

プロジェクト 406*

中川 圭介†

概要 1991 から9年間に研究室の学生達が行った計算機システム開発のプロジェクトについて報告する。プロジェクトの名前は、それが行われた電気通信大学情報工学科の西1-406室に因んで、開発したチップ、計算機、システムソフトウェアの名前に406を付加したことによる。

1980年頃米国の大学で集積回路の製作のコースがあることを知り[1]、実際に集積回路を作ってみたいと考えていた。それは、担当していた論理設計の講義を変える必要があるかどうかを実際の経験をもとに決めたかったためである。しかし、そのころ、日本の環境は集積回路を大学で試作するような状況でなかった。

1990年に、少量の特注LSIを低価格で製作するヨーロッパの企業ES2(European Silicon System)の仲介をする日本の商社を紹介され、1991年に最初のLSIの試作を学生の卒業研究として行った。その後、学部学生と大学院生がRISCマイクロプロセッサの製作を計画したことから、計算機システムをリレー式に開発していくことになった。

開発は、マイクロプロセッサの製作、パーソナルコンピュータの製作、システムソフトウェアの製作に分かれているが、プロジェクト全体が最初にこのような形で計画されたわけではなく、1つの区切りがくると研究室にきた学生との議論によって次の計画が決まるというぐあいに仕事は進んだ。

このプロジェクトの基本の考えは、実用になるシステムを作るということである。実際、開発したシステムを構成する3つの部分は、備えている機能は最小限におさえてあるが、それぞれ、独立したシステムとして利用可能なものになっている。これは、計算機システムは、たとえ開発教育の一部として計画されたものであっても、細部まで本物になってはじめて意味を持つという考えによる。

1 マイクロプロセッサrj406

1.1 LSIの試作

最初の年は4ビットの演算装置を製作した。CADシステムになれることと、LSIについての感覚を養うことが目的であったが、1.4ミクロンルールで約500ゲートの論理回路が $9mm^2$ のシリコン上に収まった。3人の学部4年生が独立に設計し、それぞれ10個づつ、合計30個の製作費が120万円であった。また、製造過程を理解するために、集積回路の教科書[2]の輪講をした。

2年目は前年に作ったLSIのテストと32ビットRISCマイクロプロセッサの設計を行った。これには3人の院生(M1)と5人の学部学生が参加した。

LSIは動作のテストをしないで返される契約であったので、PC98にインターフェース回路を介してLSIをつなぎすべての動作を検査した。その結果、動作の上限が10MHzのものと20MHzのものがあり、また、2個の1部に誤動作があることが確認できた。

マイクロプロセッサの最初の設計は学部学生が行った。前期はパイプライン制御の輪講とCADを使った32ビットの演算回路の設計を、後期にVLSIの設計を実行した。このとき、 $50mm^2$ のチップを250万円

*Project 406

†Keisuke Nakagawa, kei-naka@gw2.gateway.ne.jp, 東京都品川区平塚1-15-8

で依頼できるようになった。これにはチップのテストも含まれ、1.2ミクロンルールになった。

VLSI設計の最初の仕事は、32ビットのパイプライン化されたプロセッサが 50mm^2 におさまるかどうかの概算であったが、命令セットの第一版を暫定的に決め、院生が粗い設計をもとにゲート数を概算して、単純な比例計算を行い、収まると判断して設計を開始した。論理設計、ハードウェア記述言語による入力、レイアウトを約2ヶ月で終わり 54mm^2 の面積のチップが得られた。それから配置の最適化をCADのツールを使って行い、面積を約 48mm^2 に縮小することができた。

このとき暫定的に決めたアーキテクチャと内部の制御方式は以下のようなものである：

- (1) データの型は8ビットと32ビットの整数。
- (2) ロードストアアーキテクチャ。
- (3) 分岐はレジスタの内容を直接調べて行う。
- (4) レジスタの数は32個。
- (5) 乗除算はない。
- (6) 内部の制御回路は5段のパイプラインを採用する。

1.2 アーキテクチャ

製作するマイクロコンピュータはrj406と呼ぶことにした。アーキテクチャは前節の暫定版に割り込み機能、初期設定機能と、いくつかの機能の追加を行ったものであるが、追加命令の有効性あるいは十分であることの確認をシミュレーションによって行った。

1.2.1 内部状態と実行制御

内部状態 rj406は、モニタモードとユーザモードの2つの動作状態がある。モニタモードではすべての命令を実行できるが、ユーザモードでは特権命令の実行が許されておらず、実行すると割り込みが起きる。

CPUの初期リセットと停止制御 rj406は外部からの信号`reset`と`halt`によって初期リセットと一時停止を行う。

`reset = 1`とすると内部状態がモニタモードにリセットされ`reset = 0`に戻すと8番地の命令から実行を開始する。

`halt = 1`であると命令実行が中断され、`halt = 0`にすると実行が再開される。これは記憶装置からの読み出しが遅れたときの待ち合わせなどに使う。

状態を制御する命令 内部状態の変化に関連する命令は以下に示すものである。

<code>trap</code>		トラップ: 8番地へとび、戻り番地をr31に入れる。状態をモニタモードにする。
<code>di</code>	特権	割り込み禁止
<code>ei</code>	特権	割り込み許可
<code>reti</code>	特権	モニタモードからの復帰。状態をユーザモードにする。
<code>mtsr</code>	特権	状態レジスタの内容を変える。
<code>mfsr</code>	特権	状態レジスタの内容を読む

1.2.2 割り込み

外部からの割り込みは機能を小さくし、チップの利用者が欲しい機能は外部の回路で補うことにした。これはまた、割り込みの機能をチップの利用者が選ぶ範囲を広くすることにもなっている。

割り込み動作 外部から割り込みが要求されると命令実行を中断し0番地へとぶ。同時に状態をモニターモードにし、帰り番地をレジスタ r31 にいれる。

割り込み信号 割り込み信号は禁止できるものと出来ないものを用意し、割り込んだ信号の種類とトラップに関する情報を記憶するレジスタ `ssr` がある。

1.2.3 命令アーキテクチャ

命令アーキテクチャは、SPARK, MIPS, DLX[3] を参考にして、シミュレーションの結果をもとに決定した [4]。最終的に選んだ命令は以下の表に示すものである。

分類	形式	命令
演算(レジスタ)	000000 dr[5], s1[5], s2[5] sr[6]	add, sub, or, xor, and, slt, sltu, sll, slli, srl, srli, sra, srai
演算(即値)	op[6] dr[5], s1[5], imm16[16]	addi, ori, eori, andi, slti, sltui, addhi, orhi, xorhi, andhi
ロード/ストア	op[6] dr[5], s1[5], imm16[16]	ldb, ldw, stb, stw
分岐(PC 相対)	op[6] dr[5], s1[5], imm16[16]	bpl, bmi, bez, bnz
ジャンプ(レジスタ)	000000 dr[5], s1[5], s2[5] sr[6]	jr, jalr
ジャンプ(PC 相対)	op[6] imm[26]	j, jal
トラップ	000000 dr[5], s1[5], s2[5] sr[6]	trap
特権	000000 dr[5], s1[5], s2[5] sr[6]	di(割込禁止), ei(割込許可), reti(割込から復帰), msfr, mtfir(状態レジスタの移動)

ここで、`op` は命令コード、`dr`, `s1`, `s2` は行先と2つのオペランドのレジスタ、`sr` はレジスタ命令の命令コードである。

rj406 の命令はほとんどが基本的なものであり、検討の対象となった命令は多くない。まず、ロード命令に遅延スロットをつけるか、必要なときに1クロック待つかは、遅延スロットをつけることにした。しかし、これは逆の判断のほうがよかったと考えている。

半語(16ビット)のロードとストアは実装しなかった。多重精度の演算を半語の演算で行うことにしていたが、1語のロードとシフトで十分と考えた。

上位16ビットへの即値ロード命令は実装することにしたが、さらに演算の命令を実装した。これらは便利であるが、rj406 の中ではバランスがよくないように見える。

分岐命令は、(1)レジスタの比較による分岐、(2)セット命令と1つのレジスタの内容による分岐の組み合わせ、(3)条件コードによる分岐、のうち(2)を選んだ。遅延スロットは1語にした [5]。

割り込みから帰る方法は、帰りジャンプを選び、レジスタジャンプと状態変更命令の組み合わせにはしなかった。

また、製作後に知ったことでは、条件付移動命令があればよかったと考えている。

1.2.4 入出力制御

入出力命令は作らず、アドレスの一部を入出力のための情報の記憶または制御情報の転送場所として使用し、データの交換はロード命令とストア命令によって実行する。

1.2.5 シミュレーション

シミュレーションはキャッシュのついたCPUのシミュレータ上でテスト用のプログラムを実行した[6]. シミュレータは、浮動小数点命令を含むものとrj406の命令にたいするものが用意された. テストプログラムはNEWSのCコンパイラにかけ、その出力をアセンブラで機械語にして実行した. アセンブラは遅延スロットを埋める処理の実行を選ぶことができる. シミュレータの速さはSPARK ELCの上で毎秒10000命令であった.

rj406のためのテストのプログラムは以下のものである.

qsort	クイックソート. データは乱数の表.
tree	2分木による探索. データはアセンブラのラベルの表.
delay	遅延スロット処理. アセンブラの一部.
sieve	ビット表現を使ったエラトステネスのふるい.
8queen	8クイーン.
minix	MINIXのモニタとプロセス制御[7][8].

1.3 VLSIの製作

rj406のLSIは1人の院生が約3ヶ月で設計し、ES2社に発注した. 加減算回路とシフト回路をNAND回路を基本に設計しなおしたが、記述はすべて低レベル、すなわち、ゲートとフリップフロップによって行い、32個レジスタはスタティックRAMを使っている[9]. また、割り込みが起きたときはつねに正確にその位置が分かるように工夫してある.

回路の複雑さ、チップの面積、遅れ時間は以下に示すようになった.

論理回路

制御回路	4085 ステージ
演算回路	15257 ステージ
合計	19342 ステージ

チップ面積	44.81mm ²
-------	----------------------

予測遅れ時間

平均	45ns
最大	86ns

このシステムでは、1ステージは1個または2個のトランジスタで構成され、 n 入力nandゲートは $n+1$ ステージ、フリップフロップは13ステージで作られている[10]. ステージの総数を2入力nandゲートのステージ数で割った(ゲート)数で複雑さをあらわすと、rj406では6448ゲートになる.

テスト テストには、設計の段階で行うテストと製作されたチップのテストがある. 設計の段階では、論理シミュレーションと遅れ時間などを調べる物理シミュレーションを行った. 論理シミュレーションは論理誤りとその場所を探すもので、階層的にテストを繰り返す. また、物理シミュレーションを行いながら、遅れ時間を長くする原因を調べて改善をしていった.

チップのテストのためにES2に依頼するテストデータは、各クロック毎に入力ピンに与える信号と正しく動作したときに出力ピンから得られる信号の列である. これはテストベクトルの列とも呼ばれるが、依頼できる列の長さには制限があるので、VLSIチップの正常な動作を保証するに十分な列を制限の範囲内で実現しなければならない. rj406の列の大きさは1列が41ビットで長さは1752であったが、論理テストと同じ考えで作ったより長い列にするべきであった.

設計の評価 実装についての反省として、バイトのロード命令の選択回路を内部に置くべきであること、時計を内蔵すべきであることなどがある。これらは、いずれも、予備製作のときに回路を小さくすることにこだわって実装しなかったもので、設計の最終段階で見直すべきであった。

ソフトウェアの開発の段階で、算術右シフトの桁数指定がレジスタで与えられるときシフトの桁数が正しく指定されない誤りが発見された。これは論理シミュレーションの不備による。

2 2CPU 計算機 MP406

MP406 は共通の主記憶に結合された2台のrj406 と rj406 の初期設定などの制御と入出力を行うためのパーソナルコンピュータ (HPC) で構成されている。HPC の OS は UNIX を使った。

2.1 経過

2台のCPUを持つシステムは1994年から1997年にかけて開発された。1年目はMotorola社のMC68008を使い分散記憶のテストシステムを作り、アセンブラを作っていくつかのテストを行った。プログラムは生成-消費モデルで書かれたもの考えた。ハードウェアの製作は、3年次の実験でMC68008の初期設定とインタフェースを経験していたので、順調であった。このときホスト計算機としてパーソナルコンピュータを始めて利用して、以後の使用法を確立した。

2年目はテストシステムを使って、クイックソート、三角ゲームの高速化を試みた。同時に、MIPSを複数持つ市販のシステム上でC言語により分散記憶システムのプログラムの経験を積んだ。

3年目はrj406のテストをかねて、1台のCPUのシステムを試作して使用した。速度は6MHzであることがわかった。この年のソフトウェアの開発は1CPUの試作システム上で行ったもので、rj406チップの実用テストを兼ねていた。実装方法と同時に割り込みを含めて機能の検査をいろいろ行った。とくに、割り込みを使った入出力プログラムには時間をかけてrj406の機能を確認した。

4年目に2CPUのMP406が完成した。それまでの経験から、3台の計算機がデータの交換を主記憶のみを介しておこない、その他にはたがいに割り込むための信号を実装するのがよいと決めていた。1台のCPUの試作機はCPUの接続部分などに不具合があって数回のダウンを経験したが、MP406は安定に稼動した。

2.2 アーキテクチャ

MP406では、512キロバイトの主記憶に2台のrj406と1台のパーソナルコンピュータHPCが結合され、これら3台の間は割り込み信号のみが伝えられる。そして、2台のrj406は主記憶に直接アクセスできるが、HPCは入出力機能によって主記憶にアクセスする[11]。

rj406への付加機能 2台のrj406はrj0、rj1と呼ばれ、それぞれ、自分が誰かを聞く命令と、他のrj406に割り込む命令を実行できる。また、主記憶は2つの256キロバイトの領域に分割され、前半はモニタの領域となっていて、後半のユーザ領域の命令からのアクセスは制限される。

HPCへの付加機能 HPCには、主記憶にアクセスする命令とrj406を制御するため以下の命令が付加されている。これらはすべて入出力コマンドで指令する。

(a) rj0とrj1のそれぞれにたいして、リセット(resetを1にする)命令、停止(haltを1にする)命令、割り込みを起すコマンドが用意されている。

(b) 記憶装置へのアクセスは、アドレスの指定、指定されたアドレスへの書き込み、読み出しコマンドによって行う。

入出力 入出力はすべてHPCのプログラムが代行する。データとコマンドの転送は通常の入出力装置と同様に行えばよいが、データやコマンドはとくにハードウェアを追加することなく実装された記憶領域を使う。

2.3 基本操作

MP406に特徴的な操作は、初期起動とrj406間の情報交換である。

初期起動 普通の方法は、rj406をリセットの状態にしておき、実行するプログラムを主記憶に、プログラムへのジャンプ命令を8番地に入れ、リセットを解除して(8番地から)命令の実行を始める。

データの交換 2台のrj406のデータの交換は、1方向の転送とデータの共用の2つの場合がある。前者は入出力と同じ手法で行うが、後者については、割り込みなどによる中断がなければ以下のような方法によっている。たとえば、rj0とrj1が共通の変数dに書き込みと読み出しを行うとき、rj0とrj1ののそれぞれに、アクセス権を要求しているとき1となる変数p0とp1を用意し、rj0とrj1に以下のようなアクセス権を獲得するプログラムを使う。

rj0のプログラム:

- H1: p1 = 1の間H1を繰り返しp1 = 0のときp0 = 1としてH2へ進む。
- H2: dにアクセスして仕事をする。
- H3: p0 = 0とする。

rj1のプログラム:

- L1: p0 = 1の間L1を繰り返しp0 = 0のときp1 = 1としてL2へ進む。
- L2: T時間待ちL3へ進む。
- L3: p0 = 1の間L3を繰り返しp0 = 0のときL4へ進む。
- L4: dにアクセスして仕事をする。
- L5: p1 = 0とする。

MP406の目的の1つは、2台のCPUがそれぞれ1クロックで主記憶にアクセスできることを利用して1つのプログラムをどこまで速く実行できるかを調べることであるが、いまのところ生成-消費モデルで動くクイックソートしか試していない。

2.4 テスト

システムの構造が簡単で、情報交換が1つの記憶装置を介して行われるのでテストは容易であった。使用開始の後も、記憶装置のにアクセスするプログラムを実行する程度で2年間をすごした。製作者が卒業してしかも1台しかない計算機なので卒業期にははらはらしたけれども、この間のダウンは3回で、1回はボード上のスイッチの位置の間違い、2回はコネクタ部分のハンダのゴミによるものであった。

3 s406

3.1 経過

システムソフトウェアはアセンブラ、Cコンパイラ、オペレーティングシステムの開発を目的にして、1997年から1999年までの3年間に行った。1年目はHPCのUNIX上にCによって小さなシステムを作り、生成した機械語をrj406をロードして初期設定機能によって動作させた。クロスアセンブラは自作、コンパイラ

は講義(言語処理系論)で使われた tinyC[12] のコード生成部の命令定義を書き換えて使用した。OS については UNIX のファイルシステムを使い、アセンブラ、tinyC コンパイラ、ユーザプログラムの実行するコマンドを用意した。

2年目はローダ、アセンブラ、入出力制御プログラム、そして簡単なコマンドを実行する OS を rj406 上に実現した。これらはすべて rj406 上のアセンブラで開発されたが、アセンブラの製作のときに rj406 の機能の誤りが発見された。

3年目は最後の年であったが、言語処理系(zeroC コンパイラ)とオペレーティングシステム(s406)の製作に費やされた。OSは、S1, S2, S3の3システムを作った。S1は院生が1人で自作のアセンブラを使って書いた。S2は学部学生が1人で作った。開発にはzeroCを使用した。S3は学部学生が5人で製作した。機能を制限してあるが、CPUが2台の場合への拡張も行った。

3.2 実装

3.2.1 記憶装置の割り当て

記憶領域は512MBであり、それぞれ256MBの特権領域とユーザ領域に分かれている。特権領域には、モニタ、シェル、ローダ、入出力プログラムと、プロセステーブルなどのユーザプログラムが直接アクセスできないデータが入る。また、入出力装置に直結するデータ交換のための情報もこの部分に用意される。

ユーザ領域は4分割して、それぞれに独立なプロセスのテキストとスタックを含むデータが入る。

3.2.2 S1

S1は修士2年の学生が学部4年のときに作ったアセンブラを使って1年をかけて書いた。ただし、ファイルシステムはHPCのUNIXに頼っている。1人用ではあるが、並列のマルチプログラミングが可能で、スレッドの実行も可能である。このシステムの上にzeroCを実装して使用した[13]。

システムはMINIXを参考にしていて、プロセスの生成、終了、一時停止、再開、スレッドの生成、プロセス間通信、入出力関連の機能を備えている。

3.2.3 S2

4年生が1人で作った1人用のシステム。最初はS3のメンバーであったが、割り込みを使うシステムにこだわり、アセンブラとzeroCコンパイラを使って実装したが、zeroCの機能の追加とデバッグに貢献した。

3.2.4 S3

S3は複数利用者に対応するOSである。開発のメンバーが全員経験の浅い学部生であるので、基本の機能の決定から、機能を実現する動作の決定などを段階的に行い、教科書にある基本構造が自然に現れることを見た。実装は段階的に行ったが、割り込み無しのシステムの実装までで終わった。

基本の機能と動作 ユーザーはHPCにログインしてS406コマンドを実行するとプロンプトが現れる。もしS406が起動されていないならば新しく起動してプロンプトを出す。同時にこのユーザーにたいしてシェルが起動され、ユーザーのコマンドの解釈と処理が可能になる。シェルは再入可能なコードで書かれていて、主記憶に常駐している。

また、HPCにネットワークによって結合されている計算機からはリモートログインによってHPCに入って使用するが、1つのPCが1つの端末として取り扱われる。端末には空いているスロットが割り当てられ、以後、このスロットを経由してMP406と情報を交換する。

シェルは起動されると、コマンドを要求するプロンプトを表示して、休止する。

1度に1つまたは2つのプロセスを実行する。プロセスは入力順に実行する。実行が始まったプロセスは、原則として、中断せずに最後まで実行を続ける。

ユーザーが停止コマンドを実行すると、対応するシェルは削除され、UNIXにもどりプロンプトが表示される。

実装は

- (a) トラップのみを使う、
- (b) トラップと外部割込みを使う

という2つを順に実現することを計画したが、(a)のみで終わった。しかし基本になる処理を設計する経験を積むことができたと考えている。S3の製作にたいする感想は、CPUの設計に感覚的に似ているということである。すなわち、入出力プログラム、ローダ、アセンブラなどの処理プログラムをコマンドにしたがって実行制御してゆくモニタが中心にあり、複雑さはともかく、CPUの演算実行を行う制御回路に対応しているようにみえる。

3.2.5 zeroC

zeroCはそのコンパイラを書くために必要な機能をC言語から選んだものであるが、その後OSの開発に使われたときに機能を追加した。また、並列実行のために、スレッドの定義のコマンドを持っている[14]。

基本の機能 データ型と制御文は以下のようにになっている。

(a) データの型は文字型と整数型を基本に、それらの配列、ポインタ、静的変数と外部変数の指定ができる。

(b) if文, for文, while文, do-while文, switch文は実装したが、go to文はない。

実装 コンパイラはまずHPC上でクロスコンパイラを作り、そのソースコードをクロスコンパイラにかけて生成した。全体は字句解析、構文解析、目的語生成の3つの部分に分かれている。これらを2つの部分にわけて並列実行をしたがまだ効果は不十分である。

4 プロジェクトの評価

rj406については十分に最初の目的を達成したと思っているが、事情が許せばやはり浮動小数点演算回路を持つ高速のプロセッサを作るのが理想である。1996年に東京大学大規模集積回路システム設計教育研究センター(VDEC: VLSI design and Education Center)が設置されてVLSIを作る環境は格段に改善されている。いずれかの大学で新しいプロセッサが生まれることを期待している。また、プロセッサを実際に作る時、ある制限の中で最大の速度が得られるように努力することが大切であると考えている。

MP406は最後に最も単純な構造に落ち着き、その特徴はその後のソフトウェアの開発と教育に生きたと感じている。また、MP406の製作にはrj406のインタフェースが単純であることが有用であった。MP406の基板は、プリントパターンを設計して実装することも考えたがなんとなく避けてしまった。年度末の不安感を思うと、手はかかっても2セット用意するのがよかったであろう。

ソフトウェアシステムはよく稼動したがまだ不十分である。これはその複雑さと難しさによるのであるが、経験を増やし、より長い時間をかけて開発を試みる必要があると感じている。

9年間のプロジェクトの経過を参加した学生の人数とともに示したのがつぎの表である。

	内容	院生(人)	学部生(人)
1991	演算装置の試作	0	3
1992	プロセッサVLSIの予備設計	3	4
1993	アーキテクチャの研究	5	0
1994	rj406の機能の確認と製作	2	0
	MC68008の2CPUシステムの製作	0	2
1995	並列プログラミング	0	3
1996	rj406の1CPUシステムの製作	1	0
1997	1CPUシステムのソフトウェア	0	4
	2CPUシステムの製作	1	0
1998	OSの試作	2	3
1999	OSとコンパイラの製作	2	6

このプロジェクトは学部の4年生と大学院修士課程の学生によって行われた。したがって、1年あるいは3年の間の結果をつないで行くことが要求される。結果から見ると、VLSI、計算機、ソフトウェアの製作のそれぞれが、試作、設計、実装という順序で行われ、実装されたものがその後の段階で使用された。また、4年生は多くの場合共同で1つのシステムを作り、院生は1人で1つのシステムを完成させている。

謝辞 プロジェクトを実行するとき、参加した学生諸君から多くの影響を受けた。また青柳圭祐、伊藤佳夫、大林雄次、五味智、菅原浩二、内藤祐一郎の諸君からは原稿へのコメントを受けた。rj406の製作費の多くの部分は株式会社東芝から受けた奨学寄付金による。ES2社への製作依頼にさいしてはテクノアライアンス社の吉本琢也氏にお世話頂いた。ここに記して感謝する。

参考文献

- [1] C. A. Mead and L. A. Conway: Introduction to VLSI Systems, Addison-Wesley(1980).
- [2] N. H. E. Weste and K. Eshraghian: Principles of CMOS VLSI Design, Addison-Wesley(1985).
- [3] J. L. Hennessy and D. A. Patterson: Computer Architecture-A Quantitative Approach. Second Edition, Morgan Kaufmann(1995).
- [4] 菅原浩二: rj406のアーキテクチャ, 電気通信大学情報工学専攻修士論文(1993).
- [5] 清藤麻子: 分岐命令の研究, 電気通信大学情報工学専攻修士論文(1993).
- [6] 青柳圭祐: rj406のアーキテクチャの評価, 電気通信大学情報工学専攻修士論文(1993).
- [7] A. S. Tanenbaum and A. S. Woodhull: Operating Systems: Design and Implementation Second Edition, Prentice-Hall(1997)
- [8] 五味智: マイクロプロセッサrj406とオペレーティングシステム, 電気通信大学情報工学専攻修士論文(1994).
- [9] 大林雄次: マイクロプロセッサrj406の実装, 電気通信大学情報工学専攻修士論文(1994).
- [10] Solo 1400 Databook ECPD12 Library, ES2(1990).
- [11] 内藤祐一郎: 計算機MP406の製作, 電気通信大学情報工学専攻修士論文(1997).

- [12] 渡邊坦: コンパイラの仕組み, 朝倉書店(1998).
- [13] 村永宜之: 計算機MP406のオペレーティングシステム, 電気通信大学情報工学専攻修士論文(1999).
- [14] 美濃村直之 :MP406用Cコンパイラの実装, 電気通信大学情報工学専攻修士論文(1999).

本 PDF ファイルは 2001 年発行の「第 42 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場（＝情報処理学会電子図書館）で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>