Validating Hardware Communication Architectures

The benefits of formal methods

A great many electronic tools involve hardware architectures that are becoming increasingly complex to design. This is the case of computers dedicated to scientific computing, network servers for the Internet or car controllers, to name but a few. The case for using automatic validation tools based on formal methods is thus getting stronger.

Numerous applications now call for "distributed" or "shared" electronic architectures whose tasks are shared between several elements. In order to carry out these tasks simultaneously, in "parallel," their hardware components – processors, memories – are able to communicate and exchange data interactively. For processors numbering in the several hundreds, designing such architecture becomes extremely complex. Unfortunately, such machines often manage critical systems for which no fault can be tolerated.

Certain systems work asynchronously, that is to say that each processor has its own clock that must be synchronized with the others.

One of the key points for their communication concerns the cache coherence protocol. This protocol makes it possible to optimize the management of the autonomous memory of each processor and of the system memory. Algorithm design demands that the multitude of possible system behaviors be predicted. The communication issues involved are partly similar to those of telecommunication networks. In the latter case, the protocols are now tested using formal methods implemented in automatic design and validation aid tools. Researchers from projects VASY and PAMPA have been collaborating for several years with Bull, within the DYADE EIG group, to develop protocol validation tools adapted to the top-of-the-line hardware architectures created by Bull.

Up to now, the validation of machines was done with programs written by hand and using languages specifically adapted to the simulation of a hardware architecture, but not to the validation of such complex protocols as cache coherence. The researchers' innovative idea was to use a formal language, in this particular case ISO standardized LOTOS, that had initially been developed for the validation of distributed systems, to deal with the present kind of problem automatically. According to Bull engineers, this will make it possible to significantly lower the validation time for an architecture, while increasing its quality. The first step, conducted by project VASY, is system specification. In other words, the hardware architecture and cache coherence protocol are formally described in LOTOS. Its good functioning is then checked. To do this, researchers call upon a software environment dedicated to protocol engineering, called CADP (for Caesar/Aldebaran). CADP has been developed for 10 years by project VASY in collaboration with project PAMPA and other partners. CADP makes for compilation, simulation, verification and LOTOS formal specification test synthesis. Today, nearly 200 academic and industrial sites are using CADP worldwide.

To test telecommunication protocol software architectures, researchers call for formal specifications.

Researchers from project PAMPA have used an automatic test synthesis tool initially called TGV (Test Generation with Verification technology). TGV is also the name of France's very fast trains, so it is not surprising that this validation tool is fast, which is significant when hundreds or thousands of cases must be tested. Since it is based on formal descriptions, TGV also improves test reliability. TGV was developed in collaboration with Vérimag, a joint CNRS, INPG and UJF laboratory. The tool uses CADP libraries and a formal description of the test objectives. A test case is deduced, which is a graph in which several paths must be verified. TGV was brought up to industrial standards over a two year period in collaboration with Vérilog. It has since been integrated into a SDL compiler (SDL is another standardized formal language). The first version has just been marketed under the name of Test Composer. To automatically generate hardware tests, researchers have adapted TGV to LOTOS specifications, and to the hardware to be tested. The results speak for themselves. Using CADP together with TGV, Bull engineers have modeled their cache coherence protocol architecture and have already automatically tested hundreds of behaviors. Some twenty bugs were found that had not been previously detected using older techniques, eight of which were particularly sensitive. In addition to such encouraging results, researchers are very satisfied with the industrial response to the formal methods they are advocating.

Among the programming languages that are adapted to on board systems, the other big family is that of synchronous languages: a single, global clock manages the various tasks to be performed—a sort of local parallelism.

Such languages are therefore particularly well suited to real time control systems, like car controllers, telephone switches or certain communication protocols. Here too, formal programming and verification methods are promising, if still new. A joint INRIA and Ecole des mines de Paris researcher team has been developing such a language for the last ten years. The language is called Esterel and is the work of project MEIJE. Esterel possesses an imperative syntax and a formal operational semantics, based on interpretation in a representation model comparable to electronic circuits. It is therefore adapted to the design of mixed hardware/software components, which are increasingly being used. Its semantics also allows for exhaustive simulation, an automatic verification method for finite configuration programs. This design environment for both hardware and software is marketed by a small-and medium-sized company, Simulog. It aroused the interest of some big names in on board electronics, such as Dassault Aviation, who set up a programming team to develop Esterel code, Texas Instruments, who want to rewrite their cell phone processor architecture in Esterel, Intel, Motorola and Thomson among others.

Researchers contacts:

PAMPA [1]: Claude.Jard@inria.fr
MEIJE [2]: Robert.De_Simone@inria.fr
VASY: Radu.Mateescu@inria.fr

(1) Irisa joint project with CNRS, Insa Rennes and the University of Rennes 1

(2) Joint project with Centre de Mathématiques Appliquées (CMA) the l'École des Mines de Paris