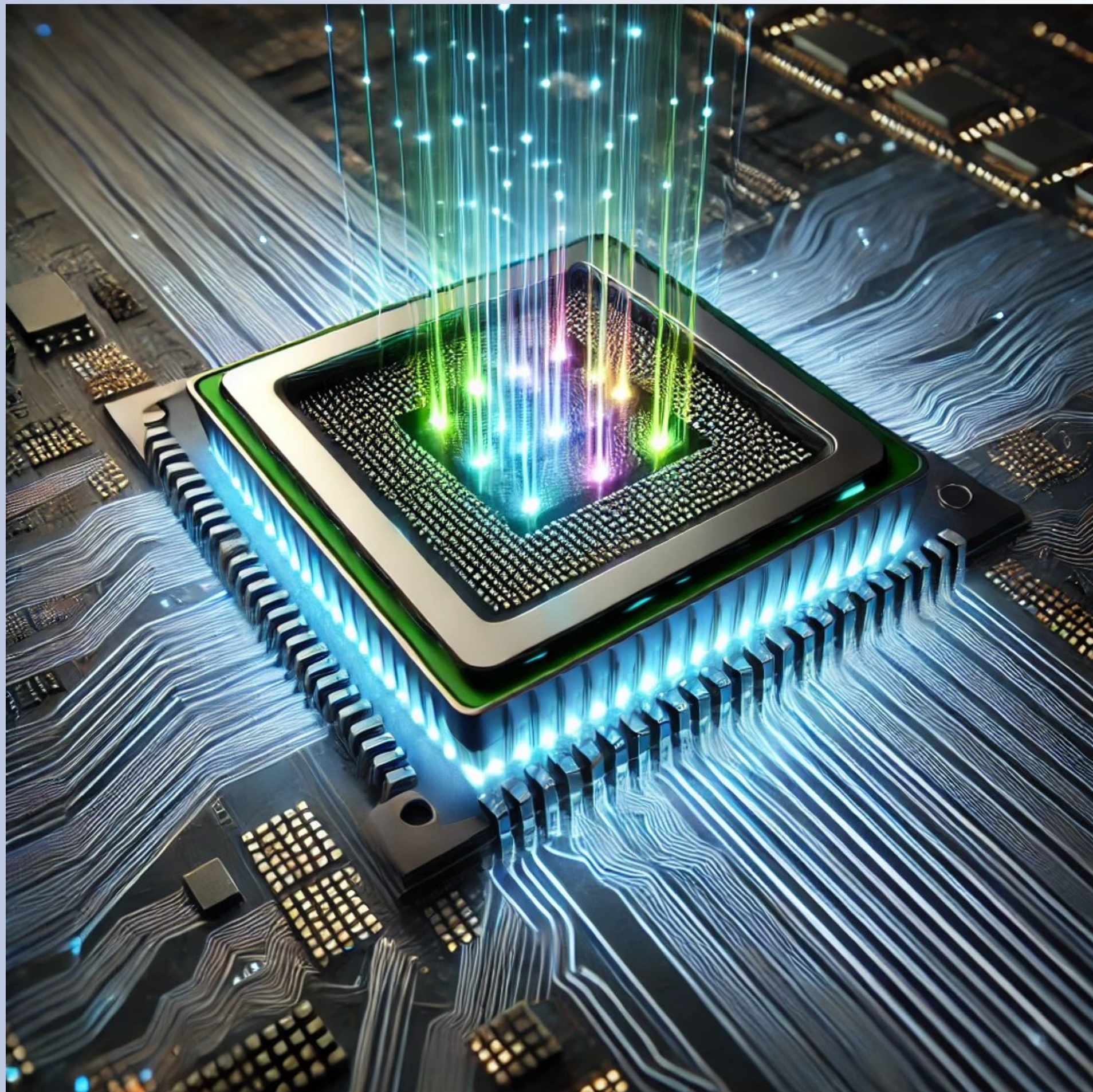


## RISC-V facilitates use of Big N multicore



### Big N opportunities for Space

Allows use of less hardened cores

N cores, M needed, F cores fail during mission

$$P(\text{functioning cores} \geq M) = P(F \leq (N-M))$$

With individual core failure probabilities  $p_i$  known probability of failures from N cores can be calculated (in simple case cumulative binomial) and N chosen to give desired probability of at least M cores still functioning at mission end.

Allows use of COTS RISC-V cores in space.

Power saving possible by core management

M cores needed but not all the time

RISC-V cores can be taken out of use until needed, brought back to use, then taken out of use again.

Big N cores can be partitioned for different purposes

Some cores allocated to use by one operating system, some cores to another OS or other software.

Need for Hypervisor can be avoided.

Single points of failure can be avoided

If the OS itself as well as the application can be distributed across the cores

### Big N Challenges

#### Hardware

- For Big N cache coherence becomes difficult but is well understood for RISC-V.
- Interrupt may need to be re-assigned when cores become faulty, well supported by RISC-V.

#### Software

##### *Application software*

For Space needs to take full advantage of concurrency

- to increase throughput
- to check cores functioning correctly.

##### Operating system

- Must allow restriction to a subset of cores.
- Must itself be distributed across allocated cores to avoid single points of failure.
- Must allow cores be taken in and out of use without disrupting scheduling.
- Must allow same task be run concurrently on multiple cores.
- Must provide read-write mutual exclusion semaphores to allow different cores use the same data concurrently.
- Should allow cores be reserved for certain tasks and tasks be restricted to certain cores.

### How big will Big N be for RISC-V?

*How soon will the opportunities and challenges be faced?*

The above and other capabilities are provided by OCEOSmp

**(not just another RTOS!)**

