# IP Core of Statistical Test Suite of FIPS 140-2

*by Akio Hasegawa, Song-Ju Kim, Communications Research Laboratory[1)]*
*and Ken Umeno, Communications Research Laboratory[1)] and ChaosWare, Inc.[2)]*
*[1)], [2)] 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan*

**Abstract :**

In this paper, we propose IP core of statistical test suite included in FIPS PUB 140-2 documentation which is published by NIST. This module can be used for self tests of encryption module. For checking our implementation, we implement this module on Xilinx Virtex II FPGA and compare results given by FPGA and PC. We have not found error.

**INTRODUCTION**

Recently, security is one of the most important□@technology in the field of Internet, wireless network, and mobile communication.□@Many IPs of encryption algorithms (for example DES, AES, RSA and so on) are developed.□@NIST (National Institute of Standards and Technology) published □gSecurity Requirements for Cryptographic Modules (FIPS PUB 140-2) (In the latest version of FIPS PUB 140-2 documentation (Dec. 3, 2002), this requirement was deleted).□h This documentation required implementing statistical random number generator tests module as one of the self tests on high level security module. Therefore, for implementing high security module, statistical test module was needed. However, there are not so much test module of these encryption core. In this paper, we propose an IP core of statistical random number generator tests core of FIPS PUB 140-2 on FPGA. This IP core is used with any encryption algorithm cores.

**STATISTICAL TEST SUITE OF FIPS PUB 140-2**

In the documentation of FIPS PUB 140-2, statistical random number generator tests are defined as follows: If statistical random number generator tests are required, a cryptographic module employing RNGs (Random Number Generators) shall perform the following statistical tests for randomness. A single bit stream of 20,000 consecutive bits of output from RNG shall be subjected to mono bit test, poker test, runs test and long runs test.

**The mono bit test**

1. Count the number of ones in the 20,000 bit stream. Denote this quantity by X .

2. The test is passed if $9{,}725 < X < 10{,}275$ .

**The poker test**

1. Divide the 20,000 bit stream into 5,000 consecutive 4 bit segments. Count and store the number of occurrences of the 16 possible 4 bit values. Denote f(i) as the number of each 4 bit value i, where .

2. Evaluate the following:

3. The test is passed if $2.16 < X < 46.17$.

**The runs test**

1. A run is defined as a maximal sequence of consecutive bits of either all ones or all zeros that is part of the 20,000 bit stream. The incidences of runs of all lengths in the sample stream should be counted and stored.

2. The test is passed if the runs that occur are each within the corresponding interval specified in the table 1.□@This must hold for both the zeros and ones.

**The long runs test**

1. A long run is defined to be a run of length 26 or more.

2. On the sample of 20,000 bits, the test is passed if there are no long runs.

In the poker test, real numbers calculation is needed for calculating equation (1). However, a real numbers calculator is more complex than an integer numbers calculator. We replace the criteria of the poker test by the following.

**The poker test (integer version)**

1. Count and store the number of occurrences of the 16 possible 4 bit values into f(i) .

2. Calculate the following:

3. The test is passed if 10,800 < X' < 230,850 otherwise f(i) > 1,256, the test is fail. Because if one of f(i) is over 1,256, X' is beyond 230,850. The second criterion is given for using small size (11x11) multiplier.

We implement these four tests, the mono bit test, the poker test (integer version), the runs test and the long runs test on FPGA.

| Length of run | Required interval |
|---|---|
| 1 | 2,315 ~ 2,685 |
| 2 | 1,114 ~ 1,386 |
| 3 | 527 ~ 723 |
| 4 | 240 ~ 384 |
| 5 | 103 ~ 209 |
| 6+ | 103 ~ 209 |

*Table 1 : Required intervals for length of runs test.*
*Runs of greater than 6 are considered to be of length 6.*

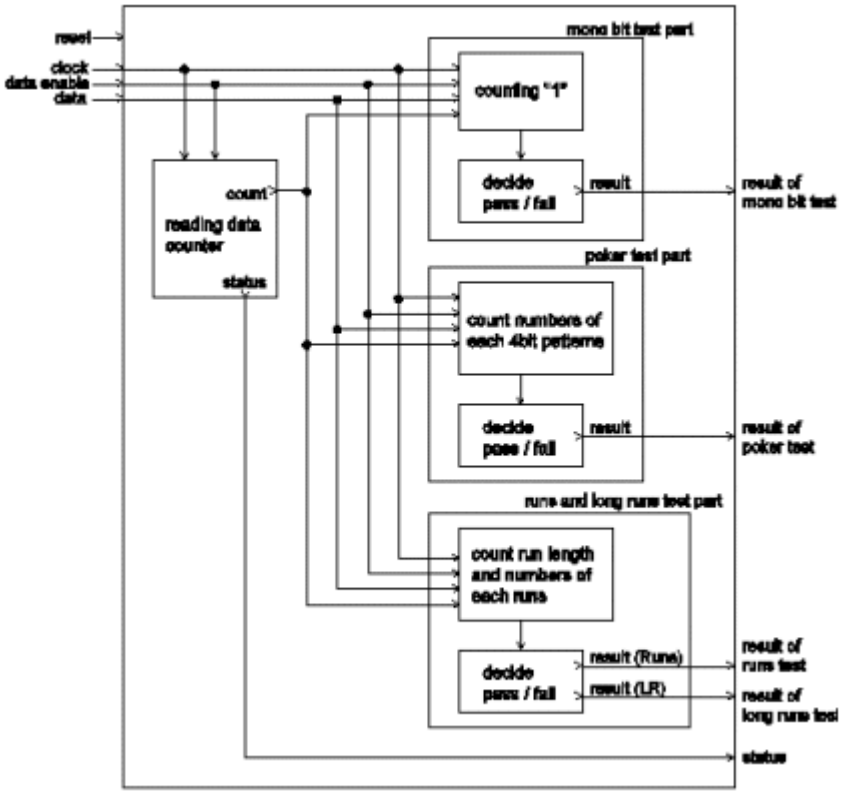


*Figure 1 : Block diagram of statistical test module of FIPS PUB 140-2.*

**FPGA IMPLEMENTATION**

This module has four inputs (clock, reset, data enable, and data signal) and five outputs (status and results of each test) as shown in Fig. 1. Sending 20,000 bit sequences with clock signal and data enable signal, this module outputs results of each test (pass or fail). This module consists of four parts, counting 20,000 bit part, mono bit test part, poker test part, and runs and longest runs test part. The mono bit test part counts □e1□f in the 20,000 bit stream and detects whether satisfying above criteria or not. Therefore, this part consists counter (for counting □e1□f) and comparators. Figure 2 shows the block diagram of the pattern count part for the poker test. This part consists three parts, storing 4 bit data part, timing trigger generator for checking pattern, and pattern checking and counting part. When the output of the trigger generator is ON, if i-th 4 bit pattern occurs, this part increments f(i) . After checking 5,000 patterns, this part outputs f(i) . Figure 3 shows a block diagram of run length counter for the runs test and the long runs test. In this figure, the outputs of data selector are as follows: First, when the input is the first bit of the bit stream, the outputs are 2 bits same as the input bit. Next, reading from 2nd to 20,000th bit, the output is the same bit. Finally, after reading 20,000 bits, this part outputs 1 bit which is inverse of 20,000th bit for informing end of run to detector. The outputs of data selector are stored 2 bit shift register. If 2 bits stored in this register, then run is continued. Thus, increments run length counter in the run detector part. On the other hand, these 2 bits are different, the run is finished. This part outputs the length of run. In the following part, the occurrence of each length of runs and the length of longest run are stored. After reading 20,000 bits, the detector checks these stored data and decides whether pass or fail. Figure 4 is a timing chart of this statistical test module. When status output is □e1□f, tests are not finished. All tests are finished, status output turns □e0□f and the result of each test appears. When the test is passed, the output becomes □e0□f and □e1□f means fail.

We implemented this module on Xilinx Virtex II FPGA using ISE 4.2 design tool. Source codes are written by VHDL. The implementation results are shown in Table 2. Moreover, we check our implementation using FPGA test module produced by Hunt Engineering as shown in fig. 5. This test module connected with PC through the PCI-bus. We send random bit sequence from PC and compare the results of each test by FPGA and PC. We do not find the difference between these results.

**USING THIS MODULE**

Figure 6 shows an example of using statistical test suite module. For testing, data_enable signal and data signal from encryption module are sent to statistical test suite module. This module outputs four result signals of each test. However, if you want to know whether □gpass□h or □gfail□h, input these signals to OR gate and check the output of OR gate. Finally, if test is failed, user module tells to the operator.

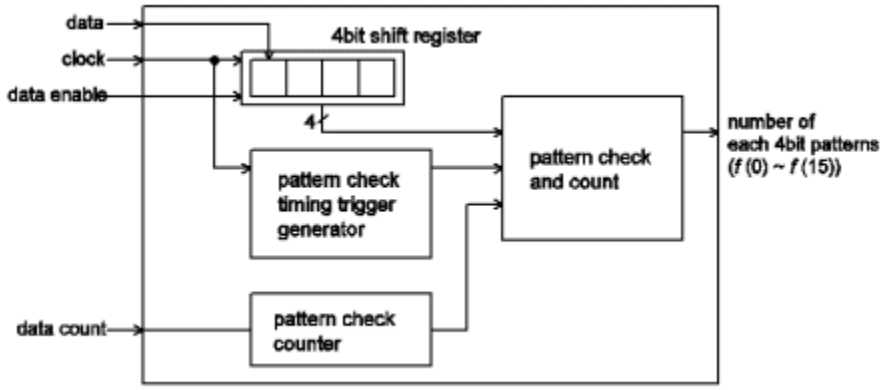


*Figure 2 : Block diagram of pattern count part for the poker test.*

**CONCLUSIONS**

In this paper, we introduce an IP core of statistical test suite of FIPS PUB 140-2 documentation on FPGA. This documentation requires implementing in high security module as one of the self tests. This statistical test suite includes four tests, the mono bit test, the poker test, the runs test, and the long run test. The poker test includes real number calculation, but implementing real number calculator on FPGA is complicated. Thus we develop the integer version of the poker test and implement it instead of original version. We compare the results of tests on FPGA and PC, we do not find error. High security module needs self tests and this IP core is used with any encryption algorithm cores. Thus high security module can be developed by encryption core and this IP core.

**REFERENCE**

[1] Security Requirements for Cryptographic Modules, FIPS PUB 140-2, NIST (2001), **http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf**.

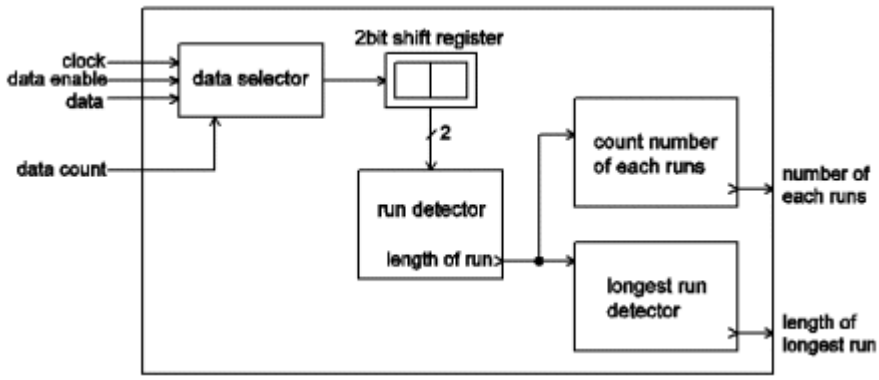


*Figure 3 : Block diagram of run length counter for the runs test and the long runs test.*

| | |
|---|---|
| Gate Size | 37k gates |
| Maximum Frequency | 113 MHz |

*Table 2 : Implementation results.*

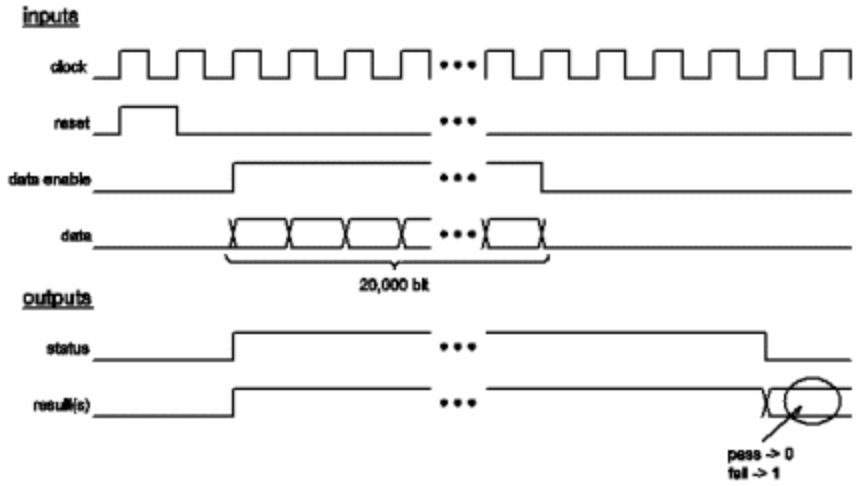


*Figure 4 : Timing chart of statistical test module.*

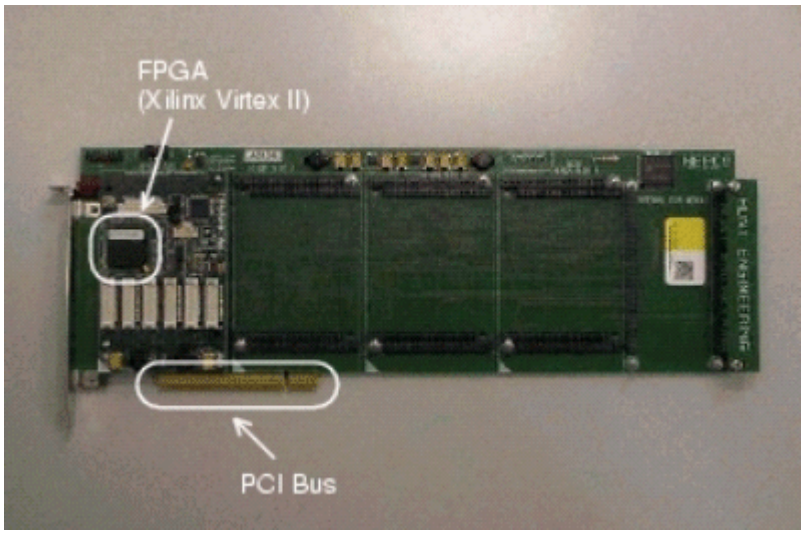


*Figure 5 : FPGA test module.*

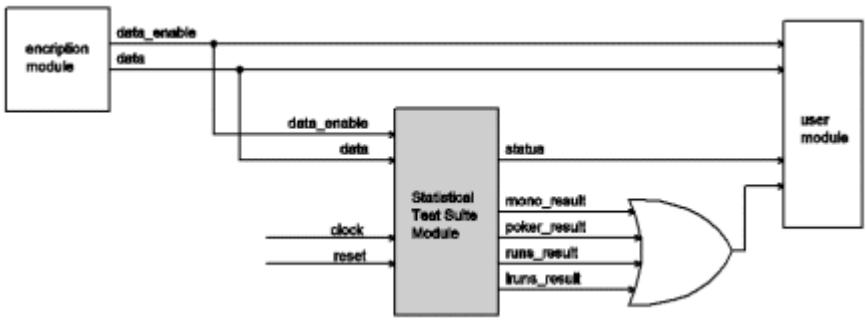


*Figure 6 : An example of using statistical test suite module.*