aiWare™ by aiMotive

Why software-driven scalable hardware enables better AD

written by: Tony King-Smith,
Executive Advisor at aiMotive

Abstract: The choice of hardware platform for tomorrow's ever-more-complex vehicles is a daunting prospect. Many designers select the most flexible hardware they can find, as it has the best chance to accommodate software developed in the years ahead. While that works well for general-purpose products like mobile phones and PCs, is it also the best choice for automotive AD (automated driving) subsystems – a far more highly constrained single-application system? For aiMotive, a deep understanding of likely future trajectories of software and algorithms driving true co-design of the software and hardware for the AD hardware platform will deliver the highest performance and safest solution while minimizing costs.

The perfect compromise

The design of computers over the past 40 years has brought us to the point where the most powerful computers are no longer in a large container-sized box but in one tiny chip. Today's smartphones have processing power that would put mainframe computers to shame, yet if we look inside the circuit boards used to build any mobile phone or PC, they seem to contain only one large chip doing all the work: the SoC (System on Chip). That one chip can do almost anything: amazing graphics, blindingly fast math, blazing communications over multiple channels, and of course, impressive AI. Indeed, more and more data centers are built not on a few powerful processors, but thousands of processors SoCs that are looking increasingly like their smartphone and PC cousins.

So surely that means the best processor for automated cars should use the same technology? With technology moving so fast, new AI algorithms coming out every few months, and consumer needs constantly changing driven by mobile apps and cloud computing, do future cars have to have the same flexibility? And is it cheap if billions of these excellent SoC-powered phones and PCs are sold every year?

”Automated driving needs to become ubiquitous if it is to realize its goals of making driving safer and more enjoyable. That means it must be ruthlessly cost-engineered while offering sufficient scalability and upgradeability to accommodate technology and algorithm innovations, as well as evolving regulatory requirements over the life of the vehicle.”
Source: Tony King-Smith, Executive Advisor, aiMotive

Not necessarily – there are some significant differences:

- A mobile phone needs to run hundreds and potentially millions of completely different apps. A vehicle's AD subsystem needs only to run one app – every second of every hour the vehicle is being used
- Mobile apps are updated every few weeks - few people have died upgrading their latest game or banking app! That's very different from updating an automotive AD app, where exhaustive validation and compliance with the latest safety regulations are essential to avoid endangering the lives of millions of people
- If an app becomes outdated or too slow over time due to old hardware, consumers can replace the phone or PC easily – perhaps every 1-2 years - and migrate everything they do to it. Replacing an entire car is a far more significant and expensive undertaking for consumers who expect their car to last at least 5-10 years or longer; they would far rather upgrade parts within it

Hardware flexibility comes at a cost – a very significant cost if your volumes are not measured in hundreds of millions or billions of units per year. That's because the semiconductor technology used to build SoCs is an industry that relies on high volume production, with substantial up-front costs for each new design. Highly flexible SoCs are very expensive to design and manufacture, so they must either ship in very high volumes or become extremely expensive. They also consume much more power than more application-optimized chips, which means they generate more heat, as flexibility means powering lots of hardware that is rarely, if ever, used.

Somehow AD engineers need to find the perfect compromise: ensuring there is only just enough hardware to do the job but sufficient to handle future upgrades within the life of a vehicle and accommodating the needs of multiple models and variants of the vehicle over its design life. Specifying the upgradeability of software-driven vehicles will become one of the most important new challenges defining the upcoming era of software-defined vehicles.

Software-driven hardware

As AD technology matures, the software becomes better understood and its performance envelope better defined. Understanding and specifying this performance and capability envelope is much easier if you control all the variables – and that's what a vertically-integrated OEM does. Industry-leading examples such as Apple or Tesla do just this by constraining the variables across their portfolio of products and aligning their hardware roadmaps to their product and software roadmaps.

When companies can align their hardware, software and product roadmaps, some powerful things happen:

- Next-generation software can be developed well in advance of the hardware, confident that all the hardware resources they need will be available; no need to rely on a 3rd party to agree with your roadmap
- Innovation is greatly accelerated, as engineers are no longer constrained by the sequential process of developing software only once the hardware is agreed upon and available
- Software development within the life of each hardware generation is constrained, so engineers focus on using the performance and capability envelope for that generation of hardware. Whatever won't fit becomes part of the requirements for the next generation of hardware platforms

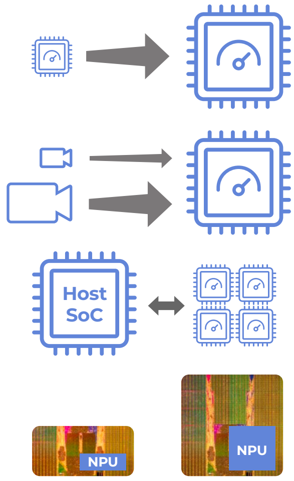
The result: shorter time to market for innovation, gaining years of advantage over competitors. And that translates to increased competitiveness

Hardware doesn't have to be fixed, even within one generation of vehicle models. While significant technical challenges exist, electronics platforms have been scalable in many markets for decades. From upgradeable memory and GPUs in PCs to the blade-based approach used by modern data centers, hardware scalability and upgradeability are not new. However, because cars are life-critical, vehicles have historically been designed as a rigidly fixed product – once completed, it never changes, which has the advantage of being thoroughly checked to be safe.

However, while fixed-function platforms are far easier to validate, they increasingly do not meet the needs of consumers and global businesses whose expectations are now accustomed to far more dynamic product development and continuous upgrades. That's one matter when the product can be upgraded every 1-2 years, but not so easy when the product must last 10-20 years. That's compounded when the vehicle relies on AI, still, an emerging technology where trends change rapidly as expertise and experience continue to build over a bewildering array of application areas.

What is a scalable AI hardware platform?

Scalability can take many forms that can work together to offer flexibility across the widest range of different vehicle models and features, and enable best possible upgrade options during the life of the vehicle.



Performance scaling
From 1 TOPS up to 1000 TOPS per chip –I common NPU IP, software APIs, optimized CNNs

Input scaling
Inputs can scale from 224x224 to 16Mpixels and beyond
High efficiency maintained or improved as input size increases

Chiplet scaling
NPU chiplets enable easy hardware upgrades

PPA scaling
Wide envelope from 1.2 to 2.5mm2 per TOPS (16nm)
Choose optimum balance between features and target area

aiMotive's aiWare4 NPU hardware IP enables chip designers to optimize their NPU accelerator chiplet or SoC design based on software requirements, and also achieve the best possible hardware IP reuse across multiple chips by using one common NPU architecture

Scalable algorithms: we can optimize the hardware by constraining our production AI software to a restricted set of algorithms. This is one of many reasons why convolution-based AI algorithms have proved so popular, as in a wide range of applications from handwriting recognition to 3D image perception, CNNs (convolutional neural networks) have proven their effectiveness. Indeed, even more, recent innovations such as transformer networks have been demonstrated to map well onto a convolution-based execution framework. Thus, a scalable convolution engine can provide highly efficient hardware while accelerating a wide range of AI algorithms.

Scalable hardware: Across multiple models within a vehicle family, there will be increasing demand for various configurations of sensors. Indeed, the sensors themselves will be increasingly upgradeable as sensor technology continues to improve. Some of the latest AD software, such as aiDrive 3.0, can "virtualize" sensor configurations, enabling the same software to work with different sensor configurations with little or no modification.

Scalable processors: if all the processing resources are integrated within one SoC, that can severely constrain what AI algorithms it can execute later. AI algorithms are evolving rapidly: some demand far more TOPS than their predecessors, while others (such as aiDrive 3.0) need significantly fewer TOPS to achieve similar performance and accuracy.

One approach is to use an NPU "chiplet" for AI acceleration, enabling the use of much smaller and simpler host processor SoCs. Just like we can add memory to some computers, NPU chiplets (using NPUs like aiWare4 designed for chiplet use) can be added to the host SoC to provide just the right amount of TOPS. Since chiplets are much simpler chips than SoCs, they do not need to be implemented in the latest silicon processes, so they can be much cheaper. And by using the latest SiP (System in Package) technologies, multiple chips can be combined into what appears to be a single chip to create an extremely small, thermally efficient, and cost-effective solution.

Scalable chiplets: once the NPU has been moved to a chiplet, then the chiplets themselves can be upgraded much more easily with little or no change to the AI software using them. For example, a small 20 TOPS chiplet manufactured in 16nm technology could be replaced by a physically identical 60-80 TOPS chiplet a few years later using more advanced 5nm process technology. This could mean hardware that used four 20 TOPS chiplets could be cost-engineered down to using only one 80 TOPS chiplet in a mid-life upgrade. Alternatively, one or more of the 20 TOPS chiplets could be replaced by one 80 TOPS chiplet as part of a sensor upgrade. This "plug and play" approach to hardware offers automotive OEMs a far greater range of hardware configuration options, increasing flexibility while reducing costs.

Scalable software: the reality is software implementing AD for any vehicle needs to operate in different operational domains – sometimes known as ODDs. This may mean that for some scenarios, such as low-speed parking, the demands on hardware are far less than when performing high-speed highway driving or navigating complex urban environments. By understanding these different ODDs early in the scalable hardware specification phase, these can be aligned with hardware fit/no-fit options (e.g., number of NPU chiplets) according to the features selected by the customer.

When co-designed with the hardware, scalable software can also ensure that any unused hardware is switched off, saving power. It can even power up hardware for short periods in "burst mode" for situations such as approaching a junction.

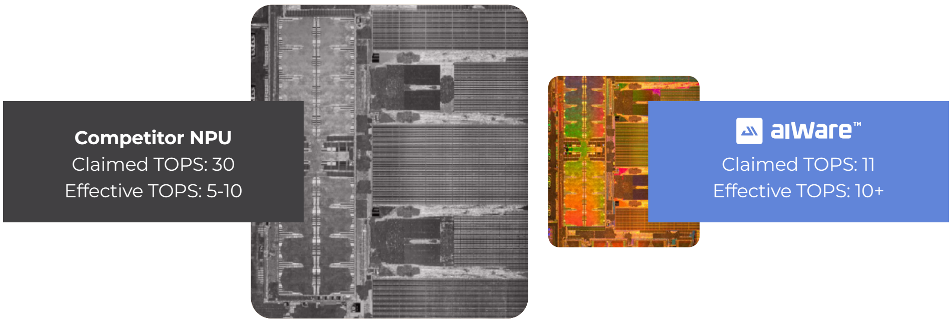
Software-driven hardware ▸ smaller chips

Another significant dividend of software-driven hardware design is that it enables hardware engineers to optimize the hardware far more effectively. This is especially important when sizing an NPU for an SoC targeting AI applications.

Chip designers worry most about PPA: Power, Performance and Area. They need to optimize all three key parameters to achieve the best possible result. And they need to get it right the first time, as creating a new chip can cost tens, if not hundreds of millions of dollars just to get the first chip prototype.

For AI hardware, how big should the NPU be? That is one of the biggest questions chip designers face since the NPU is often the largest part of the chip area, and chip area equals cost. Unfortunately, many designers rely on the overly simplistic "TOPS" measure as a key specification parameter. This ignores the efficiency of NPUs when executing AI workloads, and efficiency can vary widely. When benchmarking various SOCs, aiMotive engineers have seen well-known SoCs delivering as little as 5%-10% efficiency. That means a 50 TOPS engine can only deliver 2.5-5 TOPS usable performance. Indeed, few NPUs can reliably achieve better than 30% efficiency.

Since aiWare4 is able to achieve 85%-95% efficiency, NPUs using aiWare can be 2x to 3x smaller than other chips to deliver similar or superior performance



This is why aiWare4's industry-leading efficiency is so important for chip designers. By achieving up to 98% efficiency for well-known workloads such as Yolo, and well above 80%-85% for most CNN workloads, aiWare4-based NPUs need 2x to 3x less silicon area to deliver the same performance as most other NPUs. The key CNN workloads are understood by having software-driven hardware, enabling the NPU efficiency target to be more confidently achieved. This can translate to huge savings in the silicon area, resulting in smaller chips, lower costs, significantly lower power consumption and better performance.

Conclusions

In traditional design, hardware is fixed before the implementation of final production embedded software can begin. However, this approach will be increasingly limiting for innovation in future vehicles with AD. Using software-driven hardware design, engineers can more confidently design both ECU hardware architectures and the chips used within them to enable previously unseen scalability and configurability.

While software-driven hardware design theoretically constrains the performance and flexibility of embedded hardware platforms, in practice, this is more than compensated by enabling automotive OEMs to create a wider range of vehicles using the same core hardware and software architecture. Furthermore, by understanding the software ODD envelope, OEMs will be able to offer a wide range of hardware upgrades throughout the life of a vehicle platform, and indeed to customers within the life of their version of it. Features like hardware scalability could make all the difference for OEMs seeking new ways to create attractive consumer-friendly products in the forthcoming era of software-defined vehicles.