

## ソフトウェア CAD を指向したマイコンソフトウェア 開発支援システム

河野清尊 浅田保栄 宇野 結 清水澄代 湯川博司 丸山幸孝  
松下電器産業株式会社 中央研究所

マイクロプロセッサ（マイコン）分野専用のソフトウェア開発支援環境を専用のハードウェアおよびソフトウェアツール群により実用化した。

CPUはMC68010/68020、OSはUNIX System Vを搭載し、ソフトウェア開発専用の大型画面のグラフィック端末をRS-422インターフェースで複数台接続することにより、ソフトウェアCADツールおよびマルチウインドウ機能をマルチユーザ環境で使えるようにした。

C言語ソースプログラムをPAD図で開発できるC-PADチャートエディタ、編集中にC言語の文法チェックを行なえるC-LOED/M-CUTEエディタ、ターゲットシステムと接続したICEとの間でPAD図を用いてC言語のシンボリックデバッグが可能なC-PADデバッガ等、ソフトウェア開発の効率および信頼性を大幅に改善するソフトウェアツール群を用意することにより、ソフトウェアCAD指向の総合的な開発支援環境にアプローチした。

Software CAD oriented system dedicated for microprocessor software development ( in Japanese )

Kiyotaka Kohno Yasue Asada Musubi Uno Sumiyo Shimizu Hiroshi Yukawa Yukitaka Maruyama  
Central Research Laboratory, Matsushita Electric Industrial Co.,Ltd.  
3-15, Yagumonakamachi, Moriguchi, Osaka 570, Japan

The software CAD oriented system dedicated for microprocessor application software development has been realised. The hardware and the software was specially designed for software development.

With 68010/68020 CPU and UNIX System V, the easy to use software CAD tools and the multiwindow system can be used on each wide screen graphic display through RS-422 interfaces.

The software tools are such as, C-PAD EDITOR which is the chart editor with an easy to understand structured program logic diagram (CPAD diagram) for the C language source program, C-LOED/M-CUTE EDITOR which is the text editor with the function of the C language syntax checking, C-PAD DEBUGGER which provides an interactive debugging environment on CPAD diagram, etc. These software tools greatly improve efficiency and reliability in software development. This is the first step to approach the software CAD system for microprocessor application.

# ソフトウェアCADを指向したマイコンソフトウェア開発支援システム

河野清尊 浅田保栄 宇野 結 清水澄代 湯川博司 丸山幸孝  
松下電器産業株式会社 中央研究所

## 1. まえがき

1971年に米国インテル社が4ビットのマイクロプロセッサ（以下マイコンと呼ぶ）を発表して以来、半導体技術の進展と相まって4ビットから8ビット、16ビットさらには32ビットと、マイコンの高性能、高機能および高集積化は一段と加速して現在に至っている。

それに伴い、マイコン応用製品のソフトウェアも多様化し、複雑化しているが、ソフトウェアの開発形態は旧態依然としており、生産性、品質、再利用および保守等の面に多くの問題が生じてきている。

マイコン分野と事務処理・科学技術計算等の他分野とのソフトウェア開発における大きな違いは、マイコン分野ではデバッグにインサーキットエミュレータ（ICE）を用いることである。これはソフトウェアとハードウェアとが複雑に絡みあっていること、およびターゲットシステム上にソフトウェアのデバッグ環境が無いくることによるものである。

このようなマイコン分野のソフトウェア開発の問題として次の2点をあげることができる。

(1) ICEをソフトウェアの開発マシンとして使うには性能的に問題が多く開発効率が悪い。

(2) 開発マシン上でクロス開発した実行形式ファイルをICE上にダウンロードしてデバッグする方法では、ダウンロードに時間がかかり、かつ高級言語レベルのデバッグができないので開発効率が悪い。

（開発マシンとICEとが完全に分離している。）

このような状況において、マイコン分野のソフトウェア開発へのCAD（コンピュータによる設計支援）技術の導入、およびシステム本体とICEとの間を密結合することにより、上記問題点の解決を図るとともに、ソフトウェア開発工程における

- 生産性・品質の向上
- 重複開発の防止、共有化および再利用の促進
- ドキュメントの自動生成、保守の容易さ

等を実現するマイコンソフトウェア開発支援システムを実用化した。

このシステムの特長は次の通りである。

(1) 専用のハードウェアおよびソフトウェアCADツールの開発により、ソフトウェア開発工程における詳細設計からデバッグ・保守まで（ソフトウェア開発工程の下流部）を一貫支援した。

(2) 専用に開発した大型画面のグラフィック端末をシリアル回線（RS-422またはRS-232C）で複数台接続することにより、ソフトウェアCADツールおよびマルチウィンドウ機能をマルチユーザ環境で使えるようにした。

(3) システム本体とICEとの間をRS-422で接続することにより、データの高速転送を実現するとともに、C言語のシンボリックデバッグを実現した。（開発マシンとICEとを統合した。）

以下、第2章ではこのシステムの構成をハードウェアおよびソフトウェアの両面から述べ、第3章および第4章ではOSおよびソフトウェアCADツールを詳細に説明する。最後に第5章で今後の展開について述べる。

尚、このシステムは松下通信工業株式会社 電子計測事業部より M-Station 3500

本体 : VP-3500A  
ピットマップコントローラ : VP-3501A  
ピットマップディスプレイ : VP-3502A  
プリンタ : VP-4931A

CPUはMC68010

として商品化している。[1]

## 2. システム構成

### 2.1 ハードウェア構成

図1にこのシステムのハードウェア構成を示す。

図1に示すように、システムバスにVMEbusを採用して、標準で次の各ボードを搭載している。

（括弧はオプションである。）

- CPUボード MC68010/MC68020、MMU付き
- メモリボード 2MB (8MB)
- ディスク I/F ボード
  - HD 40MB (80MB、最大270MB)
  - FD 1MB (2MB)
- シリアル (SIO) I/F ボード
  - RS-422 2ポート (4ポート)
  - RS-232C 2ポート (4ポート)
  - セントロニクス 1ポート (2ポート)

さらに、オプションとして次のボードを用意している。

- MT I/F ボード 9トラック
- Ethernet ボード

このシステムのハードウェア面での特長は次の通りである。

- (1) 専用のグラフィック端末（ビットマップディスプレイおよびコントローラ）を開発した。
- (2) RS-422インターフェースを設けた。
- (3) シリアル回線（RS-422およびRS-232C）にグラフィック端末を接続することにより、最大8台まで接続可能にした。
- (4) ICEをRS-422に接続することにより高速のデータ転送を実現した。

また、グラフィック端末の機能および性能は次の通りである。

#### ビットマップディスプレイ

- 画面サイズ 19インチ正方形
- 解像度 1024×1024ドット
- モノクロ 4階調 無反射加工

#### ビットマップコントローラ

##### ハードウェア

- CPU MC68000 (10MHz)
- インターフェース制御

(RS-422、RS-232C、キーボード、マウス、セントロニクス)

- ウィンドウメモリ 512KB

##### ソフトウェア

- ウィンドウ制御
- プリンタ制御
- エミュレーション制御 (VT-100モード、PLOT-10モード)
- システム制御等

このグラフィック端末はRS-232Cで接続すれば汎用マシンのソフトウェアCAD端末としても使用できる。（汎用マシン上のツールも開発している。）

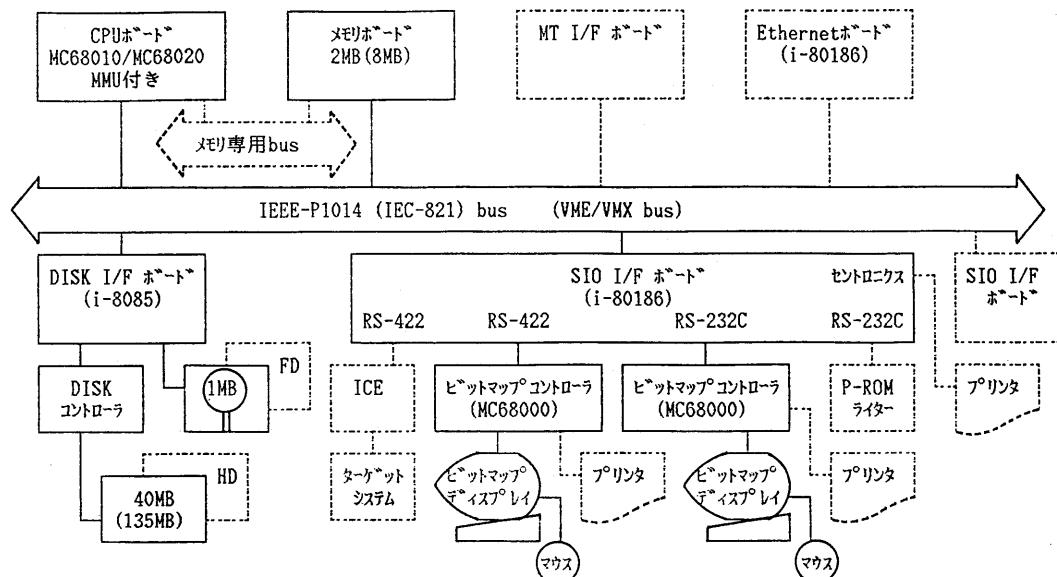


図1 ハードウェア構成

## 2.2 ソフトウェア構成

このシステムのソフトウェア構成を図2に示す。また、図3にソフトウェア開発工程および各工程に対応するソフトウェアCADツール群を示す。

OSはUNIX System Vを移植開発し、このOS上にマルチウィンドウ機能を独自開発した。

ソフトウェアCADツールは図3に示すように詳細設計から保守までを一貫支援しており、専用のグラフィック端末を用いることによりマルチユーザ環境でソフトウェアの開発が行なえる。

また、4ビットから16ビットまでの各種マイコン用のクロス言語プロセッサを開発しており、さらにその他ユーティリティも用意している。

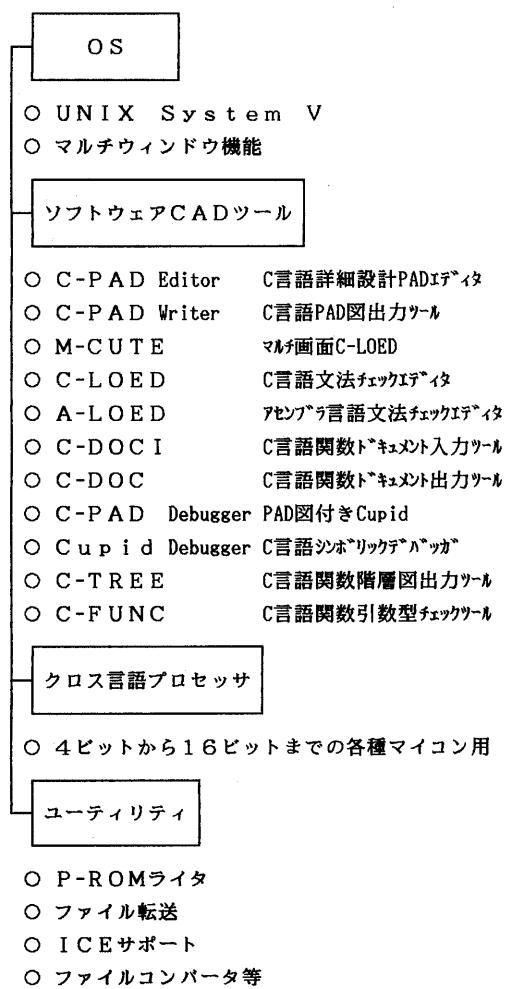


図2 ソフトウェア構成

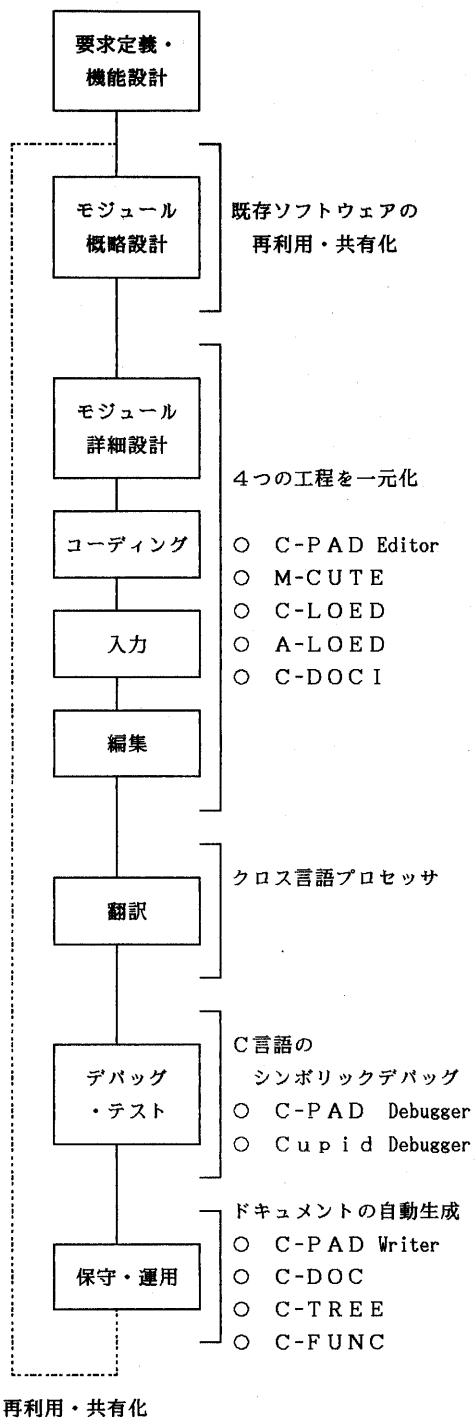


図3 ソフトウェア開発工程  
および対応するソフトウェアCADツール群

### 3. OS

#### 3.1 UNIX System V の移植開発

UNIX System V のオリジナル版は AT&T の Rel1.0 MC68000 用であり、これを MC68010 システムに移植開発した。

移植開発は、まずカーネルソースコードの解析およびオリジナルシステムの調査を行ない、その後カーネルのハードウェア依存部を抽出して、修正および新規開発箇所の仕様を検討した。

次に、移植開発環境の整備と並行してデバイスドライバの開発および単体テストを行なった。

デバイスドライバはディスク、シリアル I/F、タイマー、プリンタの他に、新規にオリジナル版には無かった MT および次の第 3.2 項で述べるマルチウィンドウ機能を実現するためのビットマップドライバを開発した。

単体テスト終了後デバイスドライバをカーネルに組込んでカーネルおよびコマンドのデバッグを行なった。デバッグに当たっては、カーネルおよび複数ユーザー プロセスを同時にデバッグできるデバッガを開発してカーネルに組むことにより効率化を図った。

また、デバッグ段階でディスク内に必要なイニシャルファイルシステムは、開発マシンの OS が System V でなかったこと、および開発マシンとターゲットシステムとのディスクが共有できなかったことから、開発マシン上に 1 つのファイルとして仮想ファイルシステムを作成してこれをターゲットシステムにダウンロードする方法を探った。

当初は MC68010 および System V Rel1.0 をベースにしていたが、既に MC68020 (MC68881 付き)、System V Rel2.0 および日本語化の基本開発を終え、現在では通信機能および日本語機能の強化等に取組んでいる。

#### 3.2 マルチウィンドウ機能

ソフトウェア CAD システム用に開発したグラフィック端末は基本的なウィンドウ制御機能を備えている。この機能を用いてシリアル回線に接続したグラフィック端末でマルチウィンドウ機能を実現するビットマップドライバおよびウィンドウマネージャを開発した。

図 4 にマルチウィンドウ環境の実現方法を示す。図 4 に示すように、グラフィック端末上に開設されたウィンドウ（現在は最大 7）を仮想端末・キーボードとみなして、これと 1 対 1 に対応する仮想端末ドライバ（ビットマップドライバ）を UNIX の回線（tty）ドライバの上位レベルに設けることにより、ユー

ザープロセスとウィンドウとを対応づけた。

各ユーザープロセスと仮想端末間のデータはシリアル回線および回線ドライバ上を混在して流れれるが、そのデータは仮想端末ドライバおよびビットマップコントローラにより、対応するユーザープロセスおよびウィンドウに振り分けられる。

このことにより、複数のグラフィック端末で同時にマルチウィンドウ機能を使える環境を UNIX に大幅な修正を加えることなく実現できた。

ウィンドウマネージャの機能は次の通りである。

- (1) ウィンドウの開設および消去
- (2) ユーザープロセスの実行および終了
- (3) ウィンドウの表示位置の移動
- (4) ウィンドウの表示位置および大きさの変更
- (5) ウィンドウへの出力の一時中断
- (6) ウィンドウの優先度の変更

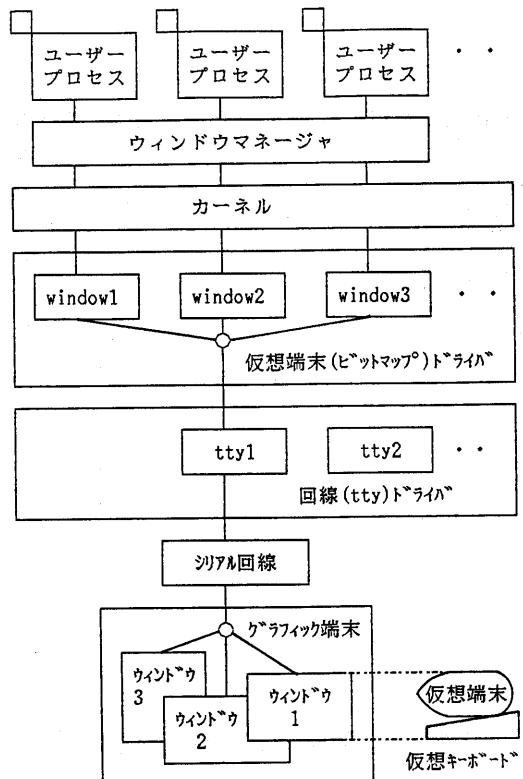


図 4 マルチウィンドウ環境の実現方法

## 4. ソフトウェアCADツール

### 4. 1 C-PAD Editor/Writer

C-PAD図は(株)日立製作所が提唱したPAD(Problem Analysis Diagram)を独自にC言語用に拡張したものである。

C-PAD Editorは図記号(以下箱と呼ぶ)として

- 関数定義の箱
- C言語キーワードに対する箱  
if-else、while、do-while、for、switch-case、break、continue、return、goto、label
- コメントの箱
- プロセスの箱

を定義して、この箱単位で編集を行なうことによりC言語ソースプログラムを効率良く開発できるチャートエディタである。また、C-PAD WriterはC言語ソースプログラムからC-PAD図を自動的にプリンタに出力するツールである。図5にC-PAD Editorの表示例を示す。

C-PAD Editorの主な機能および特長は次の通りである。

- 箱およびその内容を挿入することによりC言語ソースプログラムを自動生成する。
- C言語ソースプログラムを自動的にC-PAD図に展開して表示する。
- C-PAD図とソースプログラムとは完全に一致する。
- 強力な編集コマンドを用意している。編集コマンドを表1に示す。
- 2つのファイルの同時編集が可能である。
- 箱のみの表示を行なうことにより広範囲のプログラム構造を見ることができる。
- 図的に表示するのでプログラムの構造を容易に把握できる。
- 箱単位で編集するので構文的な誤りが生じにくい。
- ソースプログラムの大きさおよび構造的階層の深さに制限がない。

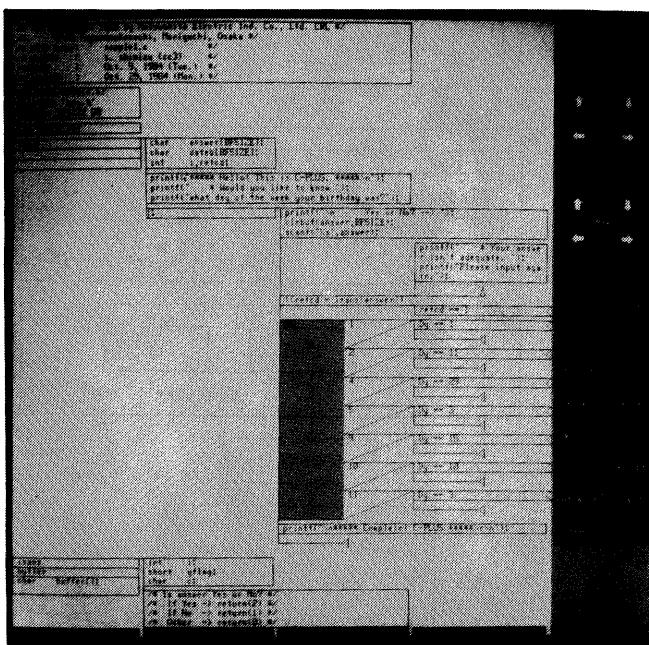


図5 C-PAD Editorの表示例

コマンド名	処理内容
help	ヘルプ
page矢印	ページ移動
箱矢印	現在の箱の移動
insert (on)	箱の挿入
delete	箱を削除
yank	箱を記憶
put (on)	記憶(削除)した箱の挿入
search	文字列を検索
kick	文字列の変換
multi	複数ファイルの編集
cute	箱の内容の修正
replace	箱の置換(削除と挿入)
change bt	箱の型変換
end	終了(書き込み)
contents	箱と内容の表示
box	箱のみの表示

表1 コマンド一覧表

C-PAD Editorの内部構造を図6に示す。

図6に示すように、C-PAD Editorは、まずC言語ソースファイルを読み込んで解析を行い、箱レベルに展開した後、抽出した情報をソース内容情報としてセットする。次に、ソース内容情報からデータ表示情報を生成するとともに、このデータ表示情報をもとにしてグラフィック端末のディスプレイ上にC PAD図を表示する。

編集時には、マウスからのコマンドの入力およびキーボードからのデータの入力に基づいて、ソース内容情報を変更するとともに、データ表示情報をもとにして表示画面を書き直す。

マウスから終了コマンドが入力されたならば、ソース内容情報からソースプログラムを生成してファイルに書き込み処理を終了する。

#### 今後の課題として

- 機能設計工程への拡張
- データ表現方法の追加
- 複数のコマンド入力の提供

等に対応していく予定である。

#### 4. 2 C-PAD Debugger

C-PAD DebuggerはRS-422（またはRS-232C）インターフェースに接続したICEを制御することにより、ターゲットシステムのプログラムをC PAD図上でデバッグできるC言語シンボリックデバッガである。これは独自開発したクロス環境のC言語シンボリックデバッガCupid[2]をこのマイコン

ソフトウェア開発支援システムに対応させ、かつユーザーインターフェース部をC PAD図を用いて使い易く改良したものである。図7にC-PAD Debuggerの表示例を示す。

C-PAD Debuggerの機能および特長は次の通りである。

- C言語専用のデバッガであるので、C言語を使うプログラマにとって使い易い。（例えば、変数名の指定にC言語の記述がそのまま使えることや、変数のビットパターン表示ができるなど。）
- 各コマンドは画面上のコマンドメニューを用いてマウスによって指定できる。
- デバッグ中には、C言語で記述されたソースプログラムのみならず、それをC PAD図に展開したものも参照できる。
- ブレークポイントの指定や参照したいソースプログラムの位置などは、表示画面上のC PAD図およびソースプログラムをマウスでポイントすることにより指定することもできる。
- 接続したICEの機能は透過的に使用できる。従って、C言語とアセンブリ言語の混在したシステムをデバッグできる。

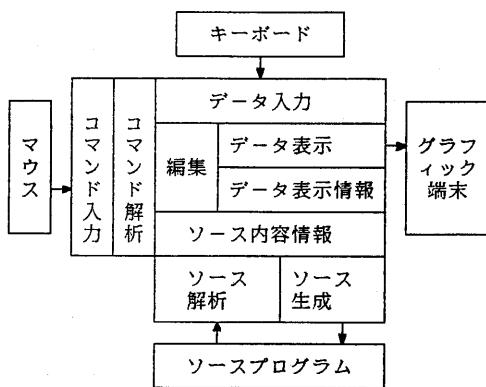


図6 C-PAD Editorの内部構造

The screenshot shows the C-PAD Debugger interface. On the left, there is a window displaying assembly code:

```
1: main()
2: {
3:     int a = 1;
4:     int b = 2;
5:     int c = 3;
6:     int d = 4;
7:     int e = 5;
8:     int f = 6;
9:     int g = 7;
10:    int h = 8;
11:    int i = 9;
12:    int j = 10;
13:    int k = 11;
14:    int l = 12;
15:    int m = 13;
16:    int n = 14;
17:    int o = 15;
18:    int p = 16;
19:    int q = 17;
20:    int r = 18;
21:    int s = 19;
22:    int t = 20;
23:    int u = 21;
24:    int v = 22;
25:    int w = 23;
26:    int x = 24;
27:    int y = 25;
28:    int z = 26;
29: }
```

On the right, there is a window showing a memory dump:

```
1: main()
2: {
3:     int a = 1;
4:     int b = 2;
5:     int c = 3;
6:     int d = 4;
7:     int e = 5;
8:     int f = 6;
9:     int g = 7;
10:    int h = 8;
11:    int i = 9;
12:    int j = 10;
13:    int k = 11;
14:    int l = 12;
15:    int m = 13;
16:    int n = 14;
17:    int o = 15;
18:    int p = 16;
19:    int q = 17;
20:    int r = 18;
21:    int s = 19;
22:    int t = 20;
23:    int u = 21;
24:    int v = 22;
25:    int w = 23;
26:    int x = 24;
27:    int y = 25;
28:    int z = 26;
29: }
```

図7 C-PAD Debuggerの表示例

## C-PAD Debuggerの内部構造を図8に示す。

図8に示すように、C-PAD Debuggerは、C言語ソースファイルと実行形式ファイルに対応するシンボル情報を参照することによりデバッグを行なう。このシンボル情報は被デバッグプログラムのクロスコンパイル／アセンブル／リンク時にオプションを指定することにより生成される。

C-PAD Debuggerのコマンドは、実行制御コマンド、変数参照コマンドおよびソースプログラム参照コマンドの3つに分類できる。

実行制御コマンドにおいては、ソースファイルの行番号および関数名を用いて実行制御アドレスを指定する。また変数参照コマンドにおいては、変数名を用いて読み書きしたい変数を指定する。この2つのコマンドを実行する実行制御部および変数参照部は、シンボル情報インターフェース部を通してシンボル情報を参照することにより、実行制御アドレス、参照すべきアドレスまたはレジスタを計算する。その後、ターゲットインターフェース部を通してICEと交信することにより、被デバッグプログラムの実行制御およびメモリ・レジスタの読み書きを行なう。

ソースプログラム参照コマンドの実行はソースファイル管理部が行なう。参照ソースファイルは実行停止時に自動的に設定されるほか、コマンドによって指定することもできる。

ターゲットシステムを新しく導入する時には、ターゲットインターフェース部を新しいシステムに対応させることによってC-PAD Debuggerを導入することができる。（クロスコンパイラ／アセンブラー／リンクカは準備しなければならない。）

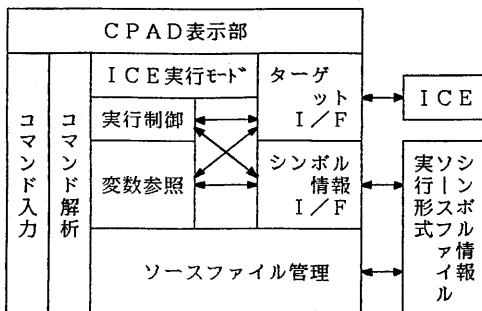


図8. C-PAD Debuggerの内部構造

## 4.3 その他のツール

### 4.3.1 M-CUTE

M-CUTE [3] は編集中にC言語の文法チェックができるマルチ画面フルスクリーンテキストエディタである。これは、独自開発したフルスクリーンテキストエディタCUTE [3] にC言語の文法チェック機能を取り入れたC-LOED [4] を、さらにグラフィック端末を用いて最大4ファイルの同時編集ができるよう機能拡張したエディタである。図9にM-CUTEの表示例を示す。

M-CUTEの特長は次の通りである。

- 複数ファイルの同時編集が可能
- 表示形式を数種類準備
- タイリング画面の採用
- ポップアップメニューによるコマンドの入力
- ファイル間での内容のコピー

### 4.3.2 C-DOC/C-DOCI

C-DOCはC言語ソースプログラムから仕様書として必要な情報を抽出して関数説明書を作成し、ディスプレイまたはプリンタに出力するツールである。

図10にC-DOCの出力例を示す。

The screenshot shows the M-CUTE editor interface with a C code file open. The code includes comments like /\* Copyright 1982 by Hitachi Electric Inc., Ltd. JP 3-6 Yagisawamachi, Horikiri, Ohta-ku, Tokyo 171. File name : adobe.c \*/. The editor features multiple windows for different parts of the code, with syntax highlighting for keywords like if, else, for, while, etc. and identifiers.

図9. M-CUTEの表示例

C言語ソースプログラムを文法的に解析することにより関数名、引数名、引数型名、ファイル名、インクルードファイル名、使用グローバル変数名および使用関数名を抽出し、また、コメント部のキーワードで始まる文字列を検索することにより作成者、作成日、戻り値および内容説明を抽出して関数説明書を作成する。

C-DOC Iは、C-DOCでコメント部より抽出する情報が入力されていないことを知らせるとともに、その情報を入力するツールである。

C-DOC/C-DOC Iは現在ではまだ静的解析ツールとしての位置付けであるが、今後は必要な情報を挿入することによりプログラムを自動的に作成できる仕様書エディタとして展開していく予定である。

## 5. 今後の展開

今回実用化したマイコンソフトウェア開発支援システムは、ソフトウェア開発工程の詳細設計から保守までのいわゆる下流部を一貫支援するものであった。今後は上流部へのアプローチとともに開発工程全体を一貫支援するシステムを目指す予定である。

このシステムの今後の展開としては、次のような項目を検討中あるいは開発中である。

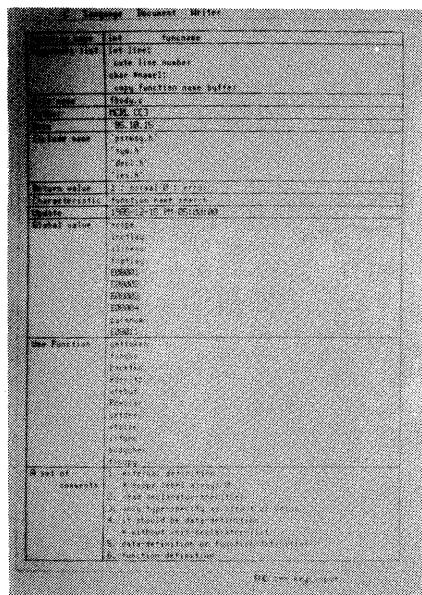


図10 C-DOCの出力例

## ソフトウェアツールの充実

### ○ PDB (Program Data Base)

詳細設計工程のソフトウェアの再利用・共有化。

### ○ ソフトウェア開発工程の上流部（要求分析から概略設計）へのアプローチ

C-PAD EditorおよびC-DOCの機能拡張。

### ○ テスト工程を支援するツール等。

## 開発工程の統合化

### ○ ツール間における共通データベースの確立。

### ○ ツールの統合。

### ○ ソフトウェア開発マネジメントシステム。

### ○ OSレベルまで含めた開発環境の構築等。

## 6. あとがき

以上、専用のソフトウェアツールおよびハードウェアを開発して、ソフトウェア開発工程における詳細設計からデバッグ・保守までを一貫支援することにより、ソフトウェアの生産性・品質を向上させるソフトウェアCADを指向したマイコンソフトウェア開発支援システムについて報告した。

尚、このシステムのプロトタイプシステムを松下通信工業株式会社で実際の商品開発に使用した結果、C-PAD Editorを用いたソースプログラムの開発および詳細設計書の作成、C-PAD Writerの出力図を用いたプログラムのレビュー等に威力を発揮して、品質の良いソフトウェアを効率よく開発できたという評価を得た。

終わりに、このシステムの研究開発にあたり協力頂いた、松下通信工業株式会社 電子計測事業部 技術部、当社 映像機器事業部 および 松下システムエンジニアリング株式会社の方々に深く感謝致します。

## <参考文献>

- [1] 「ソフトウェアCADシステム M-Station 3500」、計測と制御、Vol.25、No.9、昭和61年9月
- [2] 丸山、宇野、森原：「C言語シンボリックデバッガ CUPID」、情報処理学会第31回全国大会（昭和60年後期）予稿集 6F-8 pp.537-538
- [3] 浅田、清水、森原：「マルチ画面フルスクリーンエディタ CUTE」、情報処理学会第31回全国大会（昭和60年後期）予稿集 1F-4 pp.441-442
- [4] 宇野、森原：「CUTEテキストエディタとC言語指向エディタ CLOED」、情報処理学会第31回全国大会（昭和60年後期）予稿集 1F-5 pp.443-444