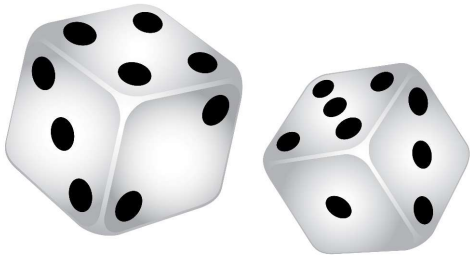
STM32 乱数発生器のプレゼンテーションへようこそ。 このプレゼンテーションでは、乱数を提供するために広く使用されているこのペリフェラルの機能について説明します。

概要



- 乱数の提供
  - 予測不可能な結果を生み出すことが望まれる場合に使用

アプリケーション側の利点

- 数のランダム性を高める
- 値の推測可能性を大幅に低減

life.augmented

STM32 製品に搭載された乱数発生器（RNG）は、予測不可能な結果を生み出すことが望まれる場合に使用される乱数を提供します。アプリケーションには、数のランダム性を高めたり、特定の値を推測する可能性を低減するというメリットが RNG によってもたらされます。

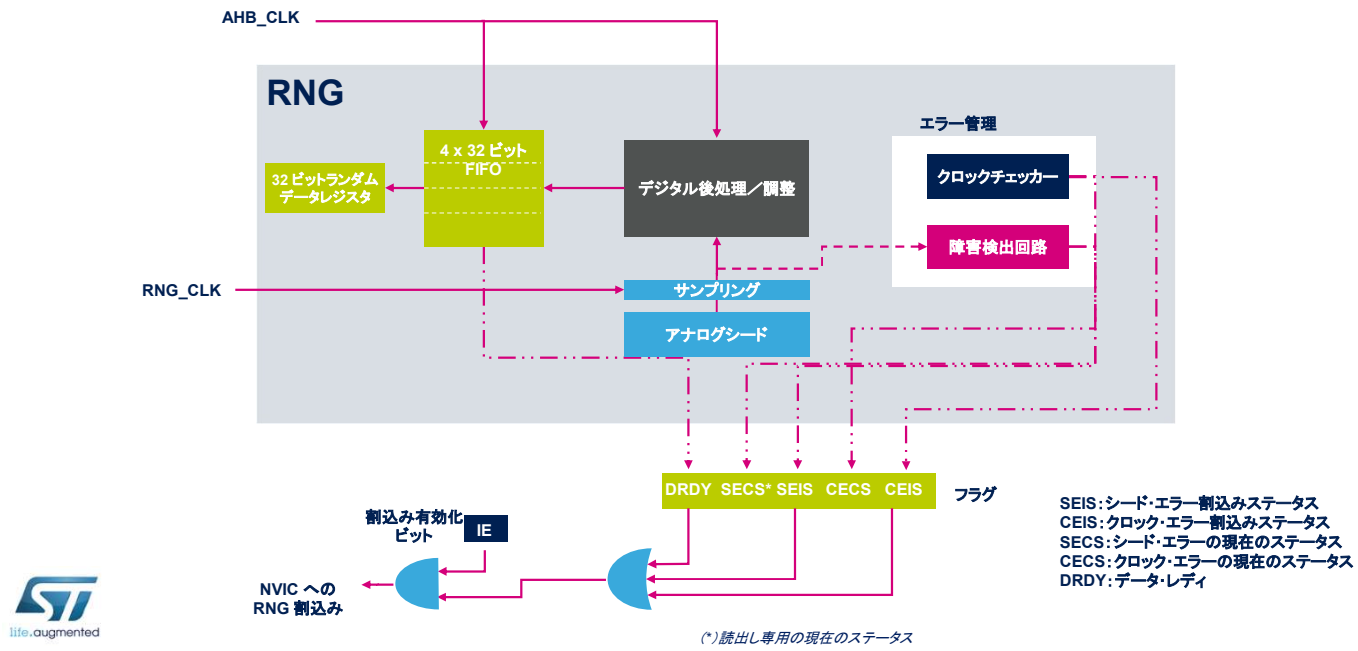
主な機能

- ノイズ·ソースに基づく32ビット乱数発生器
  - 4個の32ビット乱数を213クロック·サイクルの最小周波数で生成可能。
    - 実際の値(213 を超える場合)は、システム·クロックとRNGサンプル·クロックの比による16 x f_{AHB} / f_{RNG}となる。f_{AHB}= 64MHzかつf_{RNG}= f_{HSI16} / 8 = 2MHzである場合、サンプルは512 AHBサイクルごとに入手可能。
  - 電力消費を削減するために無効化が可能(RNG_CR では RNGEN=0)。
- 次の場合に3つのフラグをトリガ可能。
  - DRDY:有効なランダム·データが用意できた場合。
  - SECS:異常なシーケンスがシード上で発生した場合(64ビットを超える連続したビットが「0」または「1」の同一値、あるいは「01」または「10」のパターンが32回を超えて連続)。
  - *CECS*:f_{RNG} 周波数が f_{AHB} / 32未満の場合(この確認は無効化可能)。
- 3つの割込み
  - CEIS: クロック·エラーの発生を示す。
  - SEIS: シード·エラーの発生を示す。
  - DRDY: 有効なランダム·データが用意できたことを示す。

life.augmented

RNG ペリフェラルは、後で詳細に説明されるランダムな 32 ビット値を提供する連続アナログノイズに基づいています。
RNG は 213 システムクロックサイクルの最小周波数で 4 個の 32 ビット乱数を生成できます。経験則では、RNG クロックが低いほど、サンプリングされたランダムソースのエントロピーが向上します。
一連の新しいランダムデータが用意され、検証されると、データレディフラグがステータスレジスタにセットされます。このフラグは常に使用する必要があります。
RNG は、提供されたデータのランダム性の基礎検証を実行します。たとえば、同一の値(0 または 1)が 64 ビットを超えて連続する場合や、32 回を超えて連続的に 0 と 1 が交互に繰り返される場合には、シードエラーの現在のステータスフラグがセットされます。
RNG クロックが 32 分周された HCLK クロックよりも遅い場合に、クロックエラーの現在のステータスフラグがセットされます。このチェックは、特にエントロピーを最大にするために RNG クロックが低いクロック周波数で初期化される場合に、無効にできます。
割込みソースを有効化して、異常なシードシーケンスや周波数エラーを示すこともできます。

ブロック図



この RNG が単純化されたブロック図には、その基本的な機能モジュールと制御モジュールが示されています。
乱数発生器は、複数のリングオシレータで構成されるアナログ回路に基づいています。サンプリングされたリングオシレータ出力の排他的論理和をとり、計算ラウンド当たり 4 個の 32 ビット乱数を生成可能なデジタル後処理ブロックに送り込むシードを生成します。
アナログシードのサンプリングは専用 RNG クロック信号からクロック供給されるため、乱数の特性としては HCLK 周波数と無関係になります。
後処理ブロックの内容は、4 ワードの FIFO を通じてデータレジスタに転送されます。FIFO がフルになるとすぐにデータレディフラグ（DRDY）がトリガされ、それ以上のデータを RNG から読み戻すことができなくなると、自動的にリセットされます。
並行して、エラー管理ブロックにより、正しいシード動作と RNG ソースクロックの周波数が検証されます。
異常なシーケンスがシードで検出された場合や、RNG 周波数が低すぎる場合は、ステータスビットがセットされ、割込みがトリガされます。
（品質上の理由などにより）RNG クロックが AHB_CLK/32 未満に固定されている場合には、RNG 周波数エラーチェックを無効にする必要があります。

低消費電力モード

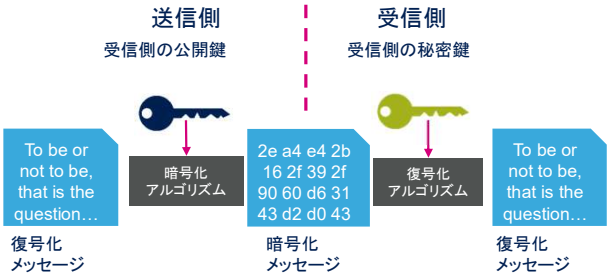
モード	RNGペリフェラルの説明
RUN	アクティブ。
SLEEP	RCCまたはRNGで無効化（RNGEN = 0）。RNGを有効にしたままにすると、RNG初期化時間によって生じる、新しいランダムサンプルが利用可能になるまでの遅延が解消される
低消費電力RUN	消費電力を最小にするためにRCCで無効化。
低消費電力SLEEP	
STOP 0/STOP 1	
STANDBY	パワーダウン状態です。ペリフェラルは、STANBYモード終了後に再初期化する必要があります。
SHUTDOWN	パワーダウン状態。ペリフェラルは、SHUTDOWNモード終了後に再初期化する必要がある。

life.augmented

真性乱数発生器は、RUN モードでのみアクティブです。初期化時間の遅延を回避するために、SLEEP モードで有効にしたままにできます。その他の低消費電力モードでは無効になり、STANDBY モードや SHUTDOWN モードでは完全にパワーダウン状態になります。

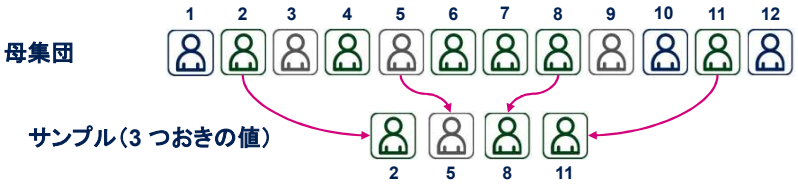
アプリケーション例

- 暗号化



- ゲーム

- 統計的サンプリング



life.augmented

RNG は、暗号、ゲーム、統計的サンプリングを含む幅広いアプリケーションに利用できます。たとえば、暗号化アルゴリズムのすべてのセキュリティは、鍵の推測を不可能にすることに結び付いています。そのため、鍵は乱数である必要があり、そうしないと攻撃者による推測が可能になります。

関連ペリフェラル

- RNGに関連したペリフェラル
  - RCC（RNGクロック制御、RNGイネーブル／リセット）
  - 割込み（RNG割込みマッピング）

life.augmented

これは、乱数発生器に関連したペリフェラルのリストです。詳細については、必要に応じてこれらのトレーニングを参照してください。

参照

- AN4230：NIST統計テストスイートを使用したSTM32マイクロコントローラの乱数生成の検証（STM32 microcontrollers random number generation validation using NIST statistical test suite）
  - AN4230は、STM32マイクロコントローラ·シリーズに組み込まれている乱数発生器ペリフェラルによって生成された数のランダム性を検証するためのガイドライン。この検証は、米国標準技術研究所（NIST）の統計テストスイート（STS）SP 800-22（公開後、2010年4月にSP800-22rev1aとして更新）に基づく。
  - NISTテストスイートは、RNGペリフェラルを組み込んだ一連のSTM32ボード上で実行。その結果は、ファームウェア·フォルダ「NIST_Test_Suite_OutputExample」に格納。

life.augmented

詳細については、NIST 統計テストスイートを使用した STM32 マイクロコントローラシリーズの乱数生成の検証に関するアプリケーションノート AN4230 を参照してください。