FPGA based System on Chip (SoC) for Space computation

Abstract:
The demand for developing reconfigurable embedded systems for space compatible instrumentations has increased in recent times. Often these systems include interdisciplinary applications domains such as signal processing, wireless communication, multimedia, evolutionary computation etc. Performing these computations using embedded processors alone, cannot achieve the desired computational capability to fulfill the requirements of massive parallelism, higher memory bandwidth, and higher execution speed. In order to meet these requirements, Field Programmable Gate Arrays (FPGAs) are used by exploiting the reconfigurable resources by implementing the algorithms in parallel or pipelined fashion using HDL languages. Beyond this flexibility, embedded processors inside in FPGA are used for running software application along with custom hardware by making use of reconfigurable resources, all on same package as System on Chip (SoC). These systems can provide the speedup of custom hardware without the cost of development for Application-Specific Integrated Circuits (ASICs).

Introduction

FPGAs are prefabricated programmable logic devices composed of Look Up Table (LUT) based programmable logic blocks connected by a programmable routing network. These devices are programmed on field as opposed to devices having internal functionality fixed and hardwired by the manufacturer such as Application Specific Integrated Circuits (ASICs) and Application-Specific Standard Parts (ASSPs). These devices include resources such as Flip-Flops (FF), Random Access Memory (RAM), Multiply and Accumulate (MAC) units, DSP48E and soft/hard microprocessor cores. It also has several third party Intellectual Properties(IPs) like Ethernet MAC controller, Digital Video Interface (DVI), High speed serial I/O's etc. Several grades (Consumer, radiation tolerant, low-power, high performance) of FPGAs are available from different vendors from Fab-less industry like Xilinx, Altera, Microsemi, Lattice Semi-conductors etc.

The primary objective of the usage of FPGAs for implementing the algorithms into hardware is to minimize the gap between idea/ algorithm development to embedded system design. The advantage of FPGAs is its reconfigurability, low design costs and low time-to-market. During the last decade, modern FPGAs are available with soft and hard embedded processor cores to enable the designer for building complex embedded applications. To enhance the execution speed of the algorithm, dedicated hardware accelerators have been developed and interfaced with the processor as a coprocessor in System on Chip (SoC) platform.

There are several preferences for choosing a platform such as Micro-Controllers (MC), Digital Signal Processors (DSP), FPGA and Application Specific Integrated Circuits (ASIC), for developing an embedded system. In order to achieve higher performance, applications need to be implemented either in multi-processors or in dedicated hardware accelerators/coprocessors. The platform selection depends on factors such as performance, power consumption, and cost per chip. The ease of tools accompanied by a specific platform to assist the developers for developing the system within the constraints of system cost and project time also plays an important role. MC and DSP platform make use of embedded software oriented methodologies to develop the system. However, the designers using FPGAs as their development platform have the ability to use the processor-based approach, developing their system partly in firmware and partly in hardware, or entirely developing the system in the hardware [3].

In future space project requires advanced technologies to miniaturize the various instruments of spacecrafts which includes those for on-board computing as well. For instance, a generic single-chip computing platform can be used on-board small spacecraft, which can be reconfigured remotely from the ground station.

FPGA based SoC design is chosen because of the following reasons:

1.Selecting Off-The-Shelf (OTS) microprocessor for a particular application which can meet all system requirements (like floating point arithmetic, power, speed, ease of tool) is time-consuming. So it is advantageous to find an alternative which allows the designer to tailor a processor and a specific set of features and peripherals for the application to be implemented. FPGA based design gives this flexibility to the designer over either MC or DSP based system.

2.The designer of FPGA based embedded system has flexibility to customize the design by adding any combination of peripherals and controllers. A unique set of peripherals can also be designed for specific applications, and the designer has privilege to add as many peripherals to meet the system requirements, which cannot be done in MC or DSP based system. Features not present in the initial phase of the design can also be added in the later part of the design.

3.Hardware and software concurrent development and coexistence on a single chip, is one of the compelling reason for choosing FPGA/SoC platform. If a segment of the algorithm is computationally complex then a custom coprocessor units can be designed to eliminate such problems.

4.These kinds of systems are preferred over ASIC based SoC solution, due to its re-programmability, Intellectual Property (IP) reuse and cheaper development cost. Al-

though ASIC has advantages over FPGA based SoC in terms of customized chip size, power, delay etc. But FPGAs are feasible solutions for prototyping a device before building an ASIC.

5.FPGA enables selection of an optimal platform of SoC configuration for a typical application involving trade-offs between flexibility, cost, performance and power consumption.

6.The FPGA's enables the change of part of hardware in run-time execution of application without changing the complete hardware by Dynamic Partial reconfiguration (DPR) which allows lower-power, dynamical hardware – change etc.

Hardware Software Co-design

In general, signal processing applications are developed using DSP processor which has dedicated hardware blocks for certain computations like MAC, multipliers, dividers etc. The disadvantages of using DSP processor is that it executes the instructions in a sequential manner thus limiting the speed of the design. So, in order to speed-up the processing of an application, parallelization is needed. This can be achieved using various ways like multi-threading of application, Graphic Processing Unit (GPU), hardware design using FPGA etc. The main difference between execution of hardware and software tasks is concurrency, which allows the hardware to execute a task much faster than the software in a processor. The designed hardware can be further accelerated by making use of the parallel and pipelined architecture techniques. This is not possible in a General Purpose Processors (GPP) and DSP processors which performs computational tasks in software by executing the application sequentially. The difference between these platforms is tabulated in Table.1. HW/SW co-design is a popular approach being used to accelerate computational intensive DSP applications.

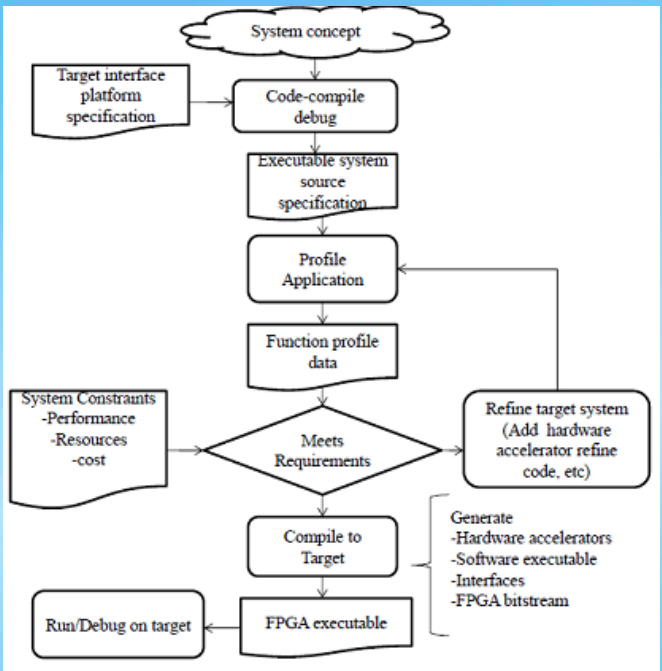

Figure 1: Hardware software co-design [7]

Table 1: Comparison of Processor/DSP, FPGA, ASIC based solutions [6]

Characteristic	Processor/ DSP	FPGA	ASIC
Programmability	High	High	Low
Development Cycle	HW+SW	HW+SW	HW
Area Efficiency	Medium	Low	High
Power Efficiency	Medium	Low	High
Efficiency	Low	Medium	High

In co-design approach, most time consuming tasks/subtasks of the application/algorithms are implemented in the hardware while the less computational intense tasks/subtasks are implemented in the embedded processor. In co-design, partitioning of the algorithm into software (SW) and hardware (HW) is a critical task. The partitioning of HW and SW is decided by the profiling results of the application as shown in Figure.1. During the profiling, the computationally intense tasks/subtasks are identified. Subsequently these are designed and implemented either in the hardwired logic or by using a dedicated co-processor unit [8-9]. FPGA based embedded system design is still relatively new compared to standard processors. So the software design tools are relatively undeveloped and difficult to debug the entire system [1-3]. There are prominent issues like IP interface, cross clock handling and memory management that imposes design bottlenecks in SoC design. So in order to resolve these things new design methodologies and easier integration methods are needed.

Programmable System on Chip design:

FPGAs no longer act as only glue logic resource in a complex hardware system, because modern FPGAs have processor units along with glue logic as processing elements. This way traditional system on board design has been replaced by SoC design. If the designer needs to develop SoC in a reconfigurable approach then FPGA device is used and the platform is named as Programmable System on Chip (PSoC). It is an integrated system with processor, peripherals, Memory, custom IP components on a single Integrated Circuit (IC) like FPGA. With a provision of including operating system, such as Linux, these systems begin to appear more like a desktop Personal Computer (PC) in terms of functionality and capability on a single IC chip. The SoC development cycle has two different platforms a) Software development b) Hardware development.

Software development platform

MATLAB and C programming language are used for algorithm development and validation. Eclipse platform GCC complier specific to Processor tool chain (ARM/ PPC/Microblaze) is used to software. Xilinx Software Development Kit (SDK) is used for profiling the algorithm in PPC440/MB processors.

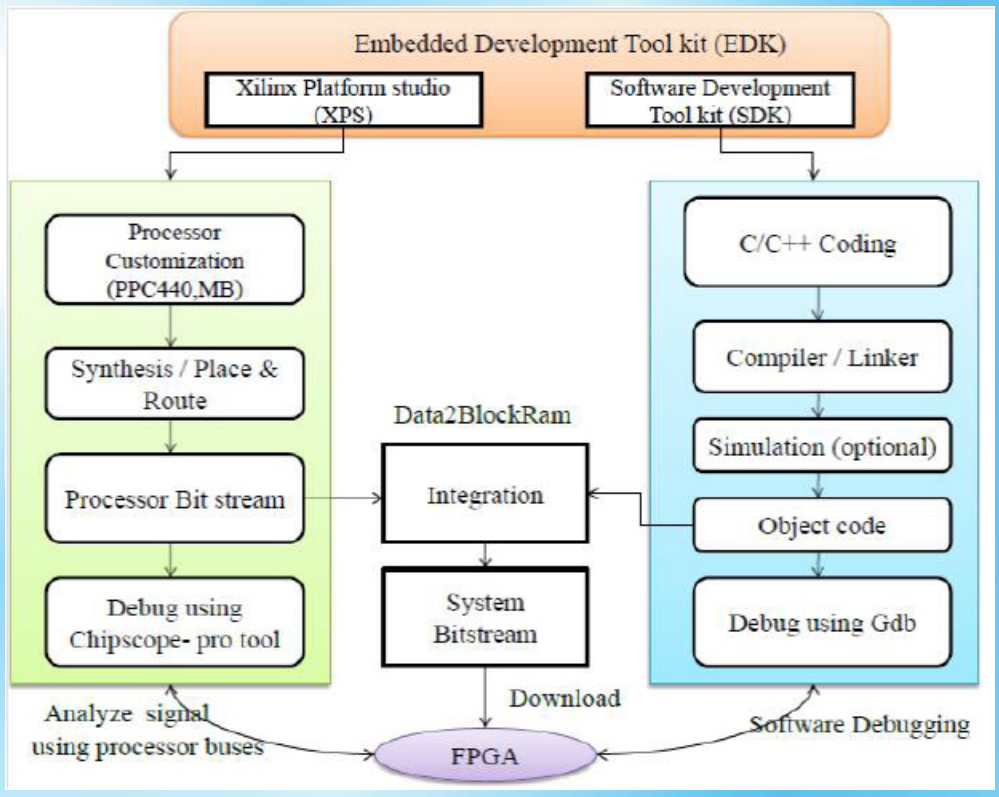

Figure 2: Hardware software co-design approach using embedded development kit (EDK) [12]

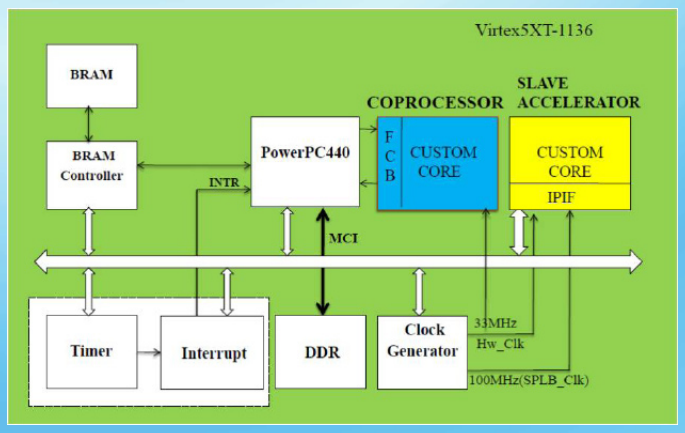

Figure 3: Basic SoC system

Hardware development platform:

The hardware development cycle in FPGA design flow uses various tools for HDL coding, simulation, synthesis, debugging tools as shown in Figure 4. Mentor graphics Modelsim, Xilinx ISim simulation tools are used for coding and simulation. For hardware development Xilinx Integrated Development Environment (IDE) with Integrated Software Environment (ISE) is used.

Embedded Development Kit (EDK) is used for building the SoC and generating custom peripherals. The PSoC platform can be designed using Xilinx platform studio (XPS) in Xilinx EDK and SDK as shown in Figure.2. In has two different design steps namely hardware (HW) and software (SW). These two steps run in parallel. In HW, Base System Builder (BSB) wizard provides an efficient way to create the FPGA based embedded system. The choice of the memory types, memory controllers, peripherals, peripheral controllers, size and type of instruction and data cache memories and size of local memory, choice of processor, bus and peripheral clock frequency are configured in BSB [11-12]. The proposed FPGA based SoC platform inside in FPGA incorporating a co-processing unit or accelerator can be interfaced to the processor through shared bus (Processor Local Bus (PLB)/AXI bus) and dedicated bus (Auxiliary Processing Unit (APU)) as shown in Figure.3.

On-chip memory and external memory are used for initializing the processor program. The Universal Synchronous and Asynchronous Receive and Transmit (UART) and Joint Test Action Group (JTAG) ports are used to monitor, debug and download the bit-stream on to the FPGA. The proposed custom/DSP IP in SoC has design objectives of high-speed in terms of operating frequency and reduced cost in terms of FPGA fabric resources. During simulation, synthesis and compilation of the embedded processor system, an appropriate optimization scheme must be selected to achieve the above design objectives. The processor internal timer along with interrupt is used for measuring the execution time of the algorithm.

Hardware accelerator

A hardware that accelerates the execution of a task as a separate unit other than processor is referred as hardware accelerator. The accelerators can be designed using ASIC or FPGA approach depending on the specification of applications. ASIC-based accelerators cannot be usually leveraged by a gained speedup due to larger design development costs and longer development cycle. Moreover, an accelerator designed for a specific application

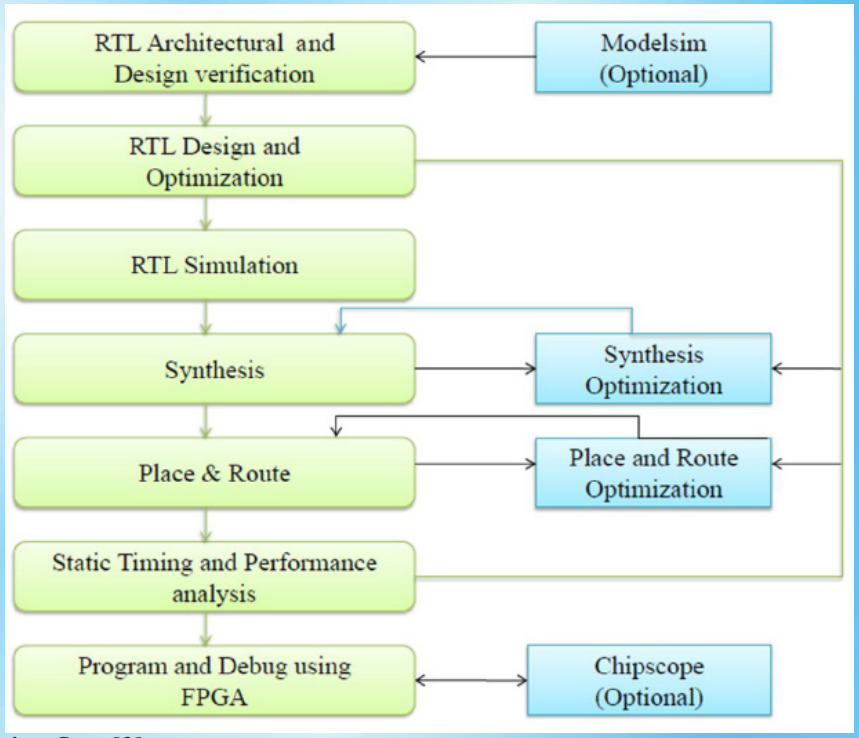

Figure 4: FPGA design flow [2]

cannot be utilized by another application. With the advent of SoC platform using FPGA, the situation has changed. The hardware rigidity is lowered by re-programmability of FPGA devices in SoC platform and it allows interfacing of designed hardware IPs with the processors for desired DSP applications. The ability of on-demand FPGA reconfiguration also enables the accelerator to adapt to the actual needs of an application executed in the processor . Several issues need to be taken into consideration while designing a SoC system. The important issues are processor to accelerator interface, mutual communication and synchronization. All these effects have crucial impact on the acceleration. The former issue comprises of communication protocol and amount of data transfer. Selection of a proper communication/interface scheme affects the quantity of the data transfer from processor to coprocessor/slave accelerator. The improper bus interface may result in slowdown of the accelerated system. Another issue is to synchronize the data transfer between processor and hardware unit by handshaking, direct or interrupt mechanism. The selection of the particular synchronization system depends on the chosen communication granularity and scheduling of the algorithm. The coprocessors can be interfaced with the processors using three different techniques i) System bus connected, ii) I/O connected, and iii) Instruction-pipeline connected.

System bus connected:

In this approach accelerators are interfaced to the processor as a slave peripheral/ slave unit (SU) using system bus i.e. Processor Local Bus (PLB) as shown in Fig. 2.4. The accelerator can transfer data and send commands to the processor through the system bus. Typically a single/multi data transactions, consumes many processor cycles due to bus arbitration. This kind of systems have two major bottlenecks i.e. insufficient peripheral bus throughput and bus arbitration, this leads to data transfer and synchronization overheads which in turn lowers the execution speed.

I/O connected:

In this approach, the accelerators are interfaced directly to a dedicated I/O port of a processor. In order to reduce the bus overhead and arbitration, a dedicated First-In-First-Out (FIFO) type of interface like Fast Simple Link (FSL) in Microblaze (MB) processor is used [11]. These interfaces are typically clocked faster than the processor bus. Often data and control are typically provided through GET or PUT instructions. This enables the bus interface to have lower latency and higher data rate compared to the system bus.

Instruction Pipeline connected:

In this approach, the accelerators with desired computing core are interfaced directly to the processor. Being coupled to the instruction pipeline, instructions not recognized by the CPU can be executed by the co-processor. This type of accelerators exposes no communication overhead and offers quick synchronization between the processor and coprocessor. The bottleneck is the implementation of the acceleration unit itself. If the critical path of the whole system goes through the acceleration unit, then the whole processor will decrease its operational speed [4-5]. Recent FPGAs include processors like ARM, PowerPC family processors, which utilize both specialized functional units and instruction set extensions for interfacing an IP. The APU interface is capable of transferring higher data volumes per second, approaching to the speed of Direct Memory Access (DMA).

Examples of SoC Applications for Space and Astronomy :

Recent internal studies at NASA's Jet Propulsion Laboratory (JPL) expect an estimate of approximately 1–5 Terabytes of raw data per day (uncompressed). Therefore, higher computational platforms are required. An FPGA/SoC approach to On-Board Data Processing enables in new Mars Science with Smart Payloads. The aim of this proposed Mars Scout Mission known as MARVEL (MARs Volcanic Emission and Life Scout) is to search for volcanic activity on Mars. In this mission one of its primary instruments, MATMOS, an FTIR spectrometer, will produce large volumes of data in short; 3-minute bursts during its on-orbit observation of sunrise and sunset. In this work data processing relies heavily on floating-point FFTs. Virtex-4 FPGA platform uses auxiliary processor unit (APU) that provides a flexible high bandwidth interface for fabric co-processor modules (FCM) to the PowerPC405 processor core. The coupling of FPU FCM with the APU provides sufficient computation power to meet MATMOS's data processing requirements when implemented in a multi-processor, dual-FPGA system. Bekker.et.al, developed MATMOS Fourier Transform Infrared (FTIR) spectrometer instrument designed to measure the Mars atmospheric composition using solar occultation from orbit. MATMOS requires high sampling rate, high dynamic range data acquisition to record time-domain interferograms which get converted to spectra on-board the space-craft in on chip FPGA. The complete design is ported into the hardcore PowerPC processor of Virtex 5-FPGA. By enabling the Floating Point Unit (FPU) of the APU an acceleration of 4x is reported. Several other works for on board signal processing for space applications have also reported the advantages of FPGA based SoC system . Fons.et.al., developed a FPGA based runtime reconfigurable coprocessor for computational applications using dynamic partial reconfiguration approach and is validated by executing several signal processing algorithms .This kind of system has good applicability for evolvable hardware based space applications.

Fiethe et. al. developed reconfigurable System-on-Chip Data Processing Units for Space Imaging Instruments. The work uses new radiation tolerant high density FPGA and processor technology enables the instrument to new system architecture and it is used in space on Venus express. An advanced SoC design integrates special functions (e.g. data compression or formatting/coding) together with the processor system completely on a single or few high density FPGAs. CASPER Astronomy computing group is using FPGA based SoC platforms for high performance computation.

Further Reading:

1. Ron Sass and Andrew G. Schmidt. Embedded Systems Design with Platform FPGAs Principles and Practices. Elsevier Inc, USA, 2010.
2. Doug Amos and Austin Lesea and Ren'e Richter. FPGA-Based Prototyping Methodology Manual Best Practices in Design-for-Prototyping. Synopsys, Inc, Mountain View, CA, USA, 2010.
3. Abdel fatah et al., FPGA-Based Real-Time Embedded System for RISS/GPS Integrated Navigation. Sensors, 12(1):115-147, 2011.
4. Ansari, Ahmad and Ryser, Peter and Isaacs, Dan. Accelerated System Performance with APU-enhanced processing. Xcell Journal, first quarter, 2005
5. Don Davis, Srinivas Beeravo, Ranjesh Jaganathan. Hardware / Software Co-design for Platform FPGAs. Xilinx Application Notes, 2005
6. Peesapati et al., FPGA-based embedded platform for fiber optic gyroscope signal denoising. International Journal of Circuit Theory and Applications, doi:10.1002/cta.1883, 2013.
7. EDK Concepts, Tools, and Technique: A Hands-On Guide to Effective Embedded System Design. Xilinx User guide, (683), 2011
8. Pingree, et al., An FPGA/SoC Approach to On-Board Data Processing Enabling New Mars Science with Smart Payloads. In Proceedings of the IEEE Conference on Aerospace, pages 1-12, mar 2007.
9. Bekker, et al., An FPGA-based data acquisition and processing system for the MATMOS FTIR instrument. In Proceedings of the IEEE Aerospace conference, pages 1-11. IEEE, 2009.
10. Fons et al, Deployment of Run-Time Reconfigurable Hardware Coprocessors IntoCompute Intensive Embedded Applications. Journal of Signal Processing Systems, 66(2):191-221, 2012.
11. Jason et al. "Evolvable Hardware for Space Applications." The 1st IntelligentSystems Technical Conference, Chicago, USA, September. 2004.
12. Fiethe, et al. "Reconfigurable system-on-chip data processing units for space imaging instruments." Proceedings of the conference on Design, automation and test in Europe. EDA Consortium, 2007.
13. CASPER https://casper.berkeley.edu/ [Accessed :23-07-2015].

P.Rangababu
Electronics & Communication Engg.
National Institute of Technology
Meghalaya
Shillong-793003
E-mail: p.rangababu@gmail.com
Contact: 09485177063