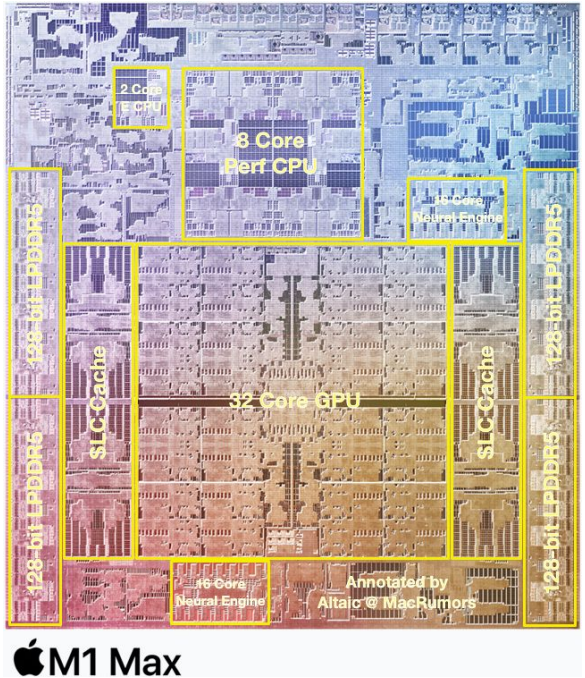
AKER

Safe and Secure SoC Access Control

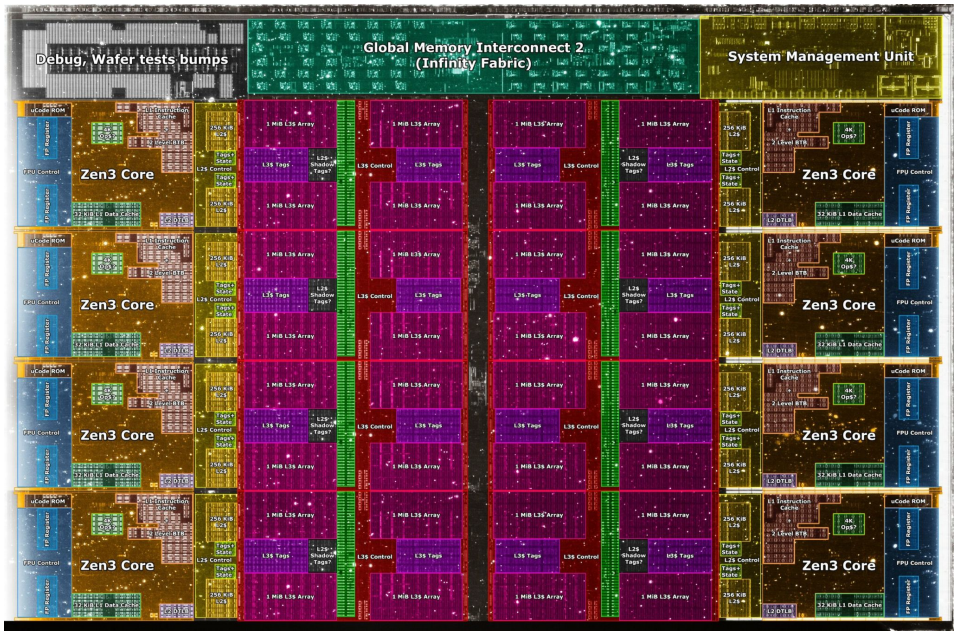
Chi Chow | Brandon Erickson | Hosein Yavarzadeh

One Last Time: SoCs and AKER's motivation

- **What is an SoC?**



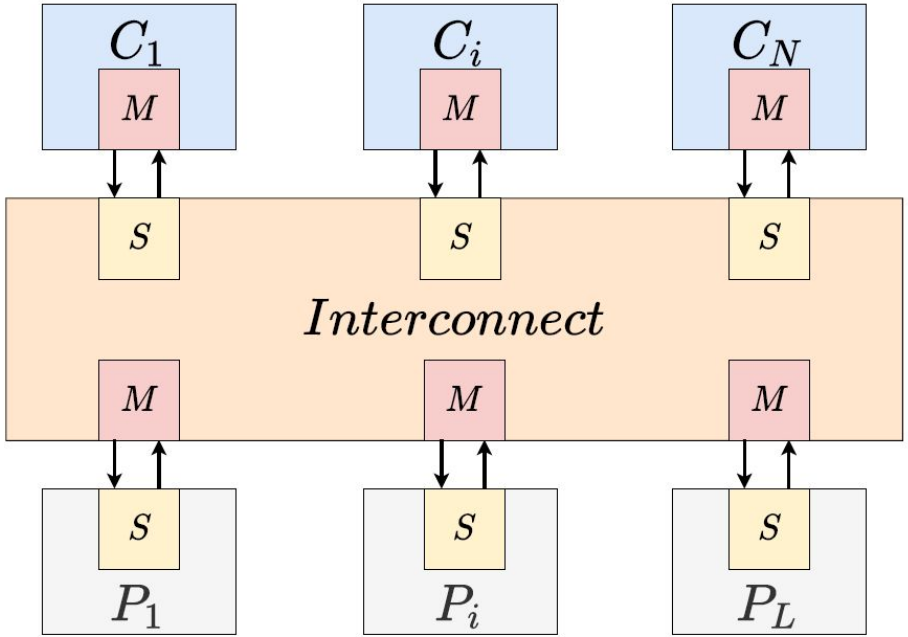
M1 Max



AMD Ryzen 5 5600X

One Last Time: SoCs and AKER's motivation

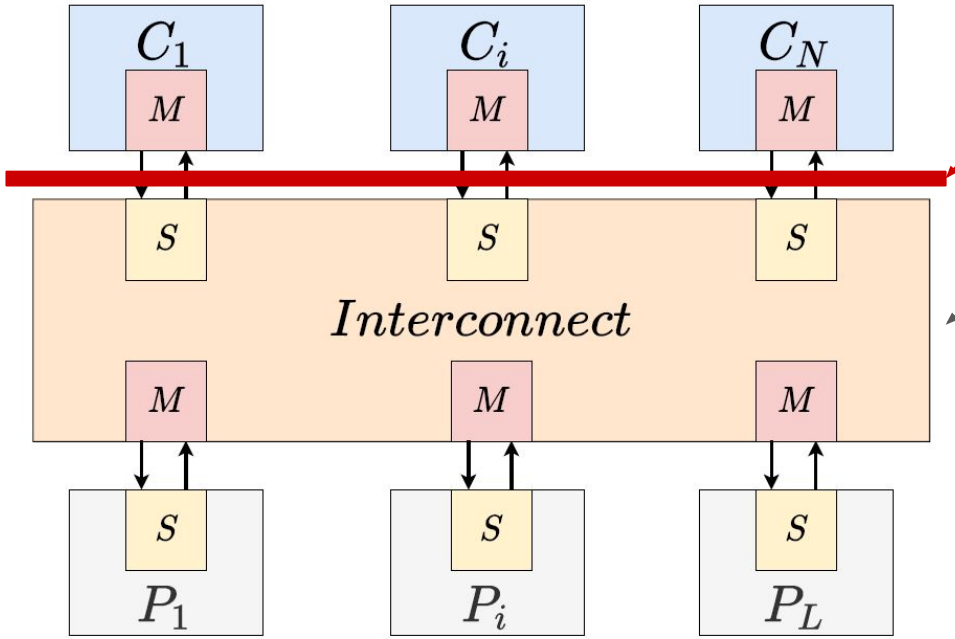
- **Why AKER?**



Standardized Interconnect Architecture on an SoC (Network on a chip)

One Last Time: SoCs and AKER's motivation

- **Why AKER?**

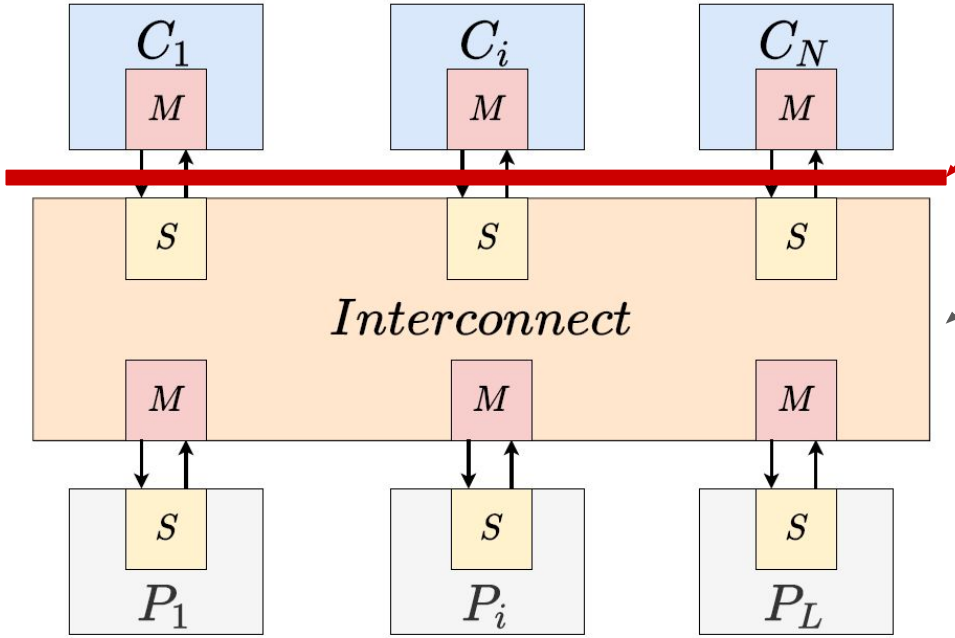


Access Control Black Magic

Standardized Interconnect Architecture on an SoC (Network on a chip)

One Last Time: SoCs and AKER's motivation

- **Why AKER?**



Access Control Black Magic

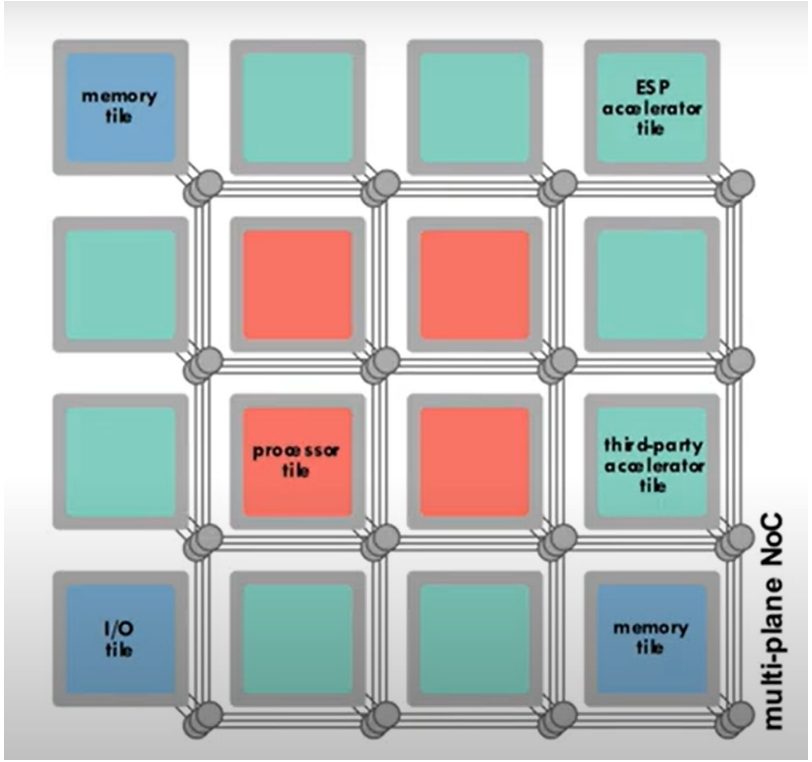
Standardized Interconnect Architecture on an SoC (Network on a chip)

Some Benefits:

- Interoperability: integrate into existing SoCs
- Flexibility: re-configure access control without hardware changes

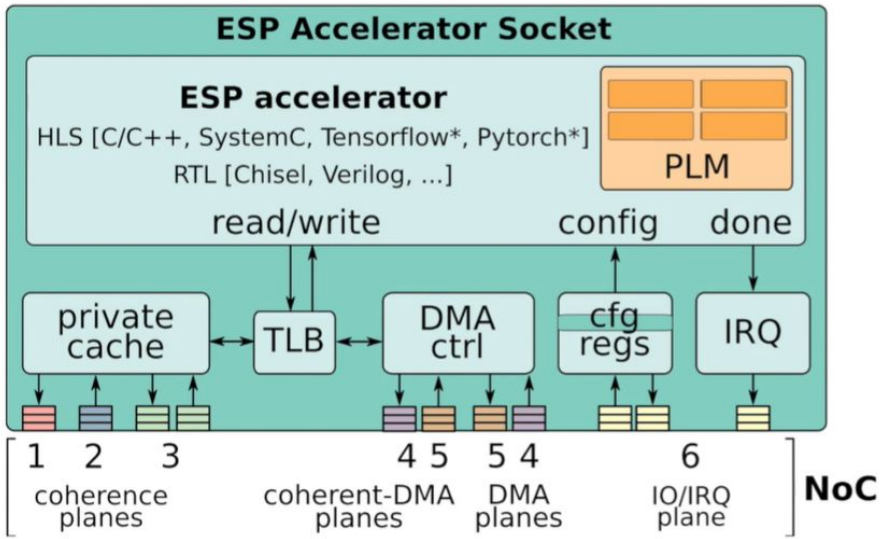
NoC Architecture Example: ESP

- An open-source platform for rapid SoC development.
- ESP **tiles** interface with the network via **sockets**.



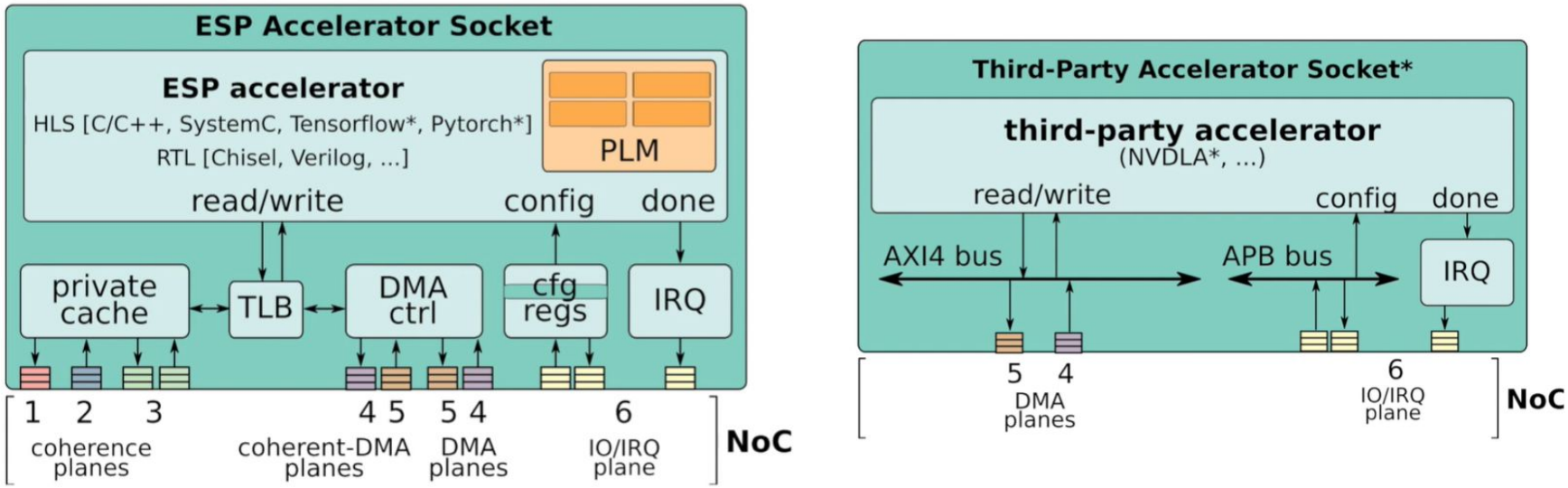
Source: https://www.esp.cs.columbia.edu/docs/

ESP: Socket Architecture



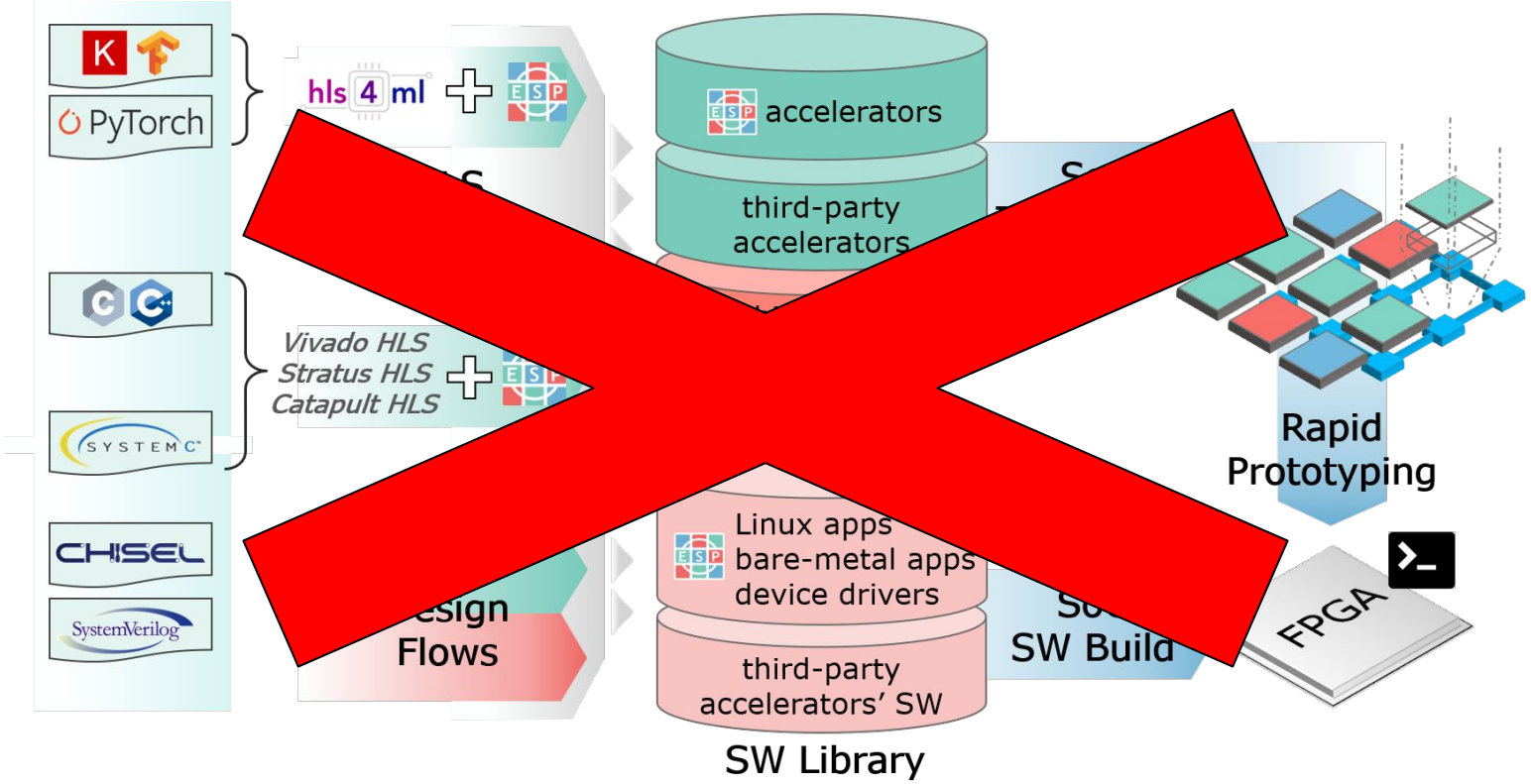
Source: https://www.esp.cs.columbia.edu/docs/

ESP: Socket Architecture



Source: https://www.esp.cs.columbia.edu/docs/

ESP Development Workflow

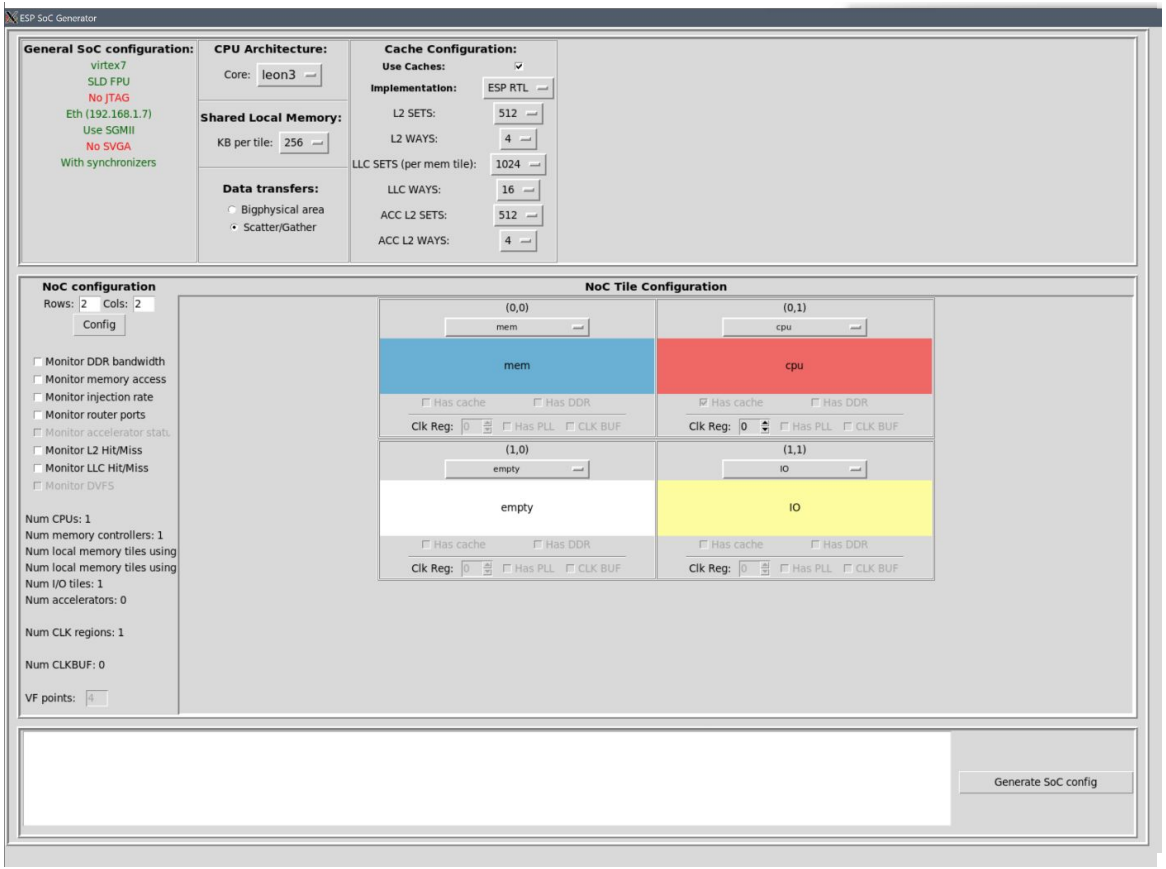


Actually very straightforward…

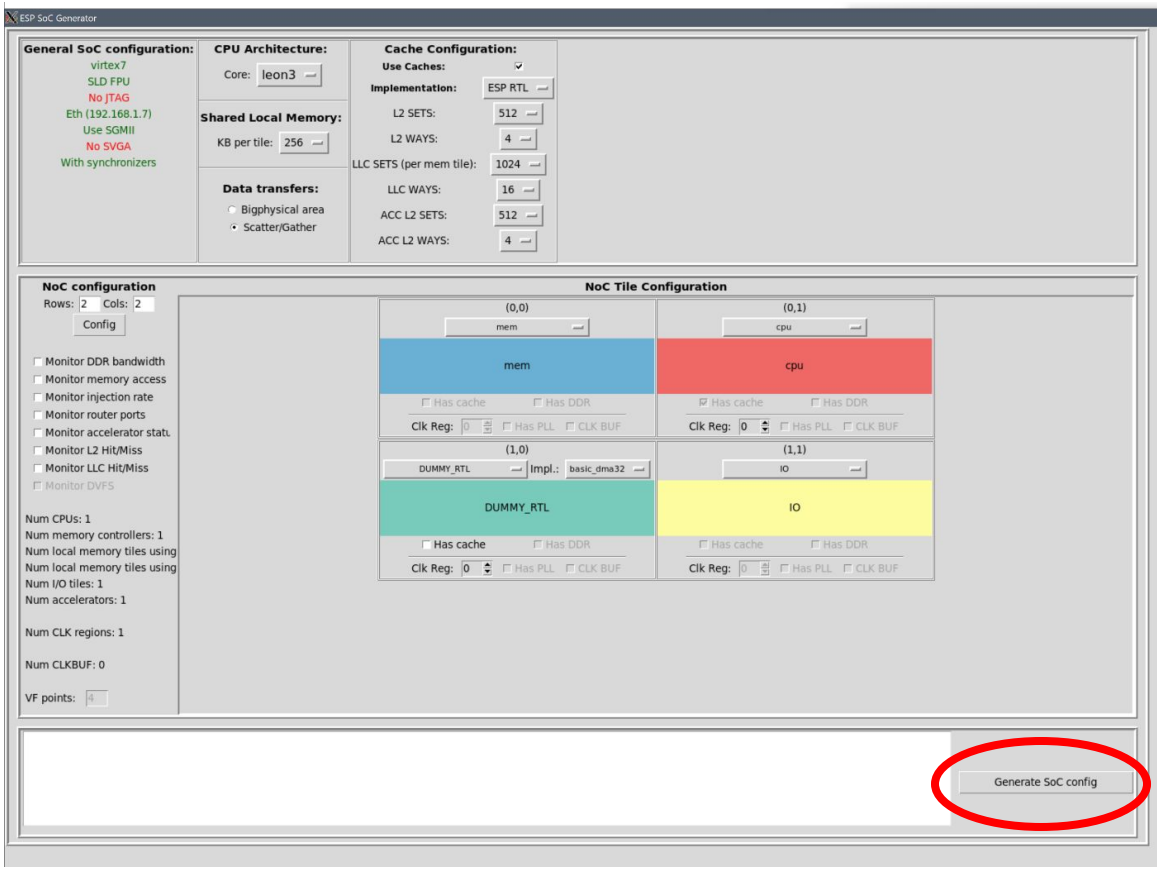
Step 1: Create Hardware Accelerator

- esp/tools/accgen/accgen.sh
- Configure resources
- For our use case, leave everything in default (we will see why later)
- Implement accelerator and import

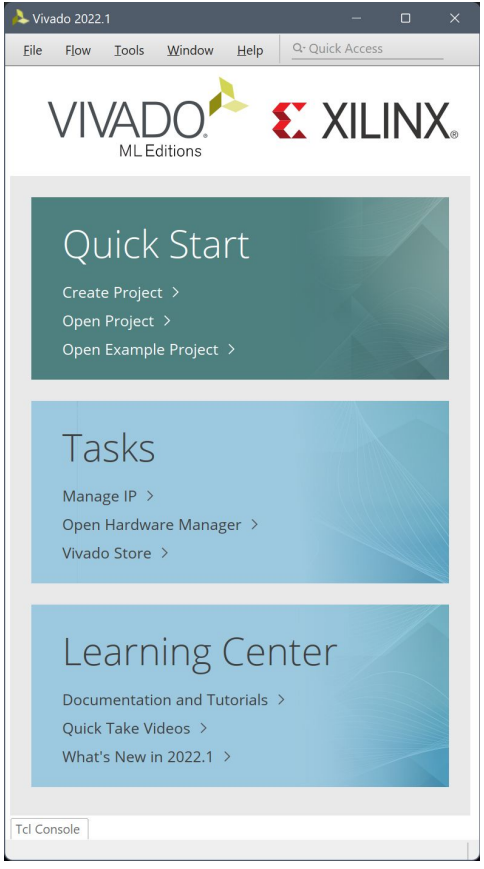
Step 2: Configure SoC



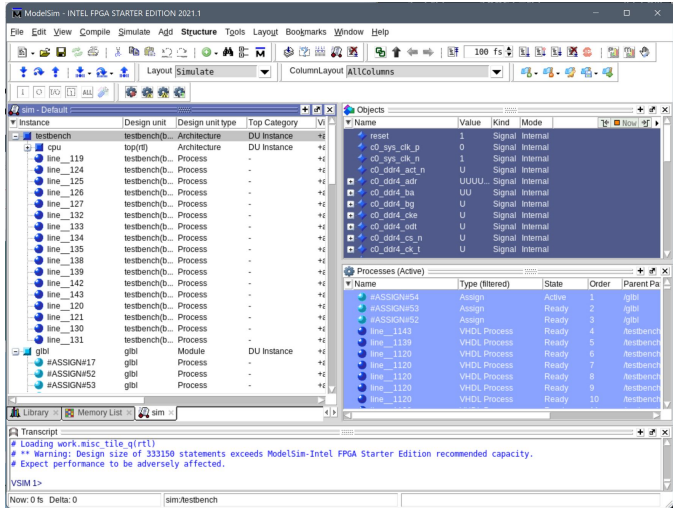
Step 2: Configure SoC



Step 3: Xilinx Vivado and Modelsim

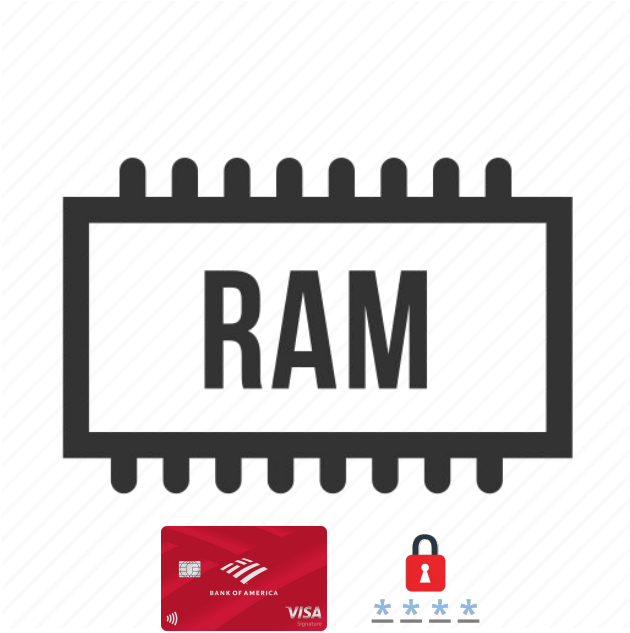


Xilinx Virtex-7 FPGA VC707



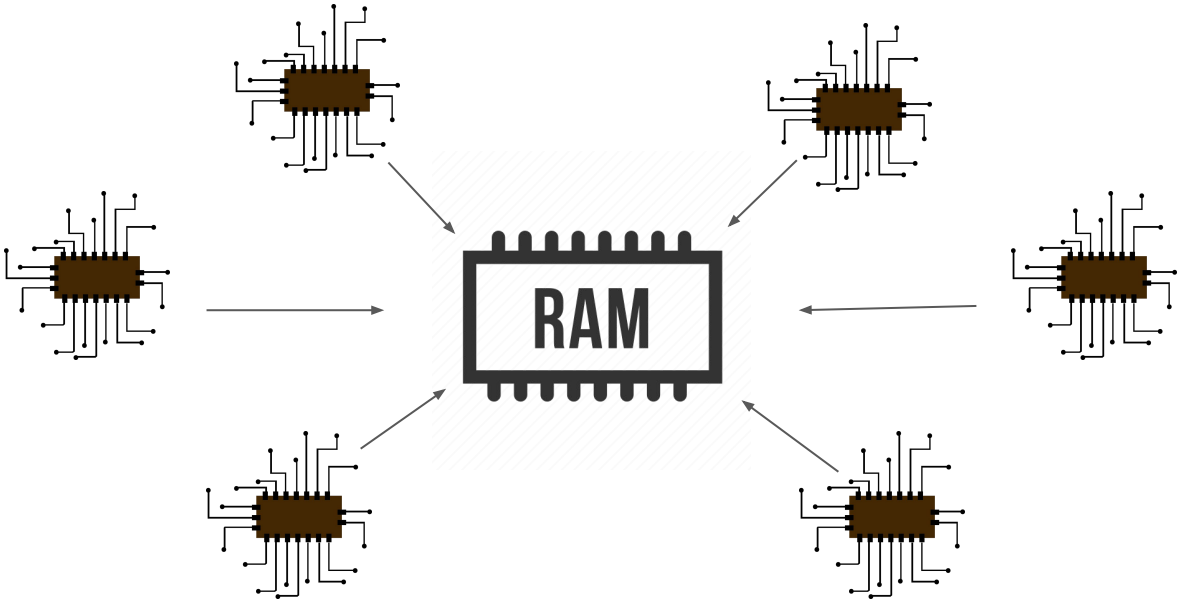
AKER: Safe and Secure SoC Access Control

Processing units store data in storage devices like RAMs.

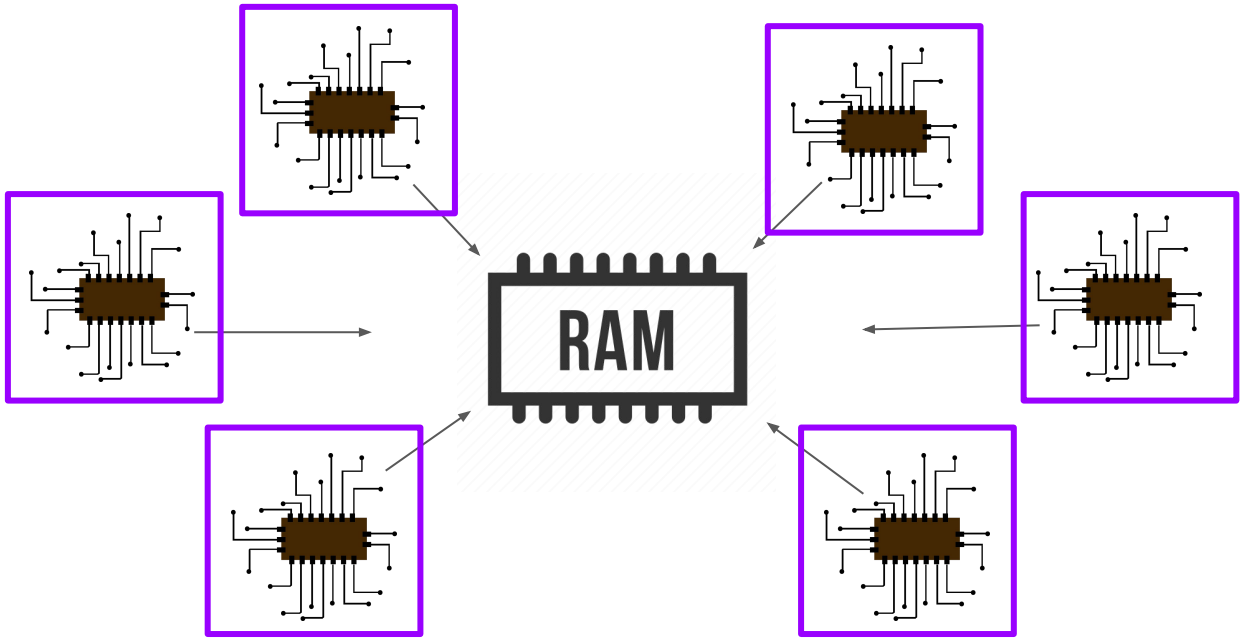


AKER

Storage Devices are shared between various processing units.



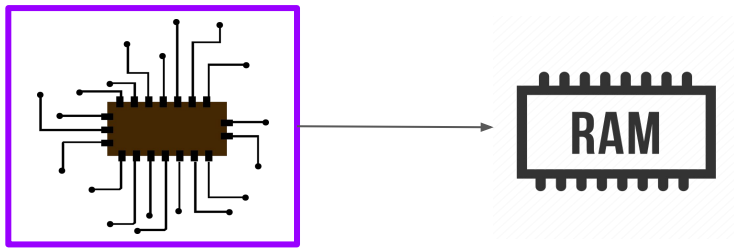
AKER: Access control wrapper



AKER: Access Control Wrapper (ACW)

Simplified version of the ACW:

```
1   Bool Addr_Check (int id, int address) {
2       if ( !trust_id(id) ) {
3           return false;
4       }
5       else {
6           if ( address < boundary ){
7               return true;
8           }
9           else {
10              return false;
11          }
12      }
13  }
```



Challenges:

- **Implementation on top of ESP**
- **Consistency**

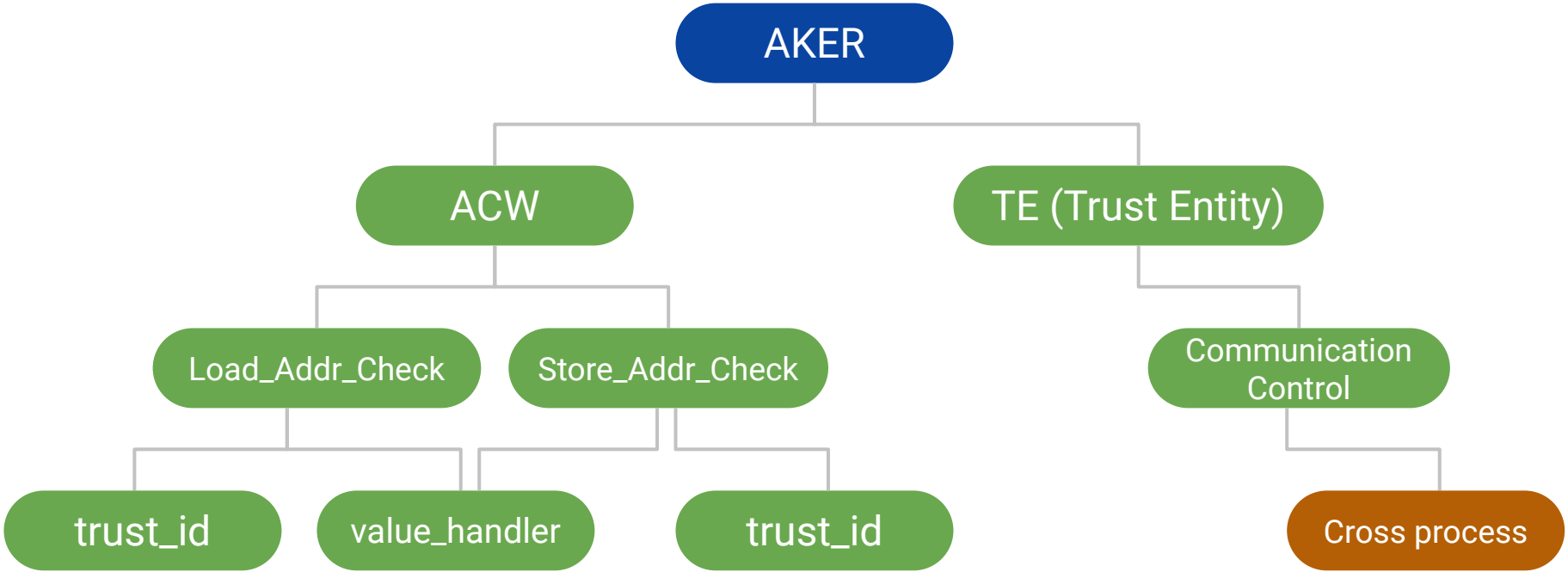
AKER: Access Control Wrapper (ACW)

Simplified version of the ACW:

```
17  Bool trust_id (int id) {
18      if (id_decoder(id)!=privileged){
19          return false;
20      }
21      return true;
22  }
```

AKER on ESP! (what did we do)

Implemented Functions:



What's next?

- "Communication Control" part >> debug!
- Security Evaluation (ongoing)
- Performance Evaluation (ongoing)

<table>
<tr><th>Week</th><th>Goals</th><th>Deliverables</th></tr>
<tr><td>5</td><td rowspan="4">Implementation of AKER on the ESP platform
(C, C++, Verilog, etc.)
● Build target ESP architecture with NoC, routing protocol
● Develop compatible AKER access control module
● Integrate AKER + ESP platform</td><td></td></tr>
<tr><td>6</td></tr>
<tr><td>7</td><td rowspan="2">Milestone Update Presentation/Report (5/17)</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td rowspan="2">Testing and Validation</td><td></td></tr>
<tr><td>10</td><td>Final Video and Report (week of 6/5)</td></tr>
</table>

Last Question?

Are all vulnerabilities in SoCs eliminated by AKER?