組み込みソフトへの数理的アプローチ

第9回 オーバラップ制御用ステート・マシンの設計と検証
――処理の多重化を検証する

藤倉 俊幸

いくつかの作業を連続して処理しなければならないときに，各作業にかかる時間がバラバラだと，最も時間のかかる作業が全体の処理速度を決めてしまう．つまりボトルネックになってしまう．このボトルネックを回避して全体の処理速度を上げる方法として，多重化によるオーバラップ制御がある．

ボトルネック部分を多重化して並列に動作させれば，全体の処理速度を上げることができる反面，並列動作を導入するために制御が複雑になる．並列性が絡んで複雑になった処理は設計やデバッグがめんどうになり，熟練が必要である．そのような領域に形式的な手法を導入できれば効果は大きい．実は数理的アプローチというのは熟練者のノウハウを解明して定式化する方法でもあるのだ．

1 オーバラップ制御とモデル化

部分的に処理を多重化して複数のトランザクションに時間差をつけて並列化して動かすことを，ここではオーバラップ制御と呼ぶことにする．同種の処理に時間差をつけて同時に流すのでオーバラップと呼んでみたが，パイプライン制御のほうがよいかもしれない．けれども，このアプリケーションの出所が工場の生産ラインなので，何とかラインという名前は紛らわしいのである．

ここでは例として，おせんべいを計量して袋詰めする工程を考える（**図1**）．この工程は，計量作業Aと袋詰め作業Bから構成されるとする．

生産ラインとしては，一定時間に何袋製造できるかが重要である．仮に，Aに20秒かかり，Bに5秒かかるとすれば，多重化しな

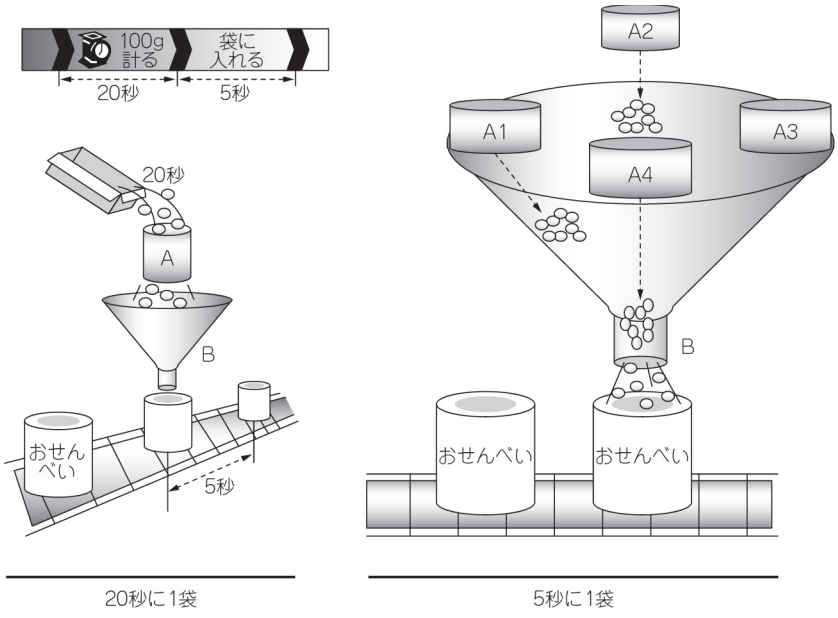

図1 オーバラップ制御の例（おせんべいの袋詰め）

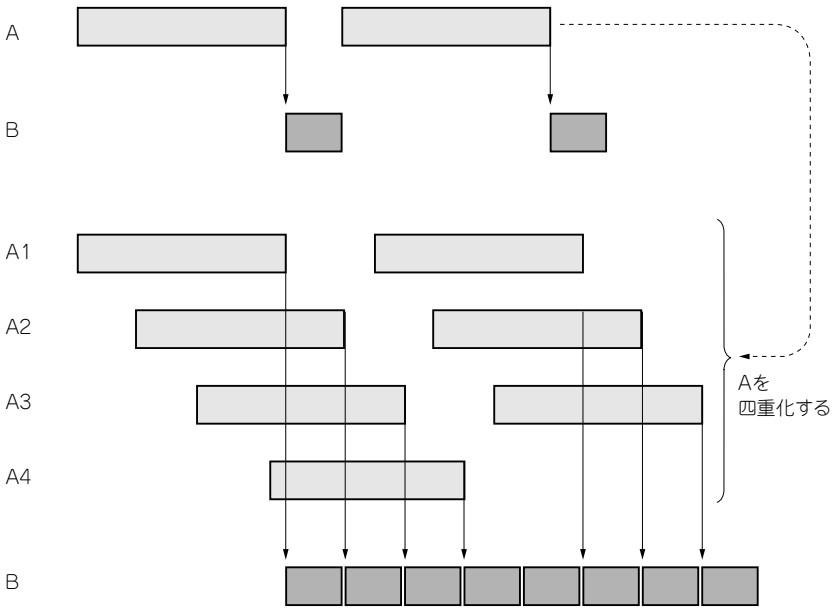

図2 動作タイミング

KEYWORD ――オーバラップ制御，並列化，タスク分割

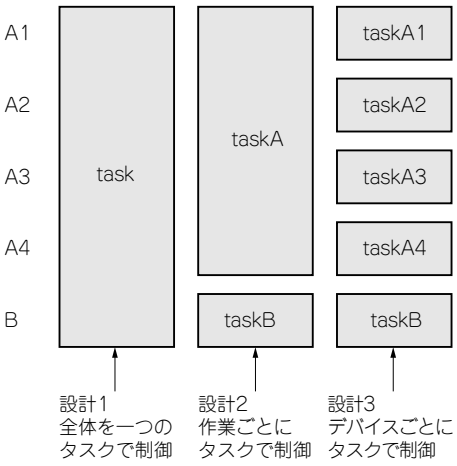

図3　タスク分割例

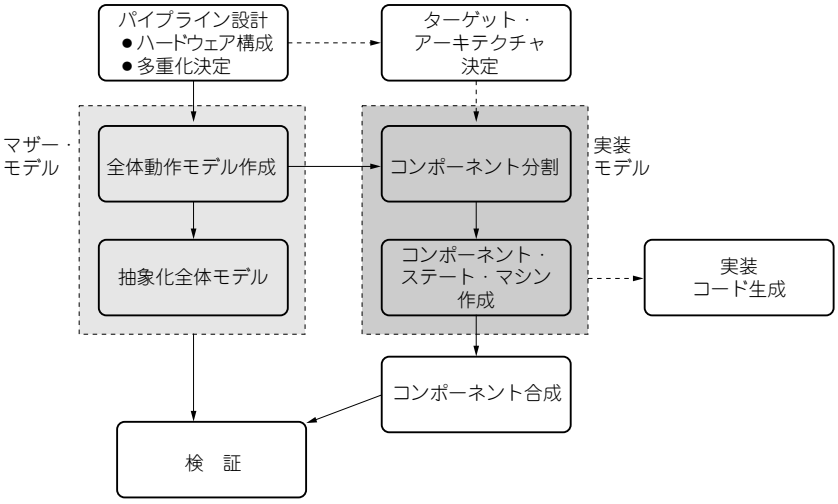

図4　設計全体の流れ

い場合，作業Aの時間が全体の速度を支配するので20秒に1袋しか作れないことになる．ところが，作業Aの作業を四重化して5秒ずつの時間差を持たせて並列に動作させると，作業Bの遊び時間はなくなって，5秒に1袋作れるようになる（**図2**）．

この制御を実現する場合，リアルタイムOSの使用を仮定すると，

- 作業Aと作業Bの全体を一つのスレッドで実装する
- 作業Aをまとめて一つのスレッドに割り当てる
- A1～A4のそれぞれに一つずつスレッドを割り当てる

などが考えられる．システム資源に余裕があれば，スレッド数を増やすことで設計を一見単純化することができるが，全体の状態が見えにくくなる．結果として，緊急停止などの対応が複雑になる．また，デバッグにも時間がかかるようになる．

静的な構造を作る場合は，クラス・レベル，言い換えればモジュール構造は**図3**の設計2で行って，実行時にはそれぞれインスタンス化した設計3で動くようにするのが一般的だろう．すると動的構造は必然的に設計3に従うことになるが，1チップ・マイコンなどの環境で動作させたい場合には，資源が少ないなどの問題が出てくるのと，緊急停止や再スタートなどの例外処理への対応がめんどうになってくる．工場などではスループットも重要だが，トラブル発生時の復旧時間も重要である．そうすると，全体を管理するようなタスクが別途必要になるなどして，結局，設計1のような構造も必要になってますます複雑化する．

そこで，必要に応じて設計1でも設計3でも生成できるようなおおもととなるマザー・モデルを作っておくとつごうが良い．ここで言いたいのは「どの設計が良いか悪いか」ではなく，「どのような設計にでも千変万化する必要がある」ということである．

最近はプロダクト・ラインに注目する人が増えてきたが，動的な構造をどうするのかに対して明確な解答を持っていない場合が多い．動的な構造について考えないままでは，プラットホームを越えた再利用は難しい．リアルタイムOSなしの8ビットCPUから，複数のCPUを分散させて使うシステムまで安全に再利用するためには，動作環境に応じて制御構造を生成できるマザー・モデルが必要である．いろいろなユーザ要求や設計制約に従って変幻自在に制御構造を生成できるようにしておく必要がある．

一般に，製造工程で使用される工業用機器では，顧客の要望によりラインの多重化はオプションとして扱われることが多く，制御プログラムは多くの多重化に対応できる必要がある．これは，いわゆるプロダクト・ラインとは異なるが，製造ラインの組み方に応じてソフトウェアのコンフィグレーションを柔軟に対応させなければならないという点はよく似ている．

設計全体の流れを**図4**に示す．**図4**の左側の全体モデルがマザー・モデルに対応する．マザー・モデルを，選定したアーキテクチャに投影して，個別モデルを生成する．

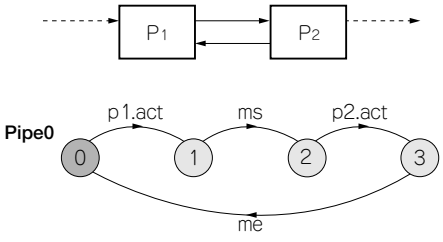

図5　パイプライン構成

2 設計手順

図1に示した例を**図5**のように抽象化して，設計・検証手順を説明する．**図1**では作業Aと作業Bの間におせんべいの自然落下プロセスが入るが，これらは現場のノウハウになるので話を簡単にするために省略する．