

## メディアプロセッサ D30V の検証手法

中木村清† 吉田豊彦† 山田朗† 佐藤尚和† 毛利篤史‡  
†三菱電機(株) システム LSI 開発研究所 ‡三菱電機(株) 情報技術総合研究所  
〒664 兵庫県伊丹市瑞原 4-1 TEL : (0727)84-7345  
E-mail : nakakimu@lsi.melco.co.jp

マルチメディア分野への適応が可能なメディアプロセッサとして、独自のアーキテクチャを採用した D30V を開発した。開発の過程においては、C 言語によるシミュレータを作成して演算能力評価およびテストケース作成を行った。機能検証および論理検証では、検証期間を短縮するために、検証方法や検証で用いるテストケース自身に工夫を凝らし、加えて論理検証ではテストベンチの論理合成による回路化を行った。

D30V のプロセッサコア部 300kTr に対して命令実行数約 32.5 万のテストケースを用いて検証を行った。その結果、バグは 127 個検出され、開発期間は 10 ヶ月、合計 45 人月でアーキテクチャ設計から論理検証までを完了することが出来た。

## Verification method of Mediaprocessor D30V

K.Nakakimura†, T.Yoshida†, A.Yamada†, H.Sato†, A.Mohri‡  
†System LSI Laboratory, Mitsubishi Electric Corporation  
‡Information Technology Laboratory, Mitsubishi Electric Corporation  
4-1 Mizuhara, Itami, Hyogo, 664 Japan TEL: (0792)84-7345  
E-mail : nakakimu@lsi.melco.co.jp

We have developed a Mediaprocessor D30V with an original architecture which would be suitable for various multimedia applications. In creating its test-cases, we utilized a functional simulator written in C and several ideas to reduce simulation time for functional-level and schematic-level verification. We synthesized its test-benches so that we can use the same method as functional-level verification for schematic-level.

We used test-cases of approximately 325,000 steps in functional verification of the D30V, which has 300,000 transistors. During of verification, 127 bugs are detected and fixed. It took 10 months from its architecture design to schematic-level verification, and total human resources accounts for 45 man-months.

### 1. はじめに

近年の著しいエレクトロニクス技術の進歩に支えられ、マイクロプロセッサの性能も年を重ねる毎に著しく向上している。この性能向上により、

従来はデジタル信号処理専用のハードウェアとシステム制御を行うマイクロプロセッサを組み合わせることで実現されていたマルチメディア分野への適用が可能となった。

しかしながら、マルチメディア処理を効率的に行うには従来のマイクロプロセッサや DSP (Digital Signal Processor) をそのまま高速化しただけでは不十分である。このため近年、新しいアーキテクチャを採用したメディアプロセッサが次々と発表されている。また、従来のマイクロプロセッサはこれらマルチメディアアプリケーションを効率的に処理するため、命令を大幅に拡張している。[1, 2]

D30V は、新しいアーキテクチャを採用したメディアプロセッサのひとつである。本報告では、そのアーキテクチャの概要および処理能力について述べる。また D30V を開発するに当たって用いた機能設計・検証手法や論理設計・検証手法について述べる。

## 2. D30V の命令セットアーキテクチャ

D30V はデジタル動画像処理を行う家電製品やパソコンのアドインカードなどのマルチメディア処理に適した VLIW 型メディアプロセッサである[3, 4]。

D30V の命令セットアーキテクチャ概要を表-1 に示す。D30V はマルチメディア処理を効率的に行うため、1つの命令中に2種類の演算を指定する VLIW アーキテクチャと1つの演算で2組のデータに同一演算を行う SIMD アーキテクチャを採用している。命令は 64 ビットの固定長フォーマットを用い、汎用レジスタを 64 個備えている。デジタル信号処理を高速に行うため、従来の DSP で広く用いられているハードウェアループ機能とモジュールアドレッシング機能も備える。ハードウェアループ機能はブロックリピート命令によりハードウェア制御で一連の命令群を分岐命令なしで規定回数繰り返す機能で分岐の実行時間を実質的にゼロにする。モジュールアドレッシングはオートインクリメントまたはオートデクリメントアドレッシングにおいてデータアドレスの更新をループさせる機能で、サーキュラバッファのアクセスに非常に有効である。

表-1 D30V のアーキテクチャ

- 命令フォーマット : 64ビット固定長(含:2コンテナ)
- 命令数 : 102サブ命令(条件付き実行)
- 並列処理 : 2ウエイVLIW + 2ウエイSIMD
- レジスタファイル : 32ビット汎用レジスタ64本、  
64ビットアキュムレータ2本、  
32ビット専用レジスタ10本
- DSP強化機能 : ハードウェアループ、  
モジュールアドレッシング

マルチメディア処理ではデータ精度が 16 ビットで十分な場合が多く、16 ビット固定小数点の算術演算が多用される。D30V の SIMD アーキテクチャはこの 16 ビット固定小数点演算を効率よく実行するために導入されたものである。D30V の各演算器は1つの 32 ビット演算または2つの独立な 16 ビット演算を行う様に設計されているため、1つのサブ命令で2組の 16 ビット演算を並列に実行することができる。この SIMD アーキテクチャによる2倍の性能向上と VLIW アーキテクチャによる2倍の性能向上により、D30V では周波数の4倍の演算性能を実現し、250MHz 動作で 1000MOPS のピーク性能を達成する。

図-1 に D30V のレジスタモデルを示す。レジスタは 32 ビットの汎用レジスタ 64 本、制御レジスタ 9 本、64 ビットのアキュムレータ 2 本から構成される。サブワード演算を効率よく実行するため各汎用レジスタは 32 ビットのレジスタ 1 本あるいは 16 ビットのレジスタ 2 本としてアクセスできる。レジスタ R0 は常にゼロであり、レジスタ R62 はサブルーチンコールのためのリンクレジスタとして使用される。また、レジスタ R63 はスタックポインタとして使用され、割り込み用スタックポインタ SPI とそれ以外の時のスタックポインタ SPU が PSW によりモード切り替えされるようになっている。

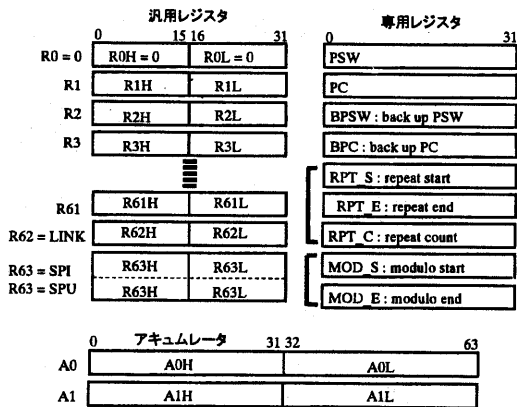


図-1 レジスタモデル

制御レジスタはプログラムカウンタ PC と PSW、それらの更新前の値でオペレーティングシステムが使用する BPC と BPSW の他、D30V の DSP 強化機能が使用する 5 本のレジスタからなる。RPT\_S、RPT\_E、RPT\_C はハードウェアループにおいて使用され、それぞれハードウェアループの開始アドレス、終了アドレス、ループ回数を保持する。MOD\_S と MOD\_E はモジュールアドレスリングで使用され、それぞれサーキュラバッファの開始アドレスと終了アドレスを保持する。アキュムレータは積和演算や精度拡張を伴う乗算で使用する。

D30V はマルチメディア処理において有効なサブワード演算を、ALU 演算、乗算、シフト演算で行う。サブワード演算では各命令でレジスタの上位側 16 ビットと下位側 16 ビットに独立な演算（ただし同一種類の演算）を行い各命令で 2 つの演算結果を得る。

また、D30V ではバック・アンバック演算をロード命令またはストア命令実行時に行う。これらの処理はバイトデータとハーフワードデータ間またはハーフワードデータとワードデータ間で可能である。例えば LD4BH 命令ではメモリ上の連続した 4 バイトの領域にある 4 つのバイトデータがそれぞれ 16 ビットデータに符号拡張され、2 つのレジスタの上位と下位の計 4 つの 16 ビット

領域にロードされる。これらのロード/ストア命令はマルチメディアデータに対しては有用である。

### 3. D30V チップアーキテクチャ

図-2 に D30V のサンプルチップのチップ写真を示す。D30V サンプルチップは D30V プロセッサコアと 32kB の命令メモリ、32kB のデータメモリより構成される。上部が命令メモリ、下部がデータメモリであり、中央部分がプロセッサコアである。命令メモリとデータメモリには SRAM を用いている。プロセッサコアには中央部のレジスタファイルを含んで 2 つの演算部 (MU: Memory I/F unit と IU: Integer unit) があり、乗算器 (Multiplier) が IU 中で大きな面積を占めている。

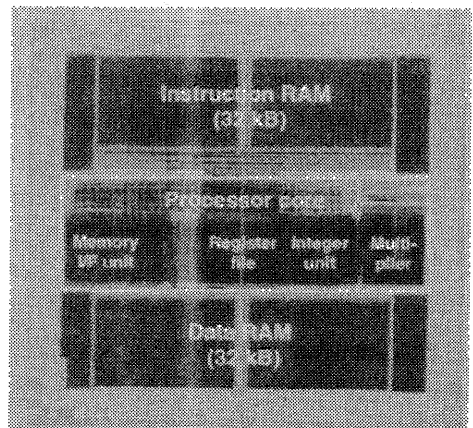


図-2 サンプルチップ写真

表-2 に D30V サンプルチップのテクノロジーを示す。このチップは 0.3 ミクロン CMOS 4 層メタル配線プロセステクノロジーで製造されており、電源電圧 2 V で 250MHz の周波数で動作する。プロセッサコア部は 300k 個のトランジスタにより構成されており、面積は 8 mm<sup>2</sup> である。64kB の SRAM をあわせたチップ全体ではトランジスタ数が 3.48M 個で面積が 37 mm<sup>2</sup> である。単相クロックによりクロック回路を単純化したことと電源電圧を 2.0V と低くしたことに加え、チップ面積、特にプロセッサコア面積を小さくした

ことにより、D30V では 250MHz 動作時で消費電力を 1.2W と非常に小さくできた。消費電力を小さくすることはメディアプロセッサを開発する上で必須の技術である。メディアプロセッサの多くは低価格システムに使用される。チップ全体の消費電力をプラスチックパッケージに格納可能な値にしない限り、チップコストをシステムが要求する値にすることはできない。

表-2 D30Vサンプルチップのテクノロジー

•プロセス	:0.3ミクロンCMOS4層メタル配線
•動作周波数	:250MHz
•ピーク性能	:1000MOPS
•チップサイズ	:6.0mm x 6.2mm
•プロセッサコアサイズ	:5.4mm x 1.5mm
•トランジスタ数	:3.48Mトランジスタ(総数) 300kトランジスタ(プロセッサコア)
•内蔵RAM	:32kB (命令) + 32kB (データ)
•電源電圧	:2.0V
•消費電力	:1.2W (typ., 2V 250MHz動作時)
•パッケージ	:135ピン PGA

#### 4. D30V の設計検証手法

D30V での設計および検証の手順を図-3 に示す。D30V は過去との互換性を一切考慮する必要のない新規に開発されるプロセッサである。そのため、チップの設計はまずアーキテクチャ設計に始まり、命令セットを決定する。さらに、決定された命令セットを用いることで、所望の演算処理能力を得られるかどうかの検討を行う必要がある。次にアーキテクチャ実現のための機能設計を行い、各機能が実現されているかどうかを確認するための機能検証を行う。その後、機能設計された機能を実現する論理回路の設計を行い、設計された論理回路が機能を実現しているかどうかを確認するための論理検証を行う。

ここで、アーキテクチャに基づく処理能力の検討には、社内でC言語によりシミュレータを作成して用いた。また、機能設計および機能検証には、Cadence Design Systems 社製の Verilog-XL を、論理設計には、Mentor Graphics 社製の

Design Architect を、論理検証には同じく、Mentor Graphics 社製の Quick Sim II を用いた。

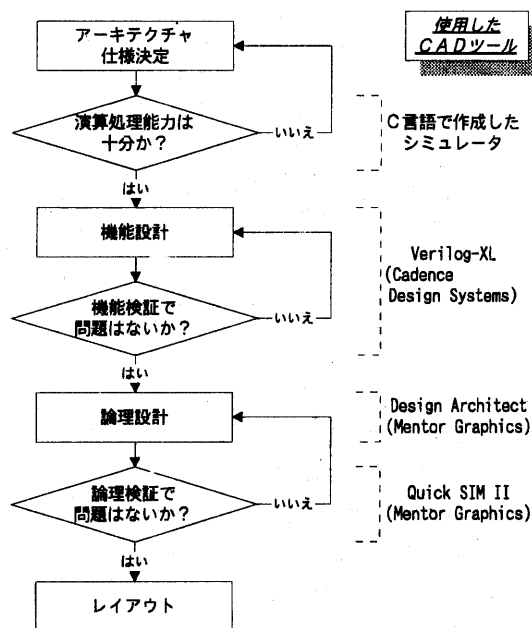


図-3 設計フローとCADツール

#### 4.1 アーキテクチャ検証

前述のようなアーキテクチャを持つ D30V に対して、実際にリアルタイムに MPEG2 デコード処理の可否を検討するために、パイプラインレベルでの処理をサポートしたパイプラインレベルのシミュレータを作成した。シミュレータはC言語で記述されており、D30V で実行可能な全ての命令のシミュレートが可能であると同時に、使われていない命令コードを使用して、テストケースのデバッグに有効なシミュレータ専用の命令もサポートしている。

このシミュレータを用いることで、処理能力の検討を行い、処理能力が不足する場合には、新たに処理能力を改善するための命令を追加し、再検討を行った。その結果、現在の D30V ではリアルタイムに MPEG2 デコード処理を行うために必要な処理能力を達成している。

シミュレータによる処理能力の見積もり結果

は、下記の通りである。

表-2 MPEG2デコード処理におけるD30Vの性能

<<ビデオ処理>> ・処理すべきビデオブロック数 : 243k ブロック/秒 ・8×8ビデオブロックのデコード性能 (1ブロック当) 逆量子化 : 174 サイクル IDCT : 500 サイクル 動き補償&画像再構築 : 80 サイクル ○総ビデオブロック処理クロック数 : 183M クロック/秒	
<<オーディオ処理>> ・処理すべきオーディオブロック数 : 1125 ブロック/秒 ・1ブロックの256点IFFT性能 : 9100 サイクル ○総IFFT処理クロック数 : 10M クロック/秒	

また、このシミュレータはチップの機能検証を行うテストケースの作成においても期待値の作成や機能モデルの動作確認のために用いることが可能であるため、非常に有用である。

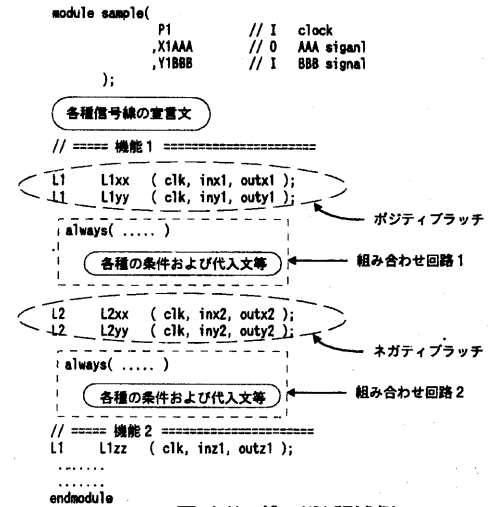
#### 4.2 機能設計

D30Vでは、Cadence Design Systems社製のVerilog-XLを用いて機能設計および機能検証を行った。よって、機能設計とはVerilogHDL (Hardware Description Language)でRTL (Register Transfer Level)記述を作成することと等価である。

VerilogHDL記述作成では、論理合成を前提としているため記述スタイルにいくつかの規則を設けた。ひとつは、ラッチを構成する部分と組み合わせ回路を構成する部分を明確に分離して、記述を行うことである。D30Vを構成する基本的な回路は、1層のクロックで動作している。そこで、図-4のようにポジティブラッチ記述 -> 組み合わせ回路1記述 -> ネガティブラッチ記述 -> 組み合わせ回路2記述 という記述スタイルとなり、VerilogHDL全体の見通しが良くなる。

ふたつめは、確実に所望のラッチを合成できるようにするためにラッチの記述をライブラリ化したことである。ラッチは、ポジティブ用とネガティブ用のラッチがあり、それぞれに対してフリ

ーランタイプとゲーテッドタイプが存在するため、合計4種類を用意した。これを図-4のようにラッチが生成されるべきである部分に記述することで、論理合成時に所望のラッチ回路を確実に得ることが可能となった。



#### 4.3 機能検証

VerilogHDLで機能設計された記述に対して、サポートする命令が正しく実行されるかどうかを検証するための機能検証においても、D30Vではいくつかの工夫を行っている。ひとつには、テストケースの生成のし易さである。前述のようにD30VではC言語で記述されたシミュレータが存在するため、機能検証のために作成したテストケースがチップの動作として期待されている動作であるかどうかを予め確認する事が出来る。そのため、機能検証の段階でフェイルするテストケースが有った場合にも、VerilogHDL記述のみをデバッグすれば良く、テストケースのデバッグ作業が不要な分だけ効率的にデバッグが可能となる。

ふたつめに、テストケース自体にも工夫を行った。チップのインターフェース信号をチェックする必要のある一部のテストケースを除いて、大部分のテストケースにおいてはレジスタへ結果を書き戻すようにテストケースを作成した。図-5

のように、テストケース作成者の期待値と、実際の VerilogHDL 記述で実現された機能での演算結果との比較を、D30V 自身の比較命令を用いて行う。この結果が不一致であった場合には、やはり D30V もサポートする条件実行のビットセット命令を用いて結果格納用に設定したレジスタのあるビットに“1”をセットする。この時、テストケースでの検証内容毎にビットの位置を振り分けることによって、どの検証でのフェイルであるかも特定することが可能となる。このようなテストケースを用いることによって、逐一チップからの出力信号を比較する必要がなくなり、結果格納用に設定したレジスタの最終的な値をチェックすることだけで、機能検証のパス/フェイルを判定することが可能となる。

```

.....
<<<検証したい機能のための演算処理>>>
.....
chkoutput:
10[repeat r32, chkb1k]
10[ldw r34, r30+, r0]          ii[nop]    <== 演算結果のロード
10[ldw r36, r31+, r0]          ii[nop]    <== 期待値のロード
10[cmpne r0, r34, r36]         ii[nop]    <== 結果の比較
10[beet/tx r60, r60, r33]      ii[nop]    <== 結果が不一致であれば、
chkb1k:                          r60のあるビットをセットする
10[rer]                          ii[nop]    <== 結果比較ループを終げる
.....

```

図-5 テストケースの記述例

また、検証のために必要な時間を短縮するためにシミュレーション上の工夫も行った。D30V は、チップ内部に2つの SRAM ブロックを内蔵しており、各メモリブロックはそれぞれ命令用 RAM とデータ用 RAM として使用される。D30V は、チップ外部とのインターフェースは、27MHz のクロックに同期して動作するが、実際の処理を行う内部の演算および制御回路は 250MHz という高速で動作する。D30V はこの 250MHz で動作する内部命令 RAM に置かれた命令コードを実行する場合に、はじめてその処理能力を発揮することが可能となる。

そのため実際の使用においては、リセット後にチップ外部に用意したプログラムコードを内部命令 RAM に転送し、転送終了後プログラムカウ

ンタの値を内部命令 RAM にジャンプさせることによって、演算処理動作を始める。ここで、問題となるのは、プログラムコードの転送は外部とのインターフェースで行われるために 27MHz に同期しており、多くのシミュレーション時間を必要とすることである。加えて、この転送を行うプログラムコードは、転送するプログラムコードのサイズ以外には違いが無く、各テストケース毎に毎回行う意味が全く無い。そこで、Verilog-XL が備えている機能であるメモリの内容を直接書き込むためのコマンドを用いることにした。このためには、機能検証で用いる SRAM メモリのモデルを VerilogHDL で記述しておく必要がある。

機能検証用に作成されたテストケースは、外部から内部命令 RAM に転送した後、実行されることを想定して作成されている。そこで、テストケース実行時に内部命令 RAM に転送されるプログラムコードのみを抽出して、Verilog-XL のコマンドで内部命令 SRAM モデルに直接書き込み可能なフォーマットに変換するスクリプトを作成した。また、内部データ RAM についてもデータの転送が必要な場合には同様の処理が行えるようになっている。

このような手法を用いることで、機能検証に必要な時間を短縮することが可能となる。

#### 4.4 論理設計

D30V の論理設計は、制御信号を生成するための制御部と、演算回路から構成されるデータバス部では異なっている。250MHz という高速で動作するためには、ALU (Arithmetic Logical operation Unit)、シフト、乗算器といった演算器は速度的にネックとなることが十分予測される。また、チップとしての処理能力は、演算器の動作速度によって左右されやすいことから、高速動作が可能な演算器が必要とされる。また、これらの演算器は設計早期の段階で機能がフィックスされることが多く、修正が比較的少ない。そのため、回路の動作速度を優先して、設計者自身が自ら手設計を行うことにした。

これに対して、制御部ではバグが発生しやすく、バグのフィックスのために、度々改訂が行われるために、機能検証の完了以前に、その仕様をフィックスすることは困難である。また、十分に高速な合成用のライブラリが準備できれば、論理合成を行っても動作速度を満たすことが可能である。よって、制御部では論理合成ツールを用いて、論理回路を得ることにした。

#### 4.5 論理検証

論理回路として実現されるのは、通常 LSI チップとなる部分のみであった。このため、論理検証ではその検証内容に応じて、チップの入力信号ピンに対しては信号をドライブし、出力信号ピンに対しては出力される信号のコンペアを行い、入出力信号ピンに対しては、入力モード時には信号をドライブし、出力モード時には出力される信号のコンペアを行っていた。これらの入力ピン用ドライブ信号や出力ピン用のコンペア信号を作成は、機能検証時の LSI チップ部分のインターフェース信号をダンプした結果を変換して行っていた。しかしながら、この手法では機能検証後にしか論理検証を行うことが出来ない。また、機能検証に用いる Verilog-XL と、論理検証に用いる Mentor Graphics 社の Quick SIM II では、信号の取り得る状態数等で完全には一致しない場合が存在する。

そこで、D30V では、論理検証においても機能検証と同じ手法を用いることにした。Quick SIM II は、メモリモデルをサポートしており、さらに Verilog-XL と同じようにメモリの内容を Quick SIM II のコマンドを使って直接書き込むことが可能である。このため、機能検証用に作成したテストケースを、機能検証と同様の手順で用いて論理検証を行える。また、LSI チップ部分のインターフェース信号の差異を吸収するために、VerilogHDL で記述されたテストベンチを合成し、図-6 のようなモデルを作成して論理検証を行うという手法を用いた。そのため、テストベンチの VerilogHDL は論理合成可能な記述スタイルを採

っている。

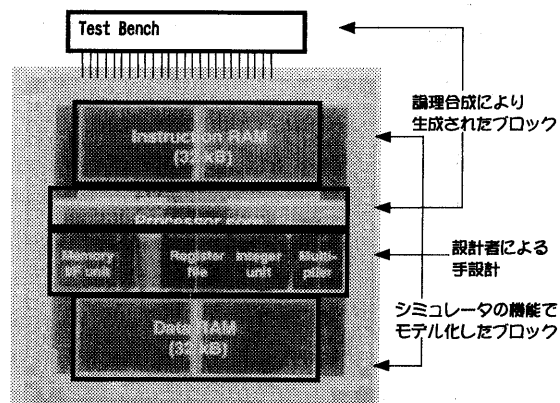


図-6 論理検証モデル

以上のように、テストベンチの合成と、論理検証用で用いるメモリモデルへの内容を直接設定する手法を用いることで、機能検証と全く同じ手法で論理検証が可能となった。そのため、従来のように論理検証用のテストケース作成のために、機能検証を行い、インターフェース信号をダンプして変換するという作業を不要とすることで、より短時間で論理検証を行うことを可能とした。

#### 5. テストケース

今回の機能検証および論理検証で用いたテストケースの総命令実行数は約 32.5 万であった。テストケースの内容の概略を表-3 に示す。

表-3 テストケース概略

テスト項目	ケース数	実行命令数
単体命令	204	105240
条件実行等	66	32882
並列実行*	15	6702
シリアル実行*	16	14194
バイパス	59	12664
単一 EIT	29	46343
複合 EIT	35	38426
バス I/F	40	68448
合計	464	324899

\*: 2 ウェイ VLIW 命令の実行順序

## 6. 検証結果および設計人工

機能検証における設計時期とバグの発生件数との関係を図-7 に示す。バグの合計は 127 件であった。

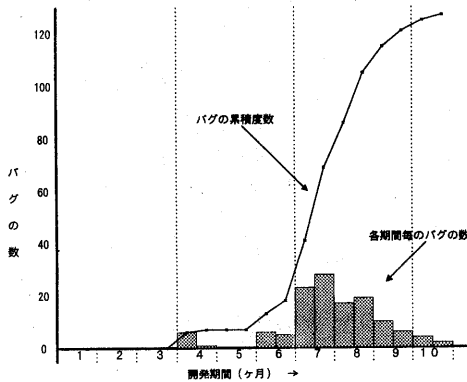


図-7 バグの発生件数

論理検証では、ALU、シフタ、乗算器等の各演算器は演算器単位での動作確認が行われていたこともあり、手設計部 (=データバス部) のバグは、わずかに 15 件程度であった。これらのバグは単なる配線ミスであり、論理検証の初期で改修がほぼ完了した。

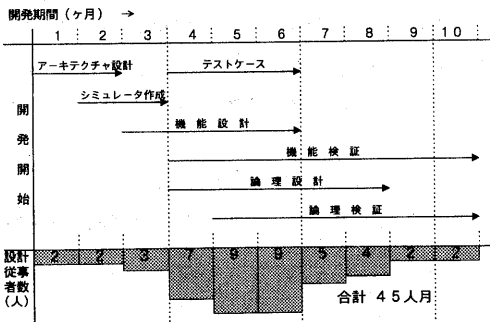


図-8 D30V開発人工

D30V では、アーキテクチャ設計から論理検証完了までの設計期間と人工を図-8 に示す。設計期間は 10 ヶ月であり、この期間に投入された人工は、45 人月であった。

## 7. まとめ

新規開発のメディアプロセッサとして D30V を開発した。開発工程において、アーキテクチャ・機能・論理の各検証実施時に、いくつかの工夫を施すことで検証時間の短縮を行った。

その結果、開発期間は 10 ヶ月、合計 45 人月でアーキテクチャ設計から論理検証までを完了することが出来た。

## 謝辞

本研究の機会を与えて頂いた三菱電機 (株) システム LSI 開発研究所 設計技術開発第二部 岩出部長に感謝します。

## 参考文献

- [1] 「'96 Hot Chips シンポジウムのデッドヒート②」、エレクトロニクス 1997 年 1 月号、pp.9-16.
- [2] 「マルチメディアがマイクロプロセッサの設計を変える」、日経エレクトロニクス 1996 年 12 月 2 日号、pp.92-100.
- [3] Holmann, E. et al. : A VLIW Processor for Multimedia Applications, Hot Chips VIII, pp.193-202 (1996).
- [4] Yoshida, T., et al. : A 2V 250MHz Multimedia Processor, ISSCC Digest of Technical Papers, pp.266-267 (1997).
- [5] 吉田豊彦, “VLIW 型メディアプロセッサ”, 情報処理, VOL.38, No.6, pp.499-506 (Jun.1997)