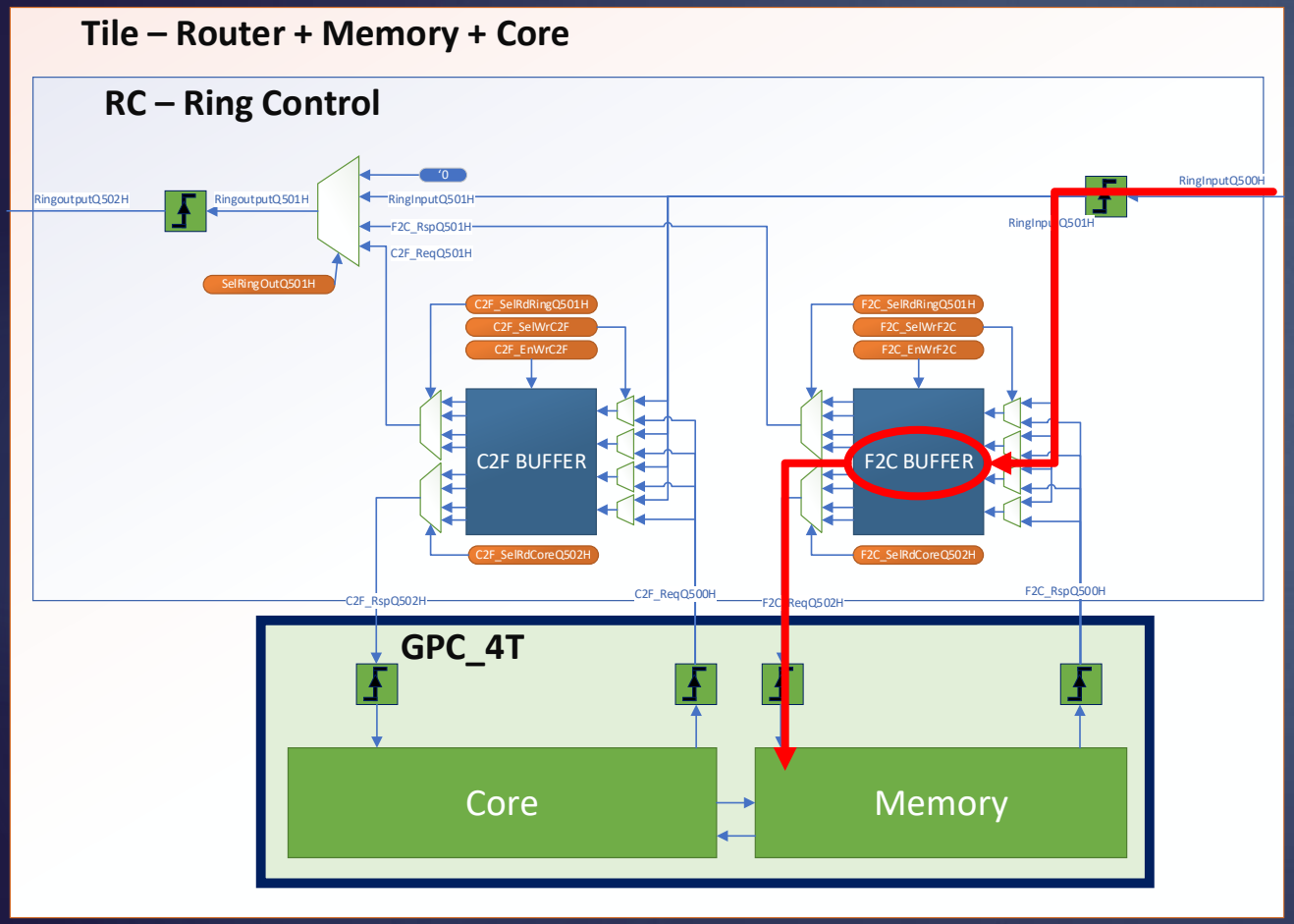
Ring-Controller



Data Path:

1. Core Request
2. Memory Response
3. Ring Input
   - Pass through

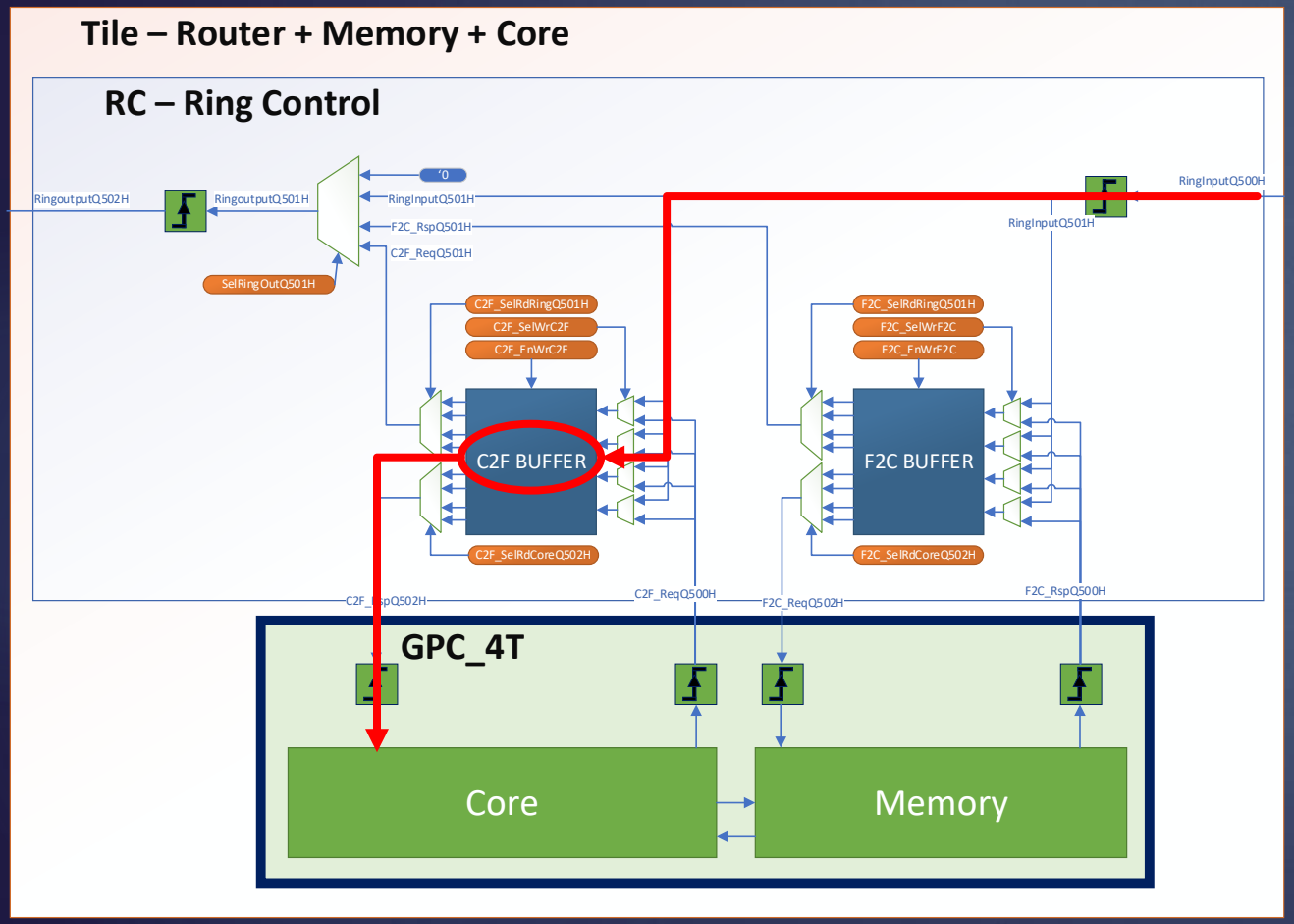
Ring-Controller



Data Path:

1. Core Request
2. Memory Response
3. Ring Input
   - Pass through
   - F2C RD/WR to local Mem

Ring-Controller

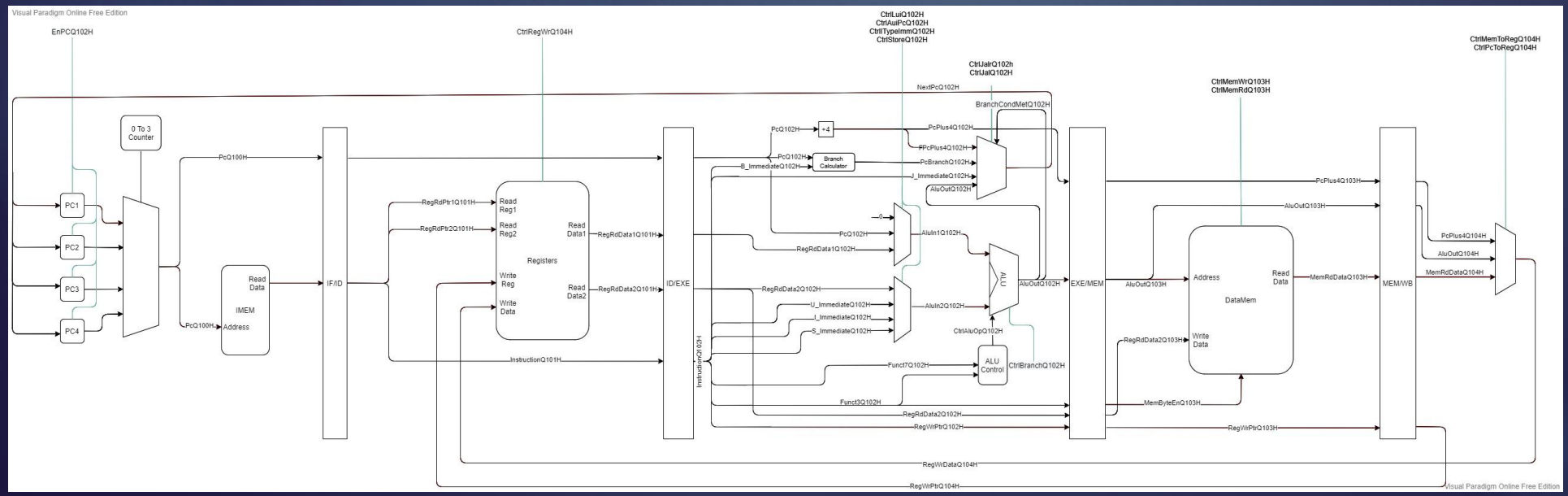


Data Path:

1. Core Request
2. Memory Response
3. Ring Input
   - Pass through
   - F2C RD/WR to local Mem
   - C2F RD_RSP

General-Purpose-Compute-Unit With 4 HW Threads

- RISCV - RV32I/E ISA compatible



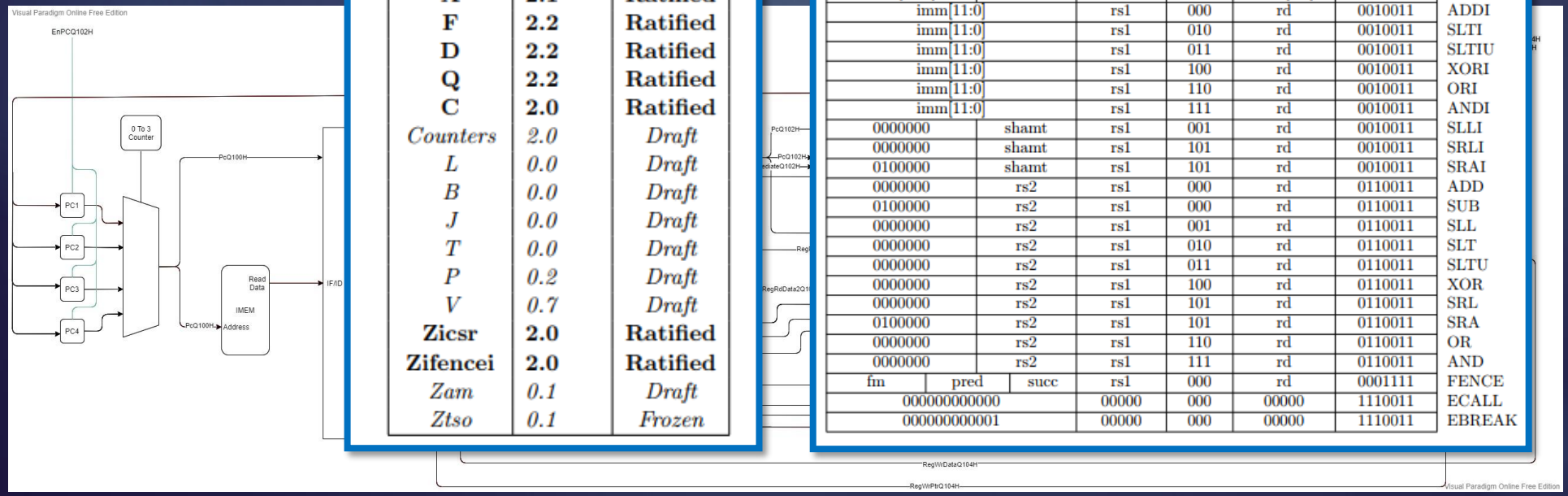
General-Purpose-Compute-Unit With 4 HW Threads

- RISCV - RV32I/E ISA compatible

Base	Version	Status
RVWMO	2.0	**Ratified**
RV32I	**2.1**	**Ratified**
RV64I	**2.1**	**Ratified**
RV32E	*1.9*	*Draft*
RV128I	*1.7*	*Draft*
Extension	**Version**	**Status**
M	**2.0**	**Ratified**
A	**2.1**	**Ratified**
F	**2.2**	**Ratified**
D	**2.2**	**Ratified**
Q	**2.2**	**Ratified**
C	**2.0**	**Ratified**
Counters	*2.0*	*Draft*
L	*0.0*	*Draft*
B	*0.0*	*Draft*
J	*0.0*	*Draft*
T	*0.0*	*Draft*
P	*0.2*	*Draft*
V	*0.7*	*Draft*
Zicsr	**2.0**	**Ratified**
Zifencei	**2.0**	**Ratified**
Zam	*0.1*	*Draft*
Ztso	*0.1*	*Frozen*

RV32I Base Instruction Set

imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:0]		rs1	100	rd	0000011	LBU
imm[11:0]		rs1	101	rd	0000011	LHU
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND
fm	pred | succ	rs1	000	rd	0001111	FENCE
000000000000		00000	000	00000	1110011	ECALL
000000000001		00000	000	00000	1110011	EBREAK



Introduction

RISCV

Idea

LOTR

RC

GPC_4T

Use-Case

Example

Execution

RISCV Tool-Chain

Verilog Simulation

GitHub

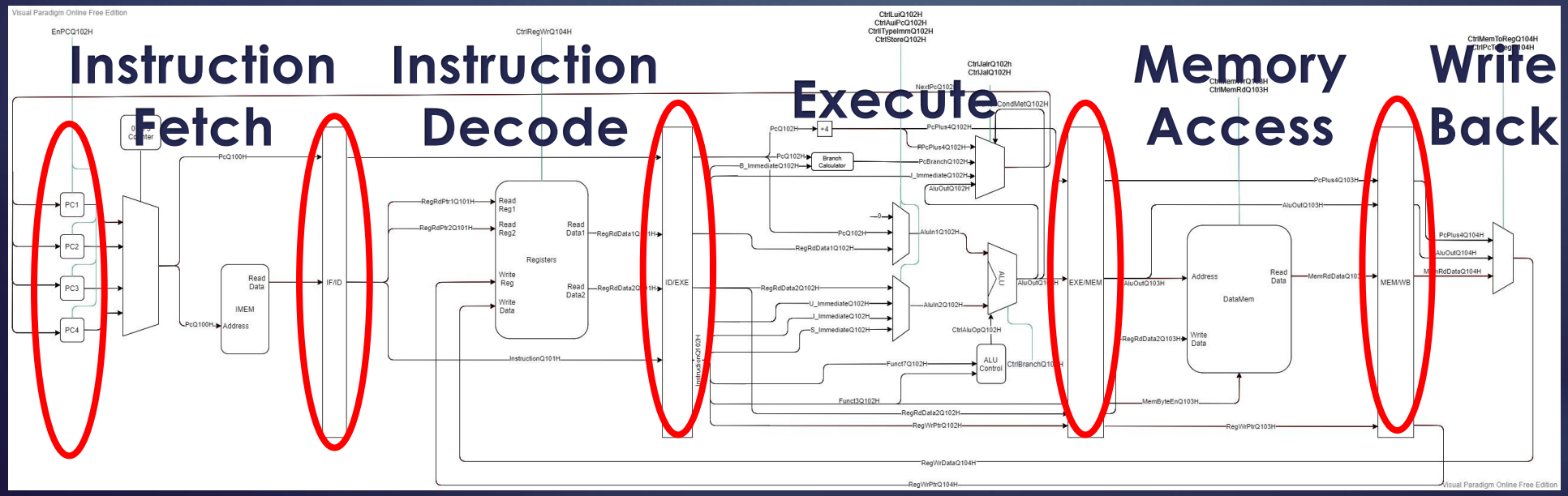
Design - RTL

Validation

POC - FPGA

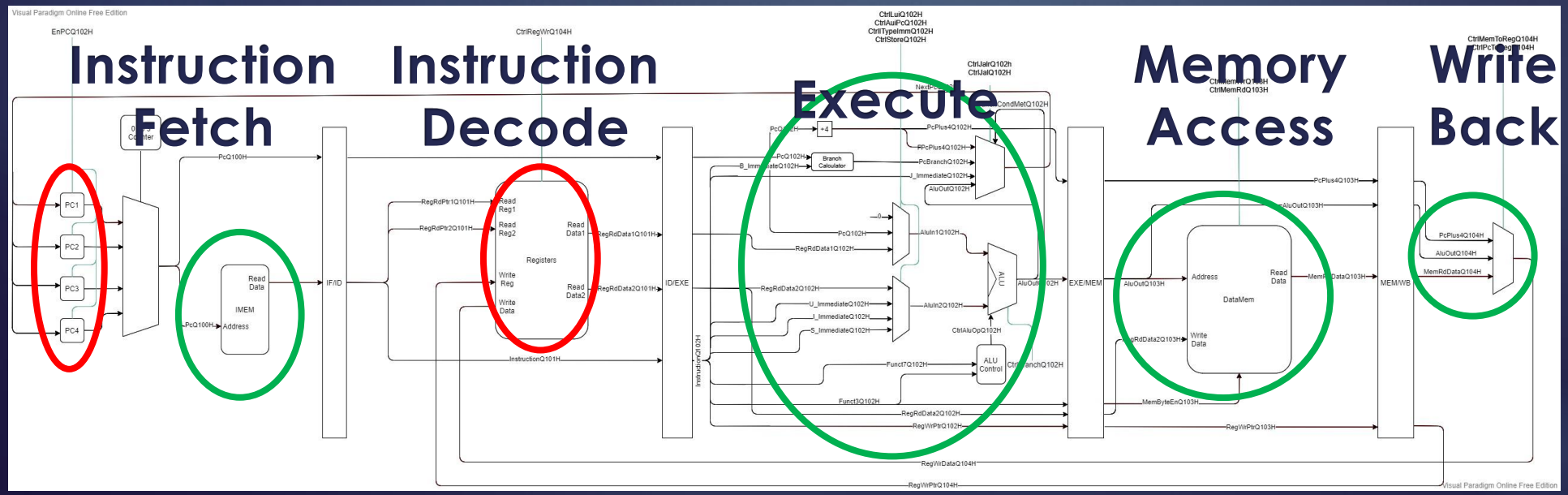
General-Purpose-Compute-Unit With 4 HW Threads

- RISCV - RV32I/E ISA compatible
- 5 Stage Pipe-Line



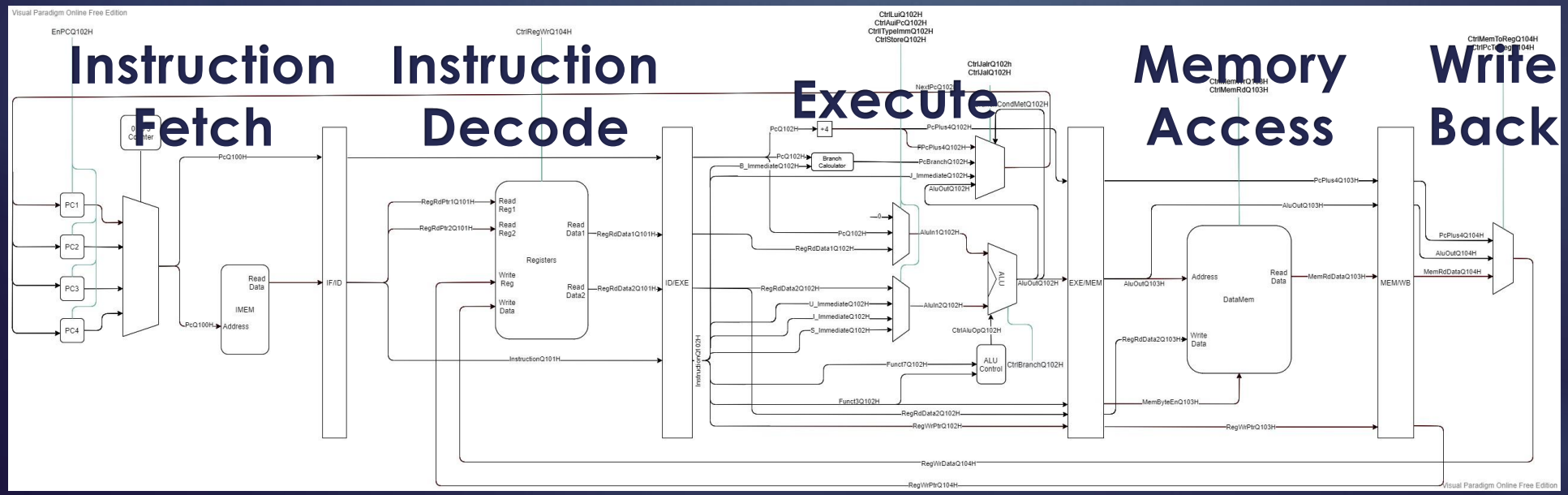
General-Purpose-Compute-Unit With 4 HW Threads

- RISCV - RV32I/E ISA compatible
- 5 Stage Pipe-Line
- 4 HW Threads
  - 4 x Program Counter && 4 x Register Files – See RED
  - Sharing Execution Logic and Memory Access – See Green
  - Thread IPC of ¼ - 1 Instruction every 4 Cycles.



General-Purpose-Compute-Unit With 4 HW Threads

- No Hazards.
- No forwarding unit.
- No Pipe Flush - no branch predictor.
- No Context Switch.
- No Interrupts.
- No Thread Scheduler
- Etc.



Introduction

RISCV

Idea

LOTR

RC

GPC_4T

Use-Case

Example

Execution

RISCV Tool-Chain

Verilog Simulation

GitHub

Design - RTL

Validation

POC - FPGA

Use-Case

- Programable Accelerators.
- Offload task from Main processor.
- Multi HW Threaded accelerator.
  - Big-Data/Neural-Network/Machine-Learning.
- Programable Controllers –
  Each controller gets an HW Thread.

Example Use-Case

- TODO - Pseudo code for multi threaded accelerator. Have “switch case” for each Thread and use the CR “WHOAMI” to move Data around the threads and Cores.

Workflow & Execution

Introduction

RISCV

Idea

LOTR

RC

GPC_4T

Use-Case

Example

Execution

RISCV Tool-Chain

Verilog Simulation

GitHub

Design - RTL

Validation

POC - FPGA

RISCV Tool-Chain

- TODO

Verilog Simulation

- TODO

GitHub

- TODO

Design - RTL

- TODO

Validation

- TODO

Proof Of Concept - FPGA

- TODO