# Appendix A

# Random Number Generators

The Electronic Gaming Machine's (EGM) Random Number Generator is a vital part of a Gaming Machine and its implementation and use must be approved by the Director. This approval will be based on a number of criteria. This appendix describes the minimum requirements for random number generators.

## 1 Minimum Range Requirement

The range of values produced by the RNG must be adequate to provide sufficient precision and flexibility when setting event outcome probabilities, i.e. so as to accurately achieve a desired expected return to player.

A 31 (minimum) bit random number generator is required for the following reasons:

- A 16 bit RNG provides inadequate precision and is extremely unlikely to be approved.
- Much theory is available on "good" 31 bit RNGs.
- Given contemporary microprocessor capabilities, there is little if any advantage in going for a range somewhere in between 16 and 31 bits.
- Implementation of high value prizes, i.e. very low probability events, may require this level of precision.

## 2 Uniform Distribution Requirement

The RNG must exhibit a highly (i.e. as exactly as possible) uniform distribution of expected outcomes over its entire outcome range.

If the RNG is periodic, this requirement implies that the RNG period is at least equal to the range, or is greater than the range by some whole multiple.

## 3 Theoretical, Experimental and Statistical Requirements

The theory of random numbers recognizes a number of general requirements for a "good", "well behaved" RNG. Such recognized requirements must be met. These include the following.

1. Uniformity of distribution.
2. Absence of periodicities in outcomes.
3. Absence of correlations between outcomes.
4. Passing various recognized statistical tests.
5. Suitability for application.

It is not considered appropriate to specify in this document the theoretical, experimental or statistical tests that may be applied to verify adequate performance of a RNG - a "good" random number generator should perform adequately for a wide range of statistical tests, not just those published in a specification. Theoretical analysis and tests are obviously specific to an individual algorithm.

# 4 Unpredictability Requirement

A particular requirement of a RNG for use in a gaming machine application is the unpredictability of the sequence of RNG outcomes. Specifically, a RNG is unacceptable in this regard if it simply cycles around one simple sequence where for any outcome $X_b$, that outcome $X_b$ always occurs and only occurs immediately after some other outcome $X_a$.

The reason for this requirement is that following a low probability game outcome (e.g. a jackpot win, major prize win, or a particular graphic game result presentation), where that game outcome might be represented by only one RNG value or a small range of RNG values, it is important that subsequent game play on that machine is unpredictable. That is, so that the machine does not subsequently go through one defined sequence of game outcomes, or one of a only a few possible sequences of game outcomes.

This unpredictability requirement results in the following requirements.

1. Minimum period requirement (4.1)
2. Background RNG activity requirement (4.2)

These requirements are defined in the following sections. At least one of these 2 requirements must be satisfied, preferably both, by an acceptable RNG.

## 4.1 Minimum Period Requirement

The period of the RNG must be much greater than its range.

Examples of algorithms that, with appropriate choices of parameters, can achieve this and other requirements include the following.

### 4.1.1 N-th Order Linear Algorithm

This algorithm is defined by the following formula.

$$X_n = \left( \sum_{i=1}^{j} a_i \times X_{n-i} \right) \bmod m$$

where: $j \geq 2$

### 4.1.2 Combined Linear Algorithm

This algorithm is defined by the following formula.

$$X_n = \left( \sum_{i=1}^{j} W_{i,n} \right) \bmod m_0$$

where: $W_{i,n} = (a_i \times W_{i,n-1} + c_i) \bmod m_i$ [3]

and: $j \geq 2$

### 4.1.3 Linear Congruential Algorithm RNGs

The popular "*Linear Congruential Algorithm*" (LCA) RNG is NOT acceptable in regard to the Minimum Period Requirement and is NOT recommended. The general form of the LCA is:

$$X_n = (a*X_{n-1} + c) \bmod m \quad [4]$$

An implementation of an LCA would be acceptable only in combination with the use of the "Background RNG Activity" feature described below.

## 4.2 Background RNG Activity Requirement

In addition to obtaining random numbers from the RNG as and when they are needed in the course of game playing, some mechanism must exist whereby additional random numbers are obtained and discarded.

This background random number generation must be frequent, i.e. having an average rate of about 10 per second or greater and should preferably be very irregular in rate rather than periodic. Somewhere in a machine's main program loop or main idle loop is an example of a good place to request additional random numbers.

# 5 Appropriate use of random numbers Requirement

## 5.1 General

The random numbers generated by the RNG must always be used in a manner that has regard to the general and specific limitations of the behaviour of the chosen RNG.

## 5.2 Random number range re-scaling

If a random number with a range shorter than that provided by the RNG is required for some purpose within the gaming machine, the method of re-scaling, i.e. converting the number to the lower range, is to be designed such that all numbers within the lower range are equally probable.

If particular random number selected is outside the range of equi-distribution of re-scaling values, it is permissable to discard that random number and select the next in sequence for the purpose of re-scaling. For a trivial example, if there is a RNG which

generates random 8 bit numbers (range 0 - 255) and a Keno number was required (re-scaling to range 0 - 79) which is determined by the formula

$$K_n = R_n \bmod 80$$

It is clear in this instance that only random numbers in the range 0 to 239 should be used. Thus if the next number in sequence was 246 it represents an unnatural bias and should be discarded and the next number in sequence picked (continuing until a number in the range 0 to 239 was reached).

In practical terms the Director is not concerned if the RNG range is 31 bits and a small number, such as in the range of 0 - 79, provides a very small bias. However, if rescaling is to be performed for very large numbers, the principles of this section must be implemented.

# 6 RNG Seeding

Random Number Generators typically require initializing using one or more seed values (i.e. the "seed-set").

## 6.1 Seed generation

The same seed-set must NEVER be used more than once, i.e. the seed-set must NEVER be re-used. The same seed-set must NEVER be loaded into more than one machine. These objectives result in the following minimum requirements for seed generation.

A seed-set must be derived using a 1-to-1 mapping from at least the following information.

1) The high order components of the date, including the year and day numbers.
2) The low order components of the time of day, including seconds and fractions of seconds.
3) A machine number unique to each EGM.
4) If a RNG is involved in the generation of the seed-set, it must be a different RNG algorithm from that used for actual game play.

The method of seed-set generation must be approved.

The Director may approve a RNG implementation where the RNG is never seeded. The following requirements must be met if this scheme of "seeding" is adopted:

1) The seeds must be held in "non-resettable" RAM or other approved storage device.
2) The background RNG Activity Requirement (4.2) must be implemented.

## 6.2 Infrequent seeding

Seeding and re-seeding must be kept to an absolute minimum. Both the method of re-seeding and the instances when it may occur must be approved by the Director. Re-seeding should not in general be under operator control. Re-seeding should not be a routine or regular practice.

# 7 RNG Test Modes

Test versions of the software should provide for production by the RNG of known defined sequences of numbers. Such a list of numbers must be able to be loaded into the machine by the testing officer. If such a facility is not provided, an In-Circuit Emulator will be used to perform these tests.

Such a test facility must not exist in the operational software.

# 8 Software RNG versus Hardware RNG

The Director recognizes that a choice may be available between a software based implementation of a mathematical pseudo random number algorithm and a hardware device that purports to actually generate random quantities.

While the Director has no disagreement in principle with either choice, it is suspected that it may be more difficult to demonstrate adherence to the various requirements above with a hardware device than with software where the algorithm can be exactly defined and hence its behaviour extensively analysed.

# 9 Chance Game Behavior

The following rules apply to the use of random number generators relative to chance game behaviour.

## 9.1 Chance game behaviour to be uncorrelated

Events of chance within games must be independent of (i.e. uncorrelated with) any other events within the game or any events within previous games.

## 9.2 Chance game behaviour not to be influenced

Events of chance within games must not be influenced, affected, controlled or determined by anything other than (in conjunction with the prevailing payout table) numerical values obtained in an approved manner from the approved RNG.

### 9.2.1 Adaptive behaviour

Events of chance within games must not be automatically influenced in any way by recent history or other statistics of player, game, EGM or venue performance.

### 9.2.2 Random number selection sequence

The numerical values from the RNG used to determine chance game events must be obtained in the normal manner and the normal sequence applicable to the type of RNG. The selection, discarding or sequence of usage of such numerical values must not be influenced in any way.

The action of background RNG generation (4.2) is considered to be part of the normal operation of a RNG incorporating such a feature, and so the requirement here does not preclude the existence of such a background RNG activity feature.

## 9.3 Chance game behaviour to be pre-determined

### 9.3.1 Chance game behaviour to be frozen

Prior to the commencement of each game play, all random behaviour to be used during a game is to be fully determined and frozen.

This requires that all random numbers (including random decisions, random events or any other random behaviour) to be used during the course of the game play are generated and recorded prior to the start of the game play.

### 9.3.2 No subsequent decisions

Subsequent to the commencement of a game play, no subsequent actions or decisions may be made that would change the behaviour of any of the events of chance within the game play other than player decision.

## 9.4 Chance game behaviour to be recorded

Prior to the commencement of each game play, sufficient information is to be recorded so as to allow all random behaviour to be used during the game to be able to be fully reconstructed in the event of a game replay for whatever reason, including all cases of game recovery following game interruption.

This requires that all pre-determined information be recorded. The manner of recording must be as for any other game replay information, that is, in an appropriately non-volatile and/or backed-up medium that will facilitate game replay and game recovery.