Prevent FPGA cloning

By Paul Dillien
High Tech Marketing

This article describes a new way of tagging designs to help to counter the fast growing trade in stolen intellectual property (IP) and cloned designs. The topic is a difficult one for the industry to discuss; recently, however, more and more voices have been raised on the issue.

An estimate of the prevalence of counterfeit electronics has been put as high as 10 per cent. (The International Chamber of Commerce website, for example, includes the statement: "Counterfeit electronics are estimated to account for 1 to 10 per cent of global electronic sales"). This is supported by the Alliance for Gray Markets and Counterfeit Abatement (AGMA), an industry group that consists of Hewlett Packard, Cisco, and other top tier electronics OEM companies, which estimates the loss to manufacturers at more than Rs.4.97 lakh crore ($100 billion). The hidden costs of damaged reputations and reliability issues for the end customer are more difficult to quantify.

One unfortunate consequence of the rise of programmable logic coupled with the decline of the ASIC is that it is now easier than ever to copy a design. Some Asian or Eastern European companies openly claim to specialise in "reverse engineering" or copying PCB layouts and memory contents. It is difficult, expensive and time consuming to reverse engineer an ASIC, but simple to copy the configuration bit stream of the most popular FPGAs as illustrated in Figure 1. It is possible to encrypt bit streams for some SRAM based FPGAs. However, the overwhelming majority of customers do not use this because of added cost, complexity and logistics. The feature is not offered in the low cost parts that most customers use for high volume applications. Non-volatile FPGAs avoid this, but comprise only a small fraction of the overall market.

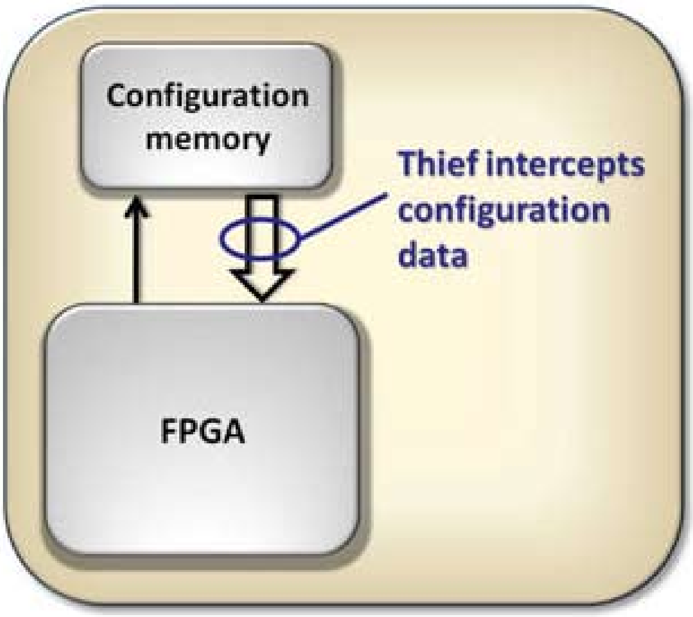

Figure1: Stealing an FPGA design is not complex.

As a result, companies can find that overnight their IP appears in the product of a competitor. The counterfeiter carries virtually no R&D cost, and can therefore undercut the legitimate supplier on price and steal market share.

The problem is how to deter theft and prove ownership of the design. This is where a product for tagging designs can help. An example of this is DesignTag from Algotronix. When buried in the FPGA bit-stream, the code of the tagging product is difficult to locate and disable—even if the fraudster knows that it is included.

Once this code is duplicated into the pirated bit-stream, it announces that the design has been copied. There is a strong parallel with the concept of marking valuables with your postal code or zip code using UV pens. It cannot prevent the theft, but allows ownership to be established afterwards.

A design-tagging product may occupy Rs.28.34 ($0.57) cents worth of silicon in an FPGA (using the 100+ list pricing), which represents 1.3 per cent of the logic resources on the chip. This is a modest overhead compared to the potential damage represented by illegal copies.

Figure 2: A product that may provide "proof-of-theft".

A more insidious problem comes from so-called "overbuilding". The background to this problem lies in the mass exodus over the last decade of companies exiting production in favour of designing products to be built by contract manufacturers (CEM). This positions the CEM in a central and critical role. The vast majority of CEMs, of course, are responsible and provide a valuable service. Unscrupulous ones, however, can supplement their profits by producing more units than they are contracted for and selling the excess onto the grey market. Again a product for tagging designs identifies the source and ownership of the design.

A further application of such product is to provide serial numbering or version identification. For example, medical, automotive, industrial, military, or aerospace manufacturers may wish to tag equipment with end customer codes or track the FPGA configuration version. Version control is especially important where upgrades are routinely applied. The design tagging product can be detected in working systems without making electrical contact and is particularly beneficial for ball grid arrays where the top of the package is far more accessible than the electrical connections. Monitoring can also be achieved without resorting to software or hardware interrupts that might affect the normal operation.

Yet another twist is to have an embedded system interact with the design tagging product so that it flags fault or status conditions. In this application, the DesignTag is programmed to output a different code that unobtrusively signals the internal condition.

So how does it work?

DesignTag is a small low-cost IP core that can be included into an FPGA as part of the design. It is a digital core with a unique code ("signature") that can be identified externally without needing to read the FPGA bit stream or internal registers.

It works by modulating the power dissipation of the host device in a predefined way. Tiny heat pulses propagate through the chip package with low attenuation. The level of the power "surge" is selected to provide a package temperature rise of only about 0.1 degrees C. The additional dissipation is typically 5mW, against an operational power consumption of >150mW for a mid-sized Spartan FPGA, which means the signal is deep below the noise floor.

The DesignTag defaults to turn off after 15 minutes of operation (the user can modify this value). This has two effects. Firstly it eliminates the small incremental power consumption, and secondly it also makes detection by a fraudster more difficult as power has to be cycled. Optionally, DesignTag can be triggered by an internal event from within the FPGA. This could be an instruction to transmit the ID code or to indicate an internal status or error condition.

DesignTag database

The code data for all DesignTags is held in a central data base. Customers can choose whether they wish to make their code "public" or "private". The advantage of a public listing is that all DesignTag Reader Software will be enabled to detect the code, thereby allowing customers, enforcement agencies, and contract manufacturers to confirm the validity of the device under test (DUT).

By comparison, private codes may be preferred when the customer does not wish any third-parties to be able to tell whether or not their product is tagged. In this case codes are distributed by the customer to trusted parties – only their reader software with these codes installed will be enabled to detect the corresponding tags.

DesignTag and FPGA

The IP core for use with FPGAs is instantiated into HDL code as a "black box" in either Verilog or VHDL. The deliverable includes the in-built code which will drive the thermal output, as well as I/O pins that are required by the circuit. The all-digital design has a small footprint, typically requiring only 256 slices in a Spartan 3 FPGA from Xilinx.

The core is only available as an encrypted EDIF netlist for security reasons. The IP includes several techniques to prevent "reverse engineering" as well as other schemes designed to mask it from being tampered with or detected.

The core is largely independent of the user design, so the impact on the design flow is minimal.

DesignTag takes an input clock in the range 2-to-250MHz to drive the timing. The clock should be derived from any available clock inside the system rather than being generated specifically for the tag IP core for two reasons. Primarily, of course, it is the lowest cost option, but also it confuses attempts to detect and disable the tag by making it appear to merge into the system.

Control inputs for the DesignTag can be driven from the embedded system or set as defaults at the design stage. For example, the core can be set to transmit continuously if required; alternatively, the core can be instructed to transmit for a specified time on power-up, and to then enter a stand-by mode. As yet another option, the transmit sequence can be initiated at any time by a Trigger input.

The core is licensed with up to four different in-built codes, which are selected by the Code Select inputs. The embedded system can program the core to transmit one of these codes on command to indicate internal status conditions. For example, conditions indicating that an overflow has occurred or a soft error was detected in memory can be signalled without interrupting the system functionality.

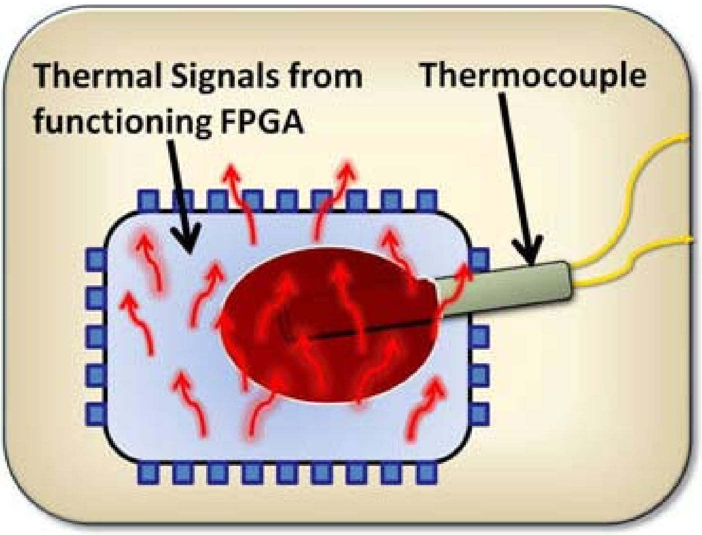

Figure 3: Detecting thermal signals with a thermocouple.

A Tamper Output can be taken from the DesignTag into the embedded system. This has two benefits. It can be used as a disable signal that impairs the system performance and confuses an attempt to disable the tag by flipping random bits in the configuration bit stream. The second benefit is that if an attacker reverse engineers the bit stream to reconstitute the design, then the DesignTag block will appear as an integral part of the design.

Detect DesignTag outputs

DesignTag codes are detected by placing a thermocouple in good thermal contact with the top of the package while the device is in operation as illustrated in Figure 3. The thermal readings are input into the DesignTag Reader Software, which decrypts the signals. Detection time for a positive recognition of a code is measured in minutes, and the system can detect and differentiate multiple tags within the same device.

Poor conditions may increase the time required, but the underlying algorithms accumulate confidence with increased sample sizes. Acquisition time can be reduced if readings are taken in still air. The self-heating of the DUT as power is applied will not prevent the tag being detected. Factors that could increase the detection time include thermally erratic conditions, poor contact between the probe and the package or multiple codes in a single design.

The DesignTag works with all popular styles of packages. However DesignTag is not recommended, at this time, for use with heatsinks or where forced air cooling is employed.

Attempting to defeat DesignTag

One of the most important features for any anti-counterfeiting scheme is the difficulty of identifying, removing, or disabling the identity tag. This article will not go into a detailed description of the in-built countermeasures against removal and reverse engineering for obvious reasons, but let's consider how a counterfeiter might approach the job of defeating DesignTag.

The first problem for a potential thief is to identify if the stolen code contains the DesignTag or not. The tiny thermal signals take the form of a 64-bit code and can only be detected by using DSP and decryption techniques. A spreading code is used to control the heat generator using a Linear Feedback Shift Register like circuit.

The spreading code generation circuit is based on the unique 'Tag ID', which acts like a cryptographic key, where each key results in a different pseudo-noise sequence. The DesignTag Reader Software will readily identify public codes, but without prior knowledge and authorisation, the software is not enabled to detect private codes, which will remain hidden.

Various aspects of the tag waveform are pseudo-random. The code is designed to frustrate attempts to decode the signals by repeated power cycling and correlating the observed signals.

Another possible attack is to obtain multiple DesignTag IP cores and compare them at the bit-stream or encrypted netlist level in order to see which specific resources have changed on the assumption that these will correspond to code bits. To frustrate this attack, many aspects of the tag IP core are varied in a random manner to create a very large numbers of differences between any two tag instances.

An attacker may try to disable the DesignTag. It is instantiated into an FPGA design as a black box at the design capture stage and once included is incorporated into the bit stream that configures the SRAM device. Users can choose to encrypt their bit stream, but most do not.

FPGA configuration bit streams typically range from 2Mb for a midrange Spartan device, through to 10's of Mb for the larger Virtex family. Fraudsters may attempt to use software to reconstruct a netlist from the bit-stream information.

While such software has been created in the past and has been reported in the technical literature it appears that such tools are not currently openly available to hackers. The output of such software would be a flat netlist of the design after mapping onto FPGA primitives; while this is a step forward from a bit-stream, it is still a long way from the original design source code.

The fraudster could also try and manipulate the design at the bit-stream level by selectively 'flipping' bits and monitoring the effect with a view to shutting off security features. This is a more practical attack since it does not depend on developing or acquiring bit-stream reverse engineering software. It does require recalculating the checksum on the FPGA bit-stream so it is not totally straightforward.

The key to defeating this attack is to make the time needed for each 'flip a bit and see what happens' cycle as long as possible and to make it difficult for an attacker to determine whether the security features have in fact been completely disabled. The fact that it takes several minutes to read a DesignTag code makes search based attacks such as this less feasible.

An advantage of DesignTag compared with FPGA design security mechanisms such as Device DNA or challenge/response schemes which involve accessing an external chip through I/O pins is that DesignTag is not tied to specific resources on the FPGA which have easily determined locations in the bitstream.

FPGAs based on Flash or antifuse technology are configured in the factory and so the code containing the DesignTag is not exposed to the end user. Any attack on the DesignTag would involve physically tampering with the device and result in destroying the FPGA.

Thermal, not electrical signals?

By this time you are probably wondering why an unconventional signalling medium was chosen.

The most obvious mechanism to signal to circuits within a chip is through package pins. However, there are important drawbacks to this mechanism in this context. To be useful to an end user, the tag mechanism has to be independent of the circuit board onto which the chip has been designed and independent of any system software.

In some cases, the developer of the circuit board and system software may be the party suspected of misuse of the IP. Also, accessing the tag should not require detailed knowledge of the system containing the suspect chip. With BGA chips and fine pitch flat packs, even the act of finding an appropriate place on a circuit board to probe a signal can be quite tricky. Accessing tag information through package pins will generally require a trained technician.

The bandwidth of a thermal path through a package is low, and well-matched to the signalling scheme adopted for DesignTag. The data transmission rate is slow, but very few bits are needed to create the code and the thermal approach gives some additional advantages.

Thermal signals are transmitted through the package and detected by a probe in contact with the package top. No interruption to the normal operation of the device is required and no access is needed to electrical connections. No additional package pins are required either so DesignTag can be added to existing products without changing their pin-out.

Readings can be taken from a device in-situ or in a test jig by a semi-skilled operator. High pin count ball grid arrays require dense PCB traces, and it can be a problem to trace out additional tracks to test points and might indicate to an attacker the presence of DesignTag.

Signalling from the tag using EMI, RF or power supply noise is also theoretically possible. A low level electrical signalling scheme would suffer from extraneous noise from the activity on the chip, fast power supply glitches and ground bounce, as well as coupling from activity on nearby PCB traces. External sources such as radio waves, mains power sources, and other EMI also cause interference.

Modern systems require multiple power supplies that must be well screened and decoupled. The supply smoothing would be fighting against the attempted signalling back out from the chip, and might compromise both the chip performance and the quality of the tag signalling. Potentially, the wide bandwidth of electrical signalling offers a much faster detection, but experimentation showed that this was more than offset by the issues discussed above.

Summary

The cost of stolen IP and cloned designs is growing fast. The DesignTag technology presented in this article provides a way to add a watermark to your design. Tagging your design provides a way of proving piracy or fraud without resorting to lengthy engineering evaluation.