

The ZK-Crypt Noise Generator Design Parameter Emulator

A GUIDE TO OPERATING THE ZK-CRYPT QUADRUPLE OUTPUT RANDOM AND DETERMINISTIC NOISE GENERATOR EMULATOR FOR ESTABLISHING FINAL DESIGN CONFIGURATIONS FOR BIS AIS 31 COMPATIBILITY AT FLEXIBLE ACCELERATED OUTPUT RATES
-OPERATES UNDER THE XP & VISTA OPERATING SYSTEMS with .PDF READERS

THE RANDOM AND DETERMINISTIC NOISE SOURCE FOR THE-

ZK-CRYPT-THE 8K GATE SYMMETRIC PERIPHERAL FOR BEST OF BREED

SINGLE STEP 32 BIT STREAM CIPHERING
WITH PAGE SYNCHRONIZATION

DUAL TRACK FEEDBACK HASH/MAC AUTHENTICATION
WITH THE MAC MIX ANTI-PREIMAGE PERMUTATION

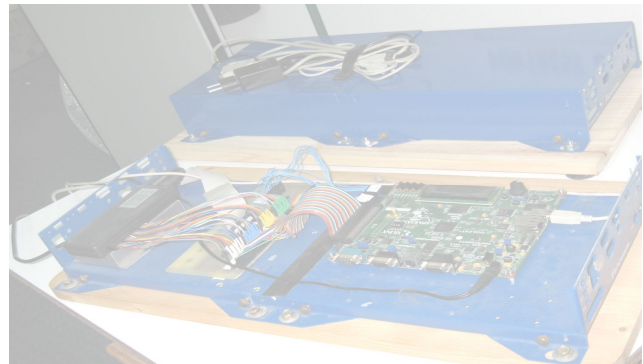
AIS 31 COMPATIBLE TRUE RANDOM NUMBER GENERATION (TRNG)
WITH A RANDOM FREQUENCY MODULATED CLOCK
AND ON-LINE ENTROPY MONITORING

all with LOW POWER, 32 BIT SINGLE STEP HIGH DIFFUSION
3 GIGA BITS/SECOND at 100 MHz OPERATION

For new NIST Guidelines-Simultaneous Authenticating and Decrypting Data
"JUXTAPOSING" ZK one DECIPHERS; ZK two COMPRESSES (HASHES)
ONE CIPHERTEXT INPUT – ONE CLEARTXT OUTPUT – ONE AUTHENTICATING TAG

For NIST Guidelines-Flexible Security and Key Lengths
"TWINNED -2S- FEEDBACK" - ZK one SWAPS FEEDBACK WITH ZK two
DOUBLE WORD DOUBLE SPEED "INFINITELY" STRONGER
CONCATENATED ENGINES FOR LIGHTNING SPEEDS

NOISE SOURCE SUBMITTERS:
CARMİ GRESSEL
AVI HECHT
MICHAEL RIVKIN
RAN GRANOT



COMPLETE SOFTWARE EMULATOR INSTALLATION INSTRUCTIONS ON PAGE 4

PC – FPGA DEMONSTRATOR AVAILABLE

JANUARY 2008
PRELIMINARY

THIS DOCUMENT AND THE NOISE EMULATOR & DEMONSTRATOR
ARE FOR LIMITED DISTRIBUTION TO DESIGNATED POTENTIAL
CLIENTS AND INSTITUTIONAL EVALUATION ONLY.

Contents

- Welcome to the ZK-Crypt AIS 31 Random Noise Emulator/Designer
- Installing the ZK-Crypt Noise Emulator
- Starting Up – Seeing it Whiz!
- Circling the Application Screen
- Instructions for Running the Emulator
 - Standard Mode – Dual Clock
 - Wander Mode – Dual Clock
 - Deterministic Mode – for Stream Cipher, Hash and ETSI Compatible TRNG
- Timing Diagrams & Flow Charts
- References and Contacting Info
- Appendix - Analytic Test Files

Welcome to the ZK-Crypt AIS 31 Random Noise Emulator/Designer

This in-house emulation was used to prove the concepts of the original noise source patent specs to be used for a widest range of Host provided (Primary Clock) frequencies. The emulator statistics are remarkably similar to the FPGA silicon generated statistics both in deterministic and in the true physical random modes.

This ZK-Crypt patented Noise Source is an all-digital high-speed component designed to be easily implemented on any logic circuit that can benefit from up to 3 unpredictable binary and one randomly missing clock signal generators operative at any host frequency.

The circuit was designed to be coupled with the ZK-Crypt Deterministic Random Number Core, which more than meets the BIS AIS 20 DRNG standard, as the ZK-Crypt DRNG is normally configured as (candidates for the next generation) a fast, robust, highest diffusion Stream Cipher or a dual orthogonal feedback track Hash or HMAC.

As this generator is typically hundreds or thousands of times faster than competing designs, the autonomous oscillator should always be in the free running mode, "pumping in" entropy into the ZK-Crypt engine, which is possibly the finest deterministic (aka pseudo) random number generator. While the oscillator is operating, the Host samples the 24 bit counter, which at each Primary Clock counts the number of positive edges of the autonomous oscillator. The first count which differs from a previous count proves that that there is a "moving" phase difference between the host oscillator and the autonomous oscillator output signal, **fr**. This corresponds to the AIS 31 initial test [schind]- ("is the device working?"). We say that not only is it working, but in all probability working very well.

As the free running noise generator continues working, the Host samples 320 bit output strings from the noise generator. Presumably the test strings are more than satisfactory (a Demerit grade of less than 50) and has not failed (a Demerit grade of more than 65). We are assured, in the ZK-Crypt engine environment, that the oscillator will not be stable, as desired, as the module's input voltage will randomly wobble between a higher and a lower average level, relating to the random clocking of the ZK-Crypt's Register Bank.

All of the analytical testing in the included demo can be performed in the same 8K gate hardware version anticipated in Smart Cards, Main Frame and mobile phone processors.

The ZK-Crypt RNG unpredictability is based on two free running oscillators/clocks; a Host provided Primary Clock, which may or may not be stable, and an autonomous FM modulated oscillator, whose frequency is proved to wobble and wander, [ppats], [wpat].

As opposed to the AIS 31 suggested method of assuring entropy, when switched on, the free running ZK-Crypt noise generator feeds untested entropy into the free running DRNG post processor. The post processor generates thousands of un-sampled 32 bit random words, before the Host completes the first proof that the random oscillator is generating a wobbling frequency (a varying phase difference between the Host Primary Clock and the autonomous oscillator). The number of sampled free running posedge autonomous clock pulses in sampled successive free running Primary Clock periods differs from one Primary Clock period to a following period. This corresponds to the AIS 31 proof that that we can assume that the noise engine is operating. In the last section of each Test Analyses, we record the changing



measured frequencies of the first 240 random oscillator clock cycles, and also the number of oscillator pulses in each of the first 240 Primary Clocks. During this pretesting we know but do not assume that the noise processor has randomized the 404 variables of the ZK-Crypt in free running RNG mode.

The second more rigorous proof, as suggested by [Schind] is to ascertain the random nibble distribution (unpredictability) in sampled 320 bit noise strings. The designer can adjust the number of string tests for any level of confidence. The designer can, if he chooses, load random or deterministic Messages into the DRNG, which would serve as a second uncorrelated noise source to the dual track feedback DRNG. (Not relevant to these tests.)

The FortressGB FPGA implementation accurately emulates both the physical and deterministic modes of the noise circuit in the ZK-Crypt engine. Using the emulator, a designer can establish safe oscillator and delay parameters for the entire range of host frequencies and the worst cases of potential oscillator duty cycles. In the final phase of proof of concept, the designer can emulate stress conditions possibly caused by over-voltage, fluctuating ambient temperature, and/or ageing degradation of semiconductor circuitry.

Most first cut design tests should be repeatable in order to focus on a design compliant with a semiconductor process. There is no need to store first cut design analyses, as a well organized **LOG FILE** stores in the filename both a summary of repeatable results of each test, and also a summary of the **DEMERIT** marks of all tests in the given **LOG FILE**. Each analysis test filename in the **LOG FILE** includes all design parameters, and a summary of the results. In general, repeatable tests, without automatic acceleration will generate stable low **DEMERIT** averages, except when because of numerical stability of variables, the emulator "seeps into" a closed high **DEMERIT** loop. In such a case, typically, as a single warning **DEMERIT** is larger than 50, the **AUTOMATIC FM ALARM ACCELERATION** switches into a higher average frequency, generally before a single failing **DEMERIT** mark has been registered.

For exhaustive testing, in both the Single and Dual Clock modes the engine must be fed external entropy, to promise maximum coverage of all possible conditions; as Marsaglia [DieH] tells us, "things happen" in random number generation. Using the Dual Clock Wander Mode on long sampling files, e.g., 200M, with varied trigger parameters, gives a nice sweep of what may happen, in practice.

Each test measures the differentials of internal and external nodes. In Dual Clock mode we first prove that there is a moving phase difference between the Host (Primary) Clock and the FM modulated autonomous Ring Oscillator. For this the Host samples the counter that registers the changing number of **fr** rising pulses in each Primary Clock period. This measure is tantamount to the AIS 31 test to prove that the device is "turned on" and operative.

The analytical output includes generation of "on-line" 4 different nibble **DEMERIT** statistics as suggested by Werner Schindler [Schind]. The noise generator outputs what has proved to be three unpredictable at worst, loosely correlated binary outputs at every clock cycle. We make the **DEMERIT** test on each of the binary string outputs, and also on the concatenation of the three outputs. The designer will see that typically, all four tests, including the concatenation, generate similar range sequences.

Our physical and emulated tests typically attain an average **DEMERIT** mark less than 20 for at least orders of magnitudes of Random Oscillator/Host frequency (**R**) configurations.

All of our tests relate to the on silicon Worst Case DMA loading test, wherein a Warning (**DEMERIT**) Trigger changes the frequency at any instance of "bad things to come", which is detected (by the Host) thousands of Primary Clock cycles after the "happening". We sample all signals at every Primary Clock period. In Automatic FM Acceleration mode, the Host shunts the FM Delay circuit, as soon as a Warning Trigger is sensed, practically in all circumstances before a **DEMERIT** above 65 occurs.

In the Dual Clock Automatic FM Alarm Accelerate mode, the device driver pseudo-randomly changes the modulated random frequencies, by shunting sections of the FM Ring Oscillator; typically, whenever a **DEMERIT** Trigger of over 50 (user definable) occurs on a single measured sequence of 320 bits. If delay parameters are completely stable (deterministic), there always is a danger that the pseudo-random output streams will be "stuck" in a 0.5 to 50K bit repeating stream sequences, with repeated **DEMERIT**s of larger than 50 (but typically less than 65 **DEMERIT**s). In such a "stuck on sequence" the Automatic Mode driver accelerates the chosen average frequency.



To compensate for ageing, unexpected changes of Host sampling frequencies, inordinate duty cycles, etc. we have included the varied means for the designer to pretest expected aberrations.

In Dual Clock mode, the unpredictable phase differences between the ring oscillator signal, **fr**, and the Host supplied Primary Clock signal provide the source of entropy. The Dual Clock generator is completely independent of the post-processing Deterministic Random Number Generator.

In the deterministic mode, the source of unpredictability is the toggled least significant bit of a pseudo-random data dependent programmable counter, Q_{TA} . The LS counter signal generates equiprobable odd and even toggled sequences of cycle lengths 4 to 15. To give the designer confidence and to show the benefits of data dependent entropy in deterministic applications; the designer may optionally disable the emulator QTA Debiasser signal. (This would not be a compliant ZK-Crypt design, but would still be subject to FortressGB pending patents.) Normally, the Single Clock will be checked, worst case, in deterministic mode, where, occasionally, periodic sequences appear, especially in the concatenated Nibble tests. In a Single Clock sequence, any "Stuck On" syndrome should be short lived. This is less likely in the more practical test, with virtually uncorrelated Q_{TA} toggling. This will enable ETSI Compatible Single Clock TRNG generation initializing with a 256 bit stored "last load generated kernel".

Every occurrence of a warning **DEMERIT** is stored in a BAD file. In deterministic sequences, (without a true random Wandering frequency or without a Host change of the FM modulation) the BAD files can be alarmingly large, as they record every **DEMERIT** happening. We have yet to find an occurrence of more than a few failures in hundreds of millions of tests where **R**, the oscillator/Host frequency ratio was more than 0.5, the Host actuated either frequency shunts or the random Wander function with a 1/1000 inc/decrease of frequency on the average of once in six sampling steps.

[Installing the ZK-Crypt Noise Emulator](#)

For successful operation you will need a PC with an XP or Vista Operating System.

To assure readable and printable results via the Microsoft Notepad we suggest installing the following font to be able to print readable LOG FILES, Bad Files and Test Analyses -

[Start](#) → [Program](#) → [Accessories](#) → [Notepad](#) → [Format](#) → [Font](#) → [Courier New #8 font](#);

or alternately, when [Browsing](#) → [Format](#) → [Font](#) → [Courier New #8 font](#).

Insert the ZK-Crypt CD ROM Noise Source Emulator Disk in the CD ROM reader;

Copy the AIS_31 Folder into the Root Folder of a Hard Drive which has spare memory;

In the subfolder [Release](#) right click on the "AIS_31_NOISE_SOURCE.exe" Icon; and, Click "Send To" → "Desktop (Create Shortcut); to put an Icon on the Desktop.

Proceed to [Starting up](#).

[Starting up](#)

Now you are ready to start. Clicking on the "AIS_31_NOISE_SOURCE" Icon on the Desktop, sets you up immediately into the (default) Standard Dual Clock mode of operation, with a new optional true random 32 bit "initialization key". You can choose to initialize with this newly generated random key; one of the 16 standard random keys (each in the standard choice list identifiable by the first Hex digit, 0 to F) accessible from the key window, or any 8 Hex digit key of a typed in override. Click the "[RUN AIS 31 TESTS](#)" Command Bar. The [STATUS PROCESS](#) changes from [READY](#) to [TESTING](#). Meanwhile, a Message Box will ask if you want to start a new [LOG FILE](#). We suggest keeping a series of similar tests in each of your [LOG FILES](#); so that your first answers will probably be "NO". ZK-Crypt will warn when you have more than 45 tests in a single [LOG FILE](#). The prefix of a Test Analysis filename, tells you that you are in one of the automatically prefixed 676 sequential [LOG FILES](#), and one of up to 52 Test Analysis files in that particular [LOG FILE](#);

e.g., the Test Analysis cbB file prefix signifies the $cb = 3 \times 52 + 2 \rightarrow$ the 158'th generated [LOG FILE](#), where $B = 26 + 2 \rightarrow$ This is the 28' th Test Analysis file is in the cb'th [LOG FILE](#).

Each file name includes the time and date of the test, and user accessible parameters.



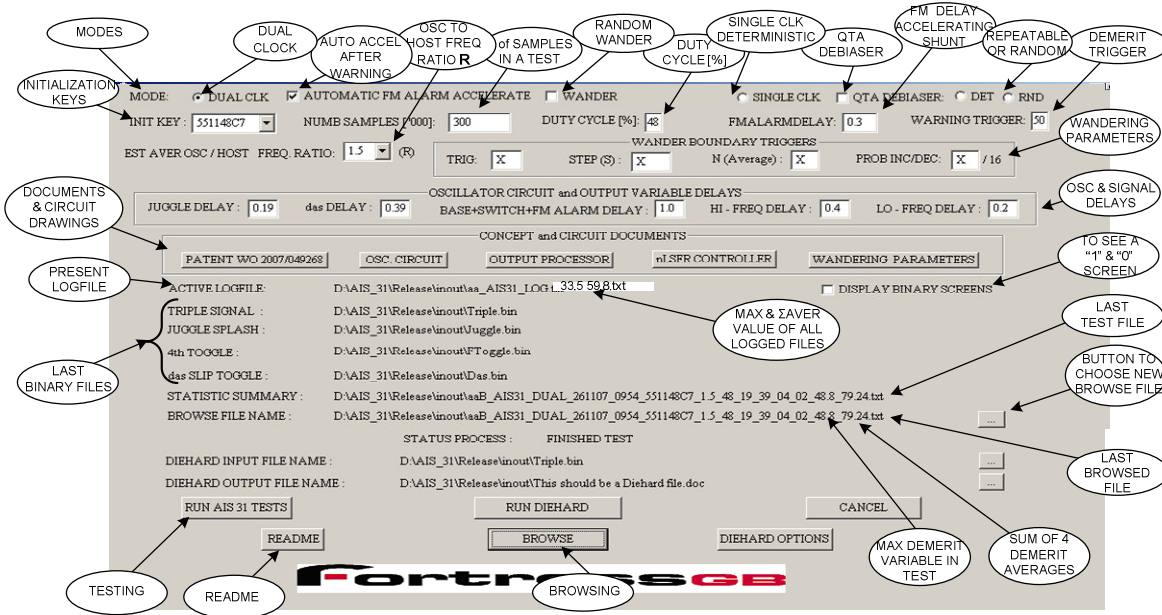
The LOG FILE filename suffix; e.g., 33.5 59.8 tells you that the maximum DEMERIT value in all of the Test Analysis files was 33.5 and the cumulative DEMERIT average of all of the tests was $59.8/4 = 14.9$.

Simultaneously, a BAD FILE log will be generated if at least one Warning DEMERIT Trigger value is sensed. All occurrences of a DEMERIT Trigger (default 50) 320 bit test are recorded with the number of the sensed Group (and the values of the following 10 test groups when in [AUTOMATIC FM ALARM ACCELERATOR](#)). The file also tells you the maximum and minimum relative oscillator period (delays) resulting from the [AUTOMATIC FM ALARM ACCELERATOR](#) Host driven shunt algorithm.

When a [TEST IS FINISHED](#) a Message Box will pop-up; click [OK](#). Now click [BROWSE](#), to see the detailed summary of the last test. In the Appendix we include annotated Analytical Test Summary files for your reference.

Circling the Application Screen:

When you Clicked the "AIS_31_NOISE GENERATOR" Icon a "tell-all" application screen pops up with default settings for a Dual Clock mode few second test generation. On the following pages we review the application settings.



The Application Screen

Modes of operation:

Dual Clock for four string True Random Number generation:

Automatically new average random frequencies are generated by Shunting Sections of the Ring Oscillator. If the **Automatic FM Alarm Accelerate** is selected; at every group occurrence of a Warning **DEMERIT** a new accelerated frequency combination is selected.

Optional Non-Repeatable **Wander** "wobbling" frequency Inc/Decrements using the **Wander Boundary TRIGGER Parameters**. If Wander is On, analysis results will not be repeatable.

Single Clock for generating 4 binary strings of data dependent typically pseudo-random numbers.

The default **QTA Debiasser** should always be selected- (the designer can optionally turn the **QTA Debiasser** off to demonstrate the need for well balanced noise signals). The repeatable (**DETerministic**) test setting may be best for first cut emulations. The **RND** setting more closely emulates real world processing with pseudo-random data feedback from the ZK-Crypt 32 Bit Word Manipulator.

Estimated Average Oscillator to Host Frequency Ratio, (R). To check reliability for all application ratios.

Typical modules will be designed to operate with a wide range of Host provided Primary Clock frequencies.

Number, #, of **Samplings** in an individual Analytical Test in Thousands. 300 K usually suffices for the first cut. AIS 31 [Schind]. For small R, a test will take 5 – 10 seconds. For DieHard tests, you may need more than 300 M samples.

Duty Cycle [%]. **fr Duty Cycles** ranging from 45% to 55% should not adversely affect results. Check before deploying a design.

FM Alarm Accelerating Delay. A shuntable section of inverters operative to increase frequency when triggered by a Warning **DEMERIT** in **AUTOMATIC FM ALARM ACCELERATE** mode.

DEMERIT Warning Trigger. The user (default 50) definable trigger which when exceeded calls for a new

combination of operative delays in the fr Autonomous Oscillator when in [AUTOMATIC FM ALARM ACCELERATE](#) mode. This mechanism assures that the Noise Generator is not stuck on a high [DEMERIT](#) sequence syndrome. Typically, [WANDER](#) mode precludes lengthy closed loop sequence syndromes. Typical Hi-[DEMERIT](#) syndromes in set frequency modes are in the range of 51-53. Click the [WANDERING PARAMETERS](#) document bar for a graphic explanation of the functions.

Boundary [Wander TRIG](#)gers. The trigger, [TRIG](#) (default 0.15), factor determines the upper and lower boundary value that changes to or from Incrementing or Decrementing of the "saw tooth" Wandering Frequency. The maximum Boundary frequency is the Base Frequency x (1 + [TRIG](#)). The initial minimum frequency, MINF is the condition of the ring oscillator wherein no section is shunted and the propagation delay is the original assumed propagation delay of the components. The minimum Boundary frequency is therefore MINF x (1 - [TRIG](#)), where we assume that the propagation delays are maximum value.

[Step \(S\)](#) is the relative part of the average [fr](#) frequency which randomly with a [Probability INCs/DECs](#) (of default 8/16) increases or decreases the average oscillator and output circuit delays. The step may occur every [N\(Aver\)](#) sample, which is the average difference in "point in time" between two possible steps. Assume [N](#) = 3 and that the frequency changes are incremental. On an average of one in 3 samples, with a probability of about one in 16/8 x [N](#) distanced samples, an increase of [Step \(S\)](#) frequency will occur.

The Relative [Oscillator & Signal Delays](#): Click the [Oscillator Circuit & Output Processor](#) Command bars. [Juggle Delay](#) is Delay which precedes the F3 Flip Flop in the [Output Processor](#). [das Delay](#) is the (default) larger Delay which precedes the F4 Flip Flop in the [Output Processor](#). The total [Base+Switrch+FM Alarm Delay](#) is not a variable and is equal to 1. In any acceleration the [FM Alarm Accelerating Delay](#) part of the [Base+Switrch+FM Alarm Delay](#) is shunted. Subsequent changes are dictated by the Host. In normal non-accelerated sequences, the [Hi-Freq Delay](#) circuit is randomly shunted about 35% of any long interval; and the [Lo-Freq Delay](#) circuit is shunted about 75% of such intervals.

To [See a "1"s & "0"s](#) scrollable [Screen](#). drawn from the last concatenated signal stream file, Triple.bin. Before running the test, click the [DISPLAY BINARY SCREENS](#). After testing, put the Triple.bin file in the [BROWSE FILE NAME](#). In the Message Window, click binary. It is valuable for granting initial confidence. Note that the distribution of lengths of literals is generally not ideal; e.g., single literals ("1"s of "0"s) should constitute about one half of the distribution.

The [Button to Choose a New BROWSE File](#)- from the [inout](#) folder in the [Release](#) subfolder. You can choose a [LOG FILE](#), an Analytic test file, a [BAD File](#) or the last [Triple.bin](#) file to be displayed for your perusal on the [BROWSE FILENAME](#) line.

After a test, the [Last Browsable Test Filename](#) appears after the [Statistic Summary](#) line. The last 4 digit decimal is the sum of the four [DEMERIT](#) marks. (Divide by 4, for average [DEMERIT](#).) The previous 3 digit decimal is the Maximum [DEMERIT](#) appearing in any of the four test results.

Clicking the [BROWSE](#) Command Bar immediately after a test get's you the last complete analyses, or to the one of your choice using the [Button to Choose a New BROWSE File](#).

Clicking the [README](#) Command Bar let's you read this document.

Clicking the [RUN AIS 31 TESTS](#) Command Bar calls for a new test, with the testing parameters seen on the screen.

The [LAST BINARY FILES](#) are temporary files which were recorded during the analysis. These files can be reconstructed for any deterministic function, using the parameters that appear in the filename.

The [TRIPLE SIGNAL \(Triple.bin\)](#) is the on-line 3 bit concatenation of the sampled bits of the: [JUGGLE SPLASH \(Juggle.bin\)](#); [4TH TOGGLE \(FToggle.bin\)](#); and, the [DAS SLIP TOGGLE \(Das.bin\)](#) files.

The [DOCUMENT & CIRCUIT DRAWINGS](#) contains four buttons which open the: [WW PATENT WO 2007/049268](#) which explains (and protects) the basic FortressGB Noise Generator; Note that much of the patentable innovation or claims on innovation may not appear in this application, and may appear in continuation patents, you can also choose a suggested AIS 31 compatible testing mode [Schind];

The FM Ring [OSCillator CIRCUIT](#);
The [OUTPUT PROCESSOR](#) which inputs the **fr** autonomous oscillating signals from the Oscillator, and the Control signals from the [nLFSR CONTROLLER](#) to output the three unbiased signals and the Missing clock ([P](#))[Random Clock](#) to the Deterministic post processing Random Number Generator;
The control unit based on two non-linear feedback shift registers, the [nLFSR CONTROLLER](#); and ,
The [WANDERING PARAMETERS](#) document graphically explains the options for getting real random **fr** frequency generation. We suggest frequent changes with small **N**, e.g., 1, 2 or 3 and small frequency Inc/Dec, e.g [Steps](#) and small [TRIG](#)ger Boundaries, e.g. 0.001 for final designs. (And, obviously exercise 300M samples- which may take many minutes. Generally, disregard the Task Manager which may announce "ZK-Crypt AIS Not Responding", while the program is operative, and "Running".

The [Initialization Key](#) list includes:

- A listing of 16 standard testing random 32 bit binary values designed for repeatable testing, each with a first Hexadecimal prefix- 0 to F for easy identification; and in the "window",
- A new random number Initialization Key generated each time the AIS_31 ...Icon is activated. Note that at each Icon activation, all parameters (the newly generated 32 bit Random Key) are set to the same programmed default values.

The designer can type in a new Key, or change any of the standard or the random initialized keys in the key window. The Initialization Key randomly assures that all binary variables of the Noise Generator will be initialized uniquely to one of 2^{32} settings after the first 100 Primary Clock steps for a given set of deterministic initialization parameters. Note that the Initialization Key used in each test is recorded in the file analysis name, to allow for repeatable testing in deterministic mode. This is sensible for extensive testing of problematic parameters found in short length tests, e.g. 300K, first cut analyses.

Click [README](#) (for this document), or click any of the [CONCEPT & CIRCUIT](#) diagrams to understand the relevance of your configuration changes.

The three modes of simulation:

- 1) The Standard AIS 31 TRNG (True Random Number Generator) mode generates repeatable pseudo and unrepeatable random controlled FM oscillations with a wobbling base frequency, causing random phase differences between the random and host clocks. The user can check out designs with projected assumptions of circuit component degradation, and learn safe parameters, e.g., max/min duty cycle, safe ratios of base oscillator/host sample frequencies, and suitable delays to decorrelate the three internally generated binary signals. In this mode, all tests are repeatable, even those which are based on random (recorded in the Analysis File) initial values. The final design should always include the [AUTOMATIC FM ALARM ACCELERATION](#). For first cut testing we suggest turning off the automatic [WARNING TRIGGER](#)ed accelerations.
- 2) The Wandering base frequency mode uses the same basic parameters as in the Standard AIS 31 mode. In addition this mode stresses the Standard parameters by forcing the base frequency to wander with a range of small and large random frequency in/decrements, with an adjustable bandwidth. The resulting wave is an aberrated saw tooth imposed on a base frequency. Both the Steps and distance between the Steps, N_{AVER} , are random functions, i.e., repeated tests generate different results. The final design tests should include the [AUTOMATIC FM ALARM ACCELERATION](#).
- 3) The deterministic Single (Host generated sampling) Clock mode maintains unpredictability to the ZK-Crypt Controller for Ciphering and Data Authentication. The FM and other physical delayed random signals are replaced by pseudo-randomness remotely affected by data in the 32 bit Word Manipulator. The only adjustable parameters are the [INITIALIZATION KEY](#), the [NUMBER OF SAMPLES](#), the default Q_{TA} [DEBIASER](#), and the [DET](#)erministic repeatable Single Clock test or the [Ra](#)Ndom unrepeatable Single Clock test. This noise source is used in ZK-Crypt Stream Cipher and Hash/MAC deterministic coding; and as a pseudorandom noise source for a random seeded TRNG. [ETSI]specs for wireless device circuitry demand that no uncorrelated random frequencies be generated on the same wireless device. For complete ETSI compliance, the user will initialize the AIS 20 compatible Deterministic Random Number Generator, DRNG, with a new Random Number.

Standard Mode – Dual Clock

- 1) Click **DUAL CLOCK** mode to test to circuit and logic parameters for generating TRNG Noise.
- 2) Choose the **NUMBER OF SAMPLEs**. Typically, you may want to experiment on 400,000 samples (type 400 in the window) to estimate results prior to a full length run. The default is **300[000] SAMPLEs**. For a full DieHard test you may need up to 350 xxx million samples. Do not be deterred by DieHard 0 or 1 p value results, be prepared to note that in many test cases we are close to an acceptable range.

Remember, you will always expect highest quality DieHard results from the ZK-Crypt Stream Cipher, or Hashing results, or in any statistical measure of any of the 10 internal Register Bank, Data Churn or Result/Feedback Processor 32 bit variables.

- 3) Choose one of the 16 default **INITIAL KEYS**. Key changes alter results, but for repeatable tests you may want to reuse the same key (remember the same first digit of the key) more than once; e.g.; for the initial default key, **16E77016** remember "1". Without **WANDER** mode, the engine may revert to a deterministic period, in which case, you can prove deterministically, that the **AUTOMATIC FM ALARM ACCELERATION** saves the day. (Discover a difficult combination of key and R without automatic **WARNING** acceleration, then discover how changing the average frequency range alleviates the situation.)
- 4) Set the **DUTY CYCLE**, (the percentage of the Oscillator period where the signal is at Logic "1"). The **DUTY CYCLE** default is **0.48**. Duty Cycle is a function of logic gate rise (typically larger) and fall (typically smaller) times and inaccurate voltage thresholds. Set the **DUTY CYCLE** default, as a function of an intended silicon implementation.
- 5) Set an average estimated ratio of frequencies, **R**, between the RANDOM OSCILLATOR and the HOST SAMPLE rate in the **EST AVER OSC/HOST FREQ RATIO** window. Note that we recommend that the RANDOM OSCILLATOR should operate at a higher frequency than the HOST SAMPLE frequency, despite the fact that excellent results can be obtained at low RANDOM OSCILLATOR frequencies. We set the default **R** ratio setting to 3.0 - assuming that the designer may know the HOST FREQUENCY options that the ultimate user may need and wants to play safe. In most instances, the designer will not know what the real ratio will be, and in addition the intervals between SAMPLEs will be pseudo or real random.

If the HOST is designed to DMA READ results at 100 MHz, and the designer wants to save energy, he may set the OSCILLATOR to a minimum average frequency of 100 MHz; e.g., the **EST AVER OSC/HOST FREQ RATIO** would then be **1/1**; as the programmer knows that he can program an automatic frequency increment by shunting the FM ALARM section of the Ring Oscillator in **AUTOMATIC FM ALARM ACCELERATE** mode should the need arise. Alternately, the designer may know that at such frequencies, the gate temperature will constantly rise, resulting in a constant change of phase, and the voltage sensitive inverters propagation delays will wobble as at each cycle the post processor alternates current consumption levels..

- 6) The Relative Delays relate to the parameters of the three main delays in the FM Ring Oscillator and the Sampling Delay of the Output Signals (see Circuit Drawings).
To display the Oscillator Delay circuit,
Click the **OSC CIRCUIT** box,
The **FM ALARM + BASE+SWITCH** is an unchangeable constant, 1.0 – whose real value is derived from the **EST AVER OSC/HOST FREQ RATIO**.
The **HI-FREQ DELAY** shortens the period of the RANDOM OSCILLATOR (by default 0.4 of the **FM ALARM + BASE+SWITCH** delay).
The **LO-FREQ DELAY** shortens the period of the RANDOM OSCILLATOR (by default 0.2 of the **FM ALARM + BASE+SWITCH** delay.)
The NOISE GENERATOR outputs 3 decorrelated random signals. To further decorrelate signals at their source, the **JUGGLE DELAY** should differ from the **das DELAY**;

e.g., the [JUGGLE DELAY](#) default is 0.19 of the [FM ALARM + BASE+SWITCH](#) delay and the [das DELAY](#) default is 0.39 of the [BASE+SWITCH](#) delay interval.

To View the Output Signal circuits- Click the [OUTPUT PROCESSOR](#) box.

To View the nLFSR Randomizing Circuit- Click the [nLFSR CONTROLER](#) box.

To View the Ring Oscillator with the Random Period Controls – Click the [OSC CIRCUIT](#) box.

- 7) Click [RUN AIS 31TESTS](#) to set the sampling machine in operation. This executes a complete test, and generates an explicit analysis. The filename tells you the parameters, wherein you have the choice of choosing initial start values from a list; from a random value that you choose; or a true random value generated by the PC. In addition, each test adds a line in the [LOG FILE](#), listing test parameters – and initializes a [BAD FILE](#), should there be at least one [DEMERIT](#) result more than the [DEMERIT](#) Trigger (Default 50). The [BAD FILE](#) records the automatically changed configuration and a summary of warning [DEMERIT](#)s before the Config change. As a [LOG FILE](#) is appended, its name changes, showing the presently averaged statistical values and the maximum single test value. In [AUTOMATIC FM OSCILLATOR](#) mode the [BAD FILE](#) displays 10 test values following a warning or failed test value, before the HOST could have shunted a section of the ring oscillator.
- 8) A POP-UP will query – [NEW LOG FILE?](#) if you are running tests with new parameters click [YES](#) or [NO](#) if you are in the middle of a test sequence with one or two changing parameters. The [READY](#) sign will change to a blinking [TESTING](#). The PC's testing time is essentially a function of the qualities of the CPU, oscillator clock ratio, [R](#), and the [NUMBER OF SAMPLES](#).

We suggest sets of log test files (typically grouped into up to 40 [TEST ANALYSIS](#) files recorded in a single [LOG FILE](#)– and typically with similar parameter tests; e.g., simultaneously change only the ratio of the average ring oscillator frequencies to the "stable" host sampling frequencies and the estimated in/decremented [DUTY CYCLE](#). Note that the final archived [LOG FILE](#) name will display the Maximum [DEMERIT](#) occurrence and the summed average of the included test files.

The program can be run in the background while you work on a WORD document. When you are near a final design, we suggest running 300-500 million sample tests in non-repeatable Wander Mode without [AUTOMATIC FM ALARM ACCELERATE](#), to see what may happen in real life without Host regulated Frequency Acceleration. Using the default settings, or any other reasonable design, you will most certainly have a [BAD FILE](#) with a few thousand warnings, and possibly several failed tests, distanced many thousands, one from the next. Looking through the [BAD FILE](#) you will see that if there are any failed [DEMERIT](#)s they are typically far distanced and that the average [DEMERIT](#) grades are exceptionally good.

- 9) When STATUS PROCESS reads [FINISHED TEST](#) click [BROWSE](#) and you can evaluate the results of a test on screen in the Notepad. To get clean report files, set the Notepad format in Courier New font with 7 pte font size.
- 10) The output of an individual test will be a comprehensive two page statistical analysis of the test, and a one line summary for the present analysis file in the easily identified file name. If any single 320 bit string was not up to snuff; e.g., a [DEMERIT](#) grade exceeding the [WARNING TRIGGER](#), with a result in the generated [BAD FILE](#) you can examine the events following the happening. In the Appendix we have included an annotated report.

Each test analysis in the Notepad format includes:

- a name which includes time, date, test parameters, and a summary of results
- a summary of the bias of internal nodes (see the [OUTPUT PROCESSOR](#) drawing)
- the proof of wobbling oscillator period – tells how many [fr](#) counts between Host samples
- detailed AIS 31 [DEMERIT](#) ratings of 9 Middle of File concurrent 320 bit Noise strings
- summary of [DEMERIT](#) distribution ratings of the 320 bit Noise sampled strings
- maximum and average values of [DEMERIT](#) for each of the Noise outputs
- the average occurrence of a missing activating (P)Random Pulse
- the percentage of time when each of the four frequency levels was active

The appended one line to the [LOG FILE](#) analysis includes - for each test:

- the filename with the configuration parameters, and

- the average and maximum DEMERIT values of the 3 single signal sample outputs & the concatenated 3 Noise output signal test values
- 11) You can [BROWSE](#) (and print out) any of the output pages. We recommend running a minimum of at least 300 million samples in each final test; wherein at each test, only 1 or 2 parameters should be incremented/decremented in the Initialization of each test sequence; e.g., increment [DUTY CYCLEs](#) from 48% to 65% then repeat while changing [JUGGLE](#) & [das](#) delays.
 - 12) You can read the output statistics of each file.
Click the right hand [box](#) on the [BROWSE FILE NAME](#) line;
scroll through and choose an individual test file or a [LOG FILE](#) Summary
Click on the file of your choice and read and print.
 - 13) Similarly, you can [RUN DIEHARD](#) and [BROWSE](#) Diehard Statistics after choosing DieHard options which appear in a pop-up box. Note, exceptionally good AIS 31 statistics rarely satisfy more than a random few of the 200 odd DieHard tests, though you may note that the sampled values are not far removed from acceptable expected range of DieHard values.
 - 14) To display test outputs in "readable binary" 1s and 0s on anyone of the 4 output tests-
Click [DISPLAY BINARY SCREEN](#) before you run the test;
and then after the test click the [BROWSE FILE NAME](#) box at the end of the line-
then choose one of the four output files – Das, FToggle, Juggle, or Triple- Click [OPEN](#)-
the chosen filename will appear in the [BROWSE FILE NAME](#) line;
now Click the [BROWSE](#) box;
Now you can up and down Scroll.

Note: **R** the estimated average frequency based ratio between the random oscillator and the Host Sampling Frequencies is a function of the (default) oscillator delays-
i.e., the default FM Alarm delay is 0.3 of the total base+switch delay;
the HI-FREQ delay is 0.4 and the LO-FREQ delay is 0.2 of the estimated final average half period.

WANDER Mode – Dual Clock - for maximum coverage and circuit stressing of the TRNG Noise Source

Stressing the Design with a user configured "Wandering" FM Oscillator. The **WANDER** mode gives a reliable sweep of the gamut of phase differences which can be expected from a set of design parameters.

Using the graphic drawing found by clicking the **WANDERING PARAMETER** command box you can estimate what might happen in a worst case real life situation. FortressGB does not know the foibles of the fab's delay circuits, but we estimate judging from our FPGA circuits, and from the wide range of **WANDER** mode tests that we have performed that we have anticipated the possible repeating syndromes.

Subsequent to a first cut design in the previous section, it is vitally necessary both to generate large **WANDER** tests on small frequency ranges to emulate normal operating conditions with fairly stable parameters and to stress the circuit in anticipation of component parameter degradation and large voltage fluctuations. **WANDER PARAMETERS** can be changed to cover short and long term degradation. Typical delay circuits will vary from the norm as a result of ageing, voltage and temperature fluctuations, etc.

The Average Base Frequency- **R**, is initialized, as in the Standard Mode by the **EST AVER OSC/HOST FREQ RATIO**. The average base **R** frequency (the **fr** pulses in the circuits) will increase and decrease in **S** sized steps where:

$$S = \text{Aver Inc/Decrement Step} - \text{Default} \rightarrow (S = | 0.001 \times R |);$$

When the **fr** frequency exceeds the upper Boundary, or is less than the lower Boundary the ragged saw tooth will reverse the present increment or decrement direction.

A frequency step may occur every **NAVER fr** clocks with a probability of **PROB/ 16** where:

NAVER= Average Number of **fr** clocks between calls to the stepping function –
Default (**NAVER**= 3) and,

PROB /16 = The Probability of an **S** sized De/Increment where $0 \leq \text{PROB} \leq 15$;
Default (**PROB** = 8) and,

- 1) Set Standard Mode Dual Clock Parameters, as in steps **1)** to **7)** in the previous section.
- 2) Click the **WANDER STRESS PARAMETER** box. Configure as in following steps 3) or 4) or run the default
- 3) Set **STEP (S)** to emulate the relative increment/decrement called for once every **NAVER** oscillator, **fr** clock signals.
- 4) Set the fraction threshold **PROB** to establish the desired probability that on the above mentioned clock the chance of an increment or decrement will happen.
- 5) Run the AIS 31 test, and **BROWSE** through the Test Results as in steps **9)** to **14)** in Standard Mode instructions.
- 6) Print interesting results. (Remember to set the Notepad font to Courier New, Bold, pte=7 for a concise easy to read report.)

Instructions for Running the ZK-Crypt Noise Emulator (Continued)

Deterministic Mode – Single Clock (For Ciphering, Hashing and ETSI compatible TRNG)

The quality of the Deterministic Noise Source is often ignored by cipher designers. As cryptanalysts search internal differentials (the probability of bias on binary variables), it is increasingly wise to choose an AIS 31 type tested source, which is virtually uncorrelated to the data in the DRNG.

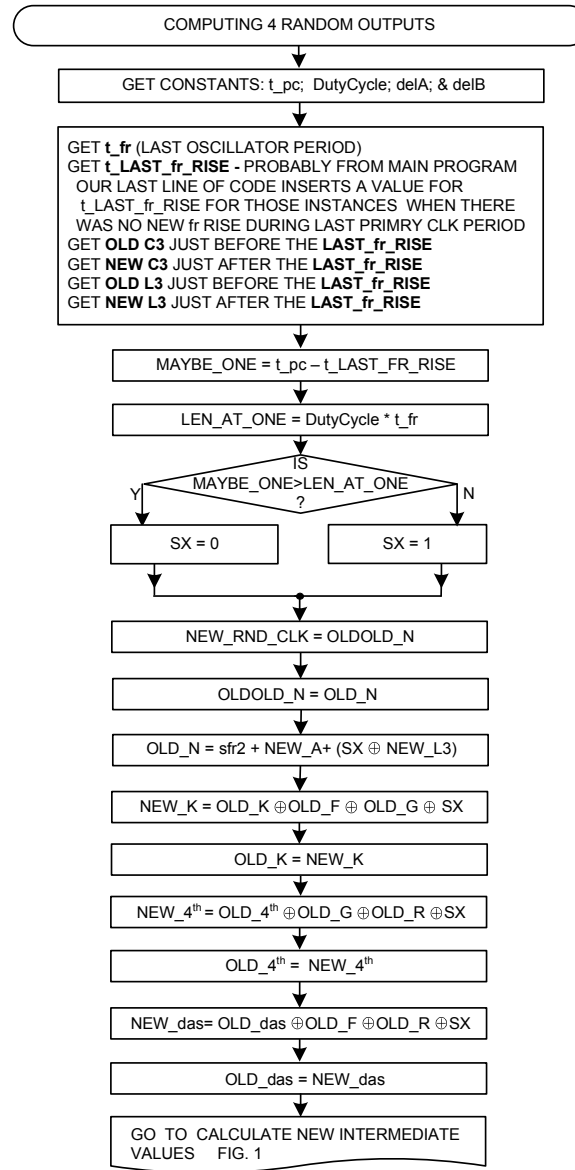
The pseudorandom noise sources of the emulator closely resemble the data regulated sources in the ZK-Crypt controller.

The two sources of randomness in the Deterministic Test are the Q_{TA} signal that emulates the LS Bit of a Control Unit Up-Counter and the (P)Random Clock Slip which "ticks" a random "1" on an average of once in about 8 Host triggered Primary Clock samples in DETERministic mode (repeatable) and about once in 16 in RANDOM non-repeatable mode. The designer might en/disable the Q_{TA} noise source by clicking the Q_{TA} DEBIASER button.

The Q_{TA} input oscillates with the Host Sampled clocks, simulating the LS bit of a 4 bit Random Up-Counter which resets at count 15; to start-counting again from an equiprobable start count at 4,5,6,.. or 15. In the real setting, whenever the (P)Random Clock misses a pulse, Q_{TA} will not be toggled. Stated differently, on an average of about once in about 8 Host clocks, there is either a double "1", e.g.,...0101101 or a double "0", e.g., ...010100101 in an otherwise oscillating ...101010... sequence.

There are only four parameters (degrees of freedom) in the Single Mode "entropy" test.

- 1) Choose the NUMB OF SAMPLES [000].
- 2) Choose one of the 16 default INITIAL KEYS or type in a Random 32 bit of your choice.
- 3) Choose DETERministic for a repeatable test or RND for a real random non-repeatable test.
- 4) See how the Q_{TA} DEBIASER affects statistics, while assuring a balanced output.
- 5) Go to step 8) of the first section run a test with an analysis of the results.



GLOBAL INPUT VARIABLES

C3=Q₁ ⊕ Q₅ ⊕ Q₇; L3=Q₂ ⊕ Q₄ ⊕ Q₈; A=Q₀ ⊕ Q₃ ⊕ Q₆;
 t_LAST_fr_RISE; t_fr;
 OLD_C3; NEW_C3;
 OLD_L3; NEW_L3
 NEW_A

At every rise of the fr signal the main program updates thus:
 The TIME of rise is recorded in t_LAST_fr_RISE
 OLD_C3 [L3] = NEW_C3 [L3] the value before the RISE
 NEW_C3 [L3] = is the value at C3 [L3] after TIME of RISE

GLOBAL VARIABLES CONSTANT DURING TEST RUN

t_pc; DutyCycle; delA; delB

GLOBAL OUTPUT VARIABLES

NEW_RND_CLK; NEW_K; NEW_4th; NEW_das

INTERNAL VARIABLES

t_nxt_Xup – TIME when AHV XOR modifier flips up
 t_nxt_Xdown – when AHV XOR modifier flips down (TRUE)
 SX= 0,1 - XORs ARE TRUE or FALSE
 t_nxt_Bup – TIME when FF3 Out modifier flips up
 t_nxt_Bdown – when FF3 Out modifier flips down (TRUE)
 SB = 0,1 - FF3 is TRUE or FALSE
 t_nxt_Aup – TIME when FF4 OUTmodifier flips up
 t_nxt_Adown – when FF4 OUT modifier flips down (TRUE)
 SA = 0,1 - FF4 is TRUE or FALSE
 out_temp- which of 2 delayed signals is sampled

+ → OR; ⊕ → XOR

0

Timing Diagrams & Flow Charts (Continued)

CONSTANTS FOR PAGE 2:

t_{pc}; the period of the PRIMARY CLOCK (PC)
 delA & delB; the delays before flip-flops FF3 & FF4
 DutyCycle; the duty cycle of the fr signal. The interval part of t_{fr} when the fr signal is at logic "1".
 Typically DutyCycle ≅ 0.5

DESCRIPTION of VARIABLES FOR PAGE 2:

t_{LAST_fr_RISE} is the time of start of the last fr oscillator interval which happened before a present Primary Clock sampling. If its value is <0, then we know how far back fr rose. If it is positive we know that it occurred in the present cycle.

fr a frequency oscillation signal of random interval t_{fr}. fr is first at logic "1" for a function of the duty cycle (DutyCycle * t_{fr}); ideally DutyCycle ≅ 0.50.

It's as if a delayed fr signal S is XORed to the delayed C3 or L3 input to FF3 or FF4. During the first logic "1" period, the signal S is a "1" which when XORed to the FFx IN causes the FFx OUTsignal to be FALSE; during the remainder of the S signal, at logic "0", FFx OUT is therefore TRUE.

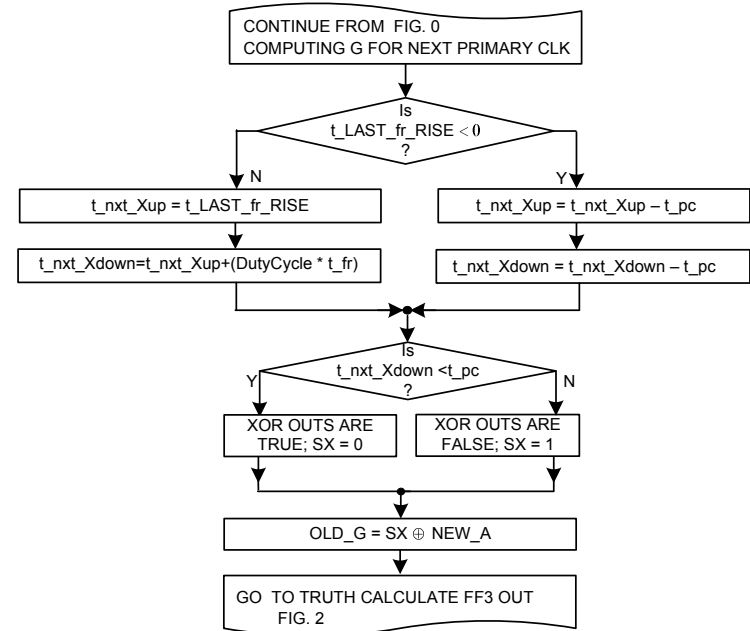
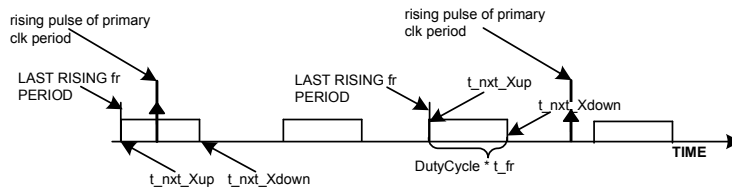
Time is measured on the abscissa. TIME is the variable. At beginning of each Primary Clock (PC) interval, "TIME" is zeroed, therefore at the end of each PC signal, TIME = t_{pc}, & immediately TIME = 0.

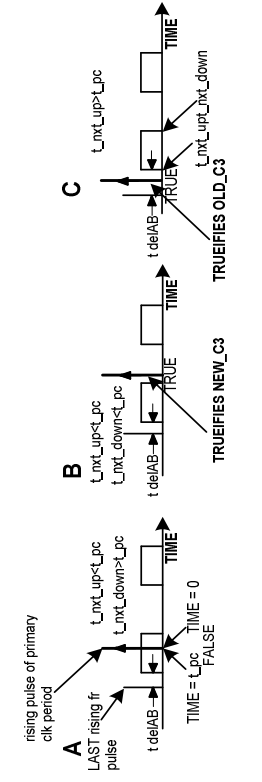
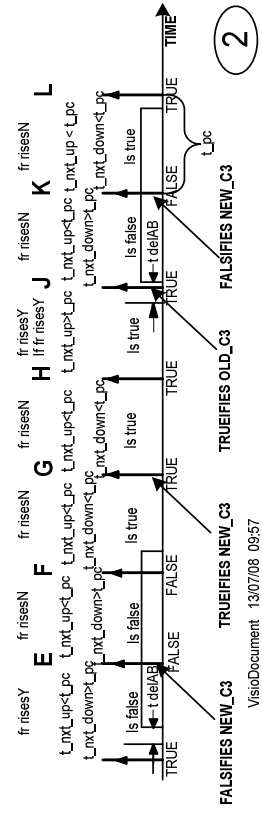
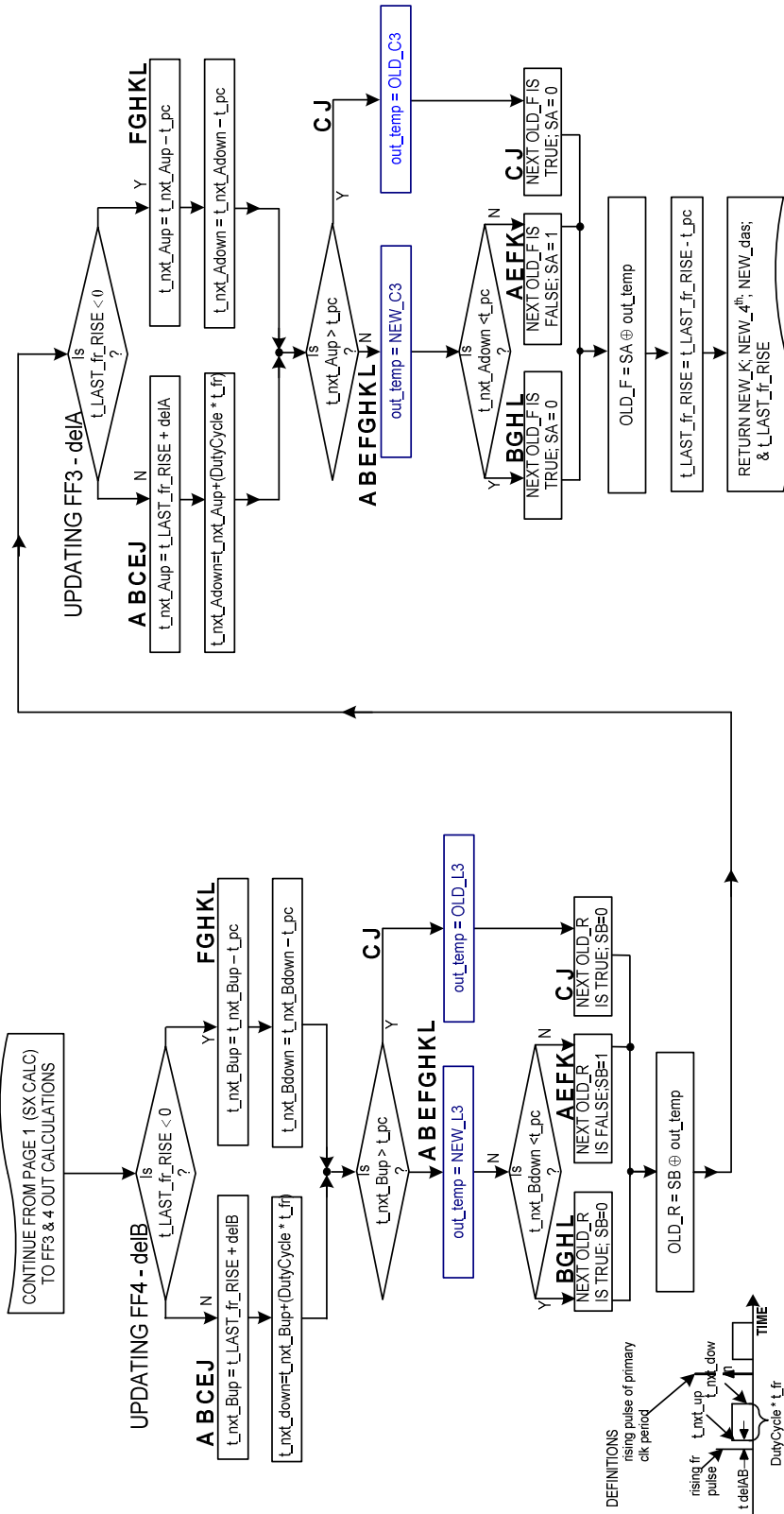
We update t_{LAST_fr_RISE} as the last function subtracting t_{pc}, as the new PC period starts at 0. This t_{LAST_fr_RISE} value will only be used if the Main Program does not insert a new record of a new LAST_fr_RISE, i.e., if t_{fr} is larger than t_{pc}, and an fr Rise does not occur in every PC period.

t_{next_up} is the TIME of start of a flip to logic "1" of the fr modifying signal Sx, delayed by delA or delB after the last rising fr signal. It is "flipped" down to "0" at TIME= t_{next_down}.

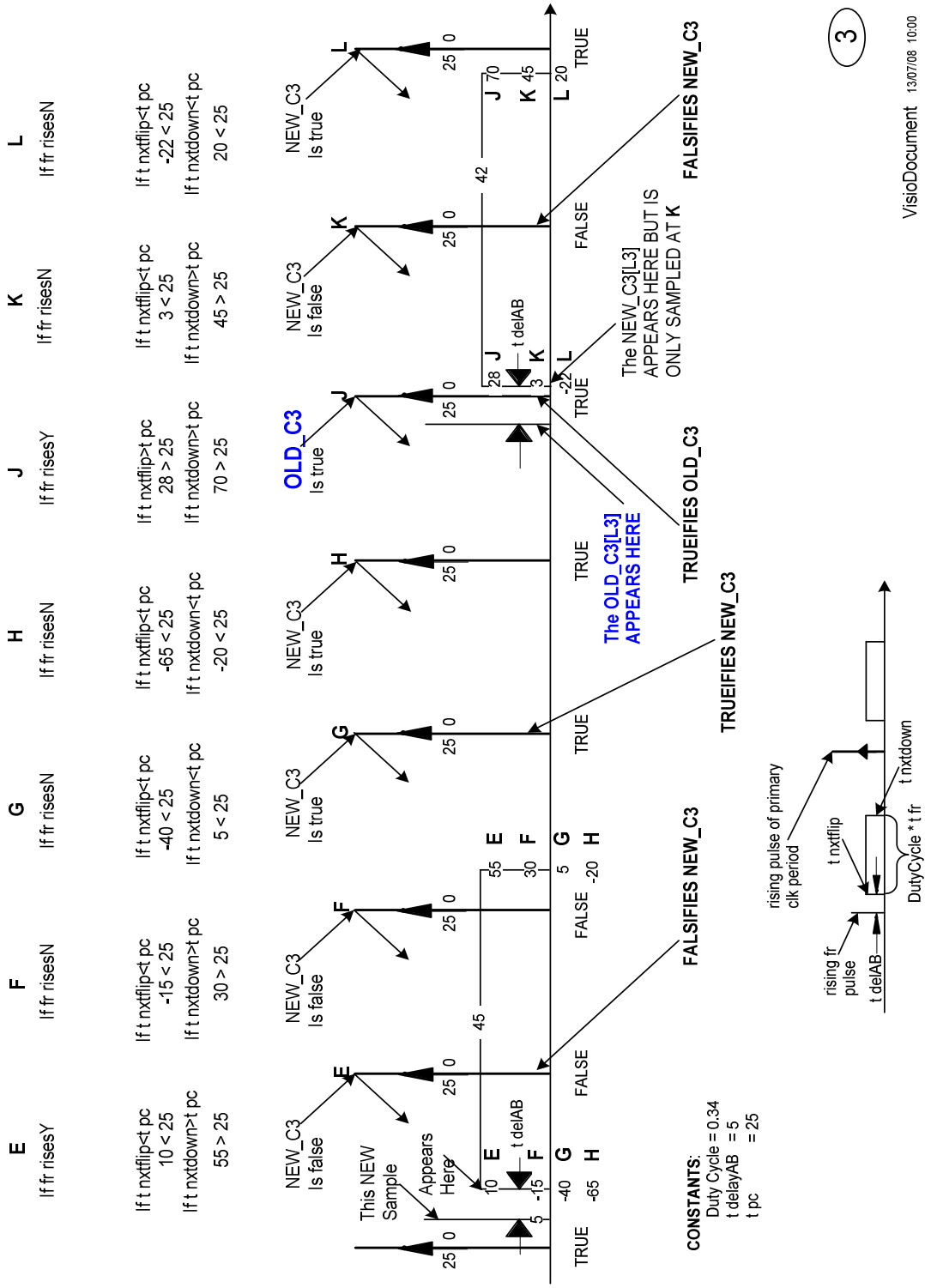
Typically, there are many fr periods in a single t_{pc}, Primary Clock period, but we are only interested in the last LAST_fr_RISE. The main program updates LAST_fr_RISE at every rise of the fr pulse, wherein variables are sampled into flip-flops.

DEFINITIONS





xxx
Timing Diagrams & Flow Charts (Continued)



3

xxx
 References:
 *[Schind] W. Schindler, "Efficient Online Tests for True Random Number Generators", CHES 2001, Springer Verlag, Berlin, 2001.
 [zk-cc] C. Gressel, A. Hecht, ZK-Crypt Circuit and Concept Drawings, www.fortressgb.com website, December 2007.
 [ppats] Provisional US Patent Applications 60/730,555 and 60/755, 821.
 *[wpat] Patent Application – WO 2007/049268, May 25, 2006.
 [ETSI] Specification No. TS 102 221 V3.0.0F-06921 European Telecom Standards Institute 2000.



*[zk-undst] C. Gressel, O. Dunkelmann, A. Hecht, "Understanding the ZK-Crypts- Ciphers for (Almost) all Reasons"
www.fortressgb.com website, December 2007.
[DieH] Marsaglia's DieHard tests, to be found at- <ftp://ftp.csis.hku.hk/pub/random/source>, 2004.

*References included in FortressGB Noise Emulator.

This is a Beta Site program based on the design which was used for the original patent spec and in the design FPGA based PC emulators which FortressGB designed and assembled. The emulator statistics are reasonably similar, both in deterministic mode and in true physical random mode, with the FPGA outputs. All facets of the noise generator are a pending WO/PCT patent.

Don't hesitate to write or call with questions, or suggestions to this preliminary version.

Contact us at -

carmi@fortressgb.com or call +972-54-7776059

Appendix A– Typical Analytic Test Files (first two are Annotated)

Statistical AIS-31 Analysis of ZK-Crypt Noise Source

Date: 20.12.07;

The file name gives you

3.0aNoFmNoWaaA_AIS_201207_1127_DUAL_NoFm_3000_046323ED_3.0_48_28._0_77.19.txt

A selected demo File Name with demo identifier 3.0bNoFmNoW which we have prefixed

and our demo program generated ID **aaA**

NoFm and without **W**, means it is Deterministic, and you can replicate. **W**ander would have indicated user chosen parameters for randomly incrementing and decrementing the average "accordion" frequency. This is definitely the worst case test. The ZK-Crypt supply voltage alternates as the engine different levels of random current consumption, which we assume will affect the propagation delays of what should be voltage sensitive inverters in the ring oscillator.

The Date and Time -20.12.07_11:27

DUAL Clocks – Both Deterministic without W (Random Wandering)

Fm- Means that the Host would initiate a Frequency Acceleration (Average Increase of about 25%), iff a single individual test registered a Demerit value of more than the Host designated value (Default 50). The NoFm setting is preferred for first cut estimations.

3000 x 1000 = 3M samples were taken at 3M consecutive Primary Clocks

046323ED is the 0'th Key in the list – at every program start, a new random option appears in the window- where the user has the option of initializing with the random key, one of the 16 listed standard keys or any typed in key of his choice.

3.0 is an estimated average of the **fr**, (the simulated random frequency) to the Host Primary Clock)

48 is chosen (default) Duty Cycle % of the **fr**, – the interval in a period when the "random" oscillating pulse is at logic "1"

28 was the highest demerit figure in any one of the test runs (see the Juggle Nibble Test → max = 28.0)

0 indicates that there were no failures → No Demerit value larger than 65

77.19 is the aggregate demerit value- it is the sum of the four average demerit values (see the four av=xx) this is much higher than can be expected with a "wandering" **fr**.

Test Parameters Sampled Outputs: 3000000 3M samples of each node were taken at 3M consecutive Primary Clocks
Estimated Average OSC/HOST Freq.Ratio 3.0; DutyCycle 0.48; Auto Demerit 50.0; This is the Demerit value that would trigger a "Warning FM" which in Fm mode would accelerate **fr** by about 25%.

Rel Delays: Juggle 0.19; DAS 0.39; Basic+Switch Delay 0.70; HiFreq 0.40; LoFreq 0.20; FmAlarm 0.3.
These are the default delay values in the oscillating and the output processor (click on the screen to observe)

The number of "1"s that were sensed at each of the nodes.

The (P)Random Clock, which is synched to the Primary Clock, missed 3M – 2,682,439 = 317561 pulses about 10.6%; the oscillator output 9.04M close to the 3.0 estimate.

of sampled '1's in test nodes: (see Drawings); (P)RandomClk 2682439; frClk 9035592;

The three permutation Primary Clock drivers, das, 4'th Toggle and Juggle Splash would ideally have pulsed 1.5M "1"s. The disparate distribution is typical of a deterministic sequence.

das 1552984; 4'th Toggle 1500045; Juggle Splash 1323577;

Of the 9.1M oscillator pulses, there should be abt 4.5 M "1"s in the 9 nLSFR cells and their derivatives.

Q8 4517800; Q7 4517809; Q6 4517785; Q4 4517795; Q5 4517785; Q3 4517794; Q2 4517795;
Q1 4517810; Q0 4517810; A 4517803; C(3) 4517784; L(3) 4517788; L(4) 4606025; B 4429558;

The three flip-flops sensing delayed signals synched to the primary clock; H and J were measured at the rise of the Primary Clock, but were modulated by the **fr** oscillator signal.

ffF3 1447111; ffF4 1358871; ffF5 1464751; H 1552996; J 1376517;



There are four levels of **fr** frequencies, randomly selected, e.g., the lowest frequency is abt 63% of the highest frequency and appeared 11.1% of the **fr** clock periods.

HiFreq10 27.0%; Freq8.3 7.4%; Freq7.1 54.5%; LoFreq6.3 11.1%;

Shown are the Schindler recommended Chi Square nibble test on strings of 320 bits; 80 nibbles of value 0 to 15. The 9 tests explicitly displayed are automatically chosen from the middle of the sample run. The Chi Square Demerit Results, in all of these were quite reasonable. The aggregate number of each of the nibble value occurrences are disparate, typical of a deterministic test, where we assume with good reason that depending on the initial condition, the Noise Generator generates a deterministic sequence. This demo was the weakest that we found where the ratio Osc/Host was equal to exactly 3 on our last short run tests length 3 million.

In some cases this can be a sequence of failures, which in virtually every case can be amended by accelerated proprietary frequency shunting.

The deviation of the nibble value occurrences is typical of tests executed without a wandering **fr** frequency; e.g., nibble 7, 0111, appeared only 6172 times, compared to nibble 13, 1101 which also consists of three "1"s and one "0" which occurred 26,467 or 2, 0010₂ which appears 114,693 times.

At each Primary Clock, three binary bits from the Juggle Toggle, 4th Toggle and das outputs are concatenated and strung together and once in 10 32 bit words are Chi Square tested. This test may give a picture of correlation between the three binary signals.

```
Nibble Frequencies of Concatenated String ...||Juggle||4thToggle||das||Juggle||4thToggle||das|...
9 tests from Test # 14061
[ 0] 8 4 3 8 5 2 8 5 3 141175 [ 8] 6 6 3 8 5 2 8 7 2 149995
[ 1] 5 4 8 4 5 6 6 5 3 149994 [ 9] 4 7 4 6 5 4 6 6 5 149994
[ 2] 4 7 7 3 8 5 6 6 6 167639 [10] 5 6 4 4 5 6 4 2 7 132349
[ 3] 4 2 5 4 4 5 4 4 4 114700 [11] 6 6 6 5 9 5 5 9 6 176463
[ 4] 7 4 7 5 5 7 5 5 6 158818 [12] 4 11 8 5 9 10 4 8 9 211756
[ 5] 1 4 4 1 3 3 3 3 2 79408 [13] 5 5 8 6 4 9 3 5 10 167640
[ 6] 4 6 4 3 6 5 4 5 6 132347 [14] 10 2 2 10 3 2 8 5 1 132353
[ 7] 1 2 4 1 1 5 1 1 5 61762 [15] 6 4 3 7 3 4 5 4 5 123527
Demerit Results = 15.6 16.0 12.4 18.4 14.4 16.0 11.6 12.4 18.4
```

The three binary driver tests evaluated separately.

```
Nibble Frequencies of Juggle Splash Toggle. 9 tests from Test # 4687
[ 0] 6 5 6 6 6 5 6 5 6 52936 [ 8] 11 8 11 11 11 9 10 9 11 97048
[ 1] 4 4 4 4 3 3 4 4 4 35292 [ 9] 5 4 5 5 5 5 4 5 5 44111
[ 2] 12 13 12 12 13 13 12 13 12 114693 [10] 6 6 6 6 6 5 5 5 6 52936
[ 3] 5 5 5 5 5 5 5 5 4 44110 [11] 3 4 4 4 4 4 4 4 4 35291
[ 4] 4 5 4 4 5 5 5 5 5 44114 [12] 6 6 6 4 4 6 6 6 6 52937
[ 5] 1 1 1 1 1 1 1 1 0 8822 [13] 2 3 3 3 3 3 3 3 2 26467
[ 6] 3 3 3 3 2 3 3 2 3 26468 [14] 4 4 3 3 4 4 4 4 4 35291
[ 7] 3 4 3 4 4 4 4 4 4 35291 [15] 5 5 4 5 4 5 4 5 4 44113
Demerit Results = 25.6 20.8 24.8 24.0 28.0 22.4 21.2 22.8 26.4
```

```
Nibble Frequencies of 4th Toggle. 9 tests from Test # 4687
[ 0] 5 4 5 5 5 4 4 4 5 44113 [ 8] 7 8 7 7 7 7 8 8 7 70582
[ 1] 2 2 2 2 2 2 2 2 2 17645 [ 9] 5 4 5 5 5 5 5 5 5 44112
[ 2] 4 4 4 3 4 4 4 4 3 35290 [10] 3 4 4 4 4 4 4 3 4 2 35289
[ 3] 1 1 1 1 1 1 1 1 1 8823 [11] 2 1 2 2 2 2 2 2 2 17645
[ 4] 8 8 7 7 7 7 8 7 8 70582 [12] 5 5 5 5 3 5 5 5 4 44113
[ 5] 4 5 5 5 6 5 6 6 6 52938 [13] 5 6 6 5 6 5 6 6 6 52935
[ 6] 9 8 8 9 9 9 8 7 9 79402 [14] 11 11 11 11 11 11 11 11 11 97048
[ 7] 5 5 4 5 5 5 3 5 5 44112 [15] 4 4 4 4 3 4 4 3 4 35291
Demerit Results = 21.2 22.0 18.4 20.0 21.2 19.6 22.0 19.2 23.2
```

```
Nibble Frequencies of das Slip Toggle. 9 tests from Test # 4687
[ 0] 2 2 2 2 2 2 2 2 1 17644 [ 8] 4 5 5 5 5 5 4 5 3 44112
[ 1] 8 8 8 7 8 7 8 7 8 70581 [ 9] 8 7 7 8 8 8 8 8 7 70580
[ 2] 7 8 8 8 8 8 7 7 8 70581 [10] 2 2 2 2 1 1 2 2 2 17646
[ 3] 2 1 2 2 2 2 2 2 2 17645 [11] 5 5 5 5 5 4 5 5 5 44114
[ 4] 6 8 8 7 8 8 7 8 7 70581 [12] 5 5 5 4 4 4 5 4 5 44114
[ 5] 1 1 1 1 1 1 0 1 1 8822 [13] 5 5 2 4 5 5 5 5 5 44115
[ 6] 3 4 4 4 4 4 4 3 4 35290 [14] 6 4 5 5 5 6 6 6 6 52937
[ 7] 9 8 9 9 7 9 9 8 9 79401 [15] 7 7 7 7 7 6 6 7 7 61757
Demerit Results = 18.4 19.2 20.8 18.4 19.2 20.4 19.6 17.6 20.4
```



```

Demerit Distribution -
Juggle Nibble Test: # FM Warning Triggers (> 50.0) 0
                   # Failed Strings (>65) 0
Count ** 0-25= 6617 ** 26-35= 2757 ** 36-45= 0 ** 46-55= 0 ** 56-65= 0 max= 28.0 av= 23.9

4th Toggle Nibble Test: # FM Warning Triggers (> 50.0) 0
                       # Failed Strings (>65) 0
Count ** 0-25= 9374 ** 26-35= 0 ** 36-45= 0 ** 46-55= 0 ** 56-65= 0 max= 23.2 av= 20.0

das Nibble Test: # FM Warning Triggers (> 50.0) 0
                 # Failed Strings (>65) 0
Count ** 0-25= 9374 ** 26-35= 0 ** 36-45= 0 ** 46-55= 0 ** 56-65= 0 max= 20.8 av= 18.6

3 signal Nibble Test: # FM Warning Triggers (> 50.0) 0
                     # Failed Strings (>65) 0
Count ** 0-25= 28124 ** 26-35= 0 ** 36-45= 0 ** 46-55= 0 ** 56-65= 0 max= 21.6 av= 14.7

Average Test Cummulative ( 77.2)/4 = 19.3 Max Test Demerit = 28.0

```

By some measures we might call these fair results, with a fairly high cumulative average of 19.3, but no failures.

The following shows the first 240 periods, assuming a 1 MHz Primary Clock, and on the right, the number of rising fr (oscillator) periods. Note the four deterministic period lengths, and the four different number of fr pulse "starts" in a single Primary (Host) Clock periods; i.e., 2, 3, 4 and 5.

```

Wobbling Accordion
if Period of Primary Clock = 1 µs, fr Periods (µs)
in 1st 240 fr Clocks
# fr Triggers in
1st 240 Primary Clocks
[ 0]= 0.380952 [ 80] = 0.238095 [160] = 0.333333 [ 0]= 2 [ 80] = 3 [160] = 3
[ 1]= 0.380952 [ 81] = 0.238095 [161] = 0.333333 [ 1]= 3 [ 81] = 3 [161] = 3
[ 2]= 0.380952 [ 82] = 0.238095 [162] = 0.333333 [ 2]= 3 [ 82] = 3 [162] = 3
[ 3]= 0.380952 [ 83] = 0.238095 [163] = 0.333333 [ 3]= 2 [ 83] = 4 [163] = 3
[ 4]= 0.380952 [ 84] = 0.238095 [164] = 0.333333 [ 4]= 3 [ 84] = 4 [164] = 3
[ 5]= 0.380952 [ 85] = 0.238095 [165] = 0.333333 [ 5]= 3 [ 85] = 2 [165] = 2
[ 6]= 0.380952 [ 86] = 0.380952 [166] = 0.333333 [ 6]= 3 [ 86] = 3 [166] = 3
[ 7]= 0.380952 [ 87] = 0.380952 [167] = 0.333333 [ 7]= 3 [ 87] = 3 [167] = 4
[ 8]= 0.380952 [ 88] = 0.380952 [168] = 0.333333 [ 8]= 3 [ 88] = 3 [168] = 4
[ 9]= 0.380952 [ 89] = 0.380952 [169] = 0.333333 [ 9]= 3 [ 89] = 3 [169] = 3
[10]= 0.380952 [ 90] = 0.380952 [170] = 0.380952 [10]= 4 [ 90] = 4 [170] = 3
[11]= 0.380952 [ 91] = 0.333333 [171] = 0.380952 [11]= 3 [ 91] = 3 [171] = 3
[12]= 0.333333 [ 92] = 0.333333 [172] = 0.380952 [12]= 2 [ 92] = 3 [172] = 3
[13]= 0.333333 [ 93] = 0.238095 [173] = 0.380952 [13]= 3 [ 93] = 3 [173] = 3
[14]= 0.333333 [ 94] = 0.238095 [174] = 0.380952 [14]= 3 [ 94] = 3 [174] = 3
[15]= 0.333333 [ 95] = 0.238095 [175] = 0.380952 [15]= 3 [ 95] = 3 [175] = 3
[16]= 0.333333 [ 96] = 0.238095 [176] = 0.380952 [16]= 3 [ 96] = 3 [176] = 3
[17]= 0.333333 [ 97] = 0.238095 [177] = 0.380952 [17]= 3 [ 97] = 3 [177] = 3
[18]= 0.380952 [ 98] = 0.238095 [178] = 0.285714 [18]= 3 [ 98] = 3 [178] = 3
[19]= 0.380952 [ 99] = 0.333333 [179] = 0.285714 [19]= 4 [ 99] = 2 [179] = 3
[20]= 0.380952 [100] = 0.380952 [180] = 0.285714 [20]= 3 [100] = 3 [180] = 5
[21]= 0.380952 [101] = 0.380952 [181] = 0.285714 [21]= 3 [101] = 3 [181] = 3
[22]= 0.285714 [102] = 0.380952 [182] = 0.285714 [22]= 3 [102] = 3 [182] = 3
[23]= 0.285714 [103] = 0.380952 [183] = 0.380952 [23]= 3 [103] = 3 [183] = 3
[24]= 0.285714 [104] = 0.380952 [184] = 0.380952 [24]= 2 [104] = 3 [184] = 3
[25]= 0.285714 [105] = 0.380952 [185] = 0.380952 [25]= 3 [105] = 3 [185] = 4
[26]= 0.285714 [106] = 0.380952 [186] = 0.380952 [26]= 3 [106] = 3 [186] = 3
[27]= 0.285714 [107] = 0.380952 [187] = 0.380952 [27]= 5 [107] = 3 [187] = 3
[28]= 0.238095 [108] = 0.380952 [188] = 0.380952 [28]= 3 [108] = 3 [188] = 3
[29]= 0.238095 [109] = 0.380952 [189] = 0.380952 [29]= 3 [109] = 3 [189] = 3
[30]= 0.333333 [110] = 0.380952 [190] = 0.380952 [30]= 3 [110] = 3 [190] = 3
[31]= 0.333333 [111] = 0.380952 [191] = 0.380952 [31]= 4 [111] = 3 [191] = 3
[32]= 0.333333 [112] = 0.380952 [192] = 0.380952 [32]= 3 [112] = 3 [192] = 4
[33]= 0.333333 [113] = 0.333333 [193] = 0.380952 [33]= 3 [113] = 2 [193] = 4
[34]= 0.380952 [114] = 0.333333 [194] = 0.380952 [34]= 3 [114] = 3 [194] = 2
[35]= 0.380952 [115] = 0.333333 [195] = 0.380952 [35]= 2 [115] = 3 [195] = 3
[36]= 0.380952 [116] = 0.333333 [196] = 0.380952 [36]= 3 [116] = 3 [196] = 3
[37]= 0.380952 [117] = 0.333333 [197] = 0.333333 [37]= 3 [117] = 3 [197] = 5
[38]= 0.380952 [118] = 0.333333 [198] = 0.333333 [38]= 3 [118] = 3 [198] = 3
[39]= 0.380952 [119] = 0.380952 [199] = 0.333333 [39]= 3 [119] = 3 [199] = 3
[40]= 0.333333 [120] = 0.380952 [200] = 0.333333 [40]= 2 [120] = 3 [200] = 3
[41]= 0.333333 [121] = 0.380952 [201] = 0.333333 [41]= 3 [121] = 3 [201] = 4
[42]= 0.333333 [122] = 0.333333 [202] = 0.333333 [42]= 3 [122] = 3 [202] = 3
[43]= 0.333333 [123] = 0.333333 [203] = 0.285714 [43]= 3 [123] = 3 [203] = 3
[44]= 0.333333 [124] = 0.333333 [204] = 0.238095 [44]= 3 [124] = 3 [204] = 3
[45]= 0.333333 [125] = 0.333333 [205] = 0.238095 [45]= 3 [125] = 3 [205] = 2
[46]= 0.333333 [126] = 0.333333 [206] = 0.238095 [46]= 2 [126] = 2 [206] = 3
[47]= 0.333333 [127] = 0.333333 [207] = 0.238095 [47]= 3 [127] = 3 [207] = 3
[48]= 0.333333 [128] = 0.333333 [208] = 0.238095 [48]= 3 [128] = 3 [208] = 3
[49]= 0.333333 [129] = 0.333333 [209] = 0.333333 [49]= 3 [129] = 2 [209] = 3
[50]= 0.333333 [130] = 0.333333 [210] = 0.333333 [50]= 3 [130] = 3 [210] = 2
[51]= 0.333333 [131] = 0.333333 [211] = 0.333333 [51]= 4 [131] = 3 [211] = 3
[52]= 0.333333 [132] = 0.333333 [212] = 0.333333 [52]= 3 [132] = 3 [212] = 3
[53]= 0.333333 [133] = 0.380952 [213] = 0.333333 [53]= 3 [133] = 3 [213] = 3
[54]= 0.333333 [134] = 0.380952 [214] = 0.333333 [54]= 3 [134] = 3 [214] = 3
[55]= 0.238095 [135] = 0.380952 [215] = 0.333333 [55]= 3 [135] = 3 [215] = 3
[56]= 0.285714 [136] = 0.380952 [216] = 0.380952 [56]= 3 [136] = 3 [216] = 2
[57]= 0.285714 [137] = 0.380952 [217] = 0.333333 [57]= 2 [137] = 3 [217] = 3

```



[58]=	0.285714	[138] =	0.380952	[218] =	0.333333	[58]=	3	[138] =	3	[218] =	3
[59]=	0.285714	[139] =	0.333333	[219] =	0.333333	[59]=	3	[139] =	2	[219] =	3
[60]=	0.285714	[140] =	0.333333	[220] =	0.333333	[60]=	3	[140] =	3	[220] =	3
[61]=	0.380952	[141] =	0.333333	[221] =	0.333333	[61]=	3	[141] =	3	[221] =	4
[62]=	0.333333	[142] =	0.333333	[222] =	0.333333	[62]=	3	[142] =	3	[222] =	3
[63]=	0.333333	[143] =	0.333333	[223] =	0.333333	[63]=	2	[143] =	3	[223] =	3
[64]=	0.333333	[144] =	0.333333	[224] =	0.333333	[64]=	3	[144] =	4	[224] =	3
[65]=	0.333333	[145] =	0.380952	[225] =	0.333333	[65]=	3	[145] =	3	[225] =	3
[66]=	0.333333	[146] =	0.380952	[226] =	0.333333	[66]=	2	[146] =	3	[226] =	3
[67]=	0.333333	[147] =	0.380952	[227] =	0.333333	[67]=	3	[147] =	3	[227] =	2
[68]=	0.380952	[148] =	0.285714	[228] =	0.333333	[68]=	3	[148] =	3	[228] =	3
[69]=	0.380952	[149] =	0.285714	[229] =	0.333333	[69]=	4	[149] =	3	[229] =	3
[70]=	0.380952	[150] =	0.285714	[230] =	0.333333	[70]=	4	[150] =	3	[230] =	3
[71]=	0.333333	[151] =	0.285714	[231] =	0.380952	[71]=	3	[151] =	3	[231] =	3
[72]=	0.333333	[152] =	0.285714	[232] =	0.333333	[72]=	3	[152] =	3	[232] =	3
[73]=	0.333333	[153] =	0.333333	[233] =	0.333333	[73]=	3	[153] =	3	[233] =	2
[74]=	0.333333	[154] =	0.333333	[234] =	0.333333	[74]=	3	[154] =	3	[234] =	3
[75]=	0.333333	[155] =	0.333333	[235] =	0.333333	[75]=	3	[155] =	3	[235] =	3
[76]=	0.333333	[156] =	0.333333	[236] =	0.333333	[76]=	3	[156] =	2	[236] =	2
[77]=	0.333333	[157] =	0.333333	[237] =	0.333333	[77]=	3	[157] =	3	[237] =	3
[78]=	0.333333	[158] =	0.333333	[238] =	0.380952	[78]=	2	[158] =	3	[238] =	3
[79]=	0.238095	[159] =	0.333333	[239] =	0.380952	[79]=	3	[159] =	4	[239] =	4



AN ANNOTATED TYPICAL TEST REPORT GENERATED BY THE ZK-CRYPT DETERMINISTIC/RANDOM NOISE GENERATOR OPERATING IN SINGLE CLOCK MODE (NO RANDOM FM MODULATED OSCILLATOR)

Explanations & Circuit Diagrams in "The ZK-Crypt Noise Generator Design Parameter Emulator"

Statistical AIS-31 Analysis of ZK-Crypt Noise Source Date: 31.12.07; This particular test is for typical cryptographic applications, where we know that the QTA signal is a good pseudo-random, Data dependent source of pseudorandomness. Test Parameters Sampled Outputs: 10,000,000 Samples; Qta In the S/W emulator we replace random Data with an external Randomization of the (P)Random Slip. This is a normal test (takes our emulator about 30 seconds).

of sampled '1's in test nodes: (see Drawings); (P)RandomClk 9218321 the missed pulse generator; NO RANDOM CLOCK -> frClk 0;

All of the following sampled Results are excellent- PROBABILITY OF 0.5 "1"s das 4999256; 4'th Toggle 5000001; Juggle Splash 5000102;

Q8 4998231; Q7 5000576; Q6 4999784; Q4 5000252; Q5 4999784; Q3 4999325; Q2 4999325; Q1 4999498; Q0 4999498; A 4999657; C(3) 5000192; L(3) 5000358; L(4) 5000358; B 4999657; ffF3 5000192; ffF4 5000357; ffF5 4999657; H 5005900; J 4996518;

Table with 16 columns showing Nibble Frequencies of 4th Toggle. 9 tests from Test # 15625. Includes demerit results: 10.8 16.8 16.4 10.8 10.0 21.2 9.2 8.4 20.4

Table with 16 columns showing Nibble Frequencies of das Slip Toggle. 9 tests from Test # 15625. Includes demerit results: 16.0 11.6 16.0 17.6 18.8 6.4 13.6 12.8 22.0

Table with 16 columns showing Nibble Frequencies of Juggle Splash Toggle. 9 tests from Test # 15625. Includes demerit results: 11.2 12.0 13.2 18.0 17.2 22.8 8.8 14.4 10.4

Table with 16 columns showing Nibble Frequencies of Concatenated String ...||Juggle||4'thToggle||das||Juggle||4'thToggle||das||... 9 tests from Test # 46875. Includes demerit results: 30.0 14.8 7.6 14.4 32.0 16.0 13.6 21.2 23.2

At each clock, the three binary signals are concatenated- so that the ideal number is 156250 x 3 = 468750

./.



The 3 binary signals tested out beautifully, with only one sample run above 50 (55.2) all averages less than 15.1.

```

Demerit Distribution -
Juggle Nibble Test: # FM Warning Triggers (> 50.0) 0
                   # Failed Strings (>65) 0
Count ** 0-25= 29813 ** 26-35= 1350 ** 36-45= 86 ** 46-55= 0 ** 56-65= 0 max= 44.0 av= 14.9

4th Toggle Nibble Test: # FM Warning Triggers (> 50.0) 0
                       # Failed Strings (>65) 0
Count ** 0-25= 29826 ** 26-35= 1360 ** 36-45= 57 ** 46-55= 6 ** 56-65= 0 max= 46.4 av= 14.8

das Nibble Test: # FM Warning Triggers (> 50.0) 1 Groups # 22274
                # Failed Strings (>65) 0
Count ** 0-25= 29773 ** 26-35= 1337 ** 36-45= 133 ** 46-55= 5 ** 56-65= 1 max= 55.2 av= 15.0

```

The concatenated tests, just about the same- (remember 3 times the number of sampled bits) the BAD FILE records all of the warning signals. Here we see that there were 8 occurrences of Demerit Results more than 50, where interval between occurrences was at least 117 test sequences.

```

3 signal Nibble Test: # FM Warning Triggers (> 50.0) 8 Groups # 5302 26213 40475 60445 60562 67894 74322 89896
                    # Failed Strings (>65) 0
Count ** 0-25= 83540 ** 26-35= 9130 ** 36-45= 992 ** 46-55= 84 ** 56-65= 3 max= 60.8 av= 16.9

```

This result shows that there is a slight correlation between the three binary concatenated symbols. Despite the problem that was noticed only once in less than 1M samples the total average is still an enviable low 16.9, with the worst signal, which appeared once in 30M tests, of 60.8

Average Test Cumulative (61.6)/4 = 15.4 Max Test Demerit = 60.8

The unweighted average of the four tests is 15.4 – excellent, and the worst test (of 600,000,000 separate statistical measures was 15.4, the weighted average would be 15.9. The single worst test was 60.8.



On the next two pages we include results using the same basic parameters, wherein the test were run for 300M samplings. Appendix B is

We leave it to the reader to assess the results. In the inout box you will find the results of problematic samplings, which in all cases either the FM acceleration and or the Wandering random results were satisfactory. However, we assume that in any event there will be a random element which may degrade the DEMERIT, but will typically improve the distribution of each of the DEMERIT nibbles. We assume, from the results, what we see when displaying the binary literals, that there is a dearth of single ones surrounded by double "0"s, and long sequences of zero in the Juggle Splash. Test counts in Cipher Mode have shown that the overall is a balance of "1"s and "0"s,)and reasonable outputs for 5 and 10, which does not affect the output of the Splash Selector, which is regulated, essentially by random data from the Data Churn.

Appendix B- R = 3 with Wander from the \AIS\Release\inout folder.

3.0cNoFmWaaC_AIS_211207_1823_DUAL_W0.15_0.001_3_NoFm_300000_04532#E_3.0_48_55.5_0_67.5.txt

Standard Wander Parameters, without accelerated FM feature.

Statistical AIS-31 Analysis of ZK-Crypt Noise Source Date: 21.12.07;

Test Parameters Sampled Outputs: 300000000

Estimated Average OSC/HOST Freq.Ratio 3.0; DutyCycle 0.48; Auto Demerit 50.0;

Rel Delays: Juggle 0.19; DAS 0.39; Basic+Switch Delay 0.70; HiFreq 0.40; LoFreq 0.20; FmAlarm 0.3.

of sampled '1's in test nodes: (see Drawings); (P)RandomClk 276637281; frClk 987611006; das 150442504; 4'th Toggle 148023692; Juggle Splash 162772866;

Q8 453097516; Q7 453097525; Q6 453097521; Q4 453097523; Q5 453097521; Q3 453097531; Q2 453097531; Q1 453097513; Q0 453097514; A 453097526; C(3) 453097517; L(3) 453097524; L(4) 461121101; B 444483939; fffF3 143952869; fffF4 149498555; fffF5 144188834; H 152802421; J 153038384; HiFreq10 25.4%; Freq8.3 9.0%; Freq7.1 54.8%; LoFreq6.3 10.8%;

Nibble Frequencies of Concatenated String ...||Juggle||4'thToggle||das||Juggle||4'thToggle||das||... 9 tests from Test #1406250

Table with 16 columns (0-15) and 8 rows of nibble frequency data for concatenated string tests.

Nibble Frequencies of Juggle Splash Toggle. 9 tests from Test # 468750

Table with 16 columns (0-15) and 8 rows of nibble frequency data for Juggle Splash Toggle tests.

Nibble Frequencies of 4th Toggle. 9 tests from Test # 468750

Table with 16 columns (0-15) and 8 rows of nibble frequency data for 4th Toggle tests.

Nibble Frequencies of das Slip Toggle. 9 tests from Test # 468750

Table with 16 columns (0-15) and 8 rows of nibble frequency data for das Slip Toggle tests.

Demerit Distribution -

Juggle Nibble Test: # FM Warning Triggers (> 50.0) 11062 Groups # 100 205 220 347 418 428 481 491 555 792 919 990 1117 1222 1237 1364 1435 1445 1498 1508 1572 1809 1936 2007 2134 2239 2254 2381 2452 2462 2515 2525 2589 2826 2953 3024 3151 3256 3271 3398 3469 3479 3532 3542 3606 3843 3970 4041 4168 4273 4288 4415 4486 4496 4549 4559 4623 4860 4987 5058 5185 5290 5305 5432 5503 5513 5566 5576 5640 5877 6004 6075 6202 6307 6322 6449 6520 6530 6583 6593 6657 6894 7021 7092 7219 7324 7339 7466 7537 7547 7600 7610 7674 7911 8038 8109 8236 8341 8356 8483



```

# Failed Strings (>65) 0
Count ** 0-25= 696901 ** 26-35= 136430 ** 36-45= 73748 ** 46-55= 29498 ** 56-65= 922 max= 55.6 av= 20.4

4th Toggle Nibble Test: # FM Warning Triggers (> 50.0) 0
# Failed Strings (>65) 0
Count ** 0-25= 806601 ** 26-35= 120759 ** 36-45= 10139 ** 46-55= 0 ** 56-65= 0 max= 40.8 av= 17.8

das Nibble Test: # FM Warning Triggers (> 50.0) 0
# Failed Strings (>65) 0
Count ** 0-25= 901547 ** 26-35= 34108 ** 36-45= 1844 ** 46-55= 0 ** 56-65= 0 max= 35.6 av= 14.4

3 signal Nibble Test: # FM Warning Triggers (> 50.0) 0
# Failed Strings (>65) 0
Count ** 0-25= 2642880 ** 26-35= 153026 ** 36-45= 16593 ** 46-55= 0 ** 56-65= 0 max= 44.0 av= 14.9

Average Test Cumulative ( 67.5)/4 = 16.9 Max Test Demerit = 55.6

```

Wobbling Accordion

```

if Period of Primary Clock = 1 µs, fr Periods (µs)
in lst 240 fr Clocks
# fr Triggers in
lst 240 Primary Clocks
[ 0]= 0.380952 [ 80] = 0.380952 [160] = 0.238333 [ 0]= 3 [ 80] = 4 [160] = 3
[ 1]= 0.380952 [ 81] = 0.380952 [161] = 0.238333 [ 1]= 3 [ 81] = 3 [161] = 2
[ 2]= 0.380952 [ 82] = 0.380952 [162] = 0.333667 [ 2]= 3 [ 82] = 3 [162] = 3
[ 3]= 0.380952 [ 83] = 0.380952 [163] = 0.333667 [ 3]= 3 [ 83] = 3 [163] = 3
[ 4]= 0.380952 [ 84] = 0.380952 [164] = 0.333667 [ 4]= 3 [ 84] = 3 [164] = 4
[ 5]= 0.333667 [ 85] = 0.380952 [165] = 0.333667 [ 5]= 4 [ 85] = 3 [165] = 3
[ 6]= 0.333667 [ 86] = 0.380952 [166] = 0.381333 [ 6]= 3 [ 86] = 3 [166] = 3
[ 7]= 0.333667 [ 87] = 0.333333 [167] = 0.381333 [ 7]= 3 [ 87] = 3 [167] = 3
[ 8]= 0.333667 [ 88] = 0.333333 [168] = 0.381333 [ 8]= 3 [ 88] = 3 [168] = 4
[ 9]= 0.333667 [ 89] = 0.333333 [169] = 0.333667 [ 9]= 3 [ 89] = 3 [169] = 3
[10]= 0.333333 [ 90] = 0.333333 [170] = 0.333333 [10]= 3 [ 90] = 3 [170] = 3
[11]= 0.333333 [ 91] = 0.333333 [171] = 0.333333 [11]= 3 [ 91] = 3 [171] = 3
[12]= 0.333333 [ 92] = 0.333333 [172] = 0.333333 [12]= 3 [ 92] = 3 [172] = 3
[13]= 0.333333 [ 93] = 0.333333 [173] = 0.333333 [13]= 3 [ 93] = 3 [173] = 3
[14]= 0.333333 [ 94] = 0.333333 [174] = 0.333333 [14]= 3 [ 94] = 3 [174] = 2
[15]= 0.238095 [ 95] = 0.333333 [175] = 0.333667 [15]= 3 [ 95] = 2 [175] = 3
[16]= 0.238095 [ 96] = 0.333333 [176] = 0.333667 [16]= 3 [ 96] = 3 [176] = 4
[17]= 0.238095 [ 97] = 0.333667 [177] = 0.333667 [17]= 3 [ 97] = 3 [177] = 4
[18]= 0.238095 [ 98] = 0.333667 [178] = 0.333667 [18]= 3 [ 98] = 4 [178] = 3
[19]= 0.238095 [ 99] = 0.333667 [179] = 0.333667 [19]= 3 [ 99] = 4 [179] = 3
[20]= 0.238095 [100] = 0.333667 [180] = 0.381333 [20]= 3 [100] = 3 [180] = 3
[21]= 0.333667 [101] = 0.333667 [181] = 0.381333 [21]= 3 [101] = 3 [181] = 3
[22]= 0.333667 [102] = 0.238333 [182] = 0.381333 [22]= 3 [102] = 3 [182] = 2
[23]= 0.333667 [103] = 0.286000 [183] = 0.381333 [23]= 3 [103] = 3 [183] = 3
[24]= 0.333667 [104] = 0.286000 [184] = 0.381333 [24]= 3 [104] = 3 [184] = 3
[25]= 0.333667 [105] = 0.286000 [185] = 0.381333 [25]= 3 [105] = 4 [185] = 3
[26]= 0.333333 [106] = 0.286000 [186] = 0.333333 [26]= 3 [106] = 3 [186] = 5
[27]= 0.333333 [107] = 0.286000 [187] = 0.333333 [27]= 2 [107] = 3 [187] = 4
[28]= 0.333333 [108] = 0.286000 [188] = 0.333333 [28]= 3 [108] = 3 [188] = 4
[29]= 0.333333 [109] = 0.238333 [189] = 0.238095 [29]= 3 [109] = 4 [189] = 3
[30]= 0.333333 [110] = 0.238333 [190] = 0.238095 [30]= 3 [110] = 4 [190] = 4
[31]= 0.333667 [111] = 0.333667 [191] = 0.238333 [31]= 3 [111] = 4 [191] = 3
[32]= 0.333667 [112] = 0.333667 [192] = 0.286000 [32]= 3 [112] = 3 [192] = 3
[33]= 0.333667 [113] = 0.333667 [193] = 0.286000 [33]= 3 [113] = 3 [193] = 3
[34]= 0.333667 [114] = 0.333333 [194] = 0.286000 [34]= 3 [114] = 4 [194] = 3
[35]= 0.333667 [115] = 0.380952 [195] = 0.381333 [35]= 4 [115] = 4 [195] = 3
[36]= 0.381333 [116] = 0.380952 [196] = 0.286000 [36]= 4 [116] = 2 [196] = 3
[37]= 0.381333 [117] = 0.380952 [197] = 0.286000 [37]= 3 [117] = 3 [197] = 3
[38]= 0.333667 [118] = 0.333333 [198] = 0.286000 [38]= 2 [118] = 3 [198] = 3
[39]= 0.333667 [119] = 0.333333 [199] = 0.286000 [39]= 3 [119] = 3 [199] = 4
[40]= 0.333667 [120] = 0.333333 [200] = 0.238333 [40]= 3 [120] = 3 [200] = 4
[41]= 0.333667 [121] = 0.333333 [201] = 0.238333 [41]= 3 [121] = 3 [201] = 3
[42]= 0.333333 [122] = 0.333333 [202] = 0.333667 [42]= 3 [122] = 2 [202] = 2
[43]= 0.333333 [123] = 0.333333 [203] = 0.333667 [43]= 3 [123] = 3 [203] = 3
[44]= 0.333333 [124] = 0.333333 [204] = 0.333667 [44]= 3 [124] = 3 [204] = 3
[45]= 0.333333 [125] = 0.333667 [205] = 0.333667 [45]= 3 [125] = 3 [205] = 3
[46]= 0.333333 [126] = 0.333667 [206] = 0.333667 [46]= 4 [126] = 3 [206] = 3
[47]= 0.333333 [127] = 0.333667 [207] = 0.333667 [47]= 3 [127] = 4 [207] = 3
[48]= 0.333333 [128] = 0.333667 [208] = 0.333667 [48]= 3 [128] = 3 [208] = 3
[49]= 0.333333 [129] = 0.333667 [209] = 0.333667 [49]= 2 [129] = 3 [209] = 3
[50]= 0.333333 [130] = 0.333333 [210] = 0.333667 [50]= 3 [130] = 4 [210] = 4
[51]= 0.333333 [131] = 0.333333 [211] = 0.333667 [51]= 3 [131] = 3 [211] = 3
[52]= 0.333333 [132] = 0.333333 [212] = 0.333667 [52]= 3 [132] = 3 [212] = 3
[53]= 0.333333 [133] = 0.380952 [213] = 0.333667 [53]= 3 [133] = 3 [213] = 3
[54]= 0.333333 [134] = 0.380952 [214] = 0.333667 [54]= 3 [134] = 4 [214] = 3
[55]= 0.333333 [135] = 0.381333 [215] = 0.333667 [55]= 3 [135] = 4 [215] = 3
[56]= 0.333333 [136] = 0.286000 [216] = 0.333667 [56]= 3 [136] = 4 [216] = 3
[57]= 0.333333 [137] = 0.286000 [217] = 0.381333 [57]= 3 [137] = 3 [217] = 3
[58]= 0.333333 [138] = 0.238333 [218] = 0.381333 [58]= 3 [138] = 3 [218] = 3
[59]= 0.333333 [139] = 0.238333 [219] = 0.381333 [59]= 3 [139] = 2 [219] = 3
[60]= 0.333333 [140] = 0.238333 [220] = 0.381333 [60]= 3 [140] = 3 [220] = 3
[61]= 0.333333 [141] = 0.333667 [221] = 0.381333 [61]= 2 [141] = 3 [221] = 3
[62]= 0.333333 [142] = 0.333667 [222] = 0.381333 [62]= 3 [142] = 3 [222] = 3
[63]= 0.333333 [143] = 0.333667 [223] = 0.381333 [63]= 4 [143] = 3 [223] = 3
[64]= 0.333333 [144] = 0.333667 [224] = 0.381333 [64]= 3 [144] = 3 [224] = 3
[65]= 0.333667 [145] = 0.333667 [225] = 0.381333 [65]= 4 [145] = 3 [225] = 2
[66]= 0.333667 [146] = 0.333667 [226] = 0.381333 [66]= 3 [146] = 2 [226] = 3

```



[67]= 0.333667	[147] = 0.381333	[227] = 0.381333	[67]= 3	[147] = 3	[227] = 3
[68]= 0.333667	[148] = 0.381333	[228] = 0.381333	[68]= 3	[148] = 3	[228] = 3
[69]= 0.333667	[149] = 0.381333	[229] = 0.381333	[69]= 3	[149] = 4	[229] = 3
[70]= 0.333333	[150] = 0.381333	[230] = 0.381333	[70]= 3	[150] = 4	[230] = 3
[71]= 0.333333	[151] = 0.381333	[231] = 0.381333	[71]= 3	[151] = 4	[231] = 5
[72]= 0.333333	[152] = 0.381333	[232] = 0.381333	[72]= 3	[152] = 3	[232] = 3
[73]= 0.333333	[153] = 0.381333	[233] = 0.381333	[73]= 3	[153] = 3	[233] = 3
[74]= 0.333333	[154] = 0.381333	[234] = 0.381333	[74]= 2	[154] = 4	[234] = 3
[75]= 0.333667	[155] = 0.381333	[235] = 0.381333	[75]= 3	[155] = 3	[235] = 3
[76]= 0.333667	[156] = 0.381333	[236] = 0.381333	[76]= 2	[156] = 3	[236] = 4
[77]= 0.333667	[157] = 0.286000	[237] = 0.381333	[77]= 3	[157] = 4	[237] = 3
[78]= 0.381333	[158] = 0.286000	[238] = 0.380952	[78]= 3	[158] = 4	[238] = 3
[79]= 0.381333	[159] = 0.286000	[239] = 0.285714	[79]= 2	[159] = 3	[239] = 2



Appendix C- R = 3 with Wander and FM acceleration taken from the \AIS\Release\inout folder.
 The following typifies a 300M FM modulated, with random propagation delays in the oscillator.
 Note the excellent dispersion in the nibbles.

3.0eFmWaaE_AIS_231207_0815_DUAL_W0.15_0.001_3_Fm_300000_04532#E_3.0_48_55.5_0_67.5.txt

Statistical AIS-31 Analysis of ZK-Crypt Noise Source Date: 23.12.07;

Test Parameters Sampled Outputs: 300000000

Estimated Average OSC/HOST Freq.Ratio 3.0; DutyCycle 0.48; Auto Demerit 50.0;

Rel Delays: Juggle 0.19; DAS 0.39; Basic+Switch Delay 0.70; HiFreq 0.40; LoFreq 0.20; FmAlarm 0.3.

of sampled '1's in test nodes: (see Drawings); (P)RandomClk 277740452; frClk 1267209416;
 das 149999894; 4'th Toggle 148670907; Juggle Splash 146492214;

Q8 581372064; Q7 581372071; Q6 581372052; Q4 581372065; Q5 581372052; Q3 581372077; Q2 581372077;
 Q1 581372061; Q0 581372062; A 581372069; C(3) 581372050; L(3) 581372058; L(4) 582418028; B 579131319;
 fffF3 148580935; fffF4 150791335; fffF5 149924752; H 152165502; J 147745194;
 HiFreq10 24.2%; Freq8.3 10.1%; Freq7.1 54.4%; LoFreq6.3 11.2%;

Nibble Frequencies of Concatenated String ...||Juggle||4'thToggle||das||Juggle||4'thToggle||das|...|

9 tests from Test #1406250

[0]	5	3	3	0	4	11	5	6	4	14886126	[8]	6	2	5	3	9	3	3	2	4	14139399
[1]	5	4	4	5	5	10	9	7	4	14579827	[9]	3	2	6	3	2	6	5	5	3	13825813
[2]	9	4	3	4	7	5	1	3	3	14027362	[10]	4	7	7	3	3	4	3	5	8	15162308
[3]	5	7	3	6	3	0	4	6	8	13960215	[11]	1	5	8	9	5	1	8	7	4	13766263
[4]	7	4	7	5	7	11	4	3	7	14460576	[12]	1	5	5	10	4	3	5	8	5	13967745
[5]	6	5	4	5	6	3	8	0	3	13699021	[13]	5	9	6	4	6	4	4	6	3	13512572
[6]	7	3	5	6	5	6	9	8	5	14184314	[14]	6	5	5	8	1	6	4	2	6	13519840
[7]	1	6	6	4	10	2	9	6	5	13273448	[15]	9	9	3	6	2	6	2	5	5	14035091

Demerit Results = 19.2 14.0 7.6 18.4 19.2 32.8 17.6 17.6 10.4

Nibble Frequencies of Juggle Splash Toggle. 9 tests from Test # 468750

[0]	5	2	8	4	7	7	7	6	8	5258080	[8]	5	4	3	3	6	4	7	4	5	4966858
[1]	5	5	6	6	3	5	6	10	4	4966902	[9]	2	4	7	7	4	7	5	8	6	4758012
[2]	6	6	4	5	3	6	4	5	5	4556377	[10]	7	8	2	1	6	5	4	4	6	4675642
[3]	3	3	5	7	3	3	6	2	2	5168633	[11]	6	6	1	5	5	6	1	3	6	4026312
[4]	3	4	9	6	11	4	6	5	6	4586287	[12]	6	2	7	6	4	9	3	8	4	5138717
[5]	4	7	7	4	8	5	3	9	4	4645731	[13]	6	9	6	4	5	3	5	3	1	4056252
[6]	6	2	3	3	5	1	2	3	2	4601224	[14]	1	9	4	0	4	7	8	2	9	4593788
[7]	3	4	5	8	3	5	9	5	7	4593808	[15]	12	5	3	11	3	3	4	3	5	4407297

Demerit Results = 19.2 16.4 15.6 21.6 14.8 12.0 14.4 19.2 14.0

Nibble Frequencies of 4th Toggle. 9 tests from Test # 468750

[0]	4	4	2	2	6	2	4	3	3	3630454	[8]	7	2	6	5	5	10	4	7	6	4996946
[1]	9	5	8	4	3	4	6	5	5	4996888	[9]	8	3	3	10	8	5	6	8	3	4952240
[2]	2	4	6	8	8	3	8	5	8	5011853	[10]	3	4	6	5	7	7	5	9	3	4780131
[3]	8	6	9	2	3	4	4	6	3	4937216	[11]	6	5	5	5	6	8	5	8	4	4526536
[4]	3	5	3	7	4	5	6	2	7	5064089	[12]	2	4	6	3	7	4	7	6	2	4885047
[5]	2	4	3	9	7	4	5	4	6	4727887	[13]	3	10	2	5	3	5	4	3	5	4578738
[6]	5	2	7	8	3	2	2	4	9	4705817	[14]	8	9	8	2	7	3	5	4	6	4757941
[7]	3	7	4	2	5	6	5	3	7	4757940	[15]	7	6	2	3	2	4	6	2	3	3690197

Demerit Results = 19.2 14.8 16.4 21.6 14.0 12.4 7.6 14.0 13.2

Nibble Frequencies of das Slip Toggle. 9 tests from Test # 468750

[0]	8	4	7	7	6	7	7	2	5	5407490	[8]	6	2	7	5	4	6	8	3	5	4899905
[1]	6	6	9	5	7	9	8	8	4	4436998	[9]	4	5	7	3	4	3	6	3	4	4287554
[2]	6	6	2	6	5	8	6	4	5	4735533	[10]	8	4	2	4	8	1	2	7	6	5183428
[3]	7	4	8	6	1	1	4	5	9	4451847	[11]	3	6	5	5	9	5	1	2	5	4735574
[4]	3	4	4	2	3	3	5	4	4	4556356	[12]	4	7	5	2	8	7	2	5	5	4168108
[5]	3	4	4	2	5	4	5	7	6	4548899	[13]	7	10	3	6	3	5	9	4	5	4556372
[6]	4	2	6	9	5	6	5	5	6	4616086	[14]	4	2	5	5	6	5	4	11	5	4571369
[7]	4	5	3	6	5	6	5	3	4	4108498	[15]	3	9	3	7	1	4	3	7	2	5735903

Demerit Results = 10.0 16.0 14.0 12.0 16.4 15.6 16.0 18.0 6.4

Demerit Distribution -

Juggle Nibble Test: # FM Warning Triggers (> 50.0) 1 Groups # 100
 # Failed Strings (>65) 0
 Count ** 0-25= 904044 ** 26-35= 33440 ** 36-45= 12 ** 46-55= 3 ** 56-65= 0 max= 50.8 av= 15.0

4th Toggle Nibble Test: # FM Warning Triggers (> 50.0) 0
 # Failed Strings (>65) 0
 Count ** 0-25= 842857 ** 26-35= 81288 ** 36-45= 12980 ** 46-55= 374 ** 56-65= 0 max= 46.0 av= 16.9

das Nibble Test: # FM Warning Triggers (> 50.0) 0
 # Failed Strings (>65) 0
 Count ** 0-25= 880359 ** 26-35= 55181 ** 36-45= 1959 ** 46-55= 0 ** 56-65= 0 max= 36.8 av= 16.2

3 signal Nibble Test: # FM Warning Triggers (> 50.0) 1 Groups # 1354
 # Failed Strings (>65) 0
 Count ** 0-25= 2609345 ** 26-35= 183273 ** 36-45= 17828 ** 46-55= 2053 ** 56-65= 0 max= 53.6 av= 16.0

Average Test Cumulative (64.2)/4 = 16.0 Max Test Demerit = 53.6



Wobbling Accordion

if Period of Primary Clock = 1 μs, fr Periods (μs)

in 1st 240 fr Clocks		# fr Triggers in 1st 240 Primary Clocks	
[0]= 0.380952	[80] = 0.380952	[0]= 3	[80] = 4
[1]= 0.380952	[81] = 0.380952	[1]= 3	[81] = 3
[2]= 0.380952	[82] = 0.380952	[2]= 3	[82] = 3
[3]= 0.380952	[83] = 0.380952	[3]= 3	[83] = 3
[4]= 0.380952	[84] = 0.380952	[4]= 3	[84] = 3
[5]= 0.333667	[85] = 0.380952	[5]= 4	[85] = 3
[6]= 0.333667	[86] = 0.380952	[6]= 3	[86] = 3
[7]= 0.333667	[87] = 0.333333	[7]= 3	[87] = 3
[8]= 0.333667	[88] = 0.333333	[8]= 3	[88] = 3
[9]= 0.333667	[89] = 0.333333	[9]= 3	[89] = 3
[10]= 0.333333	[90] = 0.333333	[10]= 3	[90] = 3
[11]= 0.333333	[91] = 0.333333	[11]= 3	[91] = 3
[12]= 0.333333	[92] = 0.333333	[12]= 3	[92] = 3
[13]= 0.333333	[93] = 0.333333	[13]= 3	[93] = 3
[14]= 0.333333	[94] = 0.333333	[14]= 3	[94] = 3
[15]= 0.238095	[95] = 0.333333	[15]= 3	[95] = 2
[16]= 0.238095	[96] = 0.333333	[16]= 3	[96] = 3
[17]= 0.238095	[97] = 0.333667	[17]= 3	[97] = 3
[18]= 0.238095	[98] = 0.333667	[18]= 3	[98] = 4
[19]= 0.238095	[99] = 0.333667	[19]= 3	[99] = 4
[20]= 0.238095	[100] = 0.333667	[20]= 3	[100] = 3
[21]= 0.333667	[101] = 0.333667	[21]= 3	[101] = 3
[22]= 0.333667	[102] = 0.238333	[22]= 3	[102] = 3
[23]= 0.333667	[103] = 0.286000	[23]= 3	[103] = 3
[24]= 0.333667	[104] = 0.286000	[24]= 3	[104] = 3
[25]= 0.333667	[105] = 0.286000	[25]= 3	[105] = 4
[26]= 0.333333	[106] = 0.286000	[26]= 3	[106] = 3
[27]= 0.333333	[107] = 0.286000	[27]= 2	[107] = 3
[28]= 0.333333	[108] = 0.286000	[28]= 3	[108] = 3
[29]= 0.333333	[109] = 0.238333	[29]= 3	[109] = 4
[30]= 0.333333	[110] = 0.238333	[30]= 3	[110] = 4
[31]= 0.333667	[111] = 0.333667	[31]= 3	[111] = 4
[32]= 0.333667	[112] = 0.333667	[32]= 3	[112] = 3
[33]= 0.333667	[113] = 0.333667	[33]= 3	[113] = 3
[34]= 0.333667	[114] = 0.333333	[34]= 3	[114] = 4
[35]= 0.333667	[115] = 0.380952	[35]= 4	[115] = 4
[36]= 0.381333	[116] = 0.380952	[36]= 4	[116] = 2
[37]= 0.381333	[117] = 0.380952	[37]= 3	[117] = 3
[38]= 0.333667	[118] = 0.333333	[38]= 2	[118] = 3
[39]= 0.333667	[119] = 0.333333	[39]= 3	[119] = 3
[40]= 0.333667	[120] = 0.333333	[40]= 3	[120] = 3
[41]= 0.333667	[121] = 0.333333	[41]= 3	[121] = 3
[42]= 0.333333	[122] = 0.333333	[42]= 3	[122] = 2
[43]= 0.333333	[123] = 0.333333	[43]= 3	[123] = 3
[44]= 0.333333	[124] = 0.333333	[44]= 3	[124] = 3
[45]= 0.333333	[125] = 0.333667	[45]= 3	[125] = 3
[46]= 0.333333	[126] = 0.333667	[46]= 4	[126] = 3
[47]= 0.333333	[127] = 0.333667	[47]= 3	[127] = 4
[48]= 0.333333	[128] = 0.333667	[48]= 3	[128] = 3
[49]= 0.333333	[129] = 0.333667	[49]= 2	[129] = 3
[50]= 0.333333	[130] = 0.333333	[50]= 3	[130] = 4
[51]= 0.333333	[131] = 0.333333	[51]= 3	[131] = 3
[52]= 0.333333	[132] = 0.333333	[52]= 3	[132] = 3
[53]= 0.333333	[133] = 0.380952	[53]= 3	[133] = 3
[54]= 0.333333	[134] = 0.380952	[54]= 3	[134] = 4
[55]= 0.333333	[135] = 0.381333	[55]= 3	[135] = 4
[56]= 0.333333	[136] = 0.286000	[56]= 3	[136] = 4
[57]= 0.333333	[137] = 0.286000	[57]= 3	[137] = 3
[58]= 0.333333	[138] = 0.238333	[58]= 3	[138] = 3
[59]= 0.333333	[139] = 0.238333	[59]= 3	[139] = 2
[60]= 0.333333	[140] = 0.238333	[60]= 3	[140] = 3
[61]= 0.333333	[141] = 0.333667	[61]= 2	[141] = 3
[62]= 0.333333	[142] = 0.333667	[62]= 3	[142] = 3
[63]= 0.333333	[143] = 0.333667	[63]= 4	[143] = 3
[64]= 0.333333	[144] = 0.333667	[64]= 3	[144] = 3
[65]= 0.333667	[145] = 0.333667	[65]= 4	[145] = 3
[66]= 0.333667	[146] = 0.333667	[66]= 3	[146] = 2
[67]= 0.333667	[147] = 0.381333	[67]= 3	[147] = 3
[68]= 0.333667	[148] = 0.381333	[68]= 3	[148] = 3
[69]= 0.333667	[149] = 0.381333	[69]= 3	[149] = 4
[70]= 0.333333	[150] = 0.381333	[70]= 3	[150] = 4
[71]= 0.333333	[151] = 0.381333	[71]= 3	[151] = 4
[72]= 0.333333	[152] = 0.381333	[72]= 3	[152] = 3
[73]= 0.333333	[153] = 0.381333	[73]= 3	[153] = 3
[74]= 0.333333	[154] = 0.381333	[74]= 2	[154] = 4
[75]= 0.333667	[155] = 0.381333	[75]= 3	[155] = 3
[76]= 0.333667	[156] = 0.381333	[76]= 2	[156] = 3
[77]= 0.333667	[157] = 0.286000	[77]= 3	[157] = 4
[78]= 0.381333	[158] = 0.286000	[78]= 3	[158] = 4
[79]= 0.381333	[159] = 0.286000	[79]= 2	[159] = 3

Appendix E– Log of Test File taken from the \AIS\Release\inout folder.

CHOSEN RESULTS – DEMONSTRATING DETERMINISTIC FAILURES & REMEDIES

File Name	Size	Size
0.1bNoFmNoW_aac_AIS_161207_0937_DUAL_NoFm_3000_046323ED_0.1_48_92_830_133.5.txt	19 KB	19 KB
0.1cFmNoW_aaw_AIS_171207_1118_DUAL_Fm_3000_046323ED_0.1_48_73.2_1_83.85.txt	15 KB	15 KB
0.1dNoFmW_aax_AIS_171207_1118_DUAL_W0.15_0.001_3_NoFm_3000_046323ED_0.1_48_103.6_611_121.4.txt	19 KB	19 KB
0.1eFmW_aay_AIS_171207_1118_DUAL_W0.15_0.001_3_Fm_3000_046323ED_0.1_48_134.8_160_83.24.txt	17 KB	17 KB
0.5bNoFmNoW_aac_AIS_161207_1114_DUAL_NoFm_3000_16E77016_0.5_48_117.2_7535_179.9.txt	19 KB	19 KB
0.5cFmNoW_aad_AIS_161207_1114_DUAL_Fm_3000_16E77016_0.5_48_72.4_1_58.19.txt	15 KB	15 KB
0.5dNoFmW_aae_AIS_161207_1115_DUAL_W0.15_0.001_3_NoFm_3000_16E77016_0.5_48_123.6_4694_156.3.txt	19 KB	19 KB
0.5eFmW_aaf_AIS_161207_1115_DUAL_W0.15_0.001_3_Fm_3000_16E77016_0.5_48_92.4_2_61.47.txt	15 KB	15 KB
1.0bNoFmNoW_aao_AIS_171207_0756_DUAL_NoFm_3000_046323ED_1.0_48_35.2_0_65.68.txt	15 KB	15 KB
1.0dNoFmW_aar_AIS_171207_0757_DUAL_W0.15_0.001_3_NoFm_3000_046323ED_1.0_48_55.6_0_64.48.txt	15 KB	15 KB
1.0eFmWaaF_AIS_231207_0922_DUAL_W0.15_0.001_3_Fm_300000_046323ED_1.0_48_50_0_55.22.txt	15 KB	15 KB
1.0eFmWBADaaF_AIS_231207_0922BAD_DUAL_W0.15_0.001_3_Fm_300000_046323ED_1.0_48_50_0_55.22.txt	3 KB	3 KB
3.0bFmNoWaaD_AIS_231207_0759_DUAL_Fm_300000_046323ED_3.0_48_28_0_77.19.txt	15 KB	15 KB
3.0cNoFmWaaB_AIS_211207_1744_DUAL_W0.15_0.001_3_NoFm_300_5C7D77DA_3.0_48_57.2_0_67.43.txt	15 KB	15 KB
3.0cNoFmWaaB_AIS_211207_1744BAD_DUAL_W0.15_0.001_3_NoFm_300_5C7D77DA_3.0_48_57.2_0_67.43.txt	2 KB	2 KB
3.0cNoFmWaaC_AIS_211207_1823_DUAL_W0.15_0.001_3_NoFm_300000_046323ED_3.0_48_55.6_0_67.51.txt	16 KB	16 KB
3.0cNoFmWBADaaC_AIS_211207_1823BAD_DUAL_W0.15_0.001_3_NoFm_300000_046323ED_3.0_48_55.6_0_67...	940 KB	340 KB
3.0dNoFmW_aal_AIS_161207_1339_DUAL_W0.15_0.001_3_NoFm_3000_046323ED_3.0_48_55.6_0_67.52.txt	16 KB	16 KB
3.0eFmWaaE_AIS_231207_0815_DUAL_W0.15_0.001_3_Fm_300000_046323ED_3.0_48_53.6_0_64.17.txt	15 KB	15 KB
3.0eFmWBADaaE_AIS_231207_0815BAD_DUAL_W0.15_0.001_3_Fm_300000_046323ED_3.0_48_53.6_0_64.17.txt	3 KB	3 KB
5.0bNoFmNoW_aaj_AIS_131207_1131_DUAL_NoFm_3000_046323ED_5.0_48_24_0_57.16.txt	15 KB	15 KB
5.0dNoFmW_aaz_AIS_171207_0805_DUAL_W0.15_0.001_3_NoFm_3000_046323ED_5.0_48_45.6_0_61.77.txt	15 KB	15 KB
10.0aNoFmNoW_aau_AIS_131207_1205_DUAL_NoFm_3000_39D75CB3_10.0_48_56.4_0_150.1.txt	16 KB	16 KB
10.0bNoFmNoW_aaa_AIS_131207_1211_DUAL_NoFm_3000_046323ED_10.0_48_95.6_9374_209.9.txt	17 KB	17 KB
10.0cFmNoW_aab_AIS_131207_1243_DUAL_Fm_3000_046323ED_10.0_48_95.6_3_113.7.txt	15 KB	15 KB
10.0dNoFmW_aac_AIS_131207_1244_DUAL_W0.15_0.001_3_NoFm_3000_046323ED_10.0_48_132_94_65.4.txt	17 KB	17 KB
10.0eFmW_aad_AIS_131207_1245_DUAL_W0.15_0.001_3_Fm_3000_046323ED_10.0_48_63.2_0_60.7.txt	15 KB	15 KB
100.0aNoFmNoW_aah_AIS_131207_1251_DUAL_NoFm_3000_046323ED_100.0_48_27.2_0_50.19.txt	15 KB	15 KB
100.0bNoFmNoW_aai_AIS_131207_1257_DUAL_NoFm_3000_16E77016_100.0_48_176_28122_568.1.txt	17 KB	17 KB
100.0cFmNoW_aaj_AIS_131207_1302_DUAL_Fm_3000_16E77016_100.0_48_176_9_89.93.txt	16 KB	16 KB
100.0dNoFmW_aak_AIS_131207_1306_DUAL_W0.15_0.001_3_NoFm_3000_16E77016_100.0_48_42.8_0_60..txt	15 KB	15 KB
100.0eFmW_aal_AIS_131207_1308_DUAL_W0.15_0.001_3_Fm_3000_16E77016_100.0_48_42.8_0_60..txt	15 KB	15 KB

The above files were chosen from the many hundreds of tests which we executed on the final circuitry, using Oscillator to Host frequency ratios from $R = 0.1$ to $R = 100$.

We note that under no circumstances should R be less than 0.5. Note the engine recovered from the random syndrome when R was equal to 0.5, using the FM shunt accelerator; accelerating the average frequency ratio to about 6.3.

We have centered our design criteria around the assumed $R = 3.0$ ratio, and note that a minimum of random frequency wobble precludes short circuit syndromes.