Stopping Hardware Trojans in Their Tracks

A few adjustments could protect chips against malicious circuitry

By Subhasish Mitra, H.-S. Philip Wong & Simon Wong

Posted 20 Jan 2015 | 21:00 GMT

Photo: Adam Voorhes; Prop Stylist: Robin Finlay

Long ago, the story goes, Greek soldiers tried for 10 years to conquer the city of Troy. Eventually, they departed, leaving behind a large wooden horse, apparently as a gift. The Trojans pulled the beautiful tribute inside. Later, a group of Greek soldiers slipped out of the horse and opened the gates for their compatriots, who easily sacked the sleeping city.

Nowadays, some 3,000 years on, a Trojan is a seemingly innocuous piece of software that actually contains malicious code. Security companies are constantly developing new tests to check for these threats. But there is another variety of Trojan—the “hardware Trojan”—that has only started to gain attention, and it could prove much harder to thwart.

A hardware Trojan is exactly what it sounds like: a small change to an integrated circuit that can disturb chip operation. With the right design, a clever attacker can alter a chip so that it fails at a crucial time (http://spectrum.ieee.org/semiconductors/design/the-hunt-for-the-kill-switch) or generates false signals. Or the attacker can add a backdoor that can sniff out encryption keys or passwords or transmit internal chip data to the outside world.

There’s good reason to be concerned. In 2007, a Syrian radar failed to warn of an incoming air strike; a backdoor built into the system’s chips was rumored to be responsible. Other serious allegations of added circuits have been made. And there has been an explosion in reports of counterfeit chips (http://spectrum.ieee.org/computing/hardware/counterfeit-chips-on-the-rise), raising questions about just how much the global supply chain for integrated circuits can be trusted.

If any such episode has led to calamity, the role of the Trojan has been kept secret. Indeed, if any potentially threatening hardware Trojans have been found, the news hasn’t yet been made public. But clearly, in the right place a compromised chip could scuttle antimissile defenses, open up our personal data to the world, or down a power plant or even a large section of a power grid.

A lot of research is still being devoted to understanding the scope of the problem. But solutions are already starting to emerge. In 2011, the United States’ Intelligence

Advanced Research Projects Activity (http://www.iarpa.gov/) (IARPA) started a new program to explore ways to make trusted chips. As part of that program, our team at Stanford University, along with other research groups, is working on fundamental changes to the way integrated circuits are designed and manufactured.

Today we try to protect against hardware Trojans by keeping careful tabs on where chips are made, limiting the opportunity for mischief by limiting who is authorized to make a chip. But if this research succeeds, it could make it practical for anyone to design and build a chip wherever they like and trust that it hasn’t been tampered with. More radically, our research could open up ways to let you use a chip even if there is a Trojan inside.

Today’s chips are so complex and costly to design and build that it’s very difficult for a single company to create them without outside help. One company might conceive and market an integrated circuit, but other companies often make critical contributions to pinning down the design. Still others may have a hand in manufacturing, packaging, and distributing the chips.

With so many cooks in the kitchen, there are multiple opportunities to meddle with the hardware. A natural place to start is at the very beginning, when a chip is being designed. Today, that’s done using sophisticated computer-aided-design software. These CAD tools are created by specialized companies that often work closely with chipmakers. The tools frequently contain millions of lines of code, and they change quickly: New algorithms are added almost continuously to help design faster, lower-power circuits. In principle, among the many thousands or perhaps millions of lines of code, it is easy to slip in a few extra ones to modify a hardware design. And there are multiple places it could be done. For one thing, routine circuit blocks, such as the accelerators used to crunch numbers for encryption and decryption, are often designed by third parties.

The other obvious time for an integrated circuit to be altered is during manufacturing. This was less of a concern decades ago, when chip manufacturing was more affordable and companies could make their own chips in their own fabrication plants, or fabs. But nowadays a new chip fab can cost upwards of US $10 billion, and research and development costs keep increasing. (http://www.pcworld.com/article/2691992/samsung-to-invest-147-billion-in-new-fab.html) Because of this very high up-front cost, most chipmakers now rely on a handful of outside foundry services, based in China, South Korea, Taiwan, and the United States, among other countries, that specialize in implementing silicon designs. Although there is no reason to suspect that any of these foundries may be adding malicious hardware, it’s impossible to exclude the possibility that they might make undesirable adjustments to the designs, potentially altering an entire batch of chips or a subset of them.

The U.S. Department of Defense is of course well aware of these vulnerabilities. To help address them, its Trusted Foundry program (http://www.dmea.osd.mil/trustedic.html) has accredited foundries, along with other links in the supply chain. The set of foundries allowed to work on these “trusted” chips is generally restricted to those in the United States. This limits access to the most advanced chips; many trusted U.S. foundry services have not been able to keep up their investments and are producing chips that are 10 years or more behind the current state-of-the-art manufacturing process. What’s more, the DOD program is focused on military chips for applications such as weapons and avionics. The integrated circuits used in such vital nonmilitary applications as medical computer systems and nuclear power plants are often made overseas and aren’t subject to the same level of supply chain scrutiny.

Ideally, what we’d like is a simple, quick, and cheap way to find out if a chip has a hardware Trojan inside. What would we be looking for? Researchers are still sorting out what kinds of hardware Trojan attacks are possible. But it’s already clear they can be hard to detect.

In one experiment (http://dl.acm.org/citation.cfm?id=1387714), conducted in 2008 at the University of Illinois at Urbana-Champaign, researchers designed a small backdoor circuit that gave access to privileged regions of chip memory. The Trojan could be used to change the process identification number of malicious software, allowing attackers to perform any operation and access any data they wish. Incorporating this Trojan added fewer than 1,000 transistors to the 1.8 million already on the chip, an increase of just 0.05 percent. And such tiny tweaks are likely to be par for the course: It doesn't take much additional circuitry to wreak havoc on a chip. In fact, it might not require any added circuitry. Recent research (http://link.springer.com/chapter/10.1007/978-3-642-40349-1_12) suggests that even slight adjustments to the electrical properties of existing transistors in a design could compromise security.

How would you find changes to the circuitry? You might think you could simply take a finished chip and look at it under a microscope. It's easy to imagine doing that back in the early 1970s, when Intel debuted its 4004 microprocessor. The 4004 had about 2,300 transistors, each measuring an optical microscope–friendly 10 micrometers or so. But today's integrated circuits are in another realm entirely. They can easily have billions of transistors, each well less than a hundredth the size of those in the 4004. While it's possible to scrutinize them with an electron microscope, the process is destructive. To get to the transistor level, you have to chemically remove or mechanically polish away the layers of metal that have been added on top of the transistors to wire everything together.

A straightforward solution to this problem is to destructively examine a representative sample of chips; if they're free of Trojans, you might conclude that all the untested chips in the batch are as well. But there is no guarantee that's the case; an attacker may have targeted only a subset of the chips in question.

Today's integrated circuits are international creatures. But trust isn't something that's built in from the start

So researchers are exploring other tests. One idea is to send different inputs into various circuits and then compare the resulting output data or the time it takes for information to move through the circuits, for example, to what you'd expect to see if the chip were operating normally. This sort of quality check can be performed after the chip is manufactured, and it could potentially detect a Trojan. But it's not the sort of thing that can be used to continuously monitor chip operation, so it wouldn't help you detect an alteration that's designed to be activated months or years down the line.

Researchers at IBM's Thomas J. Watson Research Center, in Yorktown Heights, N.Y., have been investigating a way to detect such intermittent Trojans. In this approach, a handful of chips are selected from the batch. They're put through their paces so that a "fingerprint" can be created based on such characteristics as power consumption, temperature, and electromagnetic emission. The idea is that a Trojan may have an effect on these fingerprint parameters, also called side-channel outputs, even if the Trojan isn't actively performing an attack. Statistical analysis can then be used to identify outliers. Or the fingerprinted chips can be dismantled and examined under an electron microscope to make sure they have no Trojans. If they pass that test, the fingerprints can be used to gauge the health of the rest of the chips in the batch. A mismatch in one or more of the fingerprints indicates the presence of a Trojan.

This technique has drawbacks of its own. For instance, it can't detect small circuit alterations, such as the addition of a single "exclusive OR" gate built with a handful of transistors. That's because even healthy chips display variations in the physical dimensions of their transistors and metal wiring. These variations in turn alter electrical and thermal properties from transistor to transistor. The resulting noise could easily swamp the signal created by a tiny Trojan.

Imaging could help see those smaller Trojans, and some researchers haven't given up on the idea. In 2011, a group led by Michael Bajura of the University of Southern California's Information Sciences Institute reported a way to image chips without tearing them apart. Instead of using an electron microscope, which can image only surfaces, the team directed X-rays at various angles (http://www-ssrl.slac.stanford.edu/content/science/highlight/2011-09-26/x-ray-nanotomography-imaging-circuit-integrity) through a chip. By stitching the resulting 2-D projections together, they could create a high-resolution 3-D rendering of the chip. This X-ray microscope could resolve details down to about 30 nanometers, which should reveal any added circuitry in even the most advanced chips used today. But unfortunately, the radiation can still cause damage, and a lot of work and time is needed to find departures from the chip's original design among what can easily be more than a billion transistors.

These sorts of challenges have led the U.S. government to consider a drastic measure—a change to the chipmaking process itself. The idea, called split manufacturing, is to literally split the manufacturing of chips into two steps. First, a state-of-the-art foundry would make, at the "front end of line," the smallest chip features: the layer of transistors, followed by the first (or perhaps a few) of the most finely detailed layers of metal wiring needed to connect them. In the second step, the chip would be shipped to a vetted, trusted foundry to be completed. This less advanced foundry would finish up the connections on the chip's back end—the chip's less fine layers of metal wiring and, ultimately, its connections to the outside world.

The idea is that if only the first few layers of a chip are made at an untrusted foundry, the chip will look like little more than a sea of arbitrarily connected transistors, and it will be impossible for foundry workers to know much about how the chip works. That incomprehension would in turn make it very hard for an aspiring attacker to devise a Trojan that would escape detection.

In principle, split manufacturing could be the best of both worlds: It would let a chipmaker take advantage of the advanced manufacturing capabilities of an untrusted foundry without disclosing everything about the chip. With that thinking in mind, IARPA established a new program in 2011 called Trusted Integrated Chips (http://www.iarpa.gov/index.php/research-programs/tic), dedicated to finding ways of exploiting split manufacturing.

Splitting the Check

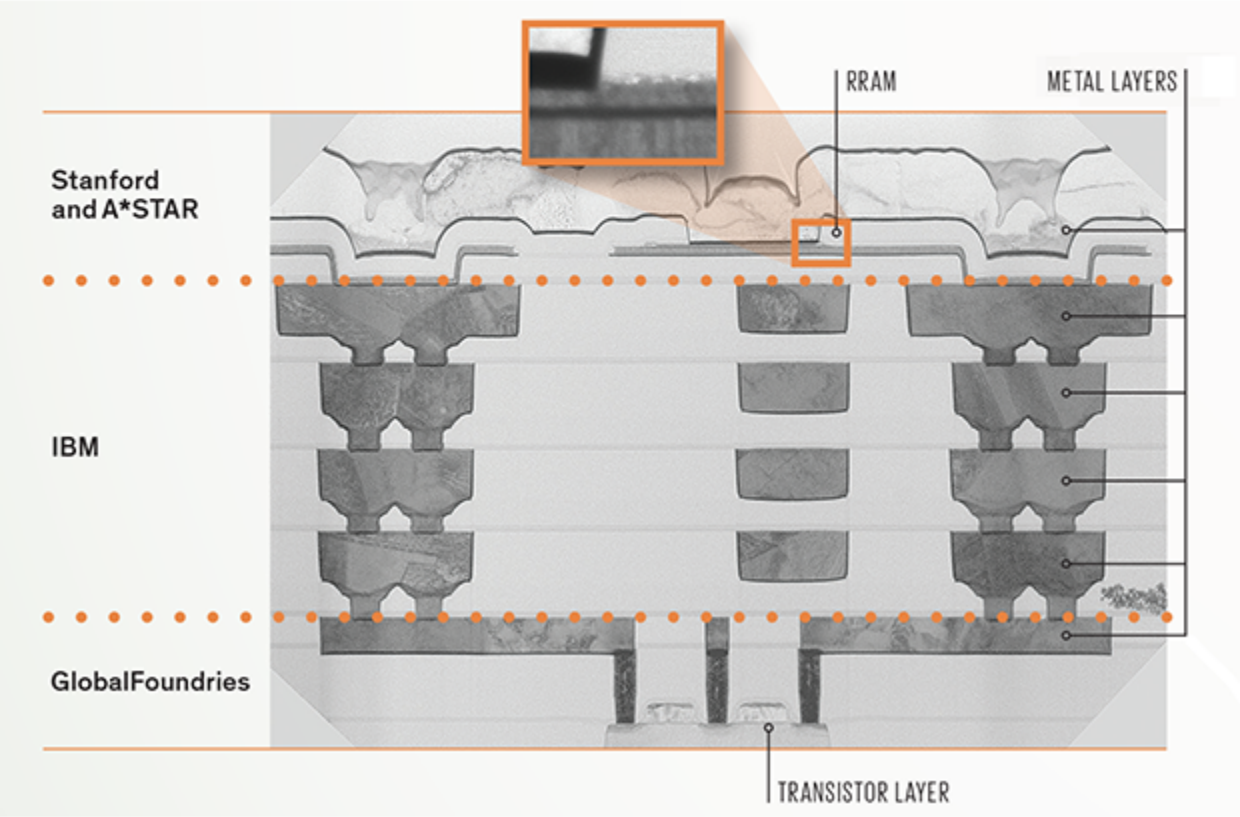


Image: A*STAR Institute of Microelectronics and Stanford University

This 130-nanometer test chip was manufactured in multiple places. The transistor layer at bottom and the first layer of metal interconnections above it were fabricated in Singapore by GlobalFoundries. The second through fifth metal layers were made by IBM in the United States, and the chip was finished by Stanford and the Agency for Science, Technology and Research (A*STAR). Checker circuitry that can hunt for abnormal operation is split among transistors in the transistor layer and resistive RAM (RRAM) switches that are sandwiched between the fifth and sixth layers of metal. These switches can be configured after the chip is made.

The IARPA program was specifically designed to address the threat of Trojans added in foundries. But we’ve actually devised an approach that expands on split manufacturing and that we think could be a complete solution to the problem—one that would address the possibility not only of Trojans introduced at the foundry but also of Trojans introduced at the design stage, a problem that split manufacturing alone can’t solve.

There will be a cost for such protection, of course, and it will be in the number of transistors. At a basic level, our approach is to build new circuitry on each chip that can monitor it on an ongoing basis, even after it's been purchased and installed. This circuitry performs a task called concurrent error detection, which has been used for years in fault-tolerant computing to detect hardware-related errors and boost the reliability of high-performance computers.

Concurrent error detection works on chunks, or blocks, of circuits on a chip. The trick to good error-checking design is to find an economical way to prove a block is operating correctly. This feat is typically accomplished by designing circuitry that runs alongside the block. The circuitry takes a computational shortcut, performing calculations on the input data to a block. If properly designed, it will generate results with the same properties as the output of that block when it's operating normally.

For example, one widely used check is the parity function, which tells you whether a computation has produced an even or odd number of 1s over a given interval. If the results of the parity function performed on the output of the checker circuitry and the one performed on the output of the circuit block match, there is a good chance that the circuit block is error-free. If there's a mismatch, it suggests something has gone awry.

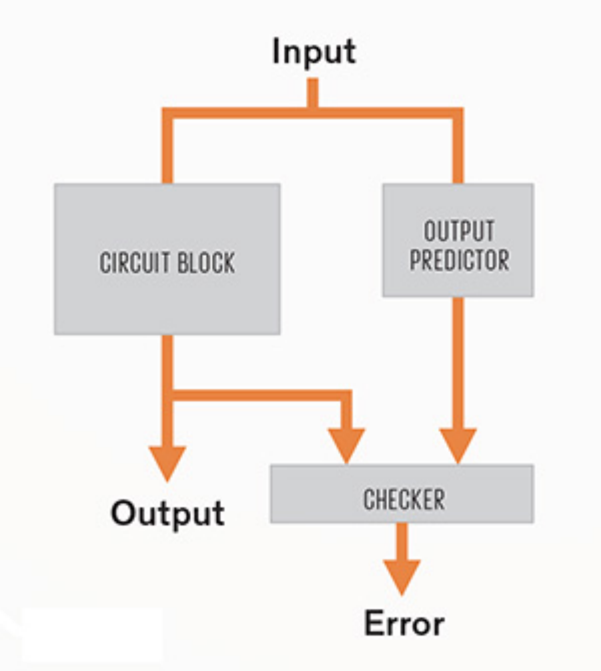
Checking for Discrepancies

This scheme works well for detecting random errors, but it could be circumvented with a carefully designed Trojan. For example, a clever attacker who knows a parity check is being used to protect the block could devise a circuit that would flip multiple bits in the output so that the number of 1s generated is preserved, masking the presence of the Trojan.

To prevent this deception, we've devised two ways to hide the functionality of the checker. The first is to change the checker design. We've devised a variation on the parity function, which we call randomized parity. An ordinary parity function might take 10 bits and count the number of 1s among them. In our approach, each checker circuit is configured to take a random sample of those 10, sampling perhaps only the second, fifth, seventh, and eighth bits in each group. The checker's sampling setting is chosen at random when the chip is being designed. And crucially, the configuration of the checker circuitry can be designed independently from the rest of the chip, which should help limit tampering at the design stage.

The second part of our approach is to obfuscate the function of the checker circuitry at the fab by building some of the connections with programmable switches that are set only after the chip is finished. That way, it won't be clear to the group fabricating the chip how the checker circuitry is designed, making it very hard to devise a workaround.

We build these switches using a form of memory called resistive (http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6193402)RAM (RRAM). An RRAM cell consists of an insulating layer sandwiched between two layers of metal. With the right choice of insulating material, RRAM can be made to switch between states of high and low resistance with a fairly small voltage, and it will keep that state even if there's no power, which means it can act as configurable wiring. We can break and complete circuits as needed by applying a voltage. (http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6177067)



The checker circuitry works by comparing characteristic information on a circuit block's output with similar information that is predicted and calculated directly by the checker circuitry.

Programmable switches themselves aren't new; many of the chips you find in TV set-top boxes, cars, and medical equipment are made with field-programmable gate arrays (FPGAs), standardized circuits that have their functionality programmed in transistor-based memory cells after they leave the assembly line. But RRAM is more attractive for this application, in part because it's compact. We've shown (http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6177067) that if FPGA memory is made with cells containing one transistor and two RRAM devices instead of the typical six-transistor form, the area of a chip can be reduced by 40 percent and its energy consumption by 28 percent. What's more, unlike traditional, transistor-based memory, RRAM can be placed among the metal connections above the active layer of transistors on a chip, and it can be inserted at whatever stage in the manufacturing of metal layers you like, even if you choose not to pursue split manufacturing.

We launched our project in 2012. In the very first phase, as a proof of concept, we manufactured our chip designs through the IARPA program, using 0.13-micrometer technology. The transistors and transistor-based checker circuitry, along with the first metal layer, were made at a GlobalFoundries fab in Singapore. Other metal layers were added at an IBM-built plant in Burlington, Vt. We then built the RRAM and last metal layer.

The process produced a handful of accelerators, built to perform data compression, and FPGAs. Graduate student Tony Wu and other members of our laboratory installed these chips on a circuit board attached to a computer in order to run tests and emulate attacks. They showed that the checker circuitry we'd built in the accelerator to run alongside the compression process took up just 1 percent of the area of the chip and did not slow it down at all. And they emulated a variety of different Trojan attacks on the chip by altering the outputs of various circuit blocks. The chip's checker circuitry could detect some 99.9998 percent of 10 million emulated hardware attacks.

This approach can detect Trojans that directly alter the operation of a circuit. It can even spot "replay" attacks. These are perpetrated by recording valid circuit block outputs and then reemitting them during an attack to give the appearance that everything is operating normally. But the checker circuits do have limitations. They can pick up data that's transmitted through the normal input-output channels of a chip. But they can't detect a Trojan that passively extracts information and then wirelessly transmits it to a third party through added radio-frequency circuitry. We're just now starting to explore protections against such an attack.

Even if split manufacturing isn't used, our checker-circuit design approach and the programmable RRAM will help make it difficult for a potential attacker to devise a hardware attack. But we think that whatever strategy is used to protect against changes—even one that combines split manufacturing, testing, and destructive imaging—it will be very hard to definitively rule out the possibility that a chip contains a hardware Trojan before it's sent on its way. Small circuits are simply too easy to hide.

Because our checker circuitry is designed to monitor a chip continuously, it will be able to watch for an attack, but it won't be able to prevent it from happening. So a key challenge now is to figure out what to do if one of these circuits detects a Trojan attack in the field. Will a simple reboot suffice, pushing off the next attempted attack for months or even years? Can we add circuitry for chip recovery, borrowing error-correcting techniques such as checkpoints to save data at critical intervals? Should we plan on building more redundancy into critical control systems to protect against the possibility of a Trojan, adding more chips to our equipment in case one has a fault?

These are questions that will become more relevant as hardware Trojan protection technology matures. We're now in the second phase of the IARPA project, which is targeting the more advanced, 28-nanometer manufacturing process, one step away from today's most advanced chips. We can start considering how the semiconductor ecosystem might be changed to make the chip supply chain more secure. Split manufacturing could help, but it could also significantly add to the cost of chips and the logistical complexity of the process. It may be that incentives and regulations will be needed to convince chip companies and foundries to adopt it.

Tackling the threat of a hardware Trojan will require tough calls and a sea change in the way we approach chip manufacturing. It could also mean we have to redefine what we mean by trust. But with the right approach, we could make attacks rare and relatively benign—and ensure that the most famous Trojan horse story remains ancient history.

*The authors would like to thank IARPA for its support, Dennis Polla and Carl McCants for their advice, Farzan Fallah for his early work on this article, and our students and postdocs at Stanford and our collaborators at A*STAR's Institute of Microelectronics, in Singapore, for their vital contributions to this research.*

About the Authors

IEEE Fellows Subhasish Mitra (http://web.stanford.edu/~subh/), H.-S. Philip Wong (http://web.stanford.edu/~hspwong/), and Simon Wong (https://profiles.stanford.edu/s-simon-wong) are all professors at Stanford University. Mitra's lab is focused on building robust nanoscale integrated circuits and systems. Before joining Stanford, he worked at Intel, where he solved a crucial problem in cost-effectively testing chips for defects. He sees a connection there to the hunt for malicious hardware. Whether a chip malfunctions because of manufacturing problems or a deliberate attack, Mitra says, the result looks much the same: "It all comes down to trust."

This article originally appeared in print as "The Trojan-proof Chip."

.