

2 個のベクタプロセッサを集積した 128 bit マイクロプロセッサの開発

鈴 置 雅 一†

システム LSI の設計は、規模が大きくなるにつれ、基本設計と検証の重要性がより高まりつつある。さらに、システム LSI のプロセッサは、その構成は特定のタイプのアプリケーションに特化したものとなる。このためシステム全体の性能を考えると、想定されるアプリケーションプログラムを切り離して考えることはできない。ここでは、次世代ゲーム機用に新規に開発された 128 bit マイクロプロセッサの概要とその仕様を決定するのに用いた手法、使用した回路の検証手順について述べる。

Development of a Microprocessor with a 128b CPU, 2 Vector Processors with 10 Floating Point Units

MASAKAZU SUZUOKI†

Basic design and simulation strategy are more important to develop the large scale system integrated LSI. Especially, the total performance of the LSI depends on the target application architecture when the chip is programmable. In this paper, the process to determine the macro architecture of the newly developed 128 bit microprocessor is introduced.

1. はじめに

いよいよ、コンピュータが本格的に家庭に導入される日が近づいているといわれている。家庭用として見たとき、コンピュータの性能とコストの比は一般消費者が受け入れられるレベルに近づいてきている。そうした思惑から、メモリに代わる半導体業界の切り札として、マイクロプロセッサを核とし、周辺デバイスを 1 チップに集積化した LSI (システム LSI) が次々に開発・発表されている。一方で汎用のマイクロプロセッサも家庭用の用途を見据えて画像音声処理用の命令の拡張を進めている。

これらのマルチメディアプロセッサでは、コンシューマ用であっても、表示するデータ自体を物理法則に沿ってシミュレーションすることが可能になりつつある。ゲームのキャラクタはあらかじめ計算された動きを再生するだけでなく、個々のオブジェクトが物理法則にのっとり、他のオブジェクトとの相互干渉を行いながら、次の動作を自律的に決定するまでになった。

こうした、かつてはスーパーコンピュータが必要であった処理を家庭に持ち込むことができるようになった背景には、半導体技術の進歩により多くの機能素子を単一のチップ上に集積できる技術 (システム LSI) が確立されたことにほかならない。

しかしながら、大規模なシステム LSI の設計は、同時に開発期間や開発費用も膨らむ。設計が完了し、レイアウトが進んだ時点で、性能上の思わぬ問題点が見つかった場合、仕様の変更は大きな痛みをとまなう。システムを複数の LSI の組み合わせで実現する場合よりもシステム LSI の方がよりトライアンドエラーの困難は大きい。したがって、システム LSI の設計では最初の基本設計の良否が非常に重要となる。

加えて、システム LSI の設計は特定の組み込み用途を想定される場合がほとんどである。つまり、プロセッサはプログラマブルでありながらも、その構成は特定のタイプのアプリケーション (ここでは、3D コンピュータグラフィックスおよび物理シミュレーション) に特化したものとなる。このためシステム全体の性能を考えると、想定されるアプリケーションプログラムを切り離して考えることはできない。どの部分をハードウェア化し、どの部分をソフトウェアで行う

† ソニーコンピュータエンタティメント
Sony Computer Entertainment, Inc.

かの判断もまた重要である。

ここでは、次世代ゲーム機用に新規に開発された 128 bit マイクロプロセッサの概要とその仕様を決定するのに用いた手法、使用した回路の検証手順、およびそこでのアプリケーション開発手順について述べる。

2. 基本設計

我々は、プロセッサの設計にあたっては、まずアプリケーションを厳密に定義した。目標設定が漠然としてしまうと、設計のブレークダウンの過程で詰めきれない部分が残る、どうしても最小公倍数的な解に落ち着きがちになると考えたためである。そこで、まず、ビデオゲーム機などの家庭用のコンピュータとして何が必要なのかが調べられた。次いでそれに基づいて実際のゲームタイトルが収集され解析された。

そもそも家庭用コンソールは、コンピュータの深い知識を持たない層が対象であり、目的がエンタテインメントであることから、通常の PC とは違う方向が求められる。第 1 にコンソールは製品寿命の途中で機能・性能が変化してはならない。コンソールの仕様の変更は、たとえそれが性能の向上であっても、開発者、ユーザともに利益にはならない。コンシューマの世界では、すべてのソフトウェア(タイトル)とハードウェアは完璧に 1 対 1 でなければいけなくて、これは、すべての CD がすべての CD プレイヤで再生できることと同じである。途中で仕様を変えられないのであれば、発売の最初の時点から可能な限りの最先端の技術を投入する必要がある。

第 2 に、ユーザは仕事でコンピュータを操作するわけではない。ユーザを魅了するための「つかみ」、つまり見た目のグラフィックスが心を打つものでなくてはならない。そのための高度なグラフィックスには多くの計算が必要とされる。

第 3 に、真の良質なエンタテインメントには、思考推論や、物理現象シミュレーションも欠かせない。多くのゲームソフトを実際に評価していくうちに、前もって用意されたテーブルを参照するのではない、計算に裏打ちされた、状況に応じて適応的に変化するキャラクターの動きこそが、ゲームやアプリケーションに奥行きをもたらすことが分かってきた。

3. プロセッサ構成

以上からコンピュータエンタテインメント機器が目指すリアルタイムの画像合成を実現し、それが長い間プラットフォームたるためには、最新の半導体技術を導入して、膨大な計算能力、特に浮動小数点演算能力が

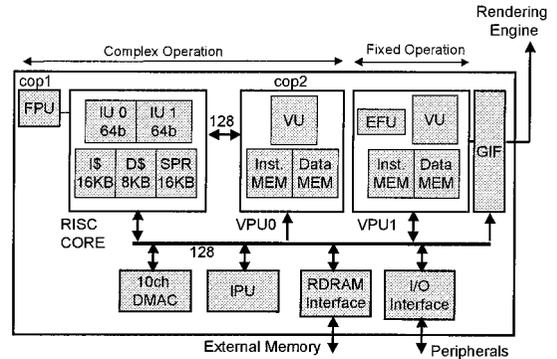


図 1 プロセッサブロック図

Fig. 1 Processor block diagram.

必要とされることが分かる。3次元空間の物体の位置と照度を計算しそれを2次元平面(ディスプレイ)に投影するだけでも十分な計算能力が必要であり、加えて、各物体の位置自体を入力に反応し再計算するためにも膨大な計算が必要なためである。

このために我々は、高いバンド幅を持つ共通バス上に複数の演算ユニットを並列に1チップ上に実装する構成をとった。プロセッサのマクロアーキテクチャを図1に示す。複数のチップに分割しなかったのは、ユニット間のバンド幅を稼ぐためと、将来のシュリンクを考慮してのことである。チップに集積するプロセッサの1つには全体の制御を行うRISCプロセッサを、他の2つには同じマイクロアーキテクチャを持つVLIW-SIMDタイプのベクタプロセッサを採用した。このほかにテクスチャデータ(3次元物体の表面を装飾するために使用される画像データ)を解凍するための画像解凍エンジンを搭載した。

LSIには128bit MPUコアに2基のベクトルユニット(Vector Processing Unit: VPU0, VPU1)および画像処理エンジン(Image Processing Unit: IPU)が集積され、各ユニットは10chのDMAコントローラ(DMAC)で接続される。これらはMIPSI準拠のコアプロセッサ(MPU)によって制御される。また外部メモリとしてはダイレクタラムバス(RDRAM)が採用された。各ユニットは独立に浮動小数点積和演算器(Floating Multiply-Adder: FMAC)と浮動小数点除算器(FDIV)を備え、各VPUはそれぞれ4個のFMACおよび1個のFDIV、MPUは浮動小数点コプロセッサ(FPU)として1個のFMACと1個のFDIVを備える。さらにVPU1は初等関数ユニット(Elementary Function Unit: EFU)として、さらに1個のFMACと1個のFDIVを持たせた。こうしてシステムには合計10個のFMACと4個のFDIVが

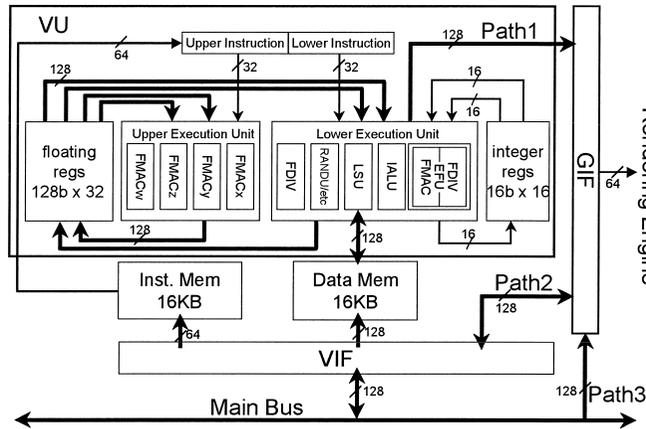


図2 VPU1ブロック図
Fig.2 VPU1 block diagram.

表1 主要諸元
Table 1 Main features.

CPU	MIPS III 2 issue super scalar with 128b multimedia extension
I Cache	16KB 2way
D Cache	8KB 2way
Scratch Pad (SP) RAM	16KB
Data Bus	128b
FPU	1FMAC + 1FDIV
VPU	5way 64b VLIW
VPU0	4FMAC + 1FDIV
VPU1	4FMAC + 1FDIV
Elementary Function Unit	1FMAC + 1FDIV
IPU	MPEG2 Decode Accelerator
DMAC	10ch
System Clock	250Mhz

表2 VPU 主要諸元
Table 2 VPU features.

Feature	VPU0	VPU1
FP Registers	128bx32	
Integer Registers	16bx16	
FMAC	Single Precision FP x4	
Multiply-Add	1 cycle	
Min/Max	1 cycle	
FDIV	Single Precision FP x1	
Divide	7 cycles	
Square Root (SQRT)	7 cycles	
Inverse SQRT	13 cycles	
Instruction Bus	64b	
coprocessor Bus	128b	-
Instruction Memory	4KB	16KB
Data Memory	4KB	16KB

集積され、これらはDMACを介してメインメモリを共有し、独立・並列に動作することができる。システムの基本仕様を表1に示す。

MPUは2ウエイ・スーパースカラ・のMIPS-IIIベースのマイクロプロセッサであり、データバスは完全に128bit幅を備える。32本の汎用レジスタはすべて128bit幅を備え、128bitのコプロセッサバスを經由して、同じく128bit幅を持つコプロセッサ(VPU0)と接続される。128bitレジスタは8×16、16×8、32×4のフィールドに分割されてそれぞれのフィールドに同じ処理を行う命令(マルチメディア命令)が追加されている。MPUは16KBの命令キャッシュ、8KBのデータキャッシュを持ち、これらはシングルサイクルでアクセスが可能である。このほかに高速のローカルメモリとして16KBのSPR(ScratchPad RAM)を備える。

VPUは、それぞれ4個のFMACおよびFDIVを備え、マイクロプログラムにより制御される。VPU

のブロック図を図2に、基本性能を表2に示す。

プログラムは、64bit LIW(Long Instruction Word)形式をとり、各命令はSIMD(Single Instruction Multiple Data)構成をとる。LIWとは1つのサイクルであらかじめ決められた複数個の命令を同時に実行できる機能であり、VPUは32bit命令を2つ同時に実行することができる。また、SIMDは、複数のデータに対して同じ操作を同時に適用できる構成であり、VPUでは、1命令で4つのFMACを同時に操作し、4次元ベクタの計算を1サイクル(スループット)で行うことができる。

動画やテクスチャ画像のような2次元画像は、IPUによって圧縮解凍される。解凍方式はMPEG2およびそのサブセットになる。IPUはテクスチャデータを解凍するだけでなく、MPEG2のデコードを行う機能を持つ。デコードにあたってはメインメモリをワークメモリとして流用する。

こうした、アーキテクチャの概要を決めるにあたっては、まずは、多くの3次元ゲームアプリケーション

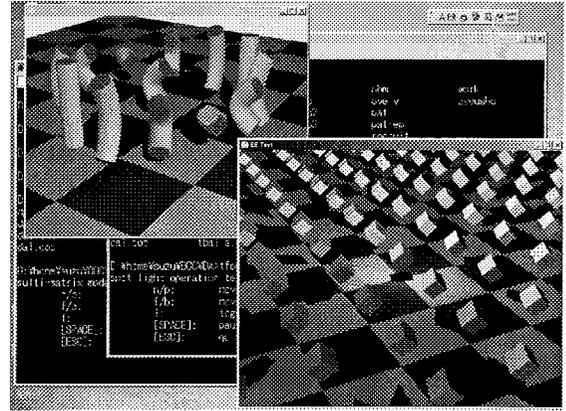
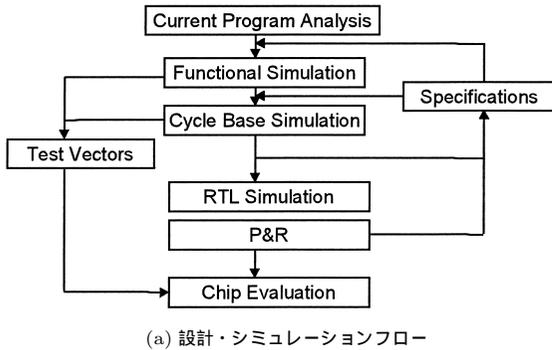


図3 ソフトウェアシミュレータ

Fig. 3 Software simulator.

が解析された。プログラムの構造を解析した結果、プログラム物体の挙動を決定する処理（挙動処理）と、決定された物体の位置を座標変換し、データを照光処理を施し2次元に投影された多角形に変換する処理（座標処理）は異なる時間スケジュールで実行されていることが分かった。

座標処理のプロセスは、外部のレンダリングエンジン（フレームバッファへ描画を行うハードウェアアクセラレータ）のポリゴン描画のパイプラインに同期して起動される。レンダリングエンジンが最大の性能を発揮できるようにするためには、レンダリングエンジンからの要求に応じて定期的にデータを供給する必要があるためである。一方、挙動処理にはそうした時間的な制約がないものの、その代わりに処理の複雑度は座標処理よりも高いものが要求される。こうした複雑な処理は浮動小数点演算能力に加えて本来のマイクロプロセッサの汎用的な性能が要求される。

この結果をふまえて、座標演算用のプロセッサと挙動処理用のプロセッサを独立に配置する構成が決定された。さらに、定型処理の多い座標演算には VLIW ベクタプロセッサ（VPU1）を、複雑な処理が必要な挙動処理には高級言語での開発が可能な汎用 RISC プロセッサとそのコプロセッサとしてもう1基のベクタプロセッサ（VPU0）が搭載されることになった。異なる性格の処理は異なる特性を持つプロセッサに割り振られる。処理の結果は最後にマージされてレンダリングエンジンへ転送される。

ここでの基本データは、現在市販されている自社のゲームプログラムを解析することで蓄積された。メインバスのトラフィックがサンプルされ、プログラムカ

ウンタが推移していく過程のみならず、キャッシュミスの割合、割込みの頻度、コンテキスト切替えのオーバーヘッドの率などの統計情報が集められた。場合によっては実際にグラフィックプログラムを作成しそれをシミュレーションしながら実際のトランザクションの統計を収集することも行った。

4. シミュレーション

こうした過程で分かったことは、システム LSI の設計にあたっては、回路設計そのものよりもむしろ、基本設計の妥当性の評価と、動作検証が従来にまして重要だということである。効率的なシミュレーションはこの評価、検証の2点で必要不可欠となる。そのため、ソフトウェアシミュレーションは、その段階に応じて多数のものが用意された。まず前章で述べた基本構成が実装され、そこで実際にプログラムを開発しながら細かい設計が詰められた。結果をもとにシミュレータはリファインされ、そのままテストベクタ作成ようの参照仮想機械として使用された。

4.1 基本設計検証

まず基本検証を目的とした論理シミュレータが作成された。ワークステーション上に実装された仮想マシン上で、設計者自身が実際に3Dグラフィックスアプリケーションを作成し、性能のバランスが決定された。テストプログラムのシミュレーション結果からボトルネックの要素が抽出され、余剰な機能は削られ律速する部分は強化された。結果はシミュレータに即座に反映され、解析が繰り返された。その過程とシミュレータの画面を図3に示す。

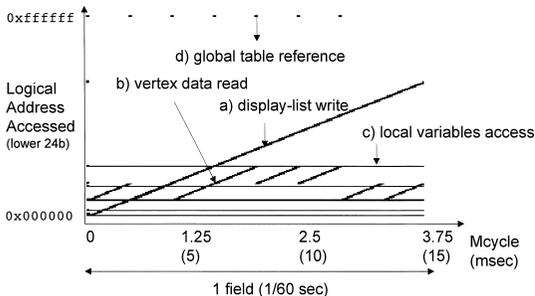


図4 メモリアクセスパターン
Fig. 4 Memory access pattern.

4.2 メモリアクセスパターン

まず、キャッシュのヒット率、メモリアクセスパターンが検討された。命令キャッシュに関しては、命令キャッシュのサイズはプログラムの最適化によって小さくすることができることが分かった。ゲームプログラムでは高機能で肥大なオペレーティングシステムを必要しないため、アプリケーション自身がキャッシュを意識したプログラムを作成することができるためである。多くのプログラムのプログラムカウンタの推移を集計することでキャッシュサイズが決定された。データキャッシュに関してはグラフィックスプログラム特有のアクセスパターンが検出された。典型的なパターンを図4に示す。

図の横軸は時間を表し、縦軸はアクセスされたメモリアドレスを示す。連続的にアドレスが増加するアクセスパターン (a) はディスプレイリスト (描画命令列) を出力するためのライトアクセスである。(b) は3次元形状データを読み出すためのリードアクセスで、同じ形状を複数回読み出しているのは、これを異なる位置に配置して表示しようとしているためである。(c) は自動変数とスタックのアクセス、(d) は広域変数のアクセスであり、(c)、(d) は通常のコンピュータプログラムのアクセスパターンとほとんど変わらない。パターン(c)、(d) に対してはデータキャッシュは有効に働くが、(a) でライトされたデータは再び参照されることはないのでキャッシングの効果はない。(b) のパターンも複数回参照されるものの、そのすべてをキャッシュに常駐できるほど容量が小さくない。こうして見ると、通常のマイクロプロセッサで行われているキャッシュシステムが、ターゲットアプリケーションでは有効に働かないことが分かる。

一方、メインプロセッサとベクタプロセッサでメモリを共有するため、キャッシュのコンシステンシも問題となった。メインメモリの内容が他のプロセッサによって非同期に更新される場合があるため、キャッシュ

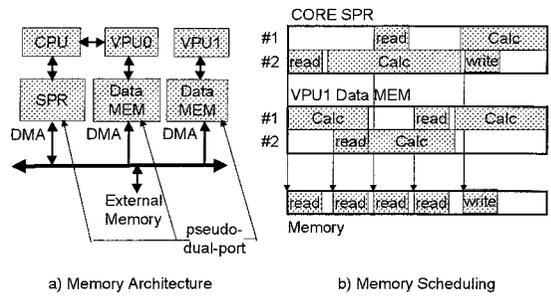


図5 メモリアーキテクチャの概要
Fig. 5 Memory architecture.

の内容とメモリの内容の整合性を保つのが難しい。仮に高度なスヌープ機能を導入した場合でも、多くのキャッシュラインがプログラムからは見えないところでフラッシュされてしまうため、プログラミングに困難をきたすことが判明した。ゲームのようなリアルタイムシステムではプログラムモジュールごとの実行サイクルの予算が厳密に管理されており、こうした点は結果としてパフォーマンスの低下を招く。

こうした結果をふまえて、図5に示すようなダブルバッファ可能なローカルメモリとそれに接続されるDMAコントローラの構成が採用された。DMAコントローラのみが各ローカルプロセッサヘータを供給し、各プロセッサがバスマスタになることはない。DMAコントローラを制御できるのはメインプロセッサだけなので、結局、メインバスのトラフィックはメインプロセッサのプログラムから一元管理できることになる。こうした構成のもとで再度シミュレータが構築され、細かいバッファサイズや転送単位が評価された。

4.3 負荷の評価

次いで、いくつかのグラフィックスのアルゴリズムがベクタプロセッサシミュレータ上でプログラムされ、オブジェクトの1頂点の座標を計算するサイクル数が評価された。高度なグラフィックス処理を行おうとすれば1頂点あたりに必要な演算量は増大していくが、その一方で頂点列を受け取ってそれを描画するレンダリングプロセッサ側の処理時間は変わらない。むしろ複雑な図形を描画するようになるにつれ、各ポリゴンの描画面積が小さくなることから、頂点あたりで見ただけの場合のレンダリング処理の負荷は軽くなる傾向にあることが分かった。このため、座標変換などの浮動小数点処理の時間がネックとなり、レンダリングプロセッサがアイドル状態にある期間が長いことが多くのプログラムで観測された。

これを解決するためにレンダリングプロセッサへのマルチパス回路が強化された。ベクタプロセッサ

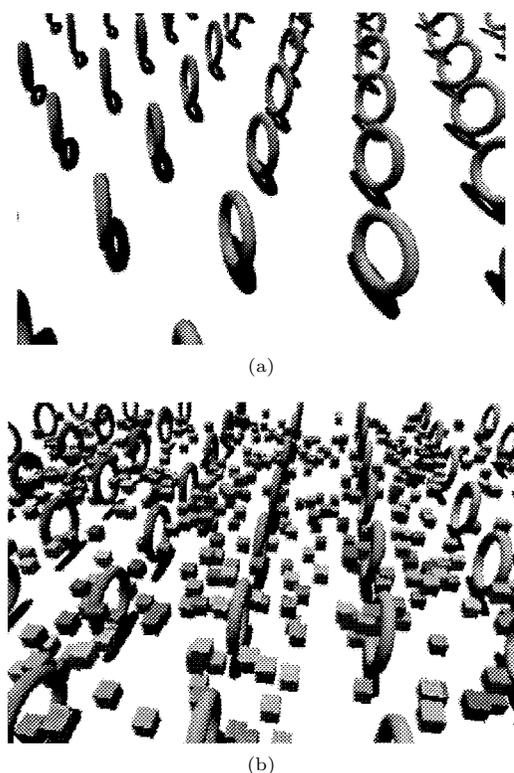


図6 複数の描画命令例の同時実行列
Fig.6 Example of multiple display list.

VPU1からの出力は直接レンダリングプロセッサへ送
出されるが、レンダリングプロセッサが待機状態にな
れば、メインメモリにあらかじめ蓄積されたディスブ
レイリスト(描画命令列)が自動的に送られる。

マルチパスのシミュレーション例を図6に示す。
VPU1が計算したオブジェクトを(a)に、メインブ
ロセッサとコプロセッサVPU0が生成し、いったん
メインメモリへバッファリングされたディスプレイリ
ストが合成された結果を(b)に示す。

4.4 機能検証

基本仕様が決定された後は、それに基づいたより正
確な論理シミュレータが作成された。これは、仮想参
照機械(God machine)として用いられ、検証ベク
タの作成とリファレンスデータの生成に用いられた。
同時に細かい性能の検討にも用いられた。論理シミュ
レータには実際にフレームバッファの表示を行うプロ
グラムが接続され、結果はハードディスクレコーダへ
蓄積された。十分なデータが集まった時点で、ハード
ディスクレコーダ内の画像が動画像として再生され、
演算精度が画質に与える影響が検討された。浮動小数
点の誤差精度、ピクセルの解像度、メモリの容量、画
像合成の品質が検証された。

4.5 ハードウェア回路シミュレータ

機能シミュレータで作成されたベクタは実行サイク
ルが小さくなるように手直しされたのち、RTLシミュ
レーションが行われ、同時にハードウェア回路シミュ
レータにもかけられた。ハードウェア回路シミュレ
ータは米国QuickTurn社の製品を使用した。

4.6 テストチップ検証

最初のテストサンプルは0.25 μ mルールで設計され
た。全体のデザインは0.25 μ mであるが速度を上げ
るためゲート長さは0.18 μ mとなる。動作周波数は、
CPUコアとベクタプロセッサでは250MHz、外部イン
タフェースを含む他の回路は125MHz。消費電力は、
250MHz動作で13W。内訳はCPUコア、ベク
タプロセッサ、他のペリフェラルで、順に42%、44%、
14%である。想定プログラムは汎用のもではなく、
このアーキテクチャに最適化されているため並列度が
高く、結果消費電力も少なくない。そのため、内部パ
ワーラインのメタル層幅は、まずは静的に評価され、
そのフィードバックを受けて部分ごとに精密に設定さ
れた。

図7にチップ写真を示す。左下3分の1が、キャッ
シュメモリを含むCPUコアであり、左上に2基のベク
タプロセッサが配置される。右側には10chのDMAコ
ントローラ、画像デコーダ、IOインタフェース、およ
びメモリインタフェースがある。メモリはDRDRAM
(DirectRambus RAM)が接続され、2chのメモリ
チャンネルが使用される。

チップの検証は、回路規模に比べて外部から観測で
きるピンには限りがあるため、テストの多くはソフト
ウェアで行われた。専用に開発されたオペレーティ
ングシステムがその上でアプリケーションプログラムを
起動し、その過程でさまざまな性能(表3)が観測さ
れた。テストに用いた基板を図8に示す。

5. まとめと課題

大規模なシステムLSIとして、浮動小数点ベクタブ
ロセッサを搭載した128bit次世代ホームエンタテイン
メント用プロセッサを開発した。設計の段階に応じ
て複数のシミュレーションを行い、基本仕様の評価と
検証ベクタの蓄積を効率良く行うことが大規模LSIの
開発では重要であることが分かった。

特に、システムLSIの課題としてターゲットアプ
リケーションの収集が重要であることも明らかになっ
た。プロセッサの性能を上げるための提案は多くな
されているが、どれを採用するかを判断するには、実際
のプログラムをシミュレータ上で走らせ、客観的な統

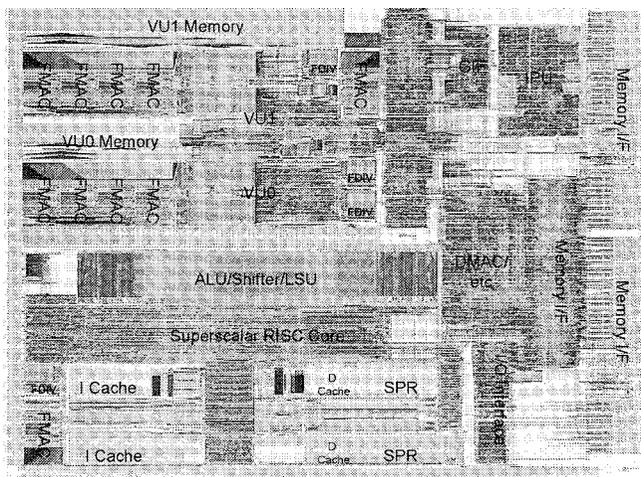


図7 チップ写真
Fig. 7 Die photograph.

表3 3DCG 計算性能
Table 3 Performance of 3D graphics.

Operation	Throughput
Effective Bus Band	1.7 Gbyte/sec
Genuine Floating Multiply	5GFLOPS
3DCG Simple Operation	
-without lighting	55M polygons/sec
-with parallel light	32M polygons/sec
-with parallel light and fog	30M polygons/sec
Bezier Surface Patch	13M polygons/sec
Image decompression (macroblock decode)	125M pixels/sec

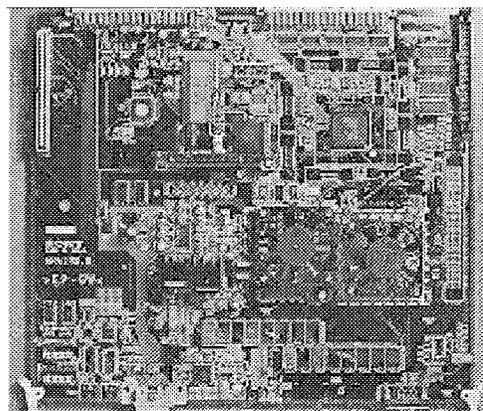


図8 評価基板
Fig. 8 Evaluation board.

計情報を集めるのが確実である。もっとも、今回の開発では、幸い開発者の多くが現行のビデオゲーム機的设计にも携わっており、現行ゲーム機上でのアプリケーションの統計を採集できた。しかし通常はこのような恵まれた状況はなかなか実現しない。また、この場合であっても、対象のプログラムがアーキテクチャ

の異なる機械（現行のゲーム機）上でのプログラムであり、収集された統計情報の普遍性には限界がある。さらに、シミュレータ上での評価は、処理時間上の都合から実行できるプログラムの規模が限られ、長時間にわたった十分な評価までには至らない。シミュレーション時間に関してはFPGAを使ったハードウェアシミュレータの期待も高かったが、使い勝手の面から実際には十分な成果を発揮できるにはいかなかった。こうした問題を解決することが、将来の非常に大規模でかつ目的特化型のシステムLSIの設計にあたって、ますます重要になる課題であると思われる

参考文献

- 1) 元岡 達, 菅野 卓ほか: VLSI コンピュータ, 岩波書店 (1985).
- 2) Patterson, D.A. and Hennessy, J.L.: *Computer Organization & Hardware/Software Interface*, Morgan Kaufmann (1994).
- 3) Foley, J., et al.: *Computer Graphics*, Addison-Wesley (1990).
- 4) Kutaragi, K., et al.: A microprocessor with a 128b CPU, 10 Floating-Point MACs, 4 Floating-Point Dividers, and an MPEG2 Decoder, International Solid State Circuits Conference '99 Digest of Technical Papers, pp.256-257 (1999).
- 5) Raam, F.M., et al.: A High Bandwidth superscalar Microprocessor for Multimedia Applications, '99 Digest of Technical Papers, pp.258-259 (1999).
- 6) Suzuoki, M., et al.: A Microprocessor with a 128-Bit CPU, Ten Floating-Point MACs, Four Floating-Point Dividers, and an MPEG-2 De-

coder., *Nov.'99 Journal of Solid State Circuits* p.1608 (1999).

- 7) Kunimatsu, A., et al.: 5.5 GFLOPS Vector Units for Emotiv Synthesis, *Proc. Hot Chips 11* (Aug. 1999).
- 8) Ide, N., et al.: 2.44 GFLOPS 300Mhz floating-point vector processing unit for high performance 3D graphics computing, *Proc. ESS-CIRC '99* (Sep. 1999).

(平成 11 年 12 月 20 日受付)

(平成 12 年 2 月 4 日採録)



鈴置 雅一 (正会員)

1987年東京大学工学部電子工学大学院卒業. 1987年ソニー(株)入社. 情報処理研究所配属. 家庭用ゲーム機 PlayStation, PlayStation2の開発を経て現在ソニーコンピュータエンタテインメント開発研究本部システム開発部所属.