Engineers increasingly vote for prototyping for embedded system debug

For the second year in a row, Byte Paradigm has conducted an online survey to understand the role of using a 'live' prototype for testing and debugging embedded systems. This year, we had a special attention to the equipment that the electronic engineer finds in its lab or on its desk to efficiently test and debug embedded systems. Results show that using a prototype is widely seen as a way to speed up debugging and that digital pattern generators are gaining success for the generation of digital input stimuli.

Why prototyping?

Mastering increasing design complexities within limited time/money budget constraints is the usual key driver for getting interested in prototyping as a testing and debugging technique.

Looking back the years, it seems that engineers and prototyping are an old couple with a long history of love and hate. Before RTL simulation became one of the most successful methodologies for design, lots of systems were developed and debugged by connecting devices on a breadboard and running extensive 'try-and-observe' test programs. EDA tools have pushed the prototyping to the late stages of the development, with the purpose to check a design already pre-validated with these tools – in theory, at least.

Prototyping has seen a new wave of interest in the design flow when it appeared that the complexity of embedded system progressively prevented from getting first-time right production. 'Seeing and feeling' a prototype enhances the confidence in one design and secures the production of an embedded system, because this one has already been used ***'for real'***.

	RTL/gate-level simulation	Test/debug executed on a prototype
Max design size	Limited by acceptable simulation time	Limited by target technology used to build the prototype
System coverage	High for purely digital system parts. Highly dependent on how the system's environment is modeled.	Complete if the prototype's environment is realistic.
Speed	Very low	Near or at system speed
Setup length	Very short. Highly flexible	Very long.
Hardware debug	Very good. High observability.	Poor observability.
Software debug	Limited to very short execution sequences.	Unlimited (at system speed!), realistic and very good observability

Figure 1: Simulation and prototyping compared

FPGA as an enabler

Engineers like the speed of execution of testing and debugging on real hardware prototypes while wishing a prototype could offer complete visibility over their design (see Figure 1 for a comparison of simulation and prototyping).

Moreover, there is no doubt that the availability of high-end FPGAs has been one of the key enablers of the come-back of prototyping in the design flow. As essential piece for prototyping, the FPGA offers all the desired flexibility, complexity and performance at a very competitive unit cost for any type of digital hardware. Coupled with embedded or separate microcontrollers, memories, I/O interfaces, and other types of peripherals, a FPGA placed on a prototyping board allows emulating virtually any embedded system before going to production. Standard prototyping boards abound on the market today – and when they can't fit the bill, making your own board does not really seem to be a daunting task.

Fitting to organizations, planning and design methodologies

Using a prototype early in the design cycle also helps shorten the overall design time while making the most of the available resources. Designing embedded system requires the complex interaction of multiple hardware and software teams of engineers and sometimes the use of elements purchased from third parties (like IP).

No hardware or software team can really afford waiting for all the 'missing pieces' to be ready to finalize a separate part of the design. Partial design must be simulated with models and emulated on prototype before being assembled into the final embedded system.

In a traditional design flow (see Figure 2), the design is debugged during the 'design phase', on workstations, with the usual EDA simulation and analysis software. Once the design is ready, it is manufactured, and the prototype is tested again. Bugs are corrected with new design phase and the process goes through a new iteration. Every engineer knows that this is a theoretical scheme that very few organizations can really afford today.

Another design flow, that uses testing and debugging on prototype **during design** is proposed at Figure 3. In this flow, validation with EDA tools and validations that use tests on a prototype 'feed' each other with debug/test results. This offers a more complete analysis and helps correcting bugs faster. The total number of iterations is reduced and – hopefully – a 'first-time-right' system comes out of the production.

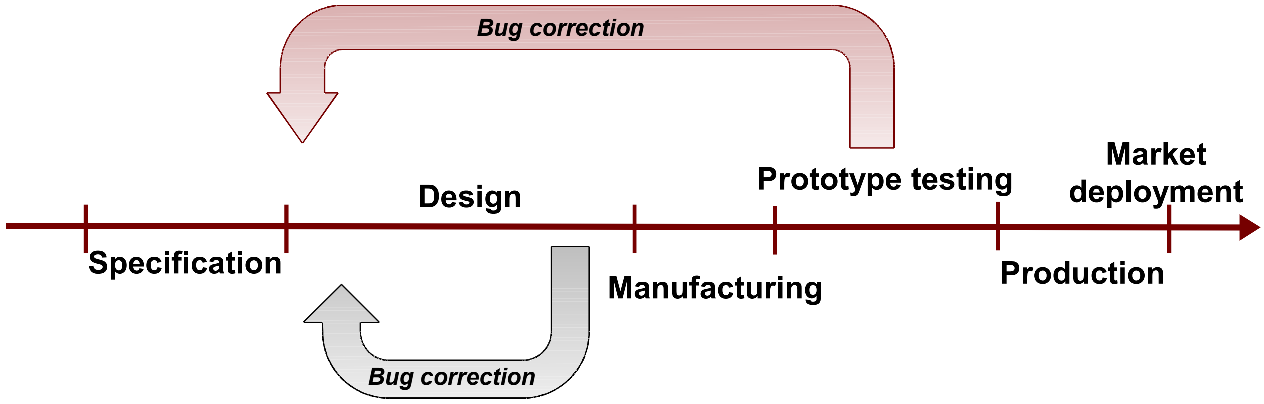

Figure 2: Sequential time line.

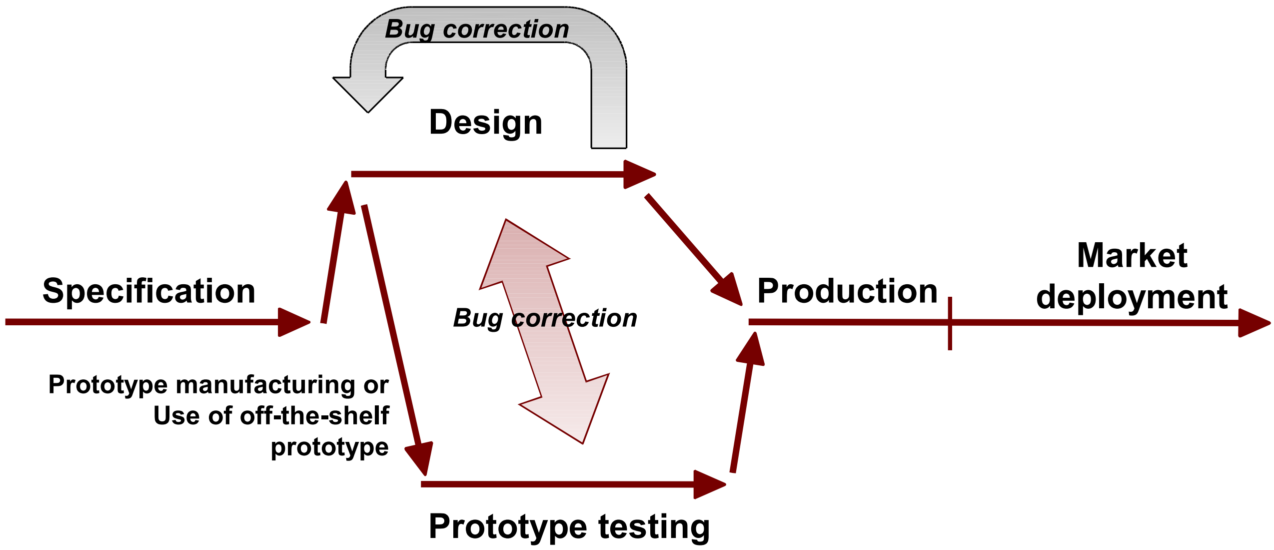

Figure 3: Time line with concurrent design and prototyping 'feeding' each other

Byte Paradigm – info@byteparadigm.com – www.byteparadigm.com
Chaussée de Namur, 119, bte 1 – B-1402 Nivelles - Belgium

Prototyping speeds up embedded system debug

In 2009 and 2010, Byte Paradigm ran an online survey among embedded system engineers. There were more than 250 respondents each year, all involved in embedded system design. In the 2 surveys, we asked the respondents to tell us their opinion on a number of proposition among which whether they thought if '**Using a hardware prototype speeds up embedded system debug'.** Results are remarkably stable from one year to another, even with a different list of respondents from one year to the other.

'Using a hardware prototype speeds up embedded system debug'

	Agree	Somewhat agree	Rather disagree	Disagree	Don't know	Response count
Year 2010	**62.6% (171)**	**31.9% (87)**	**2.9% (8)**	0.0% (0)	2.6% (7)	273
Year 2009	**61.2% (205)**	**29.3% (98)**	**2.7% (9)**	1.8% (6)	5.1% (17)	335

Table 1: Opinion on using a prototype for debug - year to year comparison

Survey results show that **90.5% of the respondents in 2009** and **94.5% of the respondents in 2010** either 'agree' or 'somewhat agree' with the statement that ***'Using a hardware prototype speeds up embedded system debug'.***

The 2010 survey especially focused on understanding whether some factors significantly changed the respondent's views on prototyping, especially:

- the performance of the embedded system being designed – the respondent was asked about the typical maximum digital frequency of his/her embedded system design.
- an experience with specific types of instruments – the respondent was notably asked whether he was familiar with using a **digital pattern generator.**

Here are the results.

'Using a hardware prototype speeds up embedded system debug' (2010)

Max digital frequency in designed embedded system	Agree	Somewhat agree	Rather disagree	Disagree	Don't know
1 MHz- 50 MHz	68.9% (42)	29.5% (18)	1.6% (1)	0.0% (0)	0.0% (0)
50 MHz – 100 MHz	58.6% (34)	39.7% (23)	1.7% (1)	0.0% (0)	0.0%(0)
100 MHz – 500 MHz	56.0% (42)	34.7% (26)	4.0% (3)	0.0% (0)	5.3% (4)
500 MHz – 1 GHz	**80.0% (24)**	6.7% (2)	10.0% (3)	0.0% (0)	3.3%(1)
More than 1 GHz	**52.9% (9)**	47.1% (8)	0.0% (0)	0.0% (0)	0.0% (0)

Table 2: Opinion on using a prototype for debug – grouped by system digital frequency range

Byte Paradigm – info@byteparadigm.com – www.byteparadigm.com
Chaussée de Namur, 119, bte 1 – B-1402 Nivelles - Belgium

'Using a hardware prototype speeds up embedded system debug' (2010)

	Agree	Somewhat agree	Rather disagree	Disagree	Don't know
Respondent uses digital pattern generator	**72.1% (62)**	25.6% (22)	1.2% (1)	0.0% (0)	1.2% (1)
Respondent does not use digital pattern generator	**58.3% (109)**	34.8% (65)	3.7% (7)	0.0% (0)	3.2% (6)

Table 3: Opinion on using a prototype for debug – grouped according to an existing previous experience with a digital pattern generator

Out of these results, it appears that engineers who use digital pattern generators more generally agree with the proposition that 'prototyping helps speed up debug' than the ones who don't use digital pattern generators.

It is hard to find a real trend according to the system's performance. There is a strong agreement in the 500 MHz to 1 GHz range but this might be specific to this specific sample of respondent. This requires further investigation.

What do we find in the lab / on the desk?

In 2009, we asked engineers who prototyped embedded systems whether they found the 'generation of inputs' more or less challenging than the 'observation of the system's outputs'. Answers showed that a large majority (83%) of them found the generation of inputs as challenging as or more challenging than the observation of the outputs (see Figure 4).

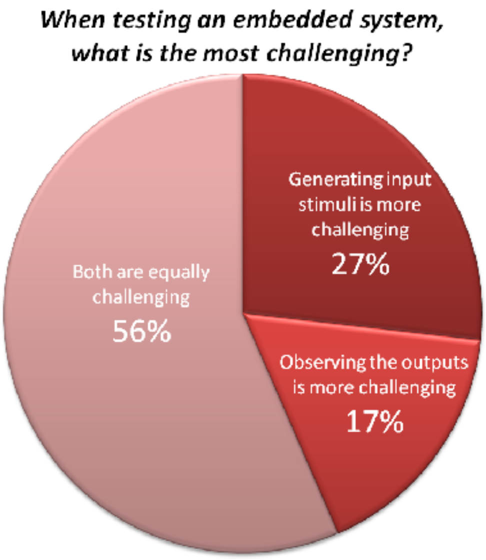

Figure 4: 2009 survey results

This year, we also tried to understand which type of tool they commonly found on their desk to test and debug their design, both for generating the inputs and for observing the design's outputs.

Oscilloscope, software and logic analyzer to observe

The chart at Figure 5 shows the number of occurrences grouped according to the maximum digital frequency of the embedded system.
There is no surprise to see the oscilloscope used by almost all our respondents for observing a system, with the couple JTAG / software emulation and then the logic analyzer following by a short distance. For embedded systems in the 100 MHz to 500 MHz range, it seems that traditional tools (oscilloscope) is often coupled with more specialized tools like embedded logic analyzers, which denotes the presence of FPGAs in the system. This does not come as a surprise either for this range of frequency.

Software, Digital Pattern Generator and... custom-made hardware to stimulate

When asked about how the embedded system is being 'stimulated' (see chart at Figure 6 below) – that is how the engineer generates the inputs of its system – we find 2 main categories:

- The most successful one: using a microcontroller, possibly with a GPIO and also a JTAG probe plus a software emulator. Software emulation environments today are feature-rich and offer a lot of flexibility, especially for software-centric (microcontroller-centric) embedded systems.
- Engineers designing embedded system in the 100 MHz- 500 MHz range also use **digital pattern generators** and **'custom set of boards and chips designed in-house'** more than in any other frequency range.

It is important to note that digital pattern generators and custom-made tools are used in combination with the other ones (for example: microcontrollers are used for stimulation in the 100 MHz-500 MHz range as well – probably for slower parts of the embedded system).

Byte Paradigm – info@byteparadigm.com – www.byteparadigm.com
Chaussée de Namur, 119, bte 1 – B-1402 Nivelles - Belgium

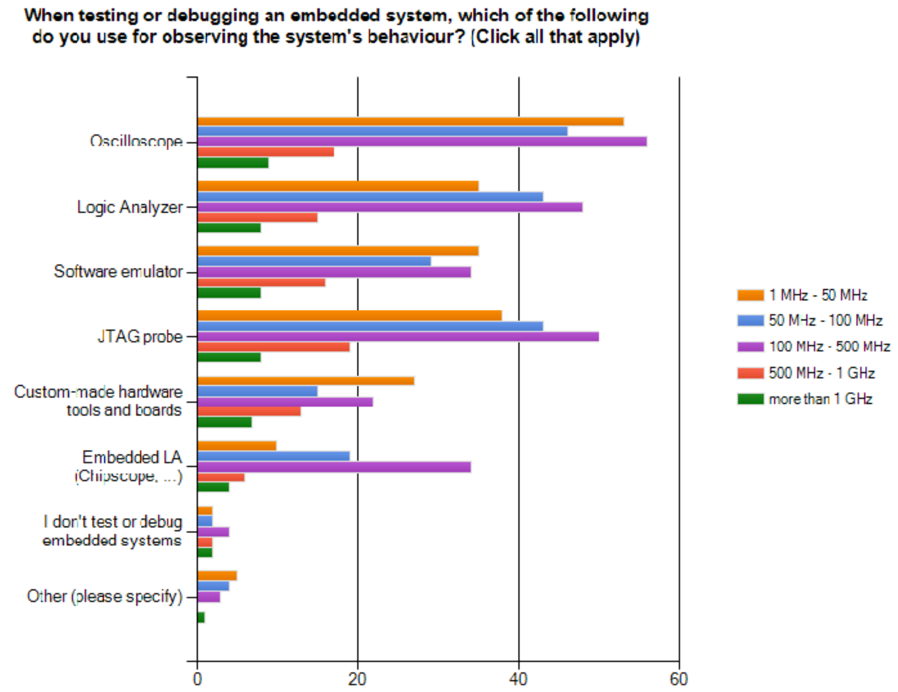

Figure 5: Types of equipment used to observe an embedded system's output, grouped by frequency range

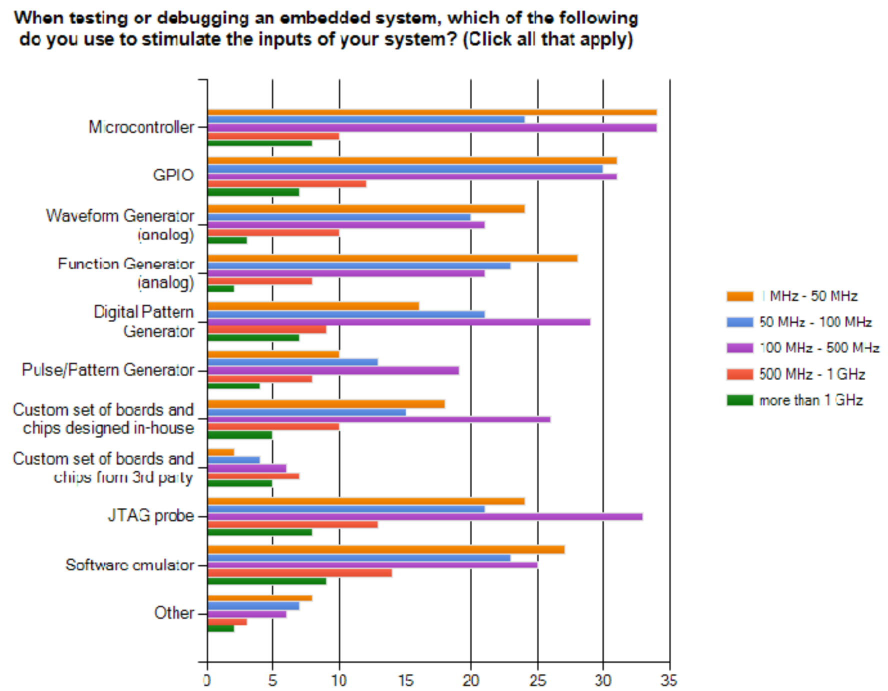

Figure 6: Types of equipment used to stimulate the inputs of an embedded system, grouped by frequency range

Debugging with a prototype is efficient if you use the right set of tools.

Results from Byte Paradigm's 2009 and 2010 surveys show that embedded systems engineers widely recognize prototyping as an efficient methodology to speed up embedded system debug and this, no matter the type of embedded system or its maximum speed.

In the constant quest for reaching shrinking time-to-markets, reducing the time spent on a prominent task such as debugging is undoubtedly of great value. Prototyping does help shorten the overall design cycle time and boost the engineer's productivity.

The embedded system engineer tends to complement its set of tools and instruments in order to overcome the challenges posed by more complex and faster technologies. In particular, the digital pattern generator – while still not being broadly adopted yet– seems to become an essential piece of equipment for the generation of digital stimulus above 100 MHz. Survey results show that engineers having an experience of digital pattern generators are more eager than other ones to use a hardware prototype to speed up debug and test on embedded systems.

Byte Paradigm

At Byte Paradigm, we are convinced that innovative, flexible and powerful digital pattern generators are one of the key elements to speed up embedded system debug on prototype. We also think that innovation in PC instrumentation can lead to boosting the designer's productivity and help design better and faster. We are passionately committed to go on that innovation path and very much interested in listening to your needs as embedded system engineer.

Contact us. Share your ideas. Tell us what you need.

About the author

Frédéric Leens is Sales and Marketing Manager at Byte Paradigm - www.byteparadigm.com

He can be reached at: frederic.leens@byteparadigm.com

Twitter: http://twitter.com/ByteParadigm

LinkedIn: http://be.linkedin.com/in/fredericleens