



本連載の最終回として、クロック同期について解説する。FlexRay ネットワークでは、接続されているノードごとにクロックを持っている。これらのノードが一つのシステムとして機能するには、クラスタとしての共通時間と時刻合わせ(同期)を行う必要がある。同期は通信準備状態(スタートアップ)で行われる。また、スタートアップ後もコミュニケーション・サイクルごとに同期補正を行うようなしくみも備えている。(編集部)

本稿では、クロック同期について説明します。

分散制御方式のネットワークでは、ノードごとにクロックを備えています。それぞれのクロックの同期をとりながらシステムとしての機能を果たします。FlexRayのプロトコルもこの制御方式をとっており、あるノードから送られてくる同期フレームのタイミングを観察して、各ノードが個々にクラスタ(ネットワーク・システム)と同期をとります。

また、FlexRayのメディア・アクセスはタイム・トリガ方式なので、ノードどうしのデータのやりとりは時間で管

理されています。ただし、クラスタ全体で絶対的な時間が定められているわけではありません。クロック同期を用いて、各ノードがクラスタに対して時間を合わせているのです(詳細は後述)。

このように、クロック同期はFlexRay ネットワーク・システムを構築するうえでとても重要です。クロック同期は電源ON後にノードがスタートアップする際に行われますが、その後で温度変動や電圧変動、クロック発生器の製造上のばらつき、経年変化などによって徐々にタイミングがそろわなくなってしまうことも考えられます。そのため、クロックの再同期のメカニズムが必要となります。

1 ノード内部の時間とクラスタの時間

図1に示すように、FlexRayの各ノード内部の時間的な構成要素はサイクル、マクロティック、マイクロティックです。

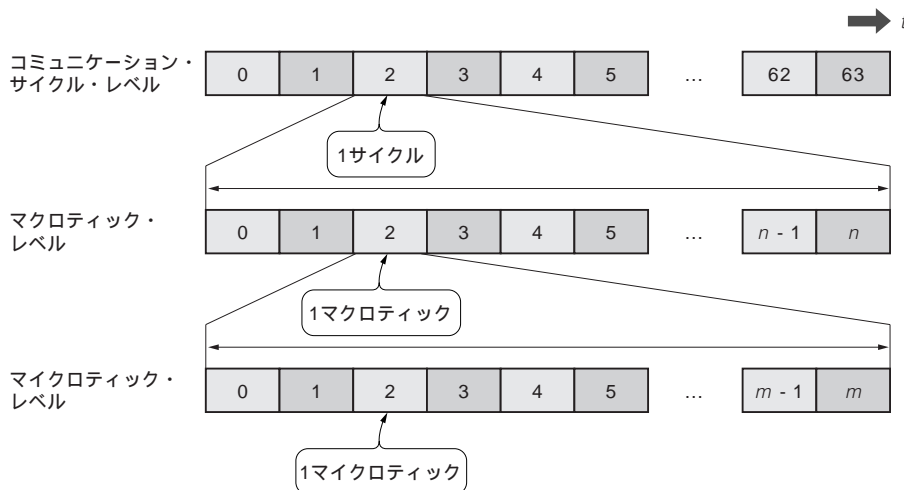


図1
タイミング階層

FlexRayの各ノード内部の時間的な構成要素は、サイクル、マクロティック、マイクロティックである。

● サイクル、マクロティック、マイクロティックの関係

マイクロティックは、FlexRayの通信コントローラのクロックによって決まる単位です。つまり、マイクロティックの長さは、ノードごとに異なるケースもあります。マイクロティックはノード内部の(ローカルな)時間の最小単位です。

マクロティックによって、各ノードの同期をとります。マクロティックの長さ(時間)は、クラスタ全体にわたって同じでなければなりません。マクロティックは整数個のマイクロティックで構成されていますが、1マクロティック当たりのマイクロティックの数は同じノード内でも異なる場合があります。

サイクルは、整数個のマクロティックから構成されます。1サイクル当たりのマクロティックの数は、一つのクラスタ内のすべてのノードで同一です。また、ある時点におけるサイクル番号は、すべてのノードで同一となります。

● 各ノードは時間についての共通な認識を持つ

先に、FlexRayプロトコルでは、クラスタとしての絶対

的な(あるいは基準となる)時間は定義されないと述べました。FlexRayでは、クラスタとしての時間を「グローバル時間」、各ノードにおける時間を「ローカル時間」と呼びます。ここで、グローバル時間とは「クラスタ内の時間に関する共通認識」であり、コールドスタート・ノード(スタートアップを開始できるノード)によって与えられます。

各ノードは独立したローカル時間、すなわちノードごとのクロックに依存する時間を持っています。前回(本誌2005年11月号、pp.87-91)でもお話ししたように、スタートアップの際に各ノードがコールドスタート・ノードと統合することで、必要な情報を受信してクロック同期(ローカル時間とグローバル時間の差を認識し、補正すること)を行います。

2 補正方法と測定方法

クロック同期は、おもに二つのプロセスからなっています。これらのプロセスは並行動作します。

1) マクロティック生成プロセス(MTG: MacroTICK Gene

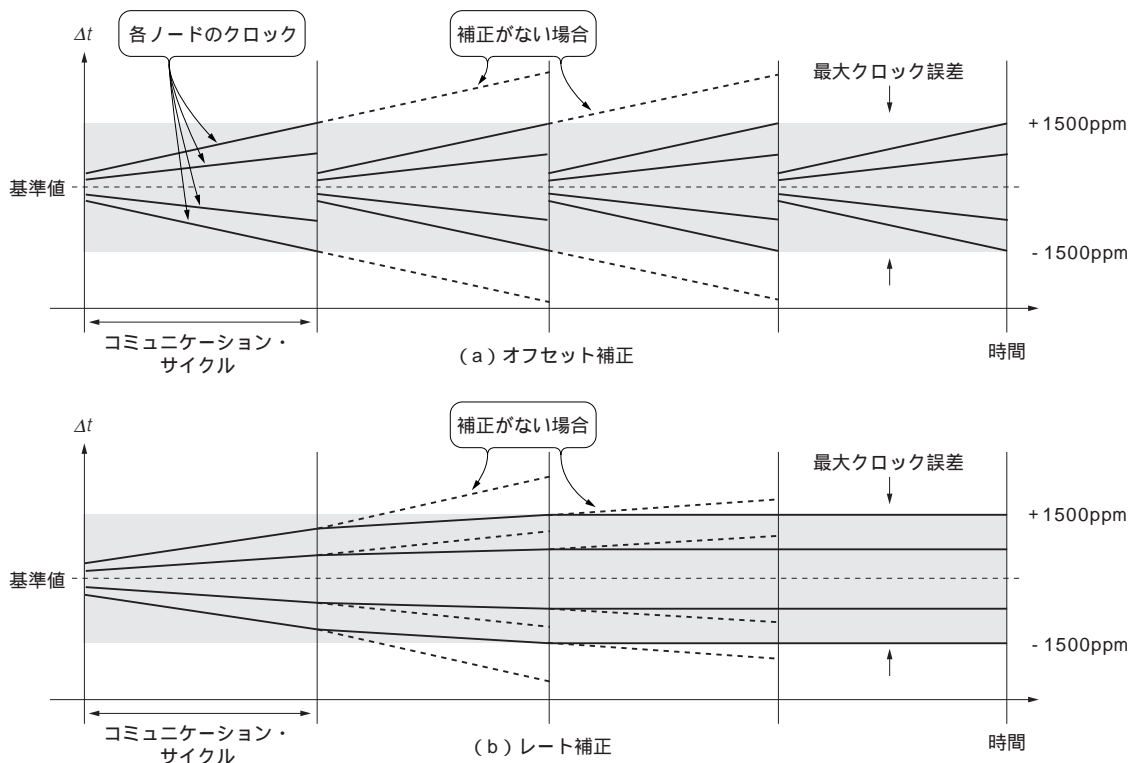


図2 クロック同期の方法

各ノードのローカル時間の同期には、オフセット(位相)補正やレート(周波数)補正を用いる方法がある。サイクルを重ねるごとに基準値(グローバル時間)からのずれは大きくなるが、補正をかけることでずれを最大誤差内に収めることができる。