Statistical Framework and Built-In Self Speed-Binning System for Speed Binning Using On-Chip Ring Oscillators

Abstract:

This paper presents a model-fitting framework to correlate the on-chip measured ring-oscillator counts to the chip’s maximum operating speed. This learned model can be included in an auto test equipment (ATE) software to predict the chip speed for speed binning. Such a speed-binning method can avoid the use of applying any functional test and, hence, result in a third-order test time reduction with a limited portion of chips placed into a slower bin compared with the conventional functional-test binning. This paper further presents a novel built-in self-speed-binning system, which embeds the learned chip speed model with a built-in circuit such that the chip speed can be directly calculated on-chip without going through any offline ATE software, achieving a fourth-order test-time reduction compared with the conventional speed binning. The experiments were conducted based on 360 test chips of a 28-nm, 0.9 V, 1.6-GHz mobile-application system-on-chip. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

Enhancement of the project:

Existing System:

One research line in speed binning is to replace the functional tests with structural tests, which requires a cheaper ATE to shift the test data at a lower frequency and applies the test with a high-speed on-chip clock. Such research works also need to transfer the structural-test Fmax to the corresponding functional-test Fmax, which is viewed as the golden answer in industrial practice. Belete et al., Zeng et al., and Cory et al. tried to use several conventional structural tests, such as different path-delay-fault tests, different transition-fault tests, and memory built-in self-speed-binning (BISB), to correlate the chip Fmax with a linear function. Chen et al used some newly defined structural performance metrics, such as flip-flop-based Fmax, patternbased Fmax, and path-based Fmax, to correlate the chip Fmax with different data-mining techniques. Chen et al.
further used multiple structural performance metrics simultaneously to derive a multi-to-one Fmax correlation. However, to obtain the result of a representative structural test still takes significant test time and its correlation to the functional-test Fmax still cannot be guaranteed for advanced process technologies.

Another research line is to add an all-digital phase-locked loop (ADPLL) in the targeted chip as a programmable clock generator and then automatically apply the embedded built-in self-testing (BIST) patterns at different frequencies by the chip itself. In this way, the requirement of ATEs can be extremely low since no test pattern needs to be sent from the ATE. Such a concept is called built-in speed-grading and built-in delay-binning and calibration. With the advance of the ADPLL design, the embedded ADPLL can be self-calibrated and more tolerant to the process variation with an acceptable area overhead. However, the applied BIST patterns are still pseudorandom patterns, which have not been validated as an effective test to correlate the chip Fmax. Another research line is to directly on-chip measure the propagation delay of each predefined critical path and then use the longest measured delay as the chip Fmax, which reduces the need of repeatedly applying the functional or structural test at different frequencies. Raychowdhury et al. proposed a process-tolerant delay monitor to measure the delay of replica critical paths. Wang et al. proposed a path-based ring-oscillator (RO) structure to measure the delay of each selected critical path with a calibration mechanism. Datta et al. and Tsai et al. applied the vernier-delay-line (DL) circuit to measure the delay between each input–output pairs. However, the preselected critical paths or input–output pairs may not remain critical to every manufactured chip due to the increasing process variations, and hence, the measured delay may not be able to directly reflect the chip Fmax.

**Disadvantages:**

- Less speed

**Proposed System:**

Conservative model binning can avoid the use of the time-consuming functional test and significantly reduce the overall test time. However, most test time of the conservative-model binning is spent on the offline software computation at the ATE (265 μs) instead of the collection of the RO counts on the chip (20 μs).
The objective of the proposed BISB system is to build an embedded circuit to on-chip calculate the learned $F_{\text{max}}$ model such that the offline software computation at the ATE can be avoided. Even though the computation with our learned $F_{\text{max}}$ model is simple from software’s point of view, to parse the measured data in the test log and load the data to the program at ATE still require extra effort. Using a built-in circuit to calculate $F_{\text{max}}$ can reduce the hassle at ATE, and hence, further save the test time. Meanwhile, we also need to make sure that the built-in circuit for calculating our learned $F_{\text{max}}$ can be economically implemented with almost the same accuracy as the software computation.

The proposed BISB system contains two major stages:

1) The model-learning stage
2) The binning-test stage.

In the model-learning stage, we need to collect all the RO counts and identify the chip Fmax by repeatedly applying the representative functional test with a fine-grain frequency resolution (5 MHz in our case) for each sampled training chip. Those collected results are the inputs to the proposed model-fitting framework.

Architecture of BISB Circuit
Fig. 1 shows the overview of the BISB circuit, which uses JTAG to communicate with the ATE. The five signals, Address, Write_en, Read_en, WDATA, and RDATA, are used to read or write the registers in the BISB circuit, such as OPM_start, OPM_cfg[7:0], OPM_En, OPM_j_count[15:0], Performance[47:0], Weight_x[15:0], and oscil_c[15:0]. Address indicates the register to be accessed. Read_en and Write_en are the read and write enable signals, respectively. WDATA and RDATA are the data to be written into and read from the selected register, respectively. The ARM core can also communicate with the BISB circuit through the Advanced Peripheral Bus interface, such that this Fmax estimation can be dynamically requested by the SoC.

Square-Root Calculator
The square-root calculator is used to compute the lower error bound of the predicted Fmax with (9) and (10), which is the key to control the defect level while not placing too many chips into a slower bin. However, to directly compute the square root of a floating point in one clock cycle may require large area overhead (easily exceeding 100k gate equivalence). Some approximation techniques, such as Taylor series or coordinate rotation digital computer (CORDIC) algorithm, have been used in practice to approximate the square root in multiple clock cycles.

Advantages:

- High speed

Software implementation:

- Modelsim
- Xilinx ISE