Testing Arithmetic Coprocessor in System Environment

Janusz Sosnowski, Tomasz Bech
Institute of Computer Science, Warsaw University of Technology,
ul. Nowowiejska 15/19, Warsaw 00-665, Poland, Email: jss@ii.pw.edu.pl

Arithmetic coprocessors (AC) are quite complex circuits and testing them is an important and not easy problem (not covered in the literature). Analyzing diagnostic software for IBM PCs we found that the testing procedures for ACs are limited to simple basic checks. Hence we decided to develop efficient test procedures in a systematic way. They are executed on the main processor and generate appropriate stimuli to AC functional blocks (e.g. instruction sequencer, data path units) and verify test responses. An important contribution of this paper is the integration of various approaches to testing and increased test observability of test results assured by on-chip event monitors and system exceptions.

Three approaches to testing has been considered: deterministic, pseudorandom and application driven. Deterministic tests are oriented to specific known fault classes adjusted to different functional blocks. Here we base on structural and behavioral description. The structural approach relates mostly to data processing blocks assuming their iterative structure (exhaustive tests for basic cells is assured). For less regular instruction sequencer higher level testing is performed. At the beginning a few selected basic instructions (test kernel) are checked exhaustively and then they are used to test the remaining instructions. Each AC instruction is verified in the following scheme: AC registers are initialized to some specific states then the instruction is executed and AC state (all registers) is verified. To detect the specified faults we use appropriate test patterns (modified m-out-of-n codes) which have to be loaded to the coprocessor registers. For each tested instruction we have to assure that the generated result is different from the correct one if any fault from the specified list appears. The considered faults relate to missing or excessive microoperations.

The main drawback of the deterministic test is that it is targeted to detect the assumed fault model which may be too simplified and sometimes not realistic. Moreover there are problems with defining the most representative fault models (closely related to logical and physical structure of the processor). So a good solution is to add pseudorandom tests (PSR). We have developed such supplementary tests for most blocks. They increase overall error coverage, however may not be effective for some fault classes covered with deterministic tests.

The developed deterministic and PSR test procedures are targeted to specified fault classes or functional blocks. Some weak point of this approach is that the applied instruction sequences may differ from typical sequences used in real programs. So not considered complex faults related to pattern sensitivity, block interaction etc. may not be sensitized. Hence we decided to add testing procedures based on typical application programs. For this purpose a benchmark of AC application program procedures has been selected.

An important issue is to monitor test execution and make test results available to the user. In our case all test procedures are performed in Windows environment. Each procedure produces messages which are displayed or stored in a file. This process can be disturbed by faults so to increase test dependability we use various error detection mechanism embedded into the system (exceptions) as well as some event monitoring hardware incorporated in new microprocessors. Typical system exceptions are: memory access violation, data misalignment, illegal instructions, overflow, underflow, divide by zero, stack faults etc. The monitored events may include: clock pulses, data read/write cycles, cache misses, taken branches, pipeline flushes etc. Some experiments proved high stability of counted events (0.1-2%), so they can be considered as an additional test signature.

The effectiveness of the developed test procedures has been verified directly in experiments with fault insertion (for some typical fault classes) and indirectly by checking various test controllability and observability measures (e.g. distribution of test stimuli applied to individual functional blocks, activity and interaction factors for various blocks, distribution of register states). System exceptions appeared in 30-60% of inserted faults. The developed test procedures assume correct operation of the main processor (20-30-% of injected faults in this processor did not influence the test result). Similar analysis applied to some commercial tests for ACs showed their significant drawbacks. The developed approach to testing is dedicated to preventive and maintenance tests. Nevertheless it is also useful in chip production as well as in certification of new AC structures.. It showed some accuracy inconsistency of K6 processor with IEEE 754 standard (for tan/arctan function).