

World's fastest simulation technology able to faithfully reproduce CPU operations

March 13 2012

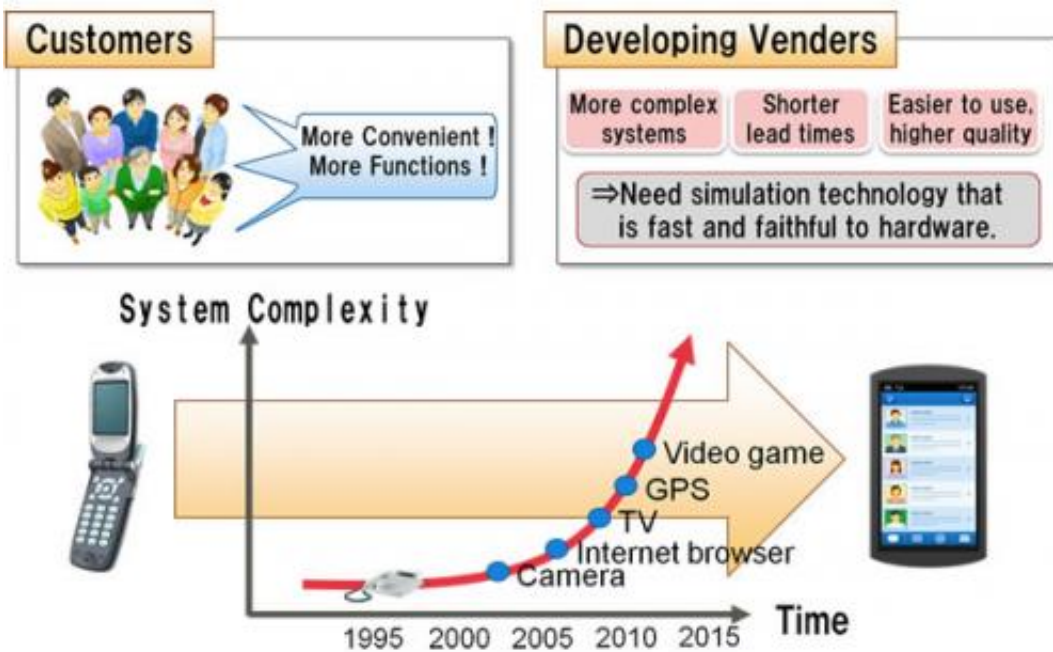


Figure 1: Technology history of the mobile phone

Fujitsu Laboratories Limited today announced that it has developed the world's fastest simulation technology for systems using the ARM computing core, widely used in mobile phones and other electronic devices. This technology is able to faithfully reproduce hardware operations with cycle-for-cycle real-time accuracy.

Systems using the ARM core have become dramatically more complex

in recent years, creating a need for a simulation technology that runs faster and with better fidelity to the hardware, for use both in testing whether the system runs as designed, and in application and [operating-system development](#). What Fujitsu Laboratories has developed is a platform for running simulations that allows for cycle-level simulations with low system overhead, based on a just-in-time compiler. The end result is that a standard PC environment can simulate at the cycle level an ARM multicore system running at faster than 100 MHz, which represents a hundredfold speedup over previous simulators. With a margin of error of only about $\pm 5\%$ relative to the actual hardware, this simulator is fast and faithful.

This technology will help to shorten the development cycle for systems and devices using ARM cores and encourage the development of a greater diversity of ARM-based systems.

Details of this technology were presented at DATE 2012 (Design, Automation, & Test in Europe), which opened March 12 in Dresden, Germany.

The ARM core is widely used in the CPUs of mobile phones and other [electronic devices](#). These devices and systems have grown vastly more complex over the past ten years, while at the same time, competition pushes manufacturers to shorten their time to market for new products. The need to create easy-to-use, high-quality products under tight lead times has led to a demand for supporting technologies that will enable rapid, accurate development of new devices and systems.

When developing systems based on the ARM core, simulations are used for advance testing as a way to accelerate the development process, in order to determine whether systems operate as designed. This has created a need for a simulation technology that runs fast and with good fidelity to the hardware.

The just-in-time compiler is a widely used approach for fast simulations of the ARM core, also used in Java runtime engines. But the JIT compiler approach does not faithfully reproduce time-based information. When cycle-level simulations are called for, other simulation technologies that run hundreds of times slower than a JIT compiler have been the only option.

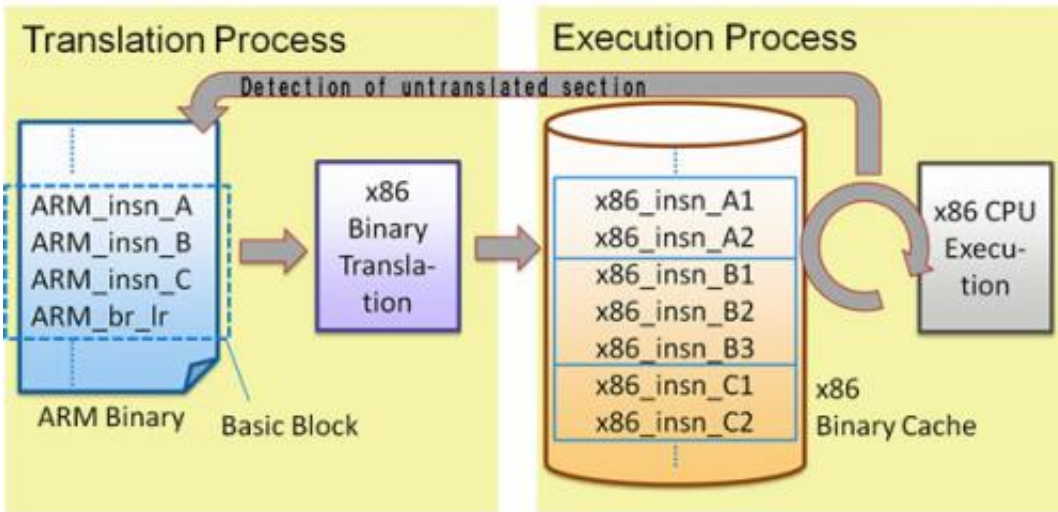


Figure 2: Flow of typical JIT compiler

Fujitsu Laboratories has developed a simulation technology that runs as fast as a JIT compiler but with precise cycle-level fidelity. This makes it possible for a standard PC environment to simulate an ARM multicore system with cycle-level fidelity at speeds greater than 100 MHz, a hundredfold speedup. Simulations run with a $\pm 5\%$ margin of error relative to running on the hardware, for fast, faithful simulations.

Key features of the technology are as follows:

1. JIT compiler technology with cycle-level precision

In order for the simulator to run fast and with fidelity to the way the code would run on its native hardware, Fujitsu Laboratories developed a real-time calculation algorithm optimized for use in a JIT compiler.

Figure 2 shows the flow of a process in a conventional JIT compiler. The program that runs on ARM is translated to a program that is executable on x86 hardware at block-level increments called basic blocks. That program is then run as an execution process. At this point, the x86-executable program is cached in memory, and as long as the cache is being hit, the execution process can continue without the translation process. Because the JIT compiler is limited to code at the function level, the number of codes per instruction is lower, and the processor load is lighter. And because cache hit rates are typically high, the program can run at high speed.

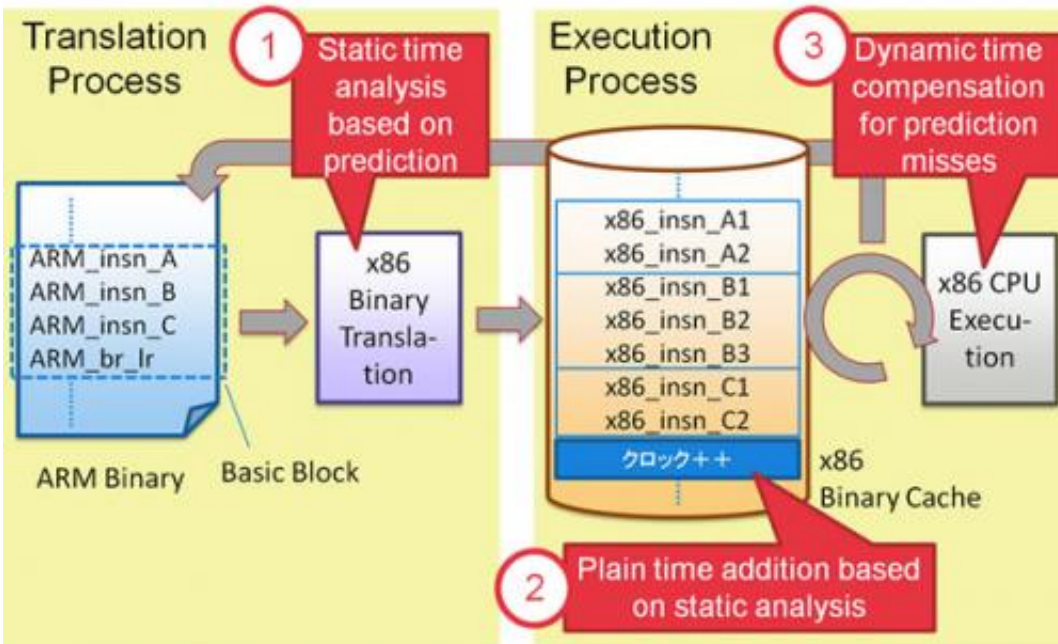
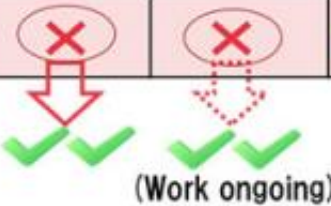


Figure 3: Flow of the newly developed JIT compiler with embedded real-time simulation

Figure 3 shows how the JIT compiler approach has been fused with the new real-time calculation algorithm. When codes are generated strictly at the cycle level, the fact that it includes time information means that the number of codes per instruction is hundreds of times greater than when codes are generated at the functional level by a JIT compiler, and execution speed is hundreds of times slower. In contrast to this, the real-time calculation approach developed by Fujitsu Laboratories uses three stages: (1) static time analysis based on predictions; (2) plain time addition based on the static analysis; and (3) dynamic time compensation in response to prediction errors. Each of these has been optimally allocated to the JIT compiler's translation process and execution process. This solves the speed problem.

As long as a translated section can be found in the cache and a prediction can be made as to the timing of an instruction's execution as decided by the static time analysis in stage 1, subsequent iterations are performed with a simple addition or subtraction of time deviations in stage 2, which is a lightweight process. This permits fast, real-time execution. If no prediction was made in stage 1, the stage-3 time analysis is performed and time compensation is applied. When an untranslated section is detected, the translation process is spawned as a new process, and stage-1 time analysis is performed. Stage-1 and stage-3 processes are time-analysis processes, which are processor-intensive, but overall they represent only a few percent of all calls, so their negative impact on execution speed is ultimately limited.

	Speed	Cycle Precision	Energy Assessment	Debugging	Cost
Actual Chip	✓✓	✓✓	✓	✗	✗
FPGA(*4)	✓	✓✓	✓	✗	✗
ESL(*5)	✓✓	✗	✗	✓✓	✗
	✗	✓✓			
CPU emulator (QEMU(*6))	✓✓	✗	✗	✓✓	✓✓



 (Work ongoing)

Figure 4: Features of the new technology

2. Multicore-compatible technology

There has been an extremely rapid adoption of multicore implementations of the ARM core, and this cycle-level simulation technology supports multiple cores. Fujitsu Laboratories' cycle-precise JIT compiler technology can perform separate real-time calculations for each core, so it can perform load-distribution that levels the load of each core, running high-speed cycle-level simulations that do not suffer in performance even when dealing with multiple cores.

This technology will help to shorten the development cycle for systems and devices using ARM cores and encourage the development of a greater diversity of ARM-based systems.

To support the development of a wider range of systems using ARM cores, [Fujitsu](#) Laboratories plans to continue work developing faster and more accurate [simulation technology](#) and to expand the range of ARM core types that can be used.

Source: Fujitsu

Citation: World's fastest simulation technology able to faithfully reproduce CPU operations (2012, March 13) retrieved 6 May 2024 from <https://phys.org/news/2012-03-world-fastest-simulation-technology-faithfully.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.