

Reducing development and testing time with System-on-Modules

Trym Sneltvedt

November 20, 2019

1 Introduction

1.1 Background

In our digital world, the demand for computing platforms with higher performance and smaller footprints increases on a daily basis. In later years a class of integrated circuits (ICs) has emerged as the go-to solution for embedded devices where performance, features and small form factor are important parameters, System on Chips (SoCs).

SoCs are fully fledged computer platforms with the necessary memory and storage for operation and powerful peripherals like field programmeable gate arrays (FPGAs) or modems for wireless communication like WiFi, Bluetooth and ZigBee. All of this is packaged on a single chip, thereby the name. In contrast to traditional microcontrollers which has been a favourite in the embedded landscape for decades, SoCs pack a bigger punch. They are equipped with more powerful microprocessors, sometimes even multiprocessors. This makes them a sought after platform for applications where high throughput are crucial, like aerospace, automotive, signal processing, communication, multimedia and military use.

A downside to the increased complexity of the electronics and the demands for smaller devices is the packaging of ICs. Integrated circuits with a high pin count very often use ball grid array (BGA) packages. These are ICs with the connectors placed in a dense grid on the bottom of the packaging, instead of on the tradition side placement. While very space efficient, this way of connecting ICs to printed circuit boards (PCBs) introduces several challenges for hardware designers. Firstly, the clearance between each pin requires PCBs with more than 2 layers. Tradition methods of PCB manufacture like copper etching or milling

is only possible (or practical) for 2 layer boards. This means prototypes has to be produced at larger production factories that has the expensive equipment available. Multi-layer boards are also more expensive. It is true that the manufacturing cost and lead time for PCBs has decreased drastically over the last years with the technological advances made, but it still imposes a high cost and increases the time to market. Another important factor is debugging. Advanced multi-layer PCBs are challenging to debug and slows down the development and prototyping phase of product development. Time-to-market is an increasingly important factor when developing embedded systems, as the market is highly competitive and disruptive.

About System on Modules (SoMs)...

1.2 Scope

This report explores the process of designing, implementing and testing an embedded system previously powered by a SoC, with a SoM. A case study was performed in cooperation with Revolve NTNU, a Formula Student team that spends one year designing, building and testing an electric race car, before attending multiple international competitions during the summer where they face off against teams from other universities worldwide.

The automotive setting makes for an ideal environment to look at the entire product development process, as the product must be quickly prototyped, verified and produced, and the deadlines are absolute. The vehicle is useless without a functioning control system.

1.3 Outline

2 Theory

When opting for a SoM for your embedded system, it is necessary to take the requirements of your system into account, and make sure your chosen SoM is capable of meeting your demands. This section discusses different features often made available on SoM's, and finishes with a comparison of several SoM's currently available on the market.

2.1 Processing unit

The processing unit is the core of the module. It is common for SoM's to utilize SoC's as their processing units, giving access to both powerful processing platforms and peripherals like programmeable logic and wireless communications.

2.2 Random access memory

High speed external memory serves as a peritcularly challenging task for hardware designers, as they call for high speed signaling, introducing concepts like *differential signaling*, *impedance matching* and *fan-out* to the PCB layout. Having sufficient RAM on the SoM is therefore a big advantage.

2.3 Non-volatile memory

When running an advanced embedded system, there might be a need for a fully fledged operating system (i.e Android for multimedia or Xenomai for real time applications). This makes it necessary with more persistent storage than in conventional embedded systems. Most commercialy available SoM's feature some sort of non-volatile memory. the most common types are NAND flash and eMMC flash.

2.4 Power electronics

Advanced SoC's often require *power sequencing* of some sort. There might be different voltage levels for internal logic and I/O pins, and the voltages has to be enabled in a specific order. This means that the hardware designer has to have dedicated circuitry for different voltage levels. This would often be switching DC-DC convertert or linear regulators. It is common for a SoC to require 3 or more different voltage levels [ref Zynq 7000 datasheet]. If the remaining design cannot use all of these voltage levels, they are essentially additional complexity which increases the cost and design time of the system.

2.5 Ethernet

2.6 Gigabit transceivers

2.7 Universal Serial Bus

2.8 SoM comparison

Table X shows and compares a small set of commercially available SoM's.

	Processing unit	RAM	Flash
Enclustra Mercury SA1	Altera Cyclone® V ARM®	1GB DDR3L	16GB eMMC flash
Enclustra Mercury ZX5	Xilinx Zynq 7000	1GB DDR3L	512MB NAND flash
Avnet PicoZed	Xilinx Zynq 7000	1GB DDR3	4GB eMMC, 128Mb QSPI flash

Table 1: SoM comparison

3 Method

3.1 PCB layout

3.2 Power supply

As with all electronics, you will have to supply it with energy for it to work as intended. In the scenario being explored in this paper, we have a SoC-based system and want to transition to a SoM for faster prototyping and iteration, we need to make sure that the power supply circuitry that was previously employed for the SoC is capable of handling a SoM. We will explore two methods for determining the applicability of an already existing power supply, simulations and analyzing an assembled circuit.

3.2.1 Simulation

3.2.2 Circuit analysis

4 Results

4.1 Electrical performance

Current draw, resiliance to noise on power supply... Compare to VCU 2019?

4.2 Computational performance

Is this even interesting? Better SoC gives better performance

4.3 Design process

By reusing software and hardware design from last year's design, issues were reduced to a minimum.

The electronic design automation (*EDA*) software used for schematics and pcb layout, *Altium Nexus* supports multiple features which shortened the design process significantly. Firstly, since VCU19 used *hierarchical design* (i.e. project is composed of multiple sheets and subsheets), large parts of the schematics developed during last years design period could be reused.

Simple subcircuits like CAN-FD transceivers could be directly copied without issue. In addition, the subsheets are modular in the sense that they appear as an electrical component with a specific set of interfaces, or *ports*. This means that sheet using the same set of ports can be intechanged freely. The modularity of the sheets was heavily used during the development of VCU20. The new processing unit features a superset of the VCU19 SoC interface, meaning they are perfectly compatible.

There is also the possibility of reusing sheets in the same project. VCU20 has 4 CAN-FD transceivers, but since the transceiver schematic is identical for all 4 interfaces, the sheet can simply be repeated.

There are also features for reuse in the PCB layout part of the design. Altium Nexus has a feature called *rooms* which is a grouping of component footprints and how they are connected. It is possible to copy the layout if a room to other rooms. The designer chooses how rooms are generated, and by default they are

created per sheet. This means that the layout for one CAN-FD transceiver can be perfected and then copied over to the other 3 transceiver rooms.