RECONFIGURABLE HARDWARE DESIGN

Adam Taylor[1],

[1] Lincoln DSP, Adam@lincolndsp.com

The timescales involved in delivering modern systems are decreasing, at the same time the complexity of these systems is increasing. To meet these challenges, we need to be able to leverage not only design reuse but also highly flexible reconfigurable systems and supporting eco systems.

Reconfigurable hardware provides the user with two potential use cases

- *Use the same base hardware, for a range of applications with customizations via Software or Programmable Logic.*
- *Update the application in the field to incorporate new standards and interfaces*

One-way, a reconfigurable hardware platform is achieved is using technologies such as System on Chip (SoC), which combines ARM cores with programmable logic. These devices provide a wide range of Industry standard interfaces SPI, I2C, Gigabit Ethernet etc. which can be used and enabled / disabled as the application requires. While the on board programmable logic enables interfacing to high performance devices such as, (ultra) wideband ADC, DAC and Image Sensors.

This paper will look at these devices, some of the architectures which can be used to create flexible hardware and the surrounding eco systems which can be used create and leverage these re configurable systems

INTRODUCTION

Development time scales are decreasing, at the same time complexity of the end solution is increasing and the budgets available from the customer are decreasing. These challenges prevent the engineering team from starting each new development with a blank sheet of paper. Instead modularity and reuse are the cornerstones of modern developments, closely coupled with software configurable identities and functionality. For instance, the wide spread adoption of software defined radio across a range of applications from Military Data Links to Satellite Communications and Radio Astronomy.

While combined modularity, reuse and software configurable identities and functionality provide a partial solution to these challenges. What is needed along with these is hardware configurability and a defined approach of how these different elements interact. Hardware reconfigurability, modularity and reuse is addressed by using System on Module and many cases Heterogeneous System on Chip devices which combine high performance ARM cores (and associated peripherals) with high performance programmable logic. This class of device was first introduced in 2010 and has seen significant adoption across a range of applications and end uses.

The Role of SoC and SoM

Programmable system on chip devices like the Xilinx Zynq and Zynq UltraScale+ are true system on chips they provide not only high performance ARM cores and programmable logic. But also, Communications interfaces, Memory Controllers, Clock generators and even Analogue to Digital Convertors. This enables the designer to optimise the solution, applications which are best suited for processors cores can be implemented with the cores, while the programmable logic allows the use to implement high performance signal generation and processing algorithms leveraging the highly parallel nature of the logic.

Like most microcontrollers within the processing system along with the actual processor cores there are several industry standard peripherals provided. From I2C and SPI controllers to Gigabit Ethernet and PCIe. This enables the provision of a range of different system interfaces when using the same basic core, however, the designer needs to be able to use this core effectively across several applications or products.

System on Modules integrates a system function on a module, which can be reused across several applications. Typically, a SoM is mounted upon an application carrier card, which contains the specific interfaces and functionality for the application at hand. When discussed in the SoC context the System on Module typically contains the SoC, supporting volatile and non-volatile memory along with clocking and power supplies and breaking out IO from the SoC such that it can be used on the mission card. Using a SoM therefore allows you to focus upon value added activities while reducing the risk of implementing complex digital circuitry. This use of a SoM therefore not only reduces the development timescales, and enables hardware platforms to be provided to the software and FPGA developers earlier in the product development cycle.

When combined SoC and SoM allow us to form the basis of our reconfigurable hardware. The flexibility of the SoC can be used to route the industry standard peripherals from the processor half of the SoC to the application interfaces as required for the application as shown in figure one. This rerouting can be performed if necessary in real time using partial reconfiguration of the programmable logic.

High performance interfaces such as (ultra) wide band ADC and DAC used in many RF applications can be implemented using the programmable logic half of the SoC. Typically, these devices use either a JESD204 multi gigabit interface or multiple LVDS lanes. Supporting these the reception or transmission of data on multiple LVDS lanes with respect to timing alignment can be challenging. This is where the specific functions within the programmable logic IO come into play, enabling the designer to adjust timing and alignment with a fine granularity and to terminate the signal on chip as well reducing the need for on board terminations.

This enables the SoC and the SoM to be able to provide any to any interfacing capability with the appropriate PHy (if required). To leverage this capability careful consideration is needed at the system level, to ensure the best modularity, reuse and reconfigurability at the hardware and software level we need to leverage a stack based approached with clear definitions between layers.

These devices also provide limited mixed signal capability with the inclusion of ADC's within the device. Capable of sampling at up to 1 MSPS these can be used to provide health monitoring and security (tamper) solutions to the overall solution.

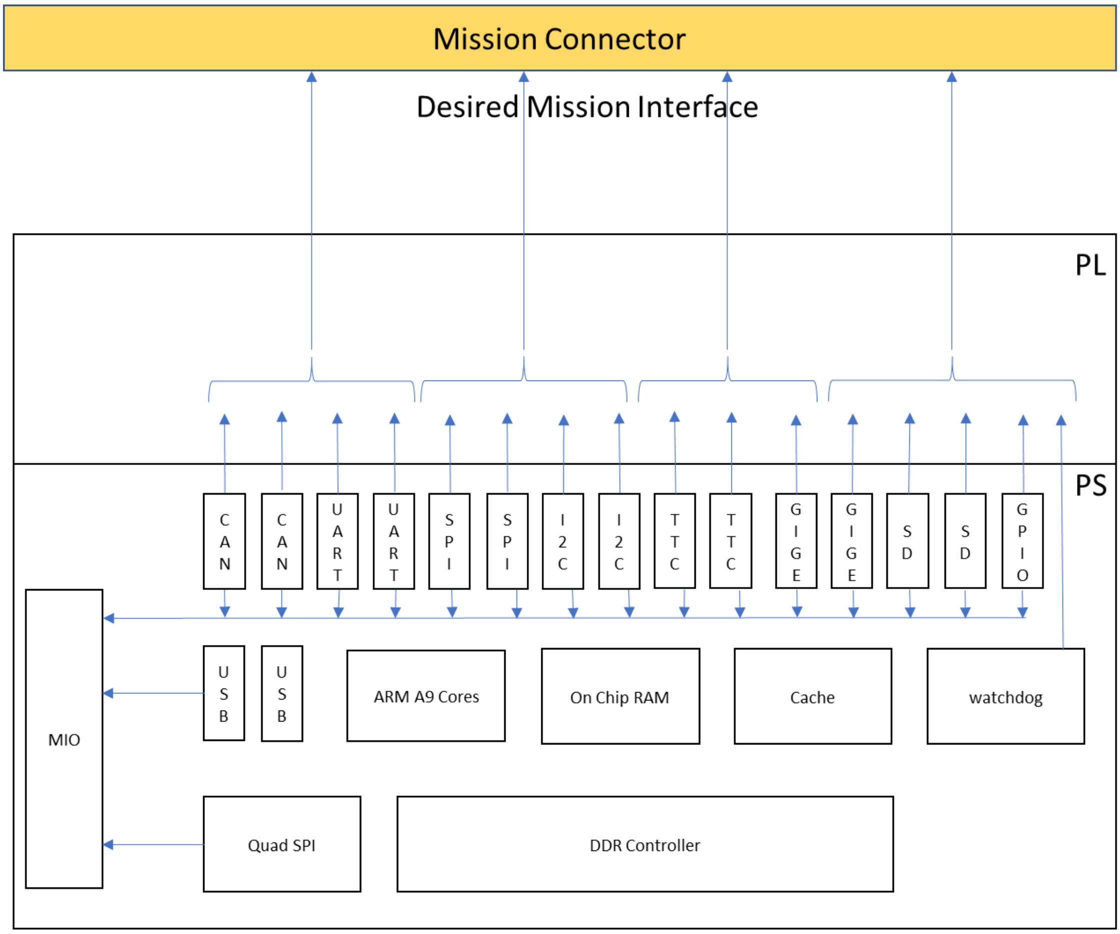

Figure z, Accommodating Interfaces Standards

Figure one, Flexible Routing.

Modularity as an Enabler

Implementing a modular stack as demonstrated within figure two provides several benefits namely: -

- Reuse of Design
- Design analysis exists for each module
- Common Architectures and Interfaces
- Integration time is reduced as layers of the stack are already validated

- Can be used at the front end for more accurate estimates
- NRE Cost is reduced

Based on the OSI stack this provides a clear definition of the role of each element of the stack, like the OSI stack not all levels are required only those necessary to implement the design solution. Figure two shows the elements circled in red which would form a reconfigurable hardware platform. The base platform (module level) is provided by the SoC mounted on a SoM while the SW Application Programmable Interface (API) and FPGA API enable the customisation of that SoM for the application at hand.

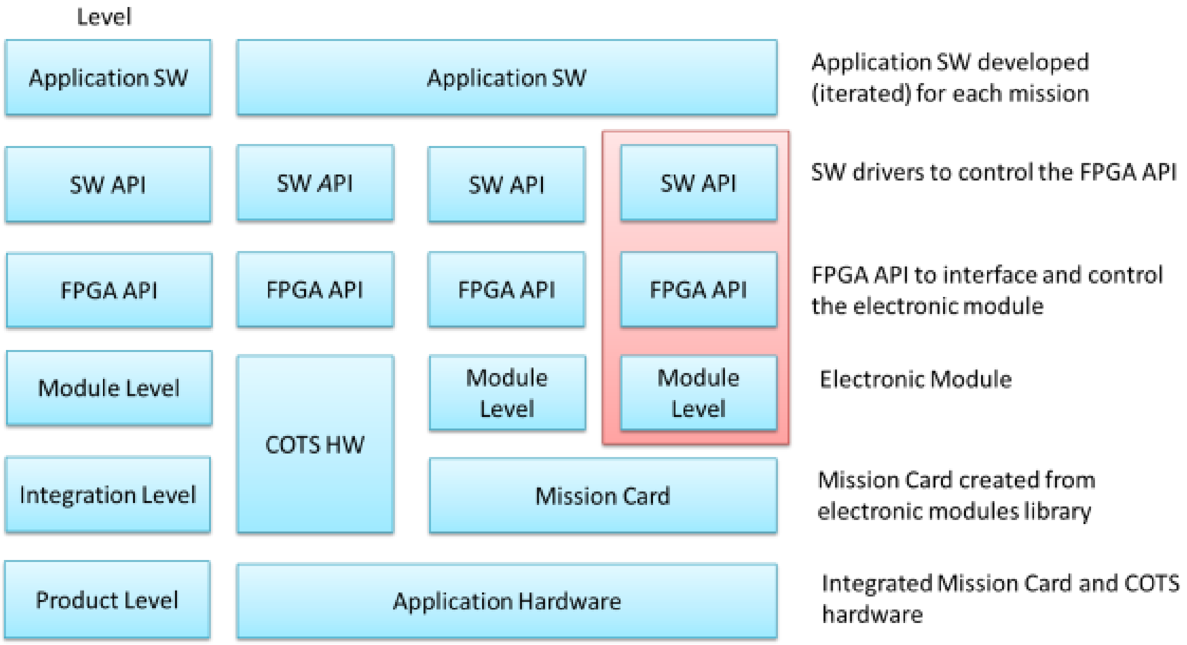
Level
Application SW
Application SW
Application SW developed (iterated) for each mission
SW API
SW API
SW API
SW API
SW drivers to control the FPGA API
FPGA API
FPGA API
FPGA API
FPGA API
FPGA API to interface and control the electronic module
Module Level
COTS HW
Module Level
Module Level
Electronic Module
Integration Level
Mission Card
Mission Card created from electronic modules library
Product Level
Application Hardware
Integrated Mission Card and COTS hardware

Figure two, A Modular Stack Example

Ensuring communication between the external world and layers of the stack is critical if the stack is to be implemented correctly and provide the maximum reconfigurability. Figure three demonstrates the different communication methods and interfaces which can be used to communicate with modules at the same level and with the higher and lower layers of the stack.

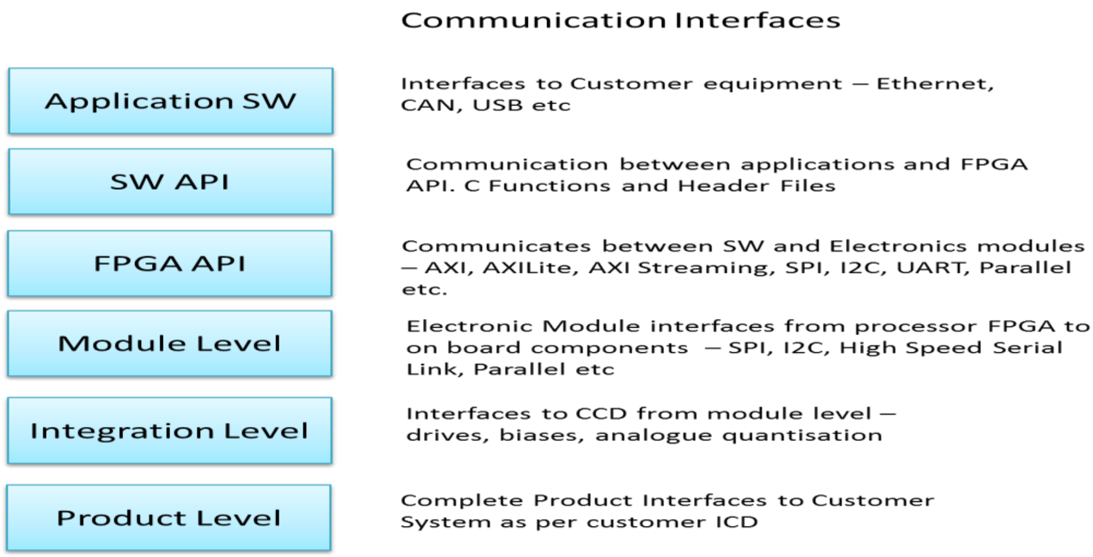
Communication Interfaces
Application SW
Interfaces to Customer equipment – Ethernet, CAN, USB etc
SW API
Communication between applications and FPGA API. C Functions and Header Files
FPGA API
Communicates between SW and Electronics modules – AXI, AXILite, AXI Streaming, SPI, I2C, UART, Parallel etc.
Module Level
Electronic Module interfaces from processor FPGA to on board components – SPI, I2C, High Speed Serial Link, Parallel etc
Integration Level
Interfaces to CCD from module level – drives, biases, analogue quantisation
Product Level
Complete Product Interfaces to Customer System as per customer ICD

Figure three, Communication within the stack

Maximum reuse within the stack is derived from the use of industry standard interfaces as much as possible. Bespoke interfaces should be limited and justification for the use of these should be clear and demonstrate the unavoidable need for such an interface.

Eco System

One of the key aspect of developing for hardware reconfigurability is the use of the correct development eco system to enable the co-development of the FPGA and SW API levels. Tools like Software Defined SoC (SDSoC), which enable the use of C / C++ to develop both the application for the processors and accelerate functions into the PL, provide the developer with significant benefits. SDSoC is a system optimising complier, which uses High Level Synthesis to translate high level C/C++ to HDL and a connectivity framework to seamlessly connect the resultant programmable logic design into the user programme. Used in this manner the developer only sees the significant increase from the use of programmable logic. Correct use of this tool requires a module level platform definition, which captures the IO interfacing requirements within the programmable logic e.g. the DAC and ADC interfaces. Figure Four demonstrates how the platform and user accelerated SDSoC code are combined at the system level enabling faster creation of the application.

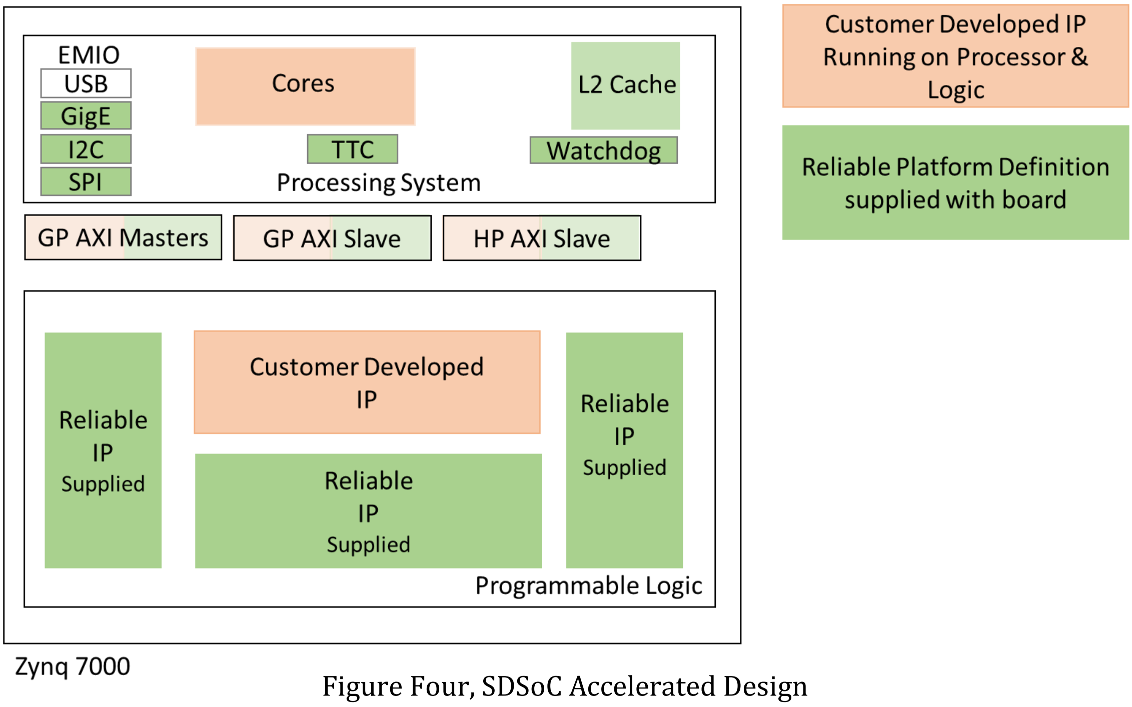

Figure Four, SDSoC Accelerated Design

This eco system can also be used with industry standard frameworks and open source libraries to develop the application software as well, which leverages the SW and FPGA API lavels.

CONCLUSION

Reconfigurable hardware provides significant benefits to developers to reduce time scales and increase re use when used in conjunctions with a well thought out reuse stack as demonstrated in this paper. Device integration is also increasing which will see the increased packaging of RF solutions within these devices enabling tighter system integration and further increasing the benefits of reconfigurability at the hardware level.