2E－4

1チップMPEG-2デコーダLSI開発における検証

鈴木和雅†, 浦本紳一‡, 服部孝†, 橋詰雅樹†

†三菱電機(株) 情報技術総合研究所
‡三菱電機(株) システムＬＳＩ事業統括部

1. はじめに

近年，CPUコアを含んだシステムLSI開発においては，その増加したLSI規模や機能に対応するための検証量の多さが問題になっている。

本稿では，今回，我々が開発を行った1チップMPEG2デコーダLSIにおける検証方法について述べる。

2. 1チップMPEG2デコーダLSI

2.1 概要

今回，我々が開発を行ったのは，MPEG2のオーディオ，ビデオ，システムの各デコーダを1チップに収めたMPEG2デコーダLSIである。内部にはCPUコアを含んでおり，このCPUコアでオーディオデコード，及び，システムシステムデコードのS/W処理を行う。ビデオデコードに関してはH/Wで処理を行う。

外部に16Mbit SDRAMを1つ接続することにより，NTSC/PALのMPEG2 MP@MLのデコードが実現可能である。

2.2 ブロック構成

図1に，1チップMPEG2デコーダLSIの概略図を示す。

ビットストリームインターフェースより入力されるエンコードされたMPEG2ビットストリームは，内蔵CPUにおいてオーディオ，ビデオに分割された後，外部SDRAMに蓄積される。ビデオデータは，ビデオデコーダ部でデコードされビデオインターフェースから出力される。一方，オーディオデータは，内蔵CPUにおいてデコードされ，オーディオインターフェースより出力される。

制御に係るレジスタへのアクセスや，内部状態のモニタは，ホストインターフェース部を介して，外部に接続されたホストから行われる。

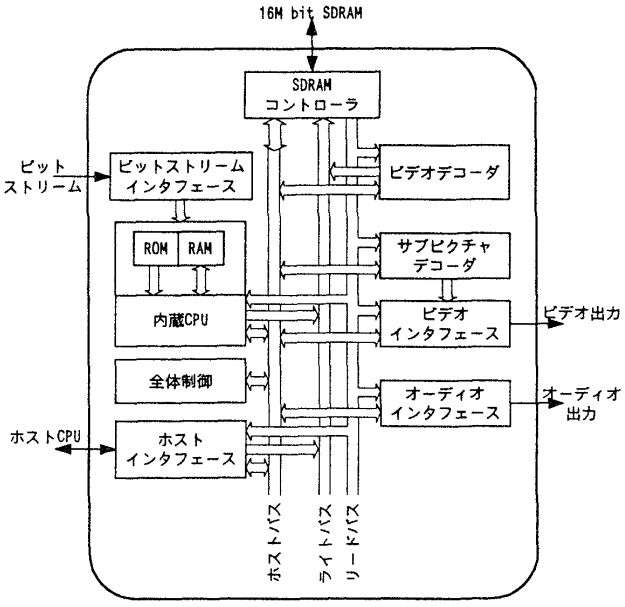

図1 1チップMPEG2デコーダLSIブロック図

3. チップレベル検証

3.1 検証方法

システムLSI開発においては，回路規模が複雑化，大規模化したために検証に費やされる時間の増大が問題となっている。

今回，1チップMPEG2デコーダLSIの検証を行うにあたって，以下の点を軸として検証環境を構築した。

(1) 統一した検証の実行環境
(2) 人為的ミスによる検証時間のロスの削減
(3) 検証項目のカテゴリー分け

(1)では，統一した検証環境を用意し，全ての検証をその検証環境上で実行するようにした。さらに，実行マシンのプラットフォームやシミュレータの種類に依存せず，同じように検証が実行できるようにした。これは，それぞれの検証環境を用意したのでは効率が悪いためと，複数の検証環境間での差異

The Verification of a 1 Chip MPEG2 Decoder LSI
Kazumasa Suzuki†, Shin-ichi Uramoto‡,
Takashi hattori† and Masaki Hashizume†,
Mitsubishi Electric Corporation,
†Information Technology R&D Center
‡System LSI Div.
5-1-1 Ofuna, Kamakura, Kanagawa,247-8501,Japan

によって生ずるトラブルをなくすためである。また，シミュレータに使用したVCSでは，検証ケース毎にコンパイルをしなくて済むようにし，単一の実行オブジェクトをマスターとして管理するようにした。

(2)では，多人数による設計，及び，検証であるため，改訂や検証実行時に発生する人為的なミスによって生ずる検証時間のロスの削減を図るため，RTL リリース用のツールやバージョン固定用のツールなどを用意し、RTL やバージョンなどの管理を行った。

(3)では，各機能毎にテストの分類を行い，テストターゲットに機能的に関与しないブロックがシミュレーションにおけるCPUパワーを消費しないようにした。また，例えば，ビデオ系のテストなどでは，一連のデータフローの中で処理を分割し，それぞれの場合についてテストのパターンを尽くすなどして，検証の効率を上げた。

3.2 検証環境

図2に今回構築した検証環境の構成図を示す。

図においてホスト制御ファイルは，外部ホストの動作を模擬するためのものであり，テストベンチ内にあるダミーホストは，この内容に従って外部ホストの動作を模擬する。

ビットストリームパターンは，エンコードされたMPEG2 ビットストリーム(ビットストリームソース)に制御情報を加えたものであり，制御情報に従ってビットストリーム入力からMPEG2デコーダLSIにビットストリームを供給していく。

アセンブラソースは，内蔵 CPU 上で実行するF/W(テストパターン)であり，アセンブラを介してメモリイメージに変換され，内部メモリにダウンロードされる。

外部SDRAMには，あらかじめ用意したメモリパターンを貼り付けることができ，外部SDRAMからデータを読み出すところから検証を実行することが可能である。

これら入力ファイルの情報や，使用 CAD，実行バージョンなどの情報は，テストケース情報ファイルに収められており，このテストケース情報ファイルをシミュレーション実行シェルに渡すことにより，その内容に従って，シミュレーションが実行される。

検証後，ビデオ・オーディオ等の出力データ，外部SDRAMへの出力データ，内部ノード等は，テストベンチ上に用意された機能によりファイルに出力され，シミュレーション実行後，テストの成否の判定に用いられる。なお，どのデータをファイルに出力するかもテストケース情報ファイルにより選択することが可能となっている。

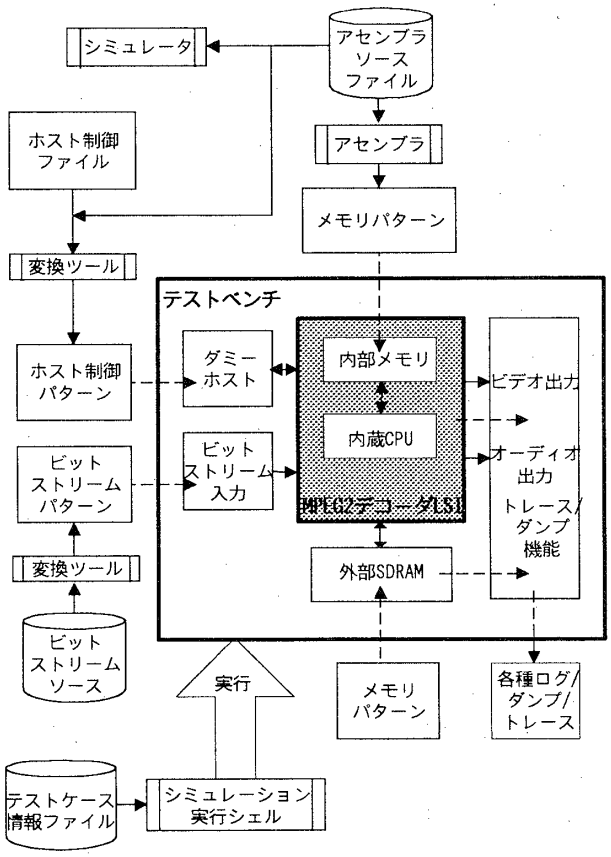

図2 検証環境構成図

3.3 シミュレーション実行環境

シミュレータには，Verilog-XL及びVCSを使用し，マシンには主にSunのUltra 2を使用した。検証に使用したテストケース数は約500で，全サイクル数は約200万サイクルである。

シミュレーション速度は，テストケースにより異なるが，VCSで約90cycle/s程度であった。

4. おわりに

今後，システム LSI はますます大規模化・高機能化し，それに伴い必要とする検証量は増加していく傾向にある。限られた時間の中で，いかに必要十分な検証を効率よく行っていくかが，今後のシステムLSI 開発における検証方法の課題として挙げることができる。