WP169 (v1.0) October 25, 2002

Could Automotive Processor Obsolescence be History?

By: Karen Parnell

Obsolescence is a concern of most design engineers and none more so than with automotive telematics equipment designers. Even though automotive electronics equipment design and development time scales have shrunk recently from 5 to 2 years, the products themselves will still need to be produced for many years and be active in the field for even longer.

The biggest obsolescence headache is that of out of date microprocessors and microcontrollers. Processors have shorter life spans than ever and are discontinued at short notice driven by the consumer market trends and the ever present need for speed enhancements. Consumer products such as games consoles and mobile phones have built in obsolescence to stimulate sales of the latest and greatest products. This built in obsolescence leads microprocessor manufacturers chasing the planned new platform introductions and high-volume sales almost guaranteed and thus propagating the obsolescence ripple effect.

© 2002 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and disclaimers are as listed at http://www.xilinx.com/legal.htm.
All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

Even if the design has been coded in "C" (which is always touted as being "portable code") there are always architecture specific instructions and features which hamper the change over between and obsolete processor to the next generation device. The change over process is further exasperated by difference package options and I/O configurations necessitating the need for a complete board re-spin. If we imagine the scenario where every Electronic Control Unit (ECU) in a car contains at least one processor and that every car contains up to 60 ECUs this leads to a major headache every time a processor is obsoleted at relatively short notice.

There are several solutions to the problem of processor obsolescence. The applicability of any given solution depends upon a number of variables, including the value of the application software, the projected life of the system and the amount of time and money available to solve the problem. The most radical and most expensive solution is to redesign the system around a new processor. Depending upon the volume of the code, a redesign can cost hundreds of man-years of time, much of it devoted to validation and testing. Not only is the huge investment in debugging and refining the existing software lost, but the solution is temporary at best. If the system has a long projected life, the same problem will recur every few years, as each new design in turn becomes obsolete.

Another solution is the last time buy (LTB), which, on the surface, appears to be the most cost-effective option. The problem is that the automotive designer must guess at how much product to buy for the life of his program. If he guesses wrong, he is faced with an even more difficult problem; a larger legacy investment that must somehow be upgraded.

Inserting a new processor along with software written to emulate the old one ispresently more good in theory than a reality. The concept is appealing and in factdoes have some operational history. The legacy software is preserved, and the process is therefore relatively cheap and fast. Once again, the solution is not permanent and, if the system has a long projected life, might have to be repeated every few years. More important, software emulation is inherently a serial process and therefore relatively slow. That means that the new processor must consume much of its performance running the emulation rather than the application. It has been shown empirically that emulation requires, on the average, about 20 clock cycles of the new processor for every legacy instruction it executes. In addition, emulation breeds further obsolescence since the processor used as the emulation engine itself will become obsolete and may force an entire rewrite of the emulator.

Soft Processor Solution

A radical but robust new solution is emerging to eradicate processor obsolescence and preserve many years of legacy code and development. The new way is to **own** the soft processor core and embed it in FPGA fabric. Not only can you port the core to multiple FPGA platforms but you can "design" the peripheral set to meet the exact design requirements thus eradicating architecture compromises and wasted peripherals.

For example, the designer may desire a processor with perhaps 10 UARTs, an interrupt controller, and access to a block of external FLASH. While many off-the-shelf processors exist that would offer multiple UARTs and the other desired peripherals, they would typically be of sufficient complexity to have numerous other peripherals that would be unused in this system. Not only is the designer paying for the additional peripherals, it is often necessary that unused peripherals in this type of processor have to be placed into a safe mode or otherwise disabled via software. This places an additional burden on the software design team, who not only have to make the used processor peripherals operate correctly, but now have to write code for the parts of the processor which are not used. It is clear that purchasing an off the shelf solution for this scenario would be highly wasteful not only in terms of initial cost, but also in wasted engineering time during the design process.

With the Xilinx MicroBlaze™ soft processor, the designer has the luxury of a different approach. They can now start with a processor core and build the peripheral set to meet their exact requirements. Silicon wastage is reduced to zero since the designer will only ever implement what they need. Software design complexity is reduced because no code need ever be written to disable unwanted processor functionality. The creation of unusual processor configurations, which can be changed at any time to suit changes in the specification, is reduced to a simple task.

Even if after five or six years of field use when the FPGA hardware may itself be nearing the end of its life then the soft processor core can simply be dropped into its new FPGA *host* utilising the same C code. The hardware platform may need some PCB modifications but the legacy code remains usable and intact.

Xilinx MicroBlaze and PicoBlaze Soft Processors

Xilinx offers both a 32-bit soft processor core called MicroBlaze and an 8-bit solution called PicoBlaze™. The PicoBlaze processor runs at speeds of 116 MHz, yet occupies a tiny footprint of just 35 Configurable Logic Blocks (CLBs). (See figure 1).

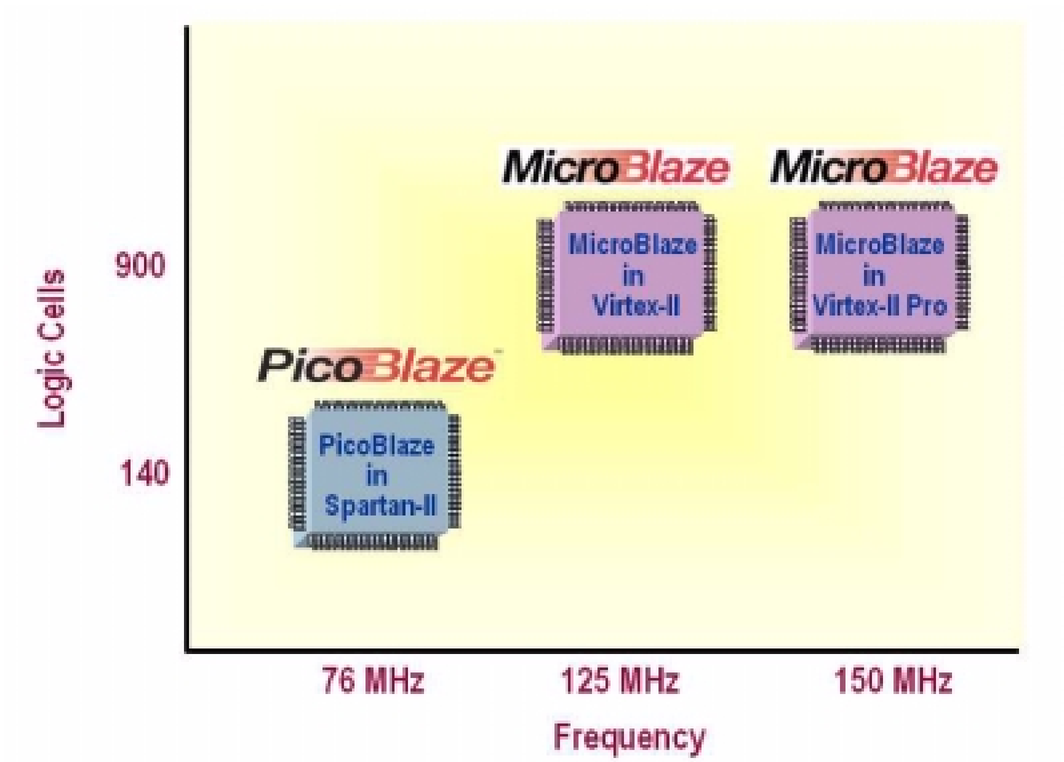

Figure 1: **Xilinx PicoBlaze and MicroBlaze Soft Processors Showing Size in Logic Cells and Speed in MHz**

The MicroBlaze 32-bit soft processor core is the industry's fastest soft processing solution and runs at 150 MHz and delivers 100 D-MIPS. It features a RISC architecture with Harvard-style separate 32-bit instruction and data busses running at full speed to execute programs and access data from both on-chip and external memory. A standard set of peripherals are also CoreConnect™ enabled to offer MicroBlaze designers compatibility and reuse.

The MicroBlaze Development Kit, including the soft processor core and a standard set of peripherals, will be available from Xilinx and its distribution partners. The kit will include a complete set of GNU-based software tools including the compiler, assembler, debugger, and linker. MicroBlaze Kits that are bought from Xilinx Distribution Partners will also include development boards that support the Virtex™-E, Virtex-II, Virtex-II Pro™, Spartan™-II, and Spartan-IIE series of FPGAs. Table 1 summarises the two processor cores. Selected Xilinx FPGAs in the new IQ Solutions range have been qualified to operate over the –40ºC to +125ºC temperature range and are targeted for use in automotive applications such as telematics systems.

Table 1: **Xilinx Soft Processors**

Soft Processor	Architecture	Bus	MIPS/ Speed	Size	FPGA Support	Support
MicroBlaze	32-bit RISC	Harvard style buses 32-bit instruction and data buses	100 D-MIPs 150 MHz	225 CLBs	Virtex Virtex-E Virtex-II Virtex-II Pro Spartan-II Spartan-IIE	MicroBlaze Developments Kit (MDK) – soft processor core, peripherals, GNU-based software tools (Compiler, assembler, debugger, and linker)
PicoBlaze	8-bit	8-bit address and data busses	35 MIPS 116 MHz	35 CLBs	Virtex Spartan-II	Free of charge reference design and application note, assembler

Conclusion

Xilinx soft processor cores such as MicroBlaze and PicoBlaze when embedded in FPGA fabric can eradicate processor obsolescence issues by providing a stable platform owned and configured by the automotive designer. In combination with the new IQ Solutions range of FPGAs, qualified to extended temperature, they are ideal for automotive applications. Not only can you benefit from the flexibility, integration and upgradeablilty offered by programmable logic but you can take advantage of a processor tailored to your design needs that will not go obsolete.

Websites

Automotive IQ Solutions www.xilinx.com/automotive
MicroBlaze Information www.xilinx.com/microblaze
PicoBlaze Information www.xilinx.com/picoblaze

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/25/02	1.0	Initial Xilinx release.