The status of EC601 project

Accomplishments

We finish the sprint 1**.** Developing a basic modular platform, creating an MVP with doing some lecture review, including: core functionalities, exploring various packaging technologies, and implementing standardized interfaces for chiplet communication.

Things not accomplished:

Further work is required to fully define the architecture. We need to explore the application performance and complexity tradeoffs for various parameters to define functional and physical modularity in detail.

More advanced cooling solutions, such as efficient thermal management materials, need to be explored. The ability to ensure system stability during high-performance operations depends on solving this team project.

It is also necessary to explore the impact of different packaging technologies on the interconnection effect of chiplets from actual product cases. Such as the increasingly mature MCM package which AMD takes, or Intel's EMIB and other technologies. In the following sections, we look at how physical connectivity will affect the development of standard interface protocols, and explore whether these protocols will allow for further improvements in chip performance and scalability.

While basic modular functionality has been demonstrated, deeper studies into specific performance and complexity trade-offs for various modularization strategies will help define the architecture more clearly. Further analysis of the packaging, electrical communication, and system performance is needed

The current Minimum Viable Product (MVP) covers basic modularity, packaging, and interface functionalities. However, enhancements to module diversity, user control interfaces, and system integration processes are still required

Challenges:

Challenges come from many areas. For example, when conducting literature reviews, the papers we read are difficult to understand and contain many technical terms, which slows down the reading speed.

Moreover, research papers outside the programming field tend to focus more on theoretical frameworks, mathematical models, or abstract principles. These can require a different kind of mental engagement compared to reading code or technical documentation. Without practical, hands-on examples like those found in programming, understanding the content can feel more abstract and harder to relate to real-world applications.

In such cases, the reading process slows down significantly because each unfamiliar term or concept needs to be researched and contextualized. This can be especially frustrating when you're reading to gather information for your own projects but find yourself spending more time learning the underlying theory rather than applying the knowledge.