Institution: **University of Oxford**

Unit of Assessment: UOA11

Title of case study: Validation of Embedded Systems with Bit-Accurate Floating Point (6)

1. Summary of the impact (indicative maximum 100 words)

Embedded software in the transportation sector (railway, automotive and avionics) needs to meet high reliability requirements because errors may have severe consequences. Research since 2008 in the UoA has developed effective reasoning technology to provide assurance that key error types are eliminated from embedded software, and has created novel algorithms to prove its integrity. Major players such as [text removed for publication] GM and Airbus have used technology developed in the UoA to verify the absence of errors. A particular advantage of this technology is its ability to reason about floating-point arithmetic, meaning that a much wider class of properties can be verified. The technology is widely distributed via third party operating systems and tool-sets.

2. Underpinning research (indicative maximum 500 words)

Floating-point arithmetic, the computer realisation of scientific notation, is essential for many embedded and safety-critical systems in transportation industries, including the automotive and avionics industries, where inaccuracies in floating-point calculations can cause subtle changes of the control flow, potentially leading to disastrous errors. Safety-critical embedded software frequently relies on the industry standard for floating-point computation, IEEE 754, typically at single (32 bit) or double (64 bit) precision. However, the existing software technology for validating this type of arithmetic is not precise enough, and can result in inaccurate verification [5].

IEEE 754 represents numbers in the form $\pm 2^e \times 1.d_1..d_m$ where d_n are binary digits 0 or 1 and where m and the range of e are larger for double as opposed to single precision. There are special forms for representing 0 and numbers extremely close to 0.

The starting point for this research was a request from an industrial user who wanted to improve assurance of embedded safety-critical software. Professor Daniel Kroening, based since 2008 at the Department of Computer Science at the University of Oxford, immediately realised that the challenge was to develop bit-precise reasoning for IEEE floating-point arithmetic. Research that was subsequently led by Kroening, and involved colleagues in the UoA and ETH Zurich, was presented at FMCAD 2009, and described a simple and general, yet powerful, framework for building abstractions from formulas [1]. The framework was implemented as a bit-accurate, sound and complete decision procedure for IEEE-compliant binary floating-point arithmetic. The procedure, known as mixed abstractions, benefited in practice from its ability to flexibly harness both over- and under-approximations, and demonstrated the potency of the procedure for the formal analysis of floating-point software. This new approach enables the most accurate calculations and reasoning in order to achieve the highest possible precision, which in turn delivers enhanced reliability of embedded software in the transportation sector.

In 2010 Kroening devised a new, broadly applicable reasoning technique to analyse programs with loops, known as Abstract Conflict-Driven Learning (ACDL), presented at POPL 2013 [6]. ACDL was a new program analysis method that embedded an abstract domain inside the Conflict Driven Clause Learning (CDCL) algorithm of modern satisfiability (SAT) solvers. The procedure combined over-approximations of greatest fixed points with under-approximations of least fixed points to obtain more precise results than computing fixed points in isolation, and generalised implication graphs used in satisfiability solvers to derive under-approximate transformers from over-approximate ones. This provided a new method for static analysers that operate over non-distributive lattices to reason about properties that require disjunction [2].

Since 2010 Kroening and colleagues have demonstrated the usefulness of the ADCL technique on a series of difficult floating-point programs. At TACAS 2012 they presented the instance of ACDL that operates over floating-point intervals. In experiments, their analyser was consistently more precise than a state-of-the-art static analyser and significantly outperformed floating-point decision procedures [3]. Further research was presented at SAS 2012 demonstrating the first step towards a uniform framework for the design and implementation of satisfiability algorithms, static analysers and their combination [4]. At SAS 2013, Kroening's group presented research that extended the FMCAD 2012 and POPL 2013 work to support Craig interpolation [5]. The results led to the first interpolation procedure for floating-point logic and subsequently, the first interpolation-based verifiers for programs with floating-point variables. The paper included a comparison with four competing techniques, two of which were commercial tools, and demonstrated that this new approach was able to verify programs which are challenging for current verification tools [5].

3. References to the research (indicative maximum of six references)

The three asterisked outputs best indicate the quality of the underpinning research.

[1] Brillout A, Kroening D & Wahl T. Mixed abstractions for floating-point arithmetic. Formal Methods in Computer-Aided Design, 2009, pp.69-76. DOI: 10.1109/FMCAD.2009.5351141

***[2] D'Silva V, Haller L & Kroening D. Abstract conflict driven learning.** POPL '13 - Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 143-154. DOI: 10.1145/2429069.2429087
This paper describes the ACDL framework in its most general, abstract form, and features a rigorous formalization of the algorithm.

***[3] D'Silva V, Haller L, Kroening D & Tautschnig M. Numeric Bounds Analysis with Conflict-Driven Learning.** TACAS. Lecture Notes in Computer Science Vol. 7214, 2012, pp 48-63. DOI: 10.1007/978-3-642-28756-5_5
This paper presents an instance of the ACDL algorithm for intervals over IEEE floating-point arithmetic and compares it experimentally with state-of-the-art approaches.

[4] D'Silva V, Haller L & Kroening D. Satisfiability Solvers Are Static Analysers. SAS. Lecture Notes in Computer Science Vol. 7460, 2012, pp 317-333. DOI: 10.1007/978-3-642-33125-1_22

***[5] Brain M, D'Silva V, Griggio A, Haller L and Kroening D. Interpolation-Based Verification of Floating-Point Programs with Abstract CDCL.** SAS. Lecture Notes in Computer Science Vol. 7935, 2013, pp 412-432. DOI: 10.1007/978-3-642-38856-9_22
This paper extends the ACDL framework to Craig Interpolation, and features a comparison on challenging floating-point programs with two commercial tools.

[6] Vijay D'Silva, Leopold Haller, and Daniel Kroening. Abstract Conflict-Driven leaning. Symposium on Principles of Programming Languages (POPL), 2013
This paper describes a new reasoning technique to analyse programs with loops.

Grants

Funding in excess of £4 million since 2008 from EPSRC, European Union, Intel Corp, Microsoft Research, [text removed for publication] Texas Instruments, Semiconductor Research Corporation, UK Technology Strategy Board and UK Defence Science and Technology Laboratory.

4. Details of the impact (indicative maximum 750 words)

Attempts to use automated formal verification in industrial-scale examples go back several decades, the motivation being that bringing verification into the software engineering process should ultimately replace most or all testing, create more reliable systems, and save both costs and time. This has proved a more challenging activity than was first hoped, though improvements both in verification technology such as model checkers (tools which explore a system's reachable states and report any errors found) and the computers available for running this technology are now enabling real progress.The innovative reasoning technology developed in the UoA offers significantly enhanced accuracy in computation and has resulted in important benefits to the transportation sector, particularly in the areas of verification and validation (V&V) of embedded systems and safety certification. CBMC, an existing tool developed by Kroening, has been substantially upgraded in the UoA to support the new reasoning for floating-point operations and is now used extensively in the sector.

V&V of embedded safety-critical software is a crucial but expensive step to ensure safe operation in the transport sector. The industry estimates it to account for 50-80% of the sticker price of an aircraft, for example. Existing technology yields error reports for some software artefacts that are in fact safe; the inspection of these by engineers is costly and time consuming. In addition, the test patterns prescribed by industry safety standards previously had to be engineered manually, incurring delays and further cost. The reasoning technology described above aids V&V processes in two ways: by eliminating 'false alarm' error reports, and by generating software test patterns that satisfy industry requirements. The novel bit-accurate precision of the new method enables, for the first time, proof of those software artefacts that could not be shown to be safe with existing technology. This reduces the cost of V&V of embedded systems, as well as the time required to carry out the V&V necessary to obtain the final certification and validation required for operation [A]. This time-saving can translate into substantial time-to-market advantages and has resulted in very significant benefits to the global transportation sector including reduction in recall rates for the automotive industry and reduced testing times for embedded systems reducing the need for costly traditional code testing methods.

Safety certification cases can be based either on engineering and validation processes, or on evidence. The software technology developed at Oxford University is unique in that it can support both styles. It has been applied in numerous use-cases supporting safety certification, including those listed below. Other key benefits unique to this software technology include:

- the ability to generate evidence in the form of test inputs that can be validated independently;
- the ability to determine whether necessary test evidence exists; this saves time and effort in cases where a particular test input does not exist;
- the bit-precision of the approach avoids the frustrating experience of test inputs that fail to deliver the desired outcome when applied to the actual car or airplane.

Since late 2009 the new technology in V&V and safety certification has been adopted by a number of major systems vendors, especially in the automotive sector, as follows:

- Tata Consultancy Services (TCS), based in India, is the largest IT employer in the world and primarily targets the automotive market. TCS has integrated the enhanced CBMC into its AutoGen test generation tool, and states that ‘to the best of our knowledge CBMC is the only tool that supports floating point operations that match the precision of the target platform and yet scales to industry size code’. They also confirm that CBMC has enabled them to identify bugs and unreachable code that would otherwise have been missed, and save ‘a large amount of time’. Additionally TCS has used CBMC to improve the precision of static analysis, reducing around 68% (and in one case 100%) of false positives generated, and leading to ‘a substantial saving in manual efforts overall’ [A].
- General Motors in India has used the reasoning technology available via CBMC to validate C programs generated from Simulink diagrams. CBMC has been integrated into GM’s in-house verification tool [B].
- [text removed for publication]
- More recently (May 2013) a German automotive supplier, BTC-ES, has used the reasoning technology to make significant improvements to the V&V tool that they market; a BTC-ES research engineer reports that ‘specific floating-point issues were not taken into account within the BTC-ES internal tool chain and in the internal formats, among them floating-point casts, rounding modes, exact string representation of floating-point numbers. The latter circumstance establishes a clear and evident argument for the inconsistent tool behaviour since the semantics of the original C program was not accurately reflected within the data structures and algorithms of the BTC-ES tools. With the expertise of the Oxford team all these issues could be resolved such that sound analysis results are now obtained.’ [D]
- In the aeronautics sector the technology has been applied by Airbus where, during the CESAR project (‘Cost-efficient methods and processes for safety relevant embedded systems’ funded by the EU and Uk Technology Strategy Board), the CBMC tool was used in the X-man Verifier to verify models in the avionics industry. The project team, including staff from the UoA and from Airbus, conducted a case study on a representative avionics application – the Ground Fuel Transfer function of a large transport aircraft. It models the specific behaviours of the fuel management system when the aircraft is physically on the ground, as opposed to behaviours while the aircraft is in flight. [E].

Impact on industry standards

The research described above has led to the new IEEE SMT-LIB standard, devised and written by Kroening’s group, and the only one that exists in terms of floating-point reasoning. The standard was informed by the demonstrated capabilities of CBMC, and has had significant impact; for example, it has been implemented in Microsoft’s Z3 SMT Solver, a high-performance theorem prover. Kroening and colleagues extended Z3 to support the theory of floating-point arithmetic with an implementation consisting of two components: an SMT-LIB 2 front-end tailored to the floating-point theory, and a theory solver for bit-precise reasoning about floating-point arithmetic. The theory solver makes use of techniques such as mixed abstraction, rewriting, and bit-blasting (via the theory of bit-vectors), and is now an integral part of Z3, developed by Microsoft and available for sale on Microsoft's webpage [F].

Other applications

Other areas of industry have also used CBMC which has the new reasoning technology embedded within it to perform V&V and safety checking – e.g.

- South Korea's Advanced Power Reactor's Reactor Protection System is concerned with safety-critical systems in nuclear reactors. Research published in 2011 showed that 'the HW-CBMC reduces cost by providing automated way of establishing the consistency of HDL implementation using the ANSI-C implementation as a reference, because debugging and testing cost of the ANSI-C implementation is usually lower.' [G]
- The Air France crash of 2009, when AF447 crashed into the Atlantic killing all on board, was caused by an airspeed measurement malfunction. Researchers at Galois (a US high assurance R&D software company) and the US National Institute of Aerospace have shown that, in relation to airspeed measurement systems, 'CBMC can prove that the C code is memory-safe, including proving there are no arithmetic underflows or overflows, no division by zero, no not-a-number floating point values, no null-pointer dereferences, and no uninitialized local variables.' CBMC thus provided further reassurance about the safety of such airspeed measurement systems [H].

To give an indication of how widely available CBMC is, it is contained in the standard distribution of the Ubuntu, Fedora and Debian versions of Linux. Debian alone has over 10m installations [I].

5. Sources to corroborate the impact (indicative maximum of 10 references)

[A] Letter from TCS, held on file, corroborating impacts relating to time-saving, support for floating point operations, elimination of false alarm reports, and reduction in manual checking.
[B] General Motors paper, held on file, corroborating their use of CBMC as a verification tool.
[C] [text removed for publication]
[D] BTC-ES report, held on file, corroborating the way in which support for floating point operations led to more accurate and consistent analysis.
[E] http://www.philipp.ruemmer.org/publications/erts2-x-man.pdf and http://www.ccs.neu.edu/home/wahl/Publications/hkwltcrs12.pdf
Paper showing case study in avionics with Airbus.
[F] Microsoft's Z3 is available at:
http://www.microsoftstore.com/store/msusa/en_US/pdp/productID.253755500
[G] Lee D-A, Yoo J, Lee J-S. Equivalence Checking between Function Block Diagrams and C Programs Using HW-CBMC. Computer Safety, Reliability, and Security. Lecture Notes in Computer Science Volume 6894, 2011, pp 397-408.
http://link.springer.com/chapter/10.1007%2F978-3-642-24270-0_29 *Corroborates the use of CBMC as a verification tool in a safety-critical system in Korean nuclear reactors*.
[H] Pike L, Niller S, Wegmann N. Runtime Verication for Ultra-Critical Systems. Proceedings of the 2nd International Conference on Runtime Verication (RV 2011).
http://www.cs.indiana.edu/~lepike/pubs/pike-rv2011.pdf *Corroborates the use of CBMC as a verification tool in relation to an airspeed measurement system*.
[I] Linux Distribution evidence - Debian: http://packages.qa.debian.org/c/cbmc.html,
Ubuntu:https://launchpad.net/ubuntu/+source/cbmc,
Fedora:https://admin.fedoraproject.org/pkgdb/acls/name/cbmc