

# An interleaved multi-thread RISC-V design for SMP with dual core lockstep to support ASIL-D functional safety requirements

Jian Wei\*, Jinfu Zhao, Lei Shi, Yixuan Zhao, Mei Wang, Yasong Hu

Automotive Semiconductor Product Unit, ECARX Holdings Inc.

## Abstract

*This is to present our recent work with Exida, to certify our interleaved RISC-V design for ASIL-D, which delivers much higher performance efficiency with smaller silicon area and consuming much less power. This design is further utilized inside an automotive SoC product, which will be taped out shortly.*

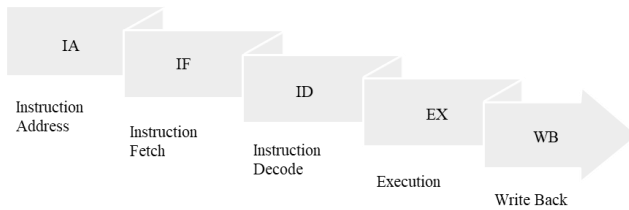
## Introduction

In the automotive industry, functional safety requirements have been typically implemented with the design of dual-core lockstep CPU architecture [5,6]. It uses two identical CPU cores, each running the exactly same set of instruction codes to lock each other with every clock cycle; comparing corresponding signals from each core can thus detect errors in one of the CPU cores.

This conventional duplication strategy costs more than two times of the silicon area, with also more than two times of power consumption [7,8].

## Interleaved Two-Thread Design

Figure 1 below shows a typical pipelined CPU core, with 5 pipeline stages: IA (Instruction Address), IF (Instruction Fetch), ID (Instruction Decode and Operand Read), EX (Execution), WB (Write Back) [1,2].



**Figure 1 5 Pipeline Stages**

We have an in-order **single-issue** implementation, which is suitable for real-time automotive microcontroller. It can deliver one instruction every cycle under ideal situation. However, there are cases lead to performance overhead.

Figure 2 shows that a stall cycle has to be introduced, thus causing performance overhead, when the next instruction's operand is the result of current instruction.

\* Corresponding author : jian.wei@ecarxgroup.com

Instruction 1	IA	IF	ID	EX	WB				
R5=R6*R7		IA	IF	ID	EX	WB			
STALL									
R4=R4+R5				IA	IF	ID	EX	WB	
Instruction 4					IA	IF	ID	EX	WB

**Figure 2 Data Dependency Causes Pipeline Stall**

As Figure 3 shows, since RISC-V does not support a delay-slot instruction following a non-sequential instruction, such as JAL or JALR; the already fetched instruction, following a non-sequential instruction, must be flushed. This will also cause at least one cycle performance overhead.

Instruction 1	IA	IF	ID	EX	WB				
JAL JALR		IA	IF	ID	EX	WB			
Instruction 3		FLUSH	—IA—	—IF—	—ID—	—EX—	—WB—		
Instruction 4				IA	IF	ID	EX	WB	

**Figure 3 Non-Sequential Instruction Flush Pipeline**

Figure 4 shows an interleaved two-thread design. It effectively avoids these performance overheads, therefore, delivers a much higher performance with more efficient silicon area, since two threads are sharing the same set of execution units.

Instruction 1 (0)	IA	IF	ID	EX	WB						
Instruction 1 (1)		IA	IF	ID	EX	WB					
R5=R6*R7 (0)			IA	IF	ID	EX	WB				
JAL JALR (1)				IA	IF	ID	EX	WB			
R4=R4+R5 (0)					IA	IF	ID	EX	WB		
Instruction 3 (1)						IA	IF	ID	EX	WB	

**Figure 4 Interleaved Two-Thread Design**

## SMP with Lock-Step for Functional Safety

We design a symmetric multiprocessing (SMP) system, while each processor is implemented with this kind of interleaved two-thread core. In each core, one of the threads is work-thread, running regular workload; the other thread is used as lock-thread, used to lock-step the work-thread in another processor. This lock-thread never accesses memory, since the memory system uses error correction coding (ECC) for functional safety. A lock-thread only inputs its data from the work-thread in another process. This further avoids cache thrashing, for both data cache and instruction cache. This is another performance advantage in addition to all the benefits mentioned before regarding this interleaved threads design.

We worked with Exida to just complete the assessment of a RISC-V32IMA based interleaved two-thread design, certified for ASIL-D functional safety requirements.

We are developing an automobile system-on-chip (SoC) product, targeting a tape out in April 2025, with TSMC 28nm technology. Figure 5 shows the primary features of this SoC. It has four RISC-V32IMAFIC inside [3,4], two of them are designed with such interleaved two-thread, providing lock-step functional safety, support autosar type of real-time operating system (RTOS).

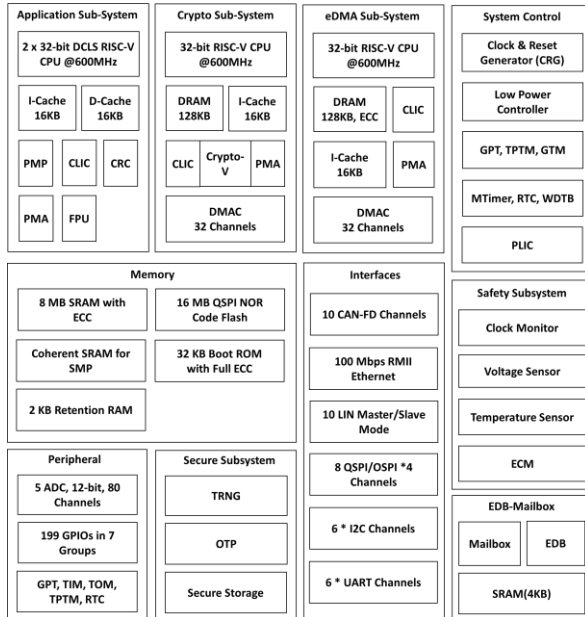


Figure 5 Primary Features Of SoC

### Further extend the design

We are designing a RISC-V 64G, targeting to support more sophisticated operating system, such as Linux. A memory management unit (MMU) requires more stages in the processor pipeline. This will make the pipeline much deeper, a total of 9-stage design. To optimize the performance, we explore an interleaved three-thread architecture, in which:

- One safe-work-thread running functional safety workload, which requires a lock-thread from another processor core to provide lock-step errors detection.
- One lock-thread, used to operate in lock-step with the safe-work-thread of another processor core.
- One independent work-thread, running regular workload with much relaxed functional safety requirements if any.

This provides great flexibility to implement a high performance multi-processor system.

## Summary

We design an interleaved two-thread RISC-V, to support SMP with lock-step functional safety. This work is certified by Exida to qualify ASIL-D requirements. This design is implemented inside an automotive MCU product. We further extend this interleaved multi-thread design to support advanced operating system, with more flexible functional safety requirements.

## References

- [1] A. Waterman, K. Asanović and J. Hauser. *The RISC-V Instruction Set Manual, Volume I: User-Level ISA*. RISC-V Foundation, 2019.
- [2] A. Waterman, K. Asanović and J. Hauser. *The RISC-V Instruction Set Manual, Volume II: Privileged Architecture*. RISC-V Foundation, 2021.
- [3] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 2017.
- [4] A. S. Waterman. *Design of the RISC-V Instruction Set Architecture*. Ph.D. dissertation, University of California, Berkeley, 2016.
- [5] M. Sarraseca, S. Alcaide, F. Fuentes, J. C. Rodriguez, F. Chang, I. Lasfar, R. Canal, F. J. Cazorla and J. Abella. "SafeLS: An Open Source Implementation of a Lockstep NOEL-V RISC-V Core." In: Proc. IEEE Int. On-Line Testing Symposium (IOLTS) (2023). doi: [10.1109/IOLTS59296.2023.10224867](https://doi.org/10.1109/IOLTS59296.2023.10224867).
- [6] K. Marcinek and W. A. Pleskacz, "Variable Delayed Dual-Core Lockstep (VDCLS) Processor for Safety and Security Applications," In: Proc. IEEE Electronics (2023), doi: [10.3390/electronics12020464](https://doi.org/10.3390/electronics12020464).
- [7] Y. Kwon, J.-J. Lee, K.-S. Shin, J.-H. Han, K.-J. Byun and N.-W. Eum. *80μW/MHz 0.68V Ultra Low-Power Variation-Tolerant Superscalar Dual-Core Application Processor*. Electronics and Telecommunications Research Institute (ETRI), 2015.
- [8] S. Cho, S. Park, S. Kim, Y. Kim, and M.-K. Lee, "CalmRISCTM-32: a 32-bit low-power MCU core," In: Proc. Asia Pacific Conference on ASICs (APASIC) (2000). doi: [10.1109/APASIC.2000.896964](https://doi.org/10.1109/APASIC.2000.896964).