

# ラップトップPC上の 機能シミュレータ : PC-FAL

## PC-FAL: Functional Simulator on Laptop Computer

市村 徹\*      木暮 誠\*\*      矢野 栄一\*      山岸 邦彦\*      関根 優年\*  
Tohru ICHIMURA      Makoto KOGURE      Eiichi YANO      Kunihiko YAMAGISHI      Masatoshi SEKINE

\* (株) 東芝 超LSI研究所  
\*\* (株) 東芝 青梅工場

\* VLSI Research Center, TOSHIBA Corporation  
\*\* OHME Works, TOSHIBA Corporation

あらまし 大型計算機上の機能論理シミュレータFALをラップトップコンピュータJ3100に移植した。本報告では、移植上の問題点および性能の評価結果について述べる。シミュレーション時間(T.A.T.: Turn Around Time)は大型計算機(15MIPS)上の機能論理シミュレータFALのそれと比較して、およそ7~8倍であった。扱える回路規模は約10Kゲートであり、小規模のシミュレーションにおいて実用レベルにある。このシミュレータを核に、機能ブロックによる論理回路の図形入力(FSET)、波形によるテストデータ入力などと共にパーソナル機能設計CADシステムPCVIEWを構成している。

Abstract This paper describes a functional simulator PC-FAL on a laptop computer J3100(180286/8MHz, 2MB memory, 20MB HDD). The turn around time for simulation is seven times longer than that of the same mixed level simulator FAL on a host machine(15MIPS). The maximum circuit size for the PC-FAL is about 10K gates. These results show PC-FAL is applicable to practical use. PC-FAL is the kernel of a personal functional design system PCVIEW, with a functional schematic entry FSET, a wave form editor and other design tools.

### 1 はじめに

1980年以降、論理LSIは加速度的に大規模化、複雑化してきている。たとえばカスタムICおよびASICでは、CPUとその周辺回路を含んだ大規模なLSIが製品化されている。それゆえ最近では、このようなLSIを用いることによって、より高度な論理システムの開発が行なわれている。

これらの論理システムの開発においては機能および論理の設計検証を早期に短期間で行なうことが必要になってくる。そのため、大まかな機能のみが決まった部分と詳細な論理設計まで進んだ部分とが混在した回路の設計検証が要求されてきた。この要求に応えるため、我々は機能回路部と論理回路部が混在した回路を扱うこ

とができる機能論理シミュレータFAL  
(Functional And Logic simulator)  
[1, 2, 3, 4]を大型計算機上に開発した。

FALは、従来機能論理設計の検証に使われてきた論理シミュレータと比較して数十倍高速である。それゆえ、実際に各種CPU、グラフィックLSI、通信制御LSI、各種信号処理LSIなどの機能設計検証に使用されてきた。これらのLSIの規模は50K~200Kゲートであり、従来使用されてきた論理シミュレータでは機能設計検証が不可能であった。

しかし大型計算機に対する負荷は増大する傾向にあり問題になってきている。この問題を解消するために実用的なパーソナルCADシステムが要求されるようになってきた。そこで我々はこうした要求に応えるために、定評のある当社のラップトップコンピュータ(J3100)

上で稼動するパーソナル機能設計CADシステム(PCVIEW)の開発を行ってきた。

最近ではEWSやパーソナルコンピュータの価格の低下と性能の向上によって、論理回路の設計検証用パーソナルCADシステム[5, 6]の実用性が次第に高くなってきている。また、ラップトップコンピュータを用いることにより複雑な機能を効率良く実現するためのアーキテクチャおよびハードウェアアルゴリズムの検討を、設計者が自分の机の上で手軽に行なうことができるようになる。以上の理由により、こうしたパーソナル機能設計CADシステムの将来性は高いと考えている。

本論文では、このPCVIEWの概要、システム構成および特徴を簡単に述べ、今回評価を行なった、PCVIEWの1サブシステムであるパーソナル機能シミュレータ(PC-FAL)について、その規模、内部構成、移植上の考慮点、評価結果について述べ、さらに現状の問題点を解析し、その解決策について論ずる。最後にPCVIEW全般にわたる長所と改良すべき内容をまとめておく。

## 2 PCVIEWの概要

PCVIEWはラップトップコンピュータJ3100(CPU:intel 80286;8MHz,主記憶:640KB+2MB, OS:MS-DOS)上でLSIの機能設計を支援するシステムである。PCVIEWは機能設計検証を効率良く行なうために必要な5つのサブシステムで構成されている(図1)。それは、(1)機能シミュレーションサブシステムPC-FAL、(2)機能図入力サブシステムFSET(Functional Schematic Entry and Translator)、(3)波形入力サブシステムTDE(Timing Diagram Editor)、(4)テストデータ作成サブシステムTDA(Test Data Assembler)、(5)マイクロプログラム作成サブシステムMICASS(Micro program Assembler)の5つのサブシステムである。以下に各サブシステムの機能と特徴について述べる。

### (1) 機能シミュレータ(PC-FAL):

PC-FALは大型計算機上で実績のある機能論理シミュレータFALの機能シミュレータ部分をJ3100に移植したもので早期検証に便利な中断点処理コマンドを備えた対話

型機能シミュレータである。またシミュレーション結果を当社標準のテストインターフェイス言語TSTL2(Toshiba Standard Testdata-interface Language 2)[7]と同じ形式で出力することができる。

### (2) 機能図入力システム(FSET):

FSETは、ハードウェアの構造・機能を表す図(機能図;図2)を画面上に直接入力し、これを機能記述言語に自動変換するシステムである。登録機能のガイドに従って、入出力信号数などをパラメータ化して機能ブロックをライブラリに登録することができる。また階層設計入力、論理式の入力等が行なえる他、機能図の階層構造の表示、ドキュメントとして機能図のプロット出力が行なえる。

### (3) 波形入力システム(TDE):

TDEは、タイミングチャートの入出力を行なうシステムである。入力操作は、マウスを使って対話的に行なうことで、波形データ(図3)を簡単に描く事ができ、入力された波形データをシミュレーション用のテストデータに変換する事によりテストデータの作成時間を大巾に短縮することができる。このシステムには標準的な波形データを登録しておいて可変部分をパラメータで置き換えて希望する波形の入力を行なえるマクロ機能がある。またシミュレーション結果の波形出力にも使用される。

### (4) テストデータアセンブラ(TDA):

TDAは、多量のテストデータの作成を容易にするためのシステムである。マクロとニーマニックを使うことにより、少量の入力データから多量のテストデータを対話型操作により生成することができる。また出力結果をTDEで読み込むことにより、波形での確認、編集やテストデータへの変換等ができる。

### (5) マイクロプログラムアセンブラ

(MICASS):MICASSはマイクロプログラムの開発を行なうためのシステムである。対話型操作、マクロ機能による入力、アセンブルの分割実行が行なえる。

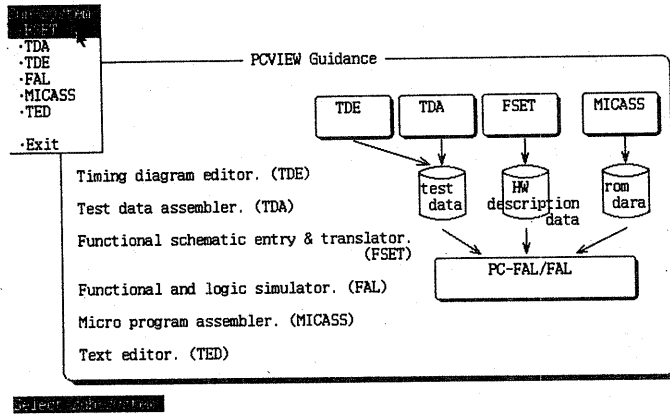


図1 PC-VIEWの構成図

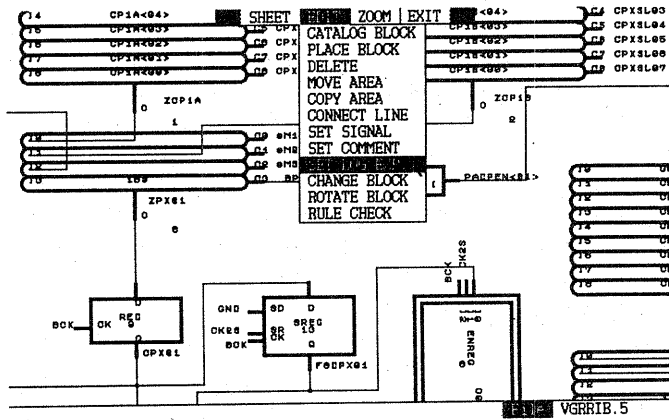


図2 機能図の入力画面

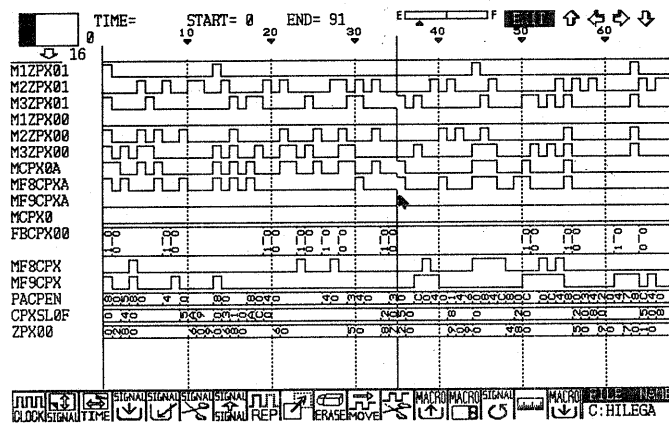


図3 タイミングチャートの編集画面

### 3. 機能シミュレータPC-FAL

#### 3.1 PC-FALの構成と規模

PC-FALでは、回路記述用の言語(HDL\*)としては状態遷移記述を含むRTL(Register Transfer Level)の記述形式と動作レベルの関数的な記述形式を使用する。PC-FALは通常のプログラム言語の処理系と同様に、記述形式に対応する2つのコンパイラ、リンカ、シミュレータおよびユーザーインターフェイスで構成されている。ユーザーインターフェイスはメニュー駆動方式のウィンドウ型でシミュレーション制御コマンドの簡易入力機能を含んでいる。

PC-FALの規模はコンパイラがC言語で8千行、リンカ、シミュレータはFortran 77で、実行文の数がそれぞれ7千、3万3千(ルーチン数は250,800)である。この中で最も大きいシミュレータは対話性重視の設計がなされているため、19コのシミュレーション制御コマンドを持ち、これらのコマンド処理部分はシミュレータ全体の約2/3を占めている。移植に際してはMS-DOSのメモリ制限(640KB)に対して、このシミュレータの大きさが問題になった。次にこのシミュレータ部分の移植に際して大ききの面で考慮した点について述べる。

#### 3.2 移植上の縮小化に関する考慮点

PC-FALシミュレータのメモリマップを図4に示す。これはPROFORTRAN-77のジャンボモデル[8]でコンパイルした結果である。シミュレータの移植に際しては、コード領域、変数領域、データ領域およびエクストラ領域の縮小化のための方策を考え、全体の必要メモリ量を見積もった。

(1) コード領域の縮小化: シミュレータのコード量は1MBであり、オーバーレイ化の必要があった。シミュレータはシミュレーション制御コマンド処理部(コマンドモニタ)とシミュレータ本体部の2つに大きく分かれる。

コマンドモニタ部分はオーバーレイのスワップによるオーバーヘッドをコマンド単位ではなくなるようにして、ルートコードが37KB、オーバーレイコードが150KBの領

\*.....H<sup>2</sup>DL(Hierarchical Hardware Design Language)

域に納まっている。

シミュレータ本体部は45コの組み込み関数命令、110コに及ぶハードウェアを模倣する命令の処理を含んでおり、これらの命令処理部分は150KBものコード量である。そのため、これらを2~3の排他的なオーバーレイに分割せざるをえなかった。このことからオーバーレイのスワップによるオーバーヘッドが生じ、速度性能上のネックになることは、ある程度予想された。そこで、これらの排他的なオーバーレイ間を渡るようなケースを少なくするために、命令の使用頻度、実行順序をもとに割り振った。

(2) 変数領域の縮小化: まず扱える回路を小さくするなど、配列のサイズを縮小して、この領域は400KBあった。PC-FALはワード単位でポインタおよび数値データを処理しているので、1ワードを4バイトから2バイトにすることにより、190KB縮小された。また、ローカル変数領域は静的に確保されるので、ルーチン間で排他的なものは共有化することで縮小化を計ることができる。実際に明らかに排他的で大きなもののみ共有化したところ18KB縮小化されて、変数領域は192KBとなった。

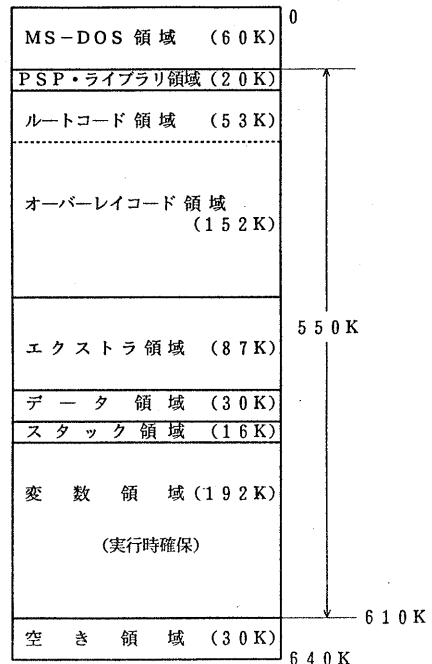


図4 PC-FALシミュレータのメモリマップ

(3) エクストラ領域の縮小化：この領域はFORMAT文や定数が格納されるが、この他にコンブロックのアドレスデータを管理する補助データがある。このためコンブロックをまとめて30ぐらい減らした結果、エクストラ領域が13KB縮小して、87KBになった。

(4) データ領域の縮小化：この領域はスタック領域と合わせて64KB以内でなければならないという制約がある。当初はこの制約に違反して64KBを越えていた。そこで調査したところオブジェクトモジュールが多くなると、この領域が大きくなることがわかった。当初800あったオブジェクトモジュールを、それらの呼び出し関係によって325までまとめた。このことによりデータ領域は30KBに縮小された。

### 3. 3 性能評価

PC-FALを適用した結果を表1に示す。この表からの大型計算機(15MIPS)上のFALシミュレータのバッチジョブ(ジョブ待ちやCPU待ちのないケース)のT. A. T. (Turn Around Time)に対してPC-FALの実行時間は、コンパイルはほぼ同程度、リンクは約3倍、シミュレーションは7~8倍といった値になっていることがわかる。CPU時間でシミュレーション時間を比較すると130~150倍となっている。

また、PC-FALが扱うことができる最大の回路規模は、回路記述の仕方によって幅があるが、回路記述の実行文の数で800~1800(ゲート数にして5K~10Kゲート)程度になる。

### 3. 4 現状の問題点と解決策

このように、現在PC-FALシミュレータは回路データのデータ領域(オブジェクトサイズ)が60KBと小さいため、シミュレ

ーション可能な回路規模が10Kゲートと小さい。これはMS-DOSのアドレス空間が1MBと小さいということが原因で、この壁を破るにはIntel 80286の保護モード[9]に移行して16MBのアドレス空間でシミュレーションする必要がある。

保護モードで動作するOSは各種存在し、それぞれについて評価を行なった。しかし、保護モード用のコードを生成するFortranの処理系は不十分なものであった。また、主流のMS-DOSをOSとして使うことで、各種ユーティリティプログラムを利用できるという利点を残すことが必要であった。これらの理由から、MS-DOSをOSとし、保護モードに移りPC-FALシミュレータを動作させた後、またMS-DOSに戻る方法を採用した。

PC-FALシミュレータを保護モードで動作させるためのプログラムを現在開発中である[10]。このプログラムは実アドレスモードで動作するオブジェクトを保護モード

表1 PC-FALの適用例

| 回 路 名                          |              | C 1                      | C 2                       | C 3          |            |
|--------------------------------|--------------|--------------------------|---------------------------|--------------|------------|
| ノ ー ド 数                        |              | 128                      | 230                       | 284          |            |
| メモリ (RAM, ROM)<br>(bit×word×個) |              | 8×8×4<br>1×8×5<br>17×8×1 | 8×8×8<br>1×8×10<br>17×8×2 | 0            |            |
| 実 行 文 数                        |              | 147                      | 297                       | 861          |            |
| オブジェクト<br>サイズ (PC)             | コード          | 13KB                     | 23KB                      | 40KB         |            |
|                                | データ          | 1KB                      | 2KB                       | 2KB          |            |
| 実<br>行<br>時<br>間<br>(秒)        | コンパイル        | NDS <sup>1</sup>         | 25 (0.69)                 | 27 (1.17)    | 59 (2.12)  |
|                                |              | PC <sup>2</sup>          | 12                        | 39           | 64         |
|                                | リン ク         | NDS <sup>1</sup>         | 23 (0.41)                 | 20 (0.45)    | 21 (0.55)  |
|                                |              | PC <sup>2</sup>          | 60                        | 59           | 74         |
|                                | シミュレー<br>ション | NDS <sup>1</sup>         | 80 (4.08)                 | 75 (4.90)    | 109 (7.39) |
|                                |              | PC <sup>2</sup>          | 501                       | 671          | 820        |
| ク ロ ッ ク の 周 期 数                |              | 114                      | 124                       | 100          |            |
| 入 力 パ タ ー ン                    |              | 50 bit×90<br>34 bit×21   | 50 bit×91<br>34 bit×21    | 21bit×310    |            |
| モ ニ タ 信 号 数                    |              | 17 (全47 bit)             | 21 (全35 bit)              | 27 (全27 bit) |            |

1 …… 15MIPSの大型計算機上でのバッチジョブのT. A. T. (Turn Around Time) 但し( )内はCPU時間

2 …… J3100/GTのRAMディスク上での実行時間 (英語DOS V.3.2, 環境変数BUFFERSの値は20)

で動作するオブジェクトに変換するものである。このプログラムを使用すると、各セグメント(コードおよびデータ)アドレスの変化に対応してディスクリプタテーブルが作成される。また、保護モードに移行することでMS-DOSのシステムコールが使用不可能となるため保護モード下で動作する各種BIOSルーチンの作成とMS-DOSへの復帰手段が必要となる。

#### 4 まとめ

PCVIEWでは図面の階層性が表現できるFSETを使用して設計者の馴染みやすいハードウェアの構造・機能を表す図(機能図)を入力することによって機能設計を行なうことができる。このことにより、機能記述用の言語による従来の設計と比較して、設計が容易になる。またラップトップコンピュータ上で機能設計が可能になるということで、設計者は自分の机の上で手軽に設計検討ができる。

今後、各サブシステム間のインターフェースの強化などによって、完備していないPCVIEWの機能の追加改良を計画中である。例えば現在のFSETでは状態制御を回路またはプル式で入力することになるが、より容易に回路のモデルを記述するための、状態遷移図などの入力機能、モニタ信号の回路図上での選択機能、シミュレーション中断時における信号値の回路図上への表示機能などがあげられる。以上述べてきたようにPCVIEWシステムの開発は完了していないが、機能設計段階において多大な労力の軽減と時間の節約が計られるものと期待され、現在試用中である。

#### 参考文献

- [ 1 ] 渡辺, 他 “機能シミュレータFALにおける中断点処理” 第29回情報処理学会全国大会 pp.1721-1722
- [ 2 ] 市村, 他 “ミックスシミュレータFALシステム” 第30回情報処理学会全国大会 pp.1945-1946
- [ 3 ] 武井, 他 “機能論理シミュレータFALを核とした設計検証システム” 電子通信学会回路とシステム研究会 CAS86-184 pp.79-86
- [ 4 ] 山岸, 他 “ミックスシミュレータFALのASIC設計への適用” 情報処理学会設計自動化研究会 38-3 (1987.7.23)
- [ 5 ] VIEWLOGIC SYSTEMS INC. “WORKVIEW トータルデザインシステムユーザリファレンスマニュアル” 1985
- [ 6 ] Daisy Systems Co. “Personal LOGICIAN System Overview” 1986
- [ 7 ] 立石, 新田 “VLSIテストデータ自動処理システムの開発” 情報処理学会設計自動化研究会 39-4 (1987.10.15)
- [ 8 ] Prospero Software “Pro FORTRAN-77ユーザリファレンスマニュアル” 1986
- [ 9 ] Intel “iAPX286 Programmer's Reference Manual”
- [ 10 ] 矢野, 他 “PC-FAL 機能シミュレータの移植” 第35回情報処理学会全国大会 pp.2261-2262