



The Advantages of FPGA's on Single Boards Computers: Increased I/O Flexibility

Implementing FPGA's (Field Programmable Gate Arrays) on embedded SBCs (single board computers) provides OEM system integrators a completely new set of tools for customizing their embedded applications into small, harsh, rugged environments. As CPU designs and FPGA technology have evolved it has become increasingly more appropriate to combine these two technologies onto a single board computer. The combination of these technologies enables SBC vendors to provide users with infinitely more flexible SBCs capable of adaptation to a broad range of unique circumstances and environments. This combination of technologies is poised to dramatically change the way embedded system designers implement COTS applications in the future. This is the dawn of a new era in OEM embedded SBCs.

Limitless Possibilities

When FPGAs are implemented on SBCs, users are able to implement the RIGHT mix of I/O for their target application. Users are empowered to re-define the SBC to meet their needs, often eliminating the need for daughter cards or companion I/O boards. The power and flexibility of an on-board FPGA lies in the fact it can be defined and then re-defined to meet the changing needs of the system or user. The days of being short a memory socket or COM port in the final release are gone. Such flexibility is accomplished by programming into the FPGA off-the-shelf IP cores or by writing custom FPGA firmware for the application. This FPGA code adds value of its own for the user, becoming the basis of IP (Intellectual Property) which enables OEMs to protect their unique and innovative measurement and control processes at a new level.

IP Cores are purchased similar to software and configured by the user for installation into the FPGA. Off-the-shelf IP cores or "stacks" are available from a wide variety of sources: the FPGA manufacturer themselves or third party vendors who have deep specialized know-how, often in narrow, niche technical fields. Some IP Cores are even available from technical communities such as PROFINET and MODBUS. And, as always, there are "free" or almost free open source downloadable cores online – but "buyer beware" – this may cost more in manpower than is saved by not buying a validated IP Core which often comes with technical support.

For OEMs wanting to write a custom IP core for their own dedicated interfaces, proprietary algorithms or specialized processing there is a large base of easy-to-use FPGA tools available. Typically designers implementing custom IP cores will write their firmware in Verilog or VHDL and using the tools provided by the FPGA manufacturer for their specific FPGA. However, a user may choose to use a third-party development tool chain which generates HDL code for a wide selection of FPGAs.

The functionality which can be incorporated into an FPGA is limitless. Users needing to have on-board DPS or to add a video controller for dual video displays can do so. Complex signal processing, motion control and machine vision can also be implemented all using the SAME off-the-shelf SBC. Hence, the well-designed combination of a Micro/sys SBC with an on-board FPGA gives the user the ability to design innovative embedded systems while significantly reducing time-to-market and R&D all the while leveraging existing IP cores to accomplish his/her unique, complex tasks.

As counterfeiting and IP theft is on the increase, concerns over what means are available to provide security for the IP inside FPGAs is also on the rise. The reprogrammable architecture of FPGAs acts as an inherent barrier to a straightforward tampering attempt to reverse engineer designs. There are, however, other means to violate the security of FPGAs which involve cloning the bitstream during configuration and/or JTAG and SEUs (single event upsets) manipulation. FPGA manufactures, aware of these highly sophisticated technical vulnerabilities generally provide additional security measures for users who are sensitive to threats at this level. Encrypted solutions can protect the “know-how” inside the FPGA adding a higher level of safety than the off-the-shelf software security application program might. This reduces the likelihood IP can be examined, explored and adopted illegally by others.

Reducing BOMs – Costs and Complexity

Whether it is components on a circuit board or daughter boards in a system, using an FPGA enables users to add functionality while reducing the BOM count and often times lowering the manufacturing costs in terms of complexity as well as BOM costs. System I/O such as serial interfaces, Ethernet controllers, user-defined I/O, legacy protocols, or feature enhancements through the use of selected IP cores from third party vendors can enhance or extend the life an embedded system while reducing component count on BOMs. The possibilities are endless for reducing hardware revisions, daughter cards and “stuffing options” on the SMT line. By including provisions for these versions of an OEM system in the FPGA, build time options are simplified and costs are reduced.

Easing the Impact of “End of Life”

For users who have experienced the pain of a panic need to re-design a system because a video controller has gone obsolete or a required protocol is no longer supported by silicon, using SBCs with on-board FPGAs is a means to extend and stretch system lifetimes. The FPGA eases end of life issues in two ways. First, FPGAs reduce the number of times an “at risk” function needs to be implemented with a low-volume, high-risk chip. For example, a specialized video controller or DSP chip can be alternatively implemented with an IP core in a generic, high volume FPGA that is standard to many applications. Most FPGA’s roadmaps extend further than specialized I/O chips which face obsolescence faster due to manufacturing costs or technological advances. Secondly, an FPGA can offer a failsafe for unexpected end-of-life bumps. A FPGA can act as a very effective and efficient “Band-Aid” for a chip which is somewhat generic but has gone obsolete suddenly while remaining critical to a system’s performance. For example, if for software reasons or other system constraints, a product needs to maintain Ethernet performance at 10 MBs rather than 100 MBs, an IP core can be an excellent solution, but only if an FPGA is on-board to act as a life saver.

FPGA's can act as a tool into the future as well as into the past for easing the on-going challenge of end-of-life challenges in the electronic industry.

Complexity and Challenges Amidst Great Rewards

FPGA technology is a fresh new technology to be matched with SBCs which will lead to highly innovative applications. And while the opportunities are exciting, it must be tempered with a respect for the challenges which still exist. First, today's FPGAs are not a solution for systems requiring analog hardware solutions. Also, implementations to expand communication devices require the SBC vendor to plan ahead for the users and provide access to transceivers. Absent this, users will be responsible for adaptor boards or "garbage boards" to implement the transceivers negating the advantages of reducing BOM count and manufacturing complexity. SBC users need to be savvy when selecting an SBC vendor to determine there is a "system" mentality to their approach in implementing FPGA's on SBCs. This includes inquiring as to what types of off-board signals are supplied and what interfaces are required to complete the transaction.

Additionally, spending some time with your SBC vendor to learn what platforms they provide for developing FPGA code is important. This research and system planning should include discovering how and under what circumstances you will write or configure your IP core, how you will download it and what interfaces are given access to by the SBC vendor. You will need to know about the interface between the FPGA and CPU to validate if your application will have the speed and signals necessary to service your IP core or application. Paying attention to the details at the system level will pay off as you begin to address implementation questions.

Attached are four models for FPGA implementations for a Micro/sys product employing a Freescale i.MX515 ARM Cortex-A8 processor and a Xilinx Spartan 6 FPGA. Aligning the system needs with the appropriate implementation model will allow designers to determine the steps involved in the design process including the critical timing issues, the required communication links, and the operating system interfaces and drivers required.

Conclusion

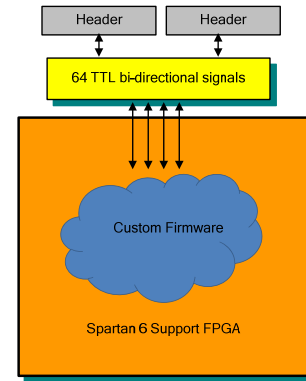
There are great rewards for the system designer who is also savvy to the possibilities FPGAs offer when implementing an embedded system. There are potential OEM system's cost savings because of reduced BOM count and simplified production control issues. FPGAs make the "feature-itis" rat race easier to manage by enabling system upgrades over a long product life cycle without the need of a complete redesign. FPGAs can be implemented in ways to maximize the protection of one's IP by embedding it in the FPGA code as well as providing potential options if there are EOL issues in the future. These FPGA rewards will make the savvy system designer a hero over the lifetime of his product's system.

One FPGA Development Kit – Four Models for Developing Firmware

Stand Alone FPGA–Simple, Easily Portable to Other Hardware

- No run-time communication with the SBC's i.MX515
- 95% of programmable space available for user
- 64 bi-directional signals mapped from FPGA to headers
- Examples of I/O implementations provided
- Ideal way to write, develop and test firmware being developed to port to user's own hardware

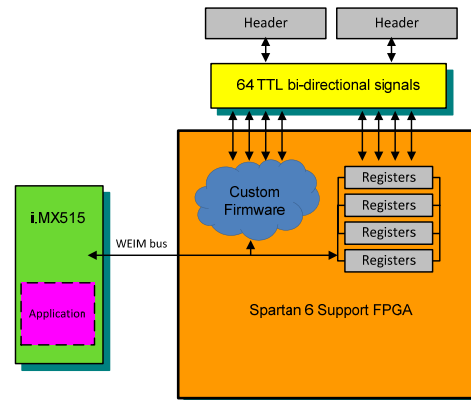
*Requires Xilinx ISE Design Software



FPGA to i.MX515 Communicate via Memory Interface Bus

- Communication between FPGA and the SBC's i.MX515 processor through on-board memory interface (WEIM)
- Uses ISE Design Software tools from Xilinx for programming
- Implement IP cores from Xilinx or 3rd party, or write custom IP cores for adding DSP, video, SATA, motor control, COMM protocols such as Ethernet, etc.
- 64 bi-directional signals mapped from FPGA to headers

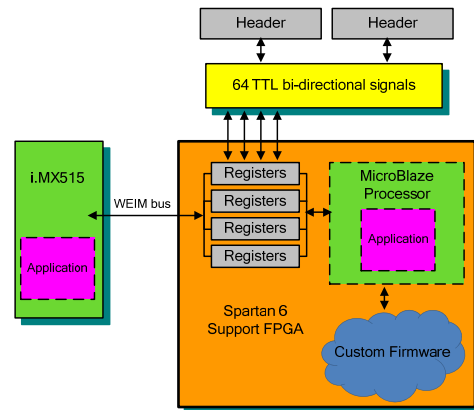
*Requires Xilinx ISE Design Software



Xilinx's MicroBlaze Microcontroller Operates in FPGA Independent of SBC's i.MX515 for Distributed Control

- Intelligent microcontroller installs inside FPGA to off-load SBC's i.MX515 of control functions
- WEIM memory interface for SBC i.MX515 updates the FPGA
- 64 bi-directional signals mapped from FPGA to headers

*Requires Xilinx EDK Design Software



Memory Interface IRQ's Enable MicroBlaze Controller and IP Cores to Run Simultaneously

- IRQ interrupts available to enhance operation of **BOTH** an IP core and the MicroBlaze controller
- Full communication to SBC's i.MX515 via WEIM interface
- 64 bi-directional signals mapped from FPGA to headers
- Reduces need for expansion boards required to implement a complex system

*Requires Xilinx EDK Design Software

