Cutting **complexity**

Collaboration cuts through algorithm complexity.

*By **Graham Pitcher**.*

Digital signal processing means different things to different people. An additional complication is that dsp is beginning to disappear as the technology is absorbed into various applications.

Dr Johannes Stahl is vp of marketing and business development for electronic system level (esl) design software specialist CoWare. He said there has always been confusion over what dsp means. "On the one hand, there are off the shelf parts, such as those from Texas Instruments. On the other hand, digital signal processing is a generic term and has nothing to do with a programmable processor.

"There's a range of things we look at in the design of dsp systems. They could be hardware or software, but what drives them all are applications requiring more bandwidth and higher performance."

In Dr Stahl's opinion, there was once a definite difference between dsps and dsp architectures. "But the fields are merging and designers don't always use classical dsp architectures any longer – even companies like TI see an end of life for that approach."

What this means, he believes, is a move to new architectures. "For example, very long instruction word (vliw) and parallel architectures will be used going forward to execute dsp algorithms."

And it's because of increasing algorithmic complexity that CoWare has announced a strategic relationship with Agility Design Solutions that is focused on accelerating the simulation and implementation of complex dsp algorithms – particularly those starting from Matlab models. Agility's mission is to reduce the time needed to develop, implement and verify signal processing algorithms.

Agilent's vp of marketing is Larry Melling. He said there was a market dynamic driving the relationship. "When you talk about dsp, you talk about a particular implementation style. But what are the right implementation styles for the algorithms being created today? We're focused on algorithmic development and understanding what kind of technology will be needed to implement those algorithms."

In Melling's view, a big algorithm once comprised 100lines of code. "Now," he continued, "we're dealing with people who have 10,000lines of Matlab code."

Agility is not only seeing algorithmic complexity growing, but also this complexity beginning to affect design cycles. "If designers can't get an implementation in a timely fashion," he continued, "the model is broken. People can't keep track of things."

Agility also recognises that algorithms are only one piece of a larger problem. "So we're providing something that facilitates more timely algorithm development," said Melling. "Something that enables a more parallel process."

Agility acknowledges that Matlab is an essential tool for signal processing algorithm developers, with its vector semantics and visualisation capabilities bringing benefits. However, Matlab programmers often need to create a C model of their algorithm. This can have a number uses, including:

- *Algorithm prototyping*. C is a popular choice for prototyping algorithms and it is easier to connect C code to input and output sources, such as video streamers.

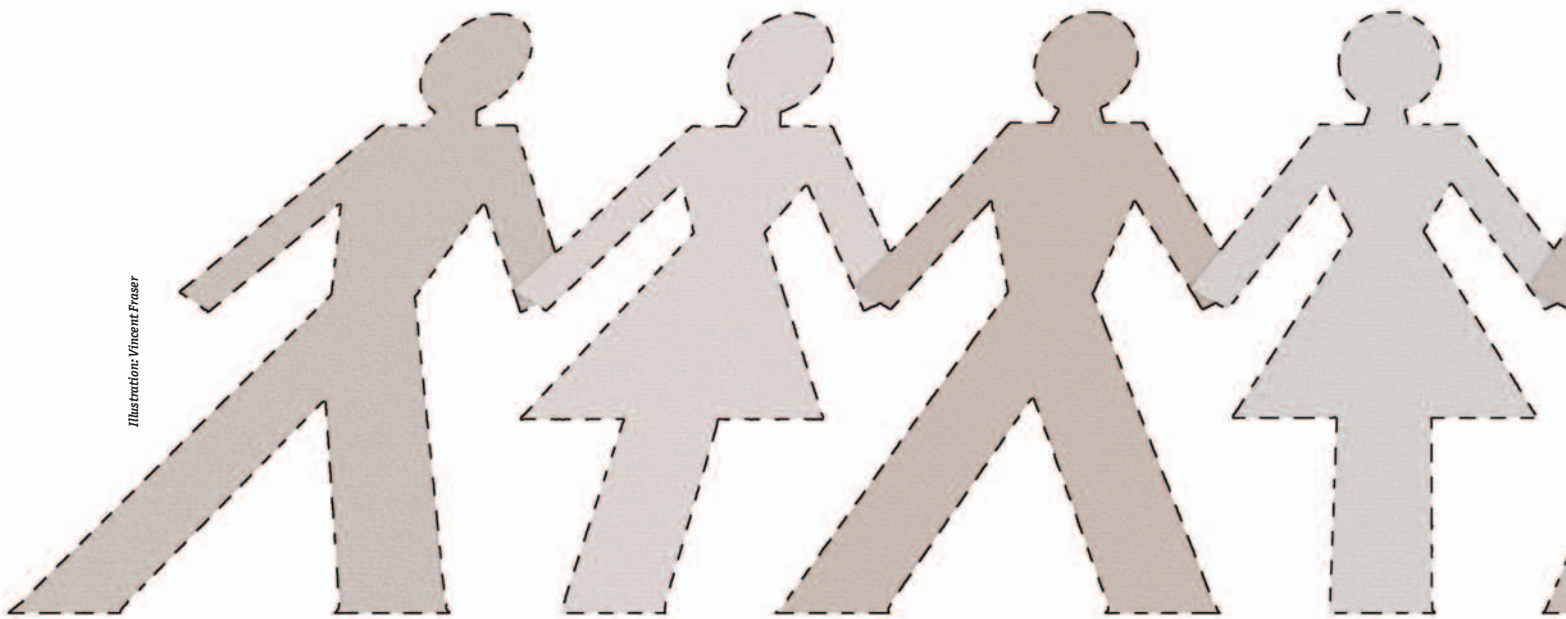

Illustration: Vincent Fraser

- *Integration into a system model*. DSP systems typically contain many components and these may be modelled in different languages. A C model allows these models to be integrated.
- *As the core of a software application*. If the algorithm is targeted at a software application, then the C model can simply be integrated with a small amount of handwritten code and compiled into the application.

Creating the C model requires developers to freeze algorithm changes temporarily and then translate their algorithms manually into C. This can be a laborious task and, as a result, developers tend to avoid making this translation until late in the design process. This increases the length of the development process and increases the cost of fixing bugs. Manual translation is also error prone, making it difficult to determine if bugs are the result of a poor algorithm or a translation error.

Agility says these challenges can be addressed by using a tool such as Agility MCS that automatically generates C code from Matlab. Using this approach, developers can continue using Matlab to modify their algorithms and to generate reference models as needed.

According to Dr Stahl: "Many of our customers start to develop dsp algorithms in Matlab, then transition into CoWare's ESL 2.0 solutions. With Agility's solutions, we are reducing the transition pain and making the entire flow from Matlab more productive."

Melling added: "Our Matlab to C technology offers a high productivity, high performance solution. We are very excited about the integration with CoWare's ESL design solutions and working with CoWare will address customers' needs for better connection of their algorithm models with established product development tools."

Boosting simulation performance

C models generated from Matlab will run inside CoWare's Signal Processing Designer, increasing simulation performance at least threefold in typical cases where the entire dsp system is explored. CoWare says this is in comparison to a traditional cosimulation approach, where an interpreted Matlab model is run using the Matlab engine. The company believes this is particularly relevant to complex wireless standards such as Long Term Evolution (LTE), where design teams are transitioning from Matlab research to production design.

CoWare users will be able to use Agility's technology to include Matlab models inside virtual hardware platforms modelled in SystemC. A C model generated by Agility's MCS product can be integrated inside CoWare Platform Architect using standard SystemC interface methodologies.

Dr Stahl noted: "DSP subsystems are very important and we had technology which allowed their simulation. We felt it was important to integrate Matlab into this simulation, but needed a better way in which to do it."

He said engineers have traditionally rewritten Matlab code as C. "But that's a problem if you have 10,000lines of code. Our partnership is allowing designers to implement Matlab IP more efficiently."

Melling noted: "What you have to remember is that one line of Matlab code translates into between five and ten lines of C, depending upon how vectorised the Matlab code is. That means there's upwards of 50,000lines of C code."

Dr Stahl explained the significance of translating Matlab code. "If you're investigating next generation wireless, these systems have perhaps 50 top level parameters and examining them requires thousands of simulations. If your simulation speed is cut by a factor of 10 because there's a piece of Matlab code in there, it makes a huge difference in how effectively you can explore the system."

A similar problem is encountered in multimedia system development. Dr Stahl noted. "In wireless systems, you're performing simulations to optimise your algorithms. You'll do that in video system development, but you'll also want to take the architecture into account.

"Most customers will take Matlab code and put it into our environment, rather than into a dsp simulation environment. It's a different design flow, but Matlab code is accelerated in both cases." ■