September 2020

Dynamic Hardware Reconfiguration For Fast Test and Validation

Alex Levin

Furquan Shaikh

Dynamic Hardware Reconfiguration For Fast Test and Validation

ABSTRACT

The design and manufacturing of products such as laptops, tablets, etc. often starts with a design firm producing a reference design that is subsequently modified by different manufacturers prior to shipping the product under their own labels. Testing the reference design is no guarantee that a shipped-out design is reliable due to the possibility of substantial differences between the reference design and the shipped-out design. This disclosure describes a single embedded device, a reconfigurable reference board with many different components connected to it, that can be used to configure different (shipped-out) platforms. The reference board enables rapid validation of a large number of variations of a given reference design, including the rapid test and integration of newly introduced components. The time-to-market, lab-space, investment in test-and-measurement equipment, engineering resources, etc. required for test and validation are thus substantially reduced.

KEYWORDS

- Test and validation
- Test and measurement
- Reference design
- System-on-chip
- Embedded device
- Embedded controller
- Firmware
- Advanced configuration and power interface (ACPI)

BACKGROUND

The design and manufacturing of products such as laptops, tablets, smartphones, smart home devices, etc. often traces the following path. A design firm produces a reference design and prototype and provides it to several manufacturing/marketing companies (brands). The brands make their own modifications to the reference design before shipping the product in large quantities under their own labels. A shipped-out design is referred to as a descendant of the reference design.

There can be substantial differences between the reference design and a shipped-out design. Different brands may utilize different types, capabilities, specifications, or vendors of components, e.g., CPUs, GPUs, DRAM, storage, displays, etc. Merely testing the reference design, even when done thoroughly, is no guarantee that a shipped-out design that is a descendant of the reference design is reliable.

Testing of designs that are to be shipped-out is currently an expensive and tedious process. Across various brands that use a given reference design and across various models of a shipped-out design marketed by a given brand, it is not uncommon for a reference brand to have hundreds to tens-of-thousands of descendant, shipped-out designs.

Even apparently small variations between different shipped-out designs, e.g., differences in DRAM capacity, demand full-scale validation. This is because, in embedded systems, hardware and software development and validation are tightly intertwined. A pair of devices, their drivers, and the operating system can interact (co-react) to create a specific load, stress particular execution-pathways, share memory, etc. in a manner that can surface a rare bug, which can nevertheless affect large numbers of customers. Full validation coverage requires testing of

every hardware variation with the same software to ensure that every combination of components in the shipped-out design is validated.

Across the design firm as well as the manufacturing/marketing companies, the testing of thousands of shipped-out designs requires significant investments in test-and-measurement equipment and consumes substantial lab space and engineering resources. Being a critical component of the design-and-manufacturing process, such traditional time-intensive testing has an impact on the overall time needed to introduce a new product to the market.

DESCRIPTION

This disclosure describes techniques of accelerated validation of every shipped-out design, e.g., validating every shipped-out software-hardware combination, with no compromise on quality of the test and validation process.

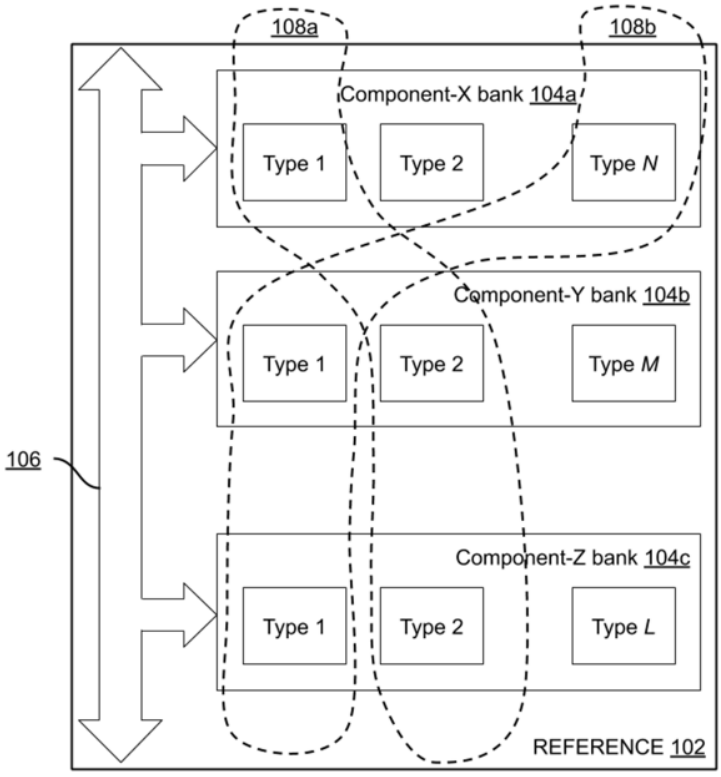

Fig. 1: A reference board comprising several component types

Per the techniques, illustrated in Fig. 1, a reference board (102) is constructed that includes all the components used in all possible shipped-out designs. The reference board is organized as follows. A bank of components-X (104a) includes *N* types of component-X. For example, component-X may be a CPU, and the *N* types of CPUs may include CPUs from different vendors, CPUs of different specifications, CPUs of different performances, etc., as found across the entire group of shipped-out designs. Similarly, the reference design includes banks of components-Y (102b), components-Z (102c), etc. For example, components-Y may include *M* types of GPUs of differing vendors, specifications, performance levels, etc., as found in the totality of shipped-out designs. Components-Z may include *L* differing types of DRAMs. Other component-banks can be banks of displays, storages, WiFi modules, embedded controllers, network interface cards, etc.

The component-banks (and their constituent components) are interconnected by a switchable bus (106), e.g., a switchable PCI bus, a multiplexed I2C bus, etc. The switchable bus enables setting up specific configurations of the reference board. For example, in one configuration (108a), a type-1 component-X (e.g., CPU), a type-2 component-Y (e.g., GPU), and another type-2 component-Z (e.g., DRAM) are interconnected. In another configuration (108b), a type-*N* component-X (e.g., CPU), a type-1 component-Y (e.g., GPU), and a type-1 component-Z (e.g., DRAM) are interconnected. In this manner, the reference board can be dynamically reconfigured to match every shipped-out design. The reference board design, per the techniques described herein, effectively enables a union set of shipped-out designs.

Per the techniques, a given shipped-out design can be tested on the reference board by reconfiguring the reference board to match the shipped-out design and by running the test-suite on the reference board in the particular configuration. In practice, reconfiguration information

can be encapsulated in firmware that can be downloaded to the reference board to automatically reconfigure the reference board and boot it, thereby readying it for testing. Per the techniques, the testing of thousands of shipped-out designs amounts to reconfiguration and re-testing of a single (or few) reference boards. The multiple workstations, test-and-measurement equipment, and engineering resources traditionally associated with multiple shipped-out designs are saved.

The reference board design described herein is effectively an embedded device, e.g., comprising ASICs (CPUs, peripherals, DRAM), firmware (BIOS, boot-loader, CMOS, etc.), and standard connectivity buses that map into plug-in ports (e.g. PCIe, I2C, I2S, SPI, etc.). Multiple devices of the same nature, e.g., two or more network-interface cards of substantially the same functionality but made by different manufacturers, are plugged into different ports of the reference board. The reference board design also has a switching topology that can be configured as necessary. Populating the reference board with multiple devices of the same nature, referred to as overstuffing the board, enables the reference board to assume a multiplicity of configurations, each corresponding to a particular shipped-out design. The techniques enable the rapid creation of a validation matrix of component-combinations, e.g., of a CPU to peripherals over known buses, where the components can be off-the-shelf devices, ASICs, peripherals, etc.

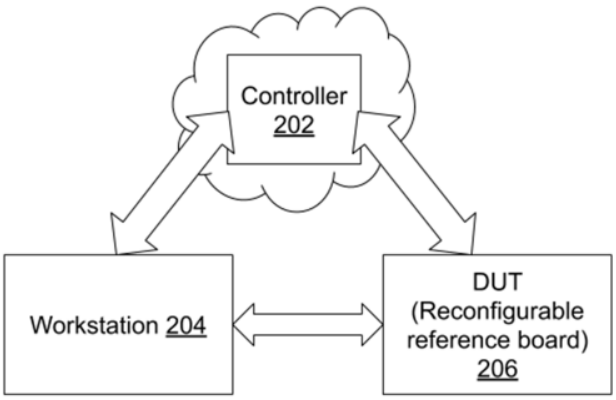

Fig. 2: Administering a test-suite to a reconfigurable reference board

Fig. 2 illustrates the administering of test-suites to a reconfigurable reference board, per the techniques of this disclosure. The reference board, or device-under-test (DUT, 206), is coupled to a workstation (204), which can be, e.g., a laptop or desktop computer. A remote, possibly cloud-based entity controller (202) is coupled to both the workstation and the reference board. The controller selects the configuration of the reference board to be tested and provides the corresponding firmware, also known as payload, to the reference board.

A software component running locally on the embedded device updates the firmware based on the payload received from the controller. The controller selects a test-suite to be run on the reference board and provides it to the workstation. The workstation exercises the test-suite on the reference board (as configured with the corresponding firmware), collects the validation results, and reports the results back to the controller. The controller parses the results and can use the parsed results to shape testing pathways.

As explained earlier, the payload downloaded to the reference board by the controller can be a firmware image that provides a certain hardware configuration. The hardware configuration specifies which of the devices physically plugged on the reference board are actually enabled, and also the modes of the devices that are enabled, e.g., the DRAM frequency, the I2C/SPI frequency, etc. The cloud-based controller selects the payload with the aim of optimizing test coverage, based on decisions made by an administrator (power user) or guided by machine-learning models.

CONCLUSION

This disclosure describes a single embedded device, a reconfigurable reference board with many different components connected to it, that can be used to configure different (shipped-out) platforms. The reference board enables rapid validation of a large number of variations of a

given reference design, including the rapid test and integration of newly introduced components. The time-to-market, lab-space, investment in test-and-measurement equipment, engineering resources, etc. required for test and validation are thus substantially reduced.