Faster Time to Market, lower Design Risk

All you need to know about Electronics Development with SoMs and Modular Hardware

A whitepaper by hema electronic and ADIUVO Engineering and Training LTD

hema electronic GmbH
Röntgenstraße 31
73431 Aalen, Germany
49 7361 9495-0
info@hema.de
hema.de

Introduction

The modern world is enabled by complex embedded systems, from vision and AI/ML inference to radar and signal processing and high speed deterministic, low latency networking. Complex systems bring with them several challenges for developers, to ensure they deliver on time / quality and cost.

At the heart of these complex systems is an embedded processor, along with the associated volatile and non-volatile memories, clocking and power solutions. As processing demands are increased by the application, so too does clock frequencies, data bandwidths and power demands. Addressing these demands requires not only engineers skilled in electronic design, high speed layout, signal and power integrity but it also increases the development time, and cost. Of course such a development no matter how experienced the team also contains an element of technical risk. If this risk materialises, it could lead to a design update which associated further costs and timescale impacts. This will of course result in late market entry or may even result in liquidated damages if the system is part of a larger system development.

Technical Risk In Development Projects

All developments include both programmatic and technical risk, programmatic risk is associated with achieving the desired development time scales while technical risk defines the risk inherent with the implementation of the technical solution. Of course should technical risks impact there will be programmatic impacts. The identification of both technical and programmatic risks requires input from the entire team both engineering and management. Each of the identified risks should have an appropriate timescale and cost impact along with a mitigation plan.

An example of a programmatic risk would be the input system requirements not being made available at the time they are needed by the development team. Such a situation would result in delaying the start of development, which pushes out the end date. Other examples of programmatic risk include the availability of skilled resources.

Technical risks however, are dictated by technical aspects of the project which may present a challenge. Examples of technical risk include the use of a new image sensor which is not supported by the standard embedded Linux solution for the target device. In this case drivers would need to be ported or written, along with a hardware and FPGA design being created.

Another technical risk would be the implementation of a DDR4 memory interface, this risk would occur late in the project once the boards came back for testing / qualification and could result in significant impact to the program.

As the project develops technical risks should be retired, for example the project plan should create a run out date for which the risk can be either retired or considered as impacted.

Reducing the technical risk on a project and retiring those subsequently identified risks enables delivery on time, quality and cost.

One measure often used to identify technical risks is the identification of the Technical Readiness Level (TRL). The TRL ranges from levels one to nine, with one being the lowest and nine the highest.

1. Basic Principles Observed - Understanding of the principles involved in the issues - Using the DDR4 memory example this would be understanding the data sheet and understanding the physical layers of the DDR4 connectivity.

2. Technology Concept - The basic concept of how the system will work, in the DDR4 example this would be identifying the pins and verifying the pin allocation and creating the schematics.

3. Experimental Proof of concept - Creation of a breadboard using the processing device and the DDR4 memory. This would require creation of schematics and the layout, procurement of parts and the manufacture, assembly and test of the breadboard.

4. Technology Validated in the lab - Successful testing in the lab, the DDR4 breadboard successfully passing the stress tests and other characterisation tests.

5. Component Technology validated in relevant environment - The breadboard being tested across temperature and vibration extremes and continuing to work to

demonstrate the DDR4 is successfully working when exposed to the operating environment.

6. System / Sub System Technology validated in relevant environment - The completed system or sub systems being designed, manufactured and tested against requirements and environment. In the DDR4 example this would be the creation of the complete solution using the DDR4 memory.

7. System Prototype demonstrated in operational environment - assembled system tested against the operational environment.

8. System complete and qualified - Assembled system has successfully completed the qualification testing against its operating requirements and environment.

9. System proven in operational environment - System deployed and working in the field.

By listing each of the technologies being developed as part of the project and assigning a TRL we are able to see where the technical risk resides. Typical developments for complex electronic solutions include assessing the TRL of designing and implementing the processing element, memories, power solutions and high speed interfacing for the desired environment. TRL does not have to be limited to physical aspects; it can also be used to aid in identifying aspects of the software development process where a low TRL exists, for example when working with operating systems, kernel drivers etc.

One method of addressing these TRL challenges, while reducing the technical risk and enabling the development of the target solution from day one is the use of modular hardware designs that consist of System on Modules and an application specific mainboard. The design risk can be reduced even further, if the mainboard as well is designed from a modular design library and thus uses well-proven hardware building blocks and circuits.

The development of the processing solution also diverts focus from the implementation of the specialist functions and capabilities or value added engineering.

What is a SOM?

A System on Module is a board-level circuit that integrates a system function in a single module. In the processing world that includes, power, clocking, volatile and non-volatile memories while also breaking out the IO lines to be able to connect the SoM with the application specific mainboard.

Within the programmable logic world, there are a range of SoMs available from several suppliers such as Avnet, Trenz, Krtkl, Xilinx, Enclustra, MyIR, Alinx and iWave. This diverse range of SoM suppliers provides the developer with a range of SoM from which to select.

The SoM becomes the heart of the system around which the reminder of the hardware is developed. A correctly architected SoM based solution provides the developer with a range of solutions including

- Reduced technical risk in development. Complex design elements such as DDR3/4 interfacing is provided and configuration and power solution / sequencing are also implemented on the SoM. Reducing the risk of a respin being required.
- Through life support and obsolescence management is addressed by the SoM provider. Components on the SoM which go obsolete or have long lead times are addressed by the manufacturer of the SoM.
- Development of the program can start earlier using a SoM and a mainboard. All the necessary board definition files are provided and verified. Software is designed on target hardware from the beginning. No evaluation board with lots of stuff is needed.

A solution based around a SoM will be developed around a three layer model, where each layer of the model builds upon the lower level.

1. Hardware - The hardware level is the lowest level of the system and contains the selection of the SoM, the design of the main board. This level will also include any prototyping stages which can be used to demonstrate and retire technical risk in the project.
2. Software - The software level is the middle level of the stack and comprises the generation of the necessary software artifacts which will be used by the SoM for example embedded Linux operating systems. This layer may also include the

identification and selection of Intellectual cores which will accelerate the programmable logic development process.

3. Application - The highest layer of the stack is the creation of the final application. This application can be either a complete FPGA design or a Heterogeneous SoC solution which includes design elements in both the programmable logic and the processing system.

Image 01: SoM based solutions are developed around a three layer model, consisting of hardware, software / IP-Cores and software applications.

Benefits of using modular hardware designs

The main benefit of using a SoM based solution and modular hardware design is that developers are able to reduce technical risk and increase the TRL of the solution at the start of development.

The SoM itself provides the developers with a solution which aligns to TRL 5 - Component Technology validated in relevant environment. The manufacturer of the SoM has designed, tested and qualified the SoM prior to marketing it. This enables the developer to start focusing on the design of the main board which contains the value added activities.

However, it is not just the physical hardware elements which are provided by the SoM manufacturer but also additional elements which aid the design and development. This includes embedded Linux operating systems for the target, schematics and user guides to describe how to design in the SoM into a main board along with the example projects and designs which can be used as a basis for development.

As the SoM is purchased off the shelf the development time is reduced, it also enables developers to also start working with the SoM on day one of the project as well to increase the TRL of the application circuits.

Using a SoM therefore provides a reduced total cost of initial engineering, starting development with a proven design with a Higher TRL. While also enabling a reduced effort in software development. If several developments are being undertaken at once it makes considerable sense to standardise on a common SoM such that the wider engineering team is able to to effectively work with the SoM and leverage any additional benefits such as reuse between developments.

All of these benefits can lead to a faster time to serial production.

While the technical, programmatic and cost benefits of using a SoM are many there are also several marketing benefits which include, being first to market with innovators title and of course opportunity for a greater market share and profit. This also enables the developers to have a head start on thinking about the next generation of product and potential in the field upgrades to the currently deployed units.

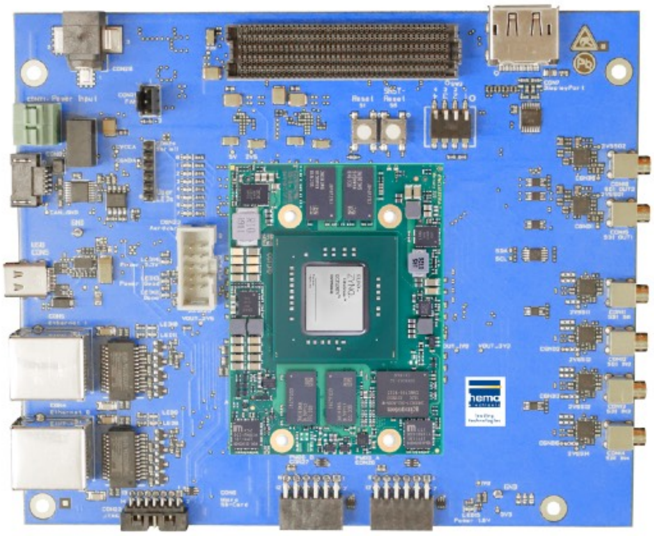

Image 02: Example of a modern embedded vision platform based on a mainboard and SoM.

Selecting a SoM for your application

Given the range of SoM available, being able to select the correct one for your application is critical and may be challenging. Selection criteria will of course vary from application to application. However, there are a number of higher level choices which can help selection

Key elements which will influence a SoM selection include:

- Processing Element - FPGA or Heterogeneous SoC e.g Zynq, Zynq MPSoC or RFSoC - Logic Resource, Processor Numbers and Performance
- Volatile Memory - DDR3/DDR3L/DDR4 - Capacity and Bus Data Width, supported interface bandwidth, provision of DDR for just PS or PL also on heterogeneous SoC.

- IO Availability - Number of IO broken out, type of IO broken out e.g. Gigabit SERIAL Links, number of high performance, high range, high density IO provided.
- Non-Volatile Memory - SD, eMMC, QSPI, Provided for the configuration of the FPGA / SoC solution.
- Clocking - Does the SoM provide flexible or reprogrammable clock generation circuits to meet diverse clocking demands.
- SoM Interfaces - JTAG, Ethernet, USB, UART etc - the interfaces which are provided directly on the SoM itself and as such do not need to be provided by the Carrier Card.
- Qualification level - is the SoM available in the necessary qualification level for example Industrial, military or automotive.

Designing a SoM into your system

Once a SoM has been selected the next step is integration of the SoM in the overall solution and design of the mainboard which mates with the SoM. To aid with the processes the SoM manufacturer normally provides an information package which can be of use to help this integration.

- SoM User Guide - Defines how to interface the SoM with a custom carrier card. This includes information on configuration, clocking, powering and otherwise interfacing with the SoM
- Mechanical design files - 3D files and mechanical drawings which define the location and positioning of connectors and mating / securing interfaces
- Constraints - The design constraints and pin locations for each of the SoM connectors
- Routing & Tracking information - Information on the signal layout of the SoM including the single or differential ended routing and trace length information to enable continuation of length matching should it be required.
- IO Banking - To provide maximum flexibility, many SoMs do not have broken out IO banks connected to a power supply. Instead the Carrier card provides the IO Bank voltage, this requires the designers to be able to correctly interface with the SoM power system to ensure correct sequencing on power up and down.

Prototyping with Modular Embedded Systems

One of the main advantages of using a SoM is the ability to start development on day one using the SoM to retire technical risk and increase the TRL. Depending on the SoM selected it may be possible to achieve this using the SoM on its own if sufficient connections are provided on the SoM itself. If the SoM does not provide the necessary interfaces, a development board can be purchased often from the SoM supplier or another vendor which breaks out the SoM IO to commonly used industry interfaces for prototyping for example PCIe, QSFP, FMC, SZGZY and Pmod along with Ethernet, USB, UART etc as required for communication and control of the Prototype solution.

The additional benefit of using a prototype carrier card is these boards are provided by the supported software solutions and examples. These examples may not be identical to the custom carrier card in development, they do however, provide a good starting point for understanding the development flow and use cases.

Close to series prototypes in just six weeks

For even faster development, companies like hema electronics provide close-to-series prototypes of electronics that can be produced in as little as six weeks. If customers decide on a SoM they would like to use for their design, they just need to define the required interfaces, IOs and additional functionalities for the mainboard. The modular concept of hema's embedded vision platform, that especially targets embedded vision projects, includes the hardware as well as middleware and a comprehensive software framework. Currently, there are already over 45 Building Blocks to choose from in a dedicated hardware library, which includes interfaces such as USB, CAN, Ethernet and Wifi / Bluetooth as well as numerous special video and sensor interfaces. Customer-specific functionalities and circuits not yet available in the library can be integrated easily as well. Thanks to a standardized connector, different system-on-modules can also be used with the same mainboard to leverage the required processing power etc and to build-up different prototypes.

Building Blocks enable fast mainboard designs

Behind each Building Block for hardware design are templates for circuit diagram and layout. The circuits are already being used successfully in numerous industrial applications. For each customer project, the function blocks are arranged on the board according to the customer's specifications and the routing is optimized. The effort required for this is significantly less than for a new development. Board format, dimensions and arrangement are freely selectable. Once all the elements have been selected, the design is completed, the hardware is manufactured, and the desired IP cores and software frameworks are installed and programmed entirely under one roof at hema electronic in Aalen. Customers thus receive their individual electronics in the shortest possible time - usually after just six weeks.

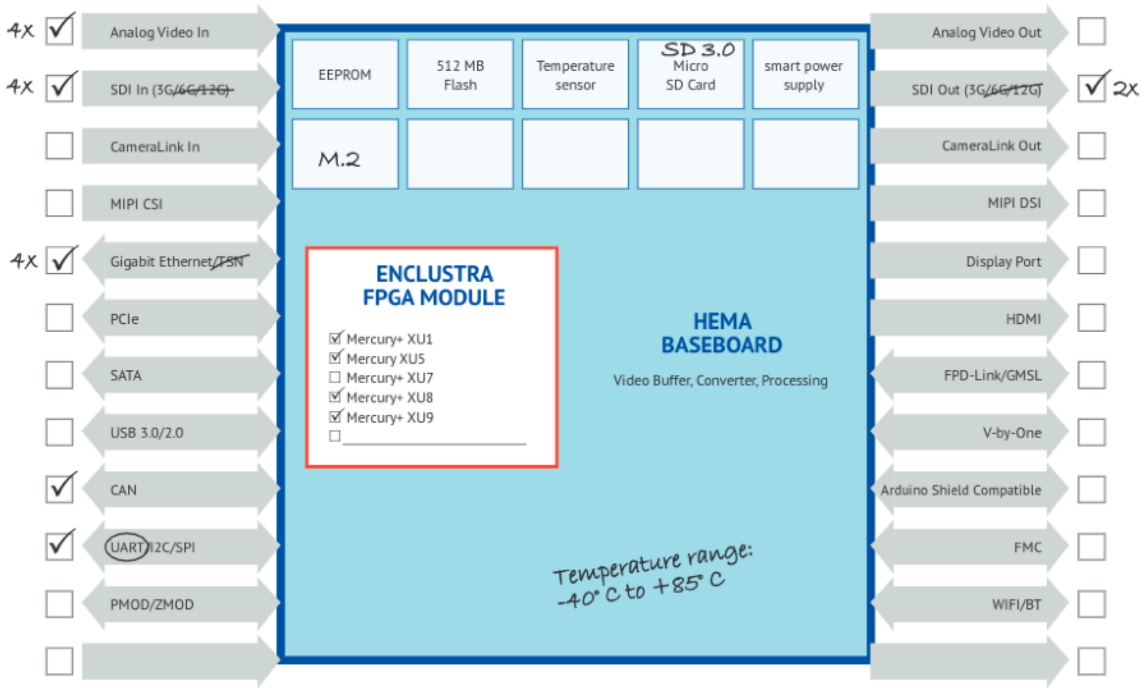

Image 03: With the hema embedded vision platform for rapid prototyping customers can choose the desired SoM, interfaces and functionalities from a library of building blocks.

Fast entry into software development

In addition to FPGA programming and an operating system for the ARM processors, hema's software offering includes comprehensive middleware for processing sensor data and specialized software libraries such as Halcon from MVTech. Combined with these comprehensive upfront hardware and software services, customers can perform functional tests and start developing their own application at the earliest possible stage.

Short paths to mass production

All circuits, components and the layout of the prototypes are suitable for industrial use. This means that even the first functional prototypes are very close to the later series product. hema provides series qualification of the electronics and production as well as comprehensive support in project and lifecycle management. The company relies on individual consulting and agreements, with which hema complements the customer's core business in a custom-fit manner. hema also attaches great importance to the robustness, durability and long-term availability of its electronics. The successful use in solutions for the defense and security industry underlines this claim.

Conclusion

Developing electronic systems is challenging, managing technical and programmatic risk can be a challenge. Using modular hardware design and a SoM provides multiple benefits to the developer, significantly reducing these risks allowing focus on the value added engineering. Using a SoM also enables a faster transition to manufacturing. Next time you are developing an application before you start from a blank sheet of paper consider the benefits a SoM would provide.

Authors

Oliver Helzle

is CEO of hema electronic GmbH. He joined the company in 2004, when he successfully completed his degree in industrial engineering, and step-by-step took over the management of the family-run company from his mother. Prior to his role as CEO, he held several management positions and successfully led customer projects in industrial automation, surface inspection and quality control, automotive and defense industry.

hema electronic is a leading development service provider for the electronics industry in the field of hardware and software design for embedded vision boards and systems for applications in industrial automation technology, defense and security technology. From consulting and conception to design (FPGAs, DSPs, embedded processors), qualification, rapid prototyping and small series production to lifecycle management, hema electronic offers everything from a single source. hema electronic effectively supports its customers in being the world market leaders of tomorrow.

o.helzle@hema.de
hema.de

Adam Taylor

Adam Taylor is a chartered engineer and fellow of the Institute of Engineering and Technology. Over his multi-decade career, he has had experience within the public and private sector, developing FPGA-based solutions for a range of applications including RADAR, nuclear reactors, satellites, cryptography and image processing.

Adam is also a Visiting Professor of embedded systems at the University of Lincoln. Education is his deepest passion and he has delivered thousands of hours' worth of training to corporate clients and casual hardware enthusiasts alike.

adam@adiuvoengineering.com
www.adiuvoengineering.com