

Appendix

# シミュレータの作成

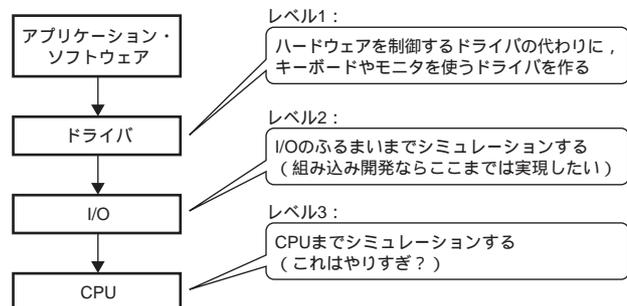
山崎長雄

ここでは、第1章で出てきたアカウント・バンクの状態遷移モデルをもとに作成したシミュレータ(アカウント・バンクの動作を模擬するソフトウェア)について説明する。シミュレータのソース・コードは、本誌のWebサイトからダウンロードできる。(編集部)

紙に書かれたモデルだけでは、十分な訓練を受けた人しか理解できません。シミュレータの優れている点は、実際に手で触れて操作できることです。紙に書いた絵や、何百枚ものプレゼンテーション資料よりもわかりやすいのは「百聞は一見にしかず」と言われるとおりです。

シミュレータを作る理由としては、実際に物を作ってから「こういうものを頼んではいけない」といった事態を避けられることが挙げられます。また、「こんな物が作れますから、やってみませんか」と企画を通したいときにも利用できます。

ここで説明するシミュレータは、ハードウェアの動作をソフトウェアで模擬(シミュレーション)するものです。シ



【図1】どこまでシミュレーションするの?

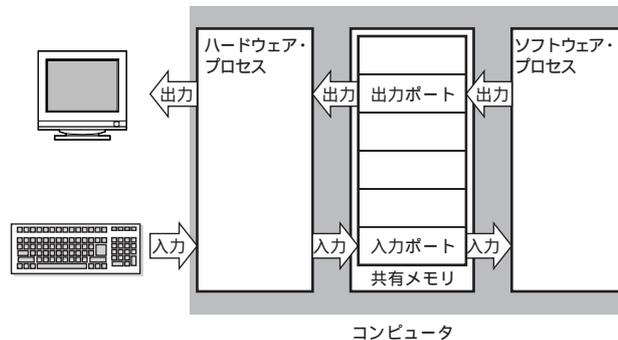
シミュレーションするレベルとして、ドライバ・レベル、I/Oレベル、CPUレベルなどがある。組み込みソフトウェアの開発なら、I/Oレベルまではシミュレーションするべきだろう。

ミュレーションするレベルは、大きく分けて三つあります(図1)。ここでは、I/Oレベルのふるまいまでシミュレーションします。

●入出力を共有メモリに仲介させる

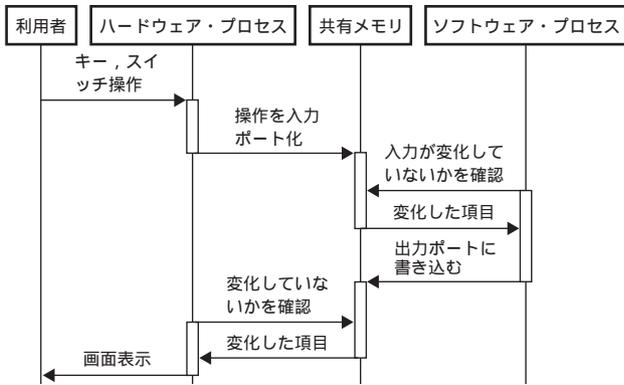
I/Oレベルをシミュレーションするためには、直接I/Oポートをビット・レベルで操作できなければなりません。できればソフトウェアが実際のハードウェアを操作しているようにプログラミングできることが理想です。ここで紹介する手法ではUNIXの共有メモリをI/Oポートに見立てて、実際のハードウェアと組み込みソフトウェアの間に近い構造をとっています。

ハードウェアをシミュレーションするプロセス(ハードウェア・プロセス)は、入力装置の役割をキーボードに割



【図2】シミュレータの構成

共有メモリ(I/O空間のように用いる)を用意して、ハードウェア・プロセスとソフトウェア・プロセスの双方から共有メモリに読み書きする。ここでハードウェア・プロセスは機器(アカウント・バンク)をシミュレーションするプログラムであり、ソフトウェア・プロセスは機器に組み込まれるソフトウェアそのものである。共有メモリに割り当てられた特定の番地を、入力ポート/出力ポートとする。ハードウェア・プロセスはキーボード入力を入力ポートにつなぎ、出力ポートを画面出力につなぐ。ソフトウェア・プロセスは入力ポートからの入力を受け付け、処理した後、出力ポートを書き換えて機器を制御する。



【図3】シーケンス図

シミュレータの利用者，ハードウェア・シミュレータ(ハードウェア・プロセス)，共有メモリ(I/Oポートの代替)，組み込みソフトウェア(ソフトウェア・プロセス)の間の入出力の取り扱いを示す。

り当て，キー入力があるとそのキーに対応した共有メモリを書き換えます(図2)。組み込みソフトウェア側のプロセス(ソフトウェア・プロセス)では共有メモリの値が変わるのを参照して，実際に入力装置から入力があったときと同じようにふるまいます。

一方，出力装置については，組み込みソフトウェア側のプロセスが出力装置を操作するつもりで共有メモリを書き換えます。ハードウェアをシミュレーションするプロセスでは，共有メモリの内容が変化するのを監視していて，実際の出力装置の役割を果たす画面の表示を更新します。

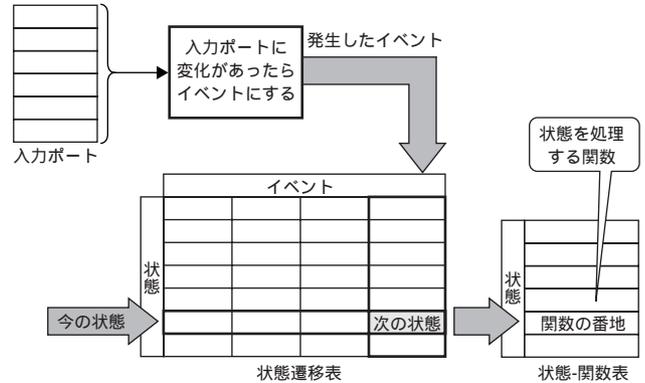
シミュレータの「利用者」，「ハードウェア・プロセス」，「共有メモリ」，「ソフトウェア・プロセス」の間の入出力がどのように処理されていくかを図3に示します。

シミュレータには以下のメリットがあります。

- ハードウェアがなくても開発を開始できる
- ハードウェア開発者に頼まなくても自分で拡張できる
- 財布に優しい
- 実際のハードウェアと同じようにI/Oにアクセスできる
- ハードウェアが手に入ったなら，ソフトウェアはポインタを変更するだけで対応できる
- ハードウェア側と組み込みソフトウェア側は別プロセスなのでリンク不要
- 組み込みソフトウェアの変数も共有メモリに置けば，ほかのプロセスを使ってモニタできる

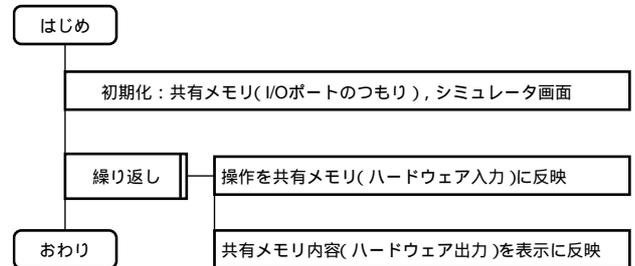
ただし，次の事項を考慮しておく必要があります。

- シミュレーションしたところまでしか確認できない
- CPUまでシミュレーションしていない。つまり，シミュ



【図4】ソフトウェア・プロセスの実装

共有メモリ(入力ポート)を確認し，変化があったらイベントを生成する。また，発生したイベントに応じて状態を遷移させる。なお，この図に示した処理の流れは，基本的に第2章の図14に示したものと同一である。



【図5】ハードウェア・プロセスの実装

シミュレータの操作を共有メモリ(入力ポート)に反映してソフトウェア・プロセスに引き渡す。また，ソフトウェア・プロセスが変更した共有メモリ(出力ポート)の内容を表示に反映する。

レータのC言語の仕様は使用するCPUのC言語の仕様と異なるかもしれない

- リアルタイム処理まではシミュレーションできない
- 外部環境も何らかの形でシミュレーションしなければならない
- ソフトウェアはもちろん，ハードウェアも設計できる人でないと，良いモデルが作れない
- 物理現象の知識も必要

●シミュレータの構造と実行環境

アカウント・バンクをシミュレーションするにあたり，組み込みソフトウェア側は第1章で作成した状態遷移表を利用して，テーブル駆動アルゴリズムで実装しています(図4)。ハードウェア側はシリアルEEPROMを単なる配列として実装し，操作を共有メモリ(入力ポート)に反映したり，共有メモリ(出力ポート)の内容を表示に反映したりしています(図5)。