情報処理学会第37回(昭和63年後期)全国大会

ＬＳＩ設計における設計データ管理の一手法

3U-8

木下美子　小野昌治　山口　高　山田一男　梁取弘司
日本電気株式会社

1．はじめに

ＬＳＩのＣＡＤは各種ＣＡＤツールの入出力となる設計データを設計の流れに従って変換、修正しながら最終的な目的データに変化させていく作業である。近年、ＣＡＤ化が進んできたＬＳＩ設計では、ＣＡＤシステムの多様化・マルチホスト化（大型機、ＥＷＳ、ＰＣ）に伴い設計データも多様化・分散化し、その管理は複雑になってきている。そこで、これらの設計データをいかに管理するかということは高品質設計を目指すための解決すべき問題となっている。

本稿では、ＬＳＩ設計での設計データ管理の問題点とそれを解決するための管理手法の一例を紹介する。

2．設計データ管理システム

現状において設計データの管理は部分的（個人毎や各ツール毎）にしかなされていないことが多い。設計データ間には多くのＣＡＤツールを介した複雑な関係があるため、互いの設計変更が伝わらないような個別管理を行なうと、全体（プロジェクトや品種）では設計データ間の整合性が保証されない。また、互いの資産流用も信頼性の面で困難な状況である。

そこで我々はこれらの問題を解決する方法として、リレーショナルデータベース（以下管理ＤＢと呼ぶ）を利用した設計データ管理システム（以下、単にシステムと呼ぶ）をＥＷＳ上に試作した。このシステムの目標を以下のように掲げた。

- 複数のユーザが利用でき、複数のプロジェクト（品種）の管理ができること
- 分散されておかれた設計データの格納場所が管理できること
- データの整合性が保証できること
- 設計流用に必要な検索用データが管理できること

3．管理手法

これらの目標を実現するために以下の管理手法を用いた。

3.1 複数ユーザ、複数品種

システムはＥＷＳのｌｏｇｉｎユーザ名をユーザ名として認識し、管理ＤＢにはユーザ管理テーブル・品種管理テーブルを設け複数ユーザ・複数品種を管理することにした。更に品種への参加ユーザを表わすテーブルも用意し、品種設計を１つのプロジェクトとして管理し易い構造とした。

また、複数のユーザが同一品種の設計を行なう場合、修正途中の不完全な設計データを他のユーザが使用してしまう危険性がある。そこで、一人のユーザが修正している間に他のユーザがアクセスできないような排他制御をするロック制御機能を設けた。

3.2 データの格納場所

設計データの論理的構造のモデルを図１に示した。品種やセルは概念であり、物理的に存在するのは各種の設計データである。設計データには種類があるのでシステム内では識別子を用いて区別する（例：回路図、ネットリストなど）。設計データは”品種”、”セル”、”種類識別子”及び、ネットワークを利用した環境であれば”ホスト名（機種名＋ノード名）”を与えると分散環境下でも一意に決定できる。そこで、各データ種類毎にこの４要素を組合せて作った名前付け規則（ネーミングルール、図２参照）を定義し、ネットワーク上でユニークな設計データ格納ファイル名を自動的に付けることにした。

管理ＤＢの中には複数の品種と複数のセルが論理的な概念として管理され、設計データとしては物理的なデータへのポインタだけが持たれている。ネーミングルールは、このポインタの役割を果たしている。

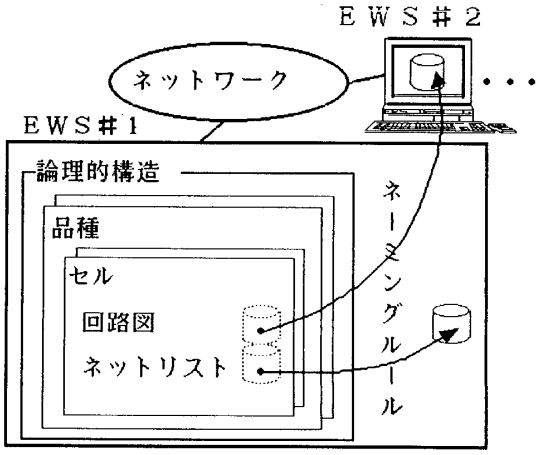

図１　分散環境でのデータ管理

ネーミングルール

識別子	ホスト	ルール
回路図	EWS#1	/usr/{品種}/{セル}/{識別子(回路図)}
ネットリスト	EWS#2	/usr/{品種}/{セル}/{識別子(ネットリスト)}

図２　ネーミングルールの一例

これによって、物理的に分散されたデータをユーザに分かりやすい論理的構造での呼び出し方法を実現した。

3.3 整合性の保証

設計データは互いに複雑な関係を持って存在する。設計データの関係には主にバージョン、階層間、フェーズ間の３つの関係がある。[1]

A Data Management Mechanism for LSI Design
Yoshiko KINOSHITA, Masaharu ONO, Takashi YAMAGUCHI,
Kazuo YAMADA, Hiroshi YANADORI
NEC Corporation

バージョンとは１つの設計データの新旧の関係である。管理方法としては図３の（ａ）のように１つのデータから派生したデータを木構造として管理する方法[1]と（ｂ）のように時間的推移として管理する方法とがある。前者は互いのバージョン間の関係を明確に表わしているが、ここまで複雑な関係が必要になる頻度は低いと考え、むしろ管理ＤＢ内の情報量や操作の効率化のためここでは後者の方法を取った。

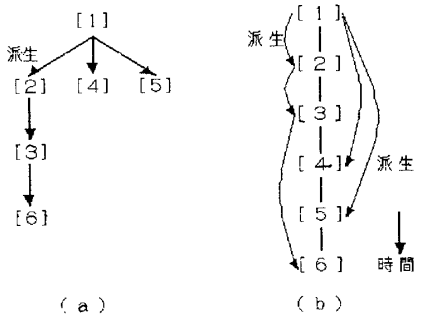

図３　バージョン管理方法

階層間とは階層設計をしたときの同種設計データ間の上位・下位の関係であり、フェーズ間とは、異種設計データ間での元データと目的データの関係である。階層間・フェーズ間では一方の変更が他方に影響を与える場合、一方のみの変更は不整合を引き起こす。例えば、下位階層の回路でピン数が変わった場合、上位階層の回路も修正しなければならないし（階層間）、回路図を変更した場合、ネットリストも更新しなければならない（フェーズ間）。このように、階層間・フェーズ間で整合性を保証するためには関連するデータを更新する必要がある。この更新は全てが自動的に実行できるものではない。そこで、整合性の保証の工夫として一つのデータが変更されたとき関連するデータに”整合性が保証できなくなった”という告知を送り更新を促すようにした。

セル管理テーブル　ROOT

データ種	作成日	更新日	作者	階層告知	フェーズ告知
回路図	88/5/15	....	kino	from LEAF	なし

セル管理テーブル　LEAF

データ種	作成日	更新日	作者	階層告知	フェーズ告知
回路図	88/5/15	....	ono	なし	なし
ネットリスト	....	....	ono	なし	from 回路図

階層テーブル

親	子
ROOT	LEAF

図４　告知（整合性の管理）

実現方法の一例を図４に概略的に示した。セル管理テーブルに告知欄をつけておき、下位セルＬＥＡＦの回路図を変更したとき、同じセルのネットリストにフェーズ間関係の告知を送り、階層テーブルを参照して上位セルＲＯＯＴに階層関係の告知を送る。そして告知を送られたデータが次に呼び出されたときシステムがユーザに更新を促す。告知がなければ他のデータとの整合は取れている。

このようにデータの信頼性を高めるための整合性の保証を実現した。

3.4 データ検索

設計流用や部品化を促進するために付加情報を管理し、検索できるようにした。管理ＤＢに用意した品種管理テーブル、セル管理テーブル、設計データ管理テーブルに作成者、作成日、機能表現などの検索項目を設け、さらに、自由にコメントを記入できるようなメモ書きファイルを設けた。ここでメモ書きファイルも前述のネーミングルールによって管理される。これらの情報の検索は設計流用に有効である。データの検索例を図５に示す。

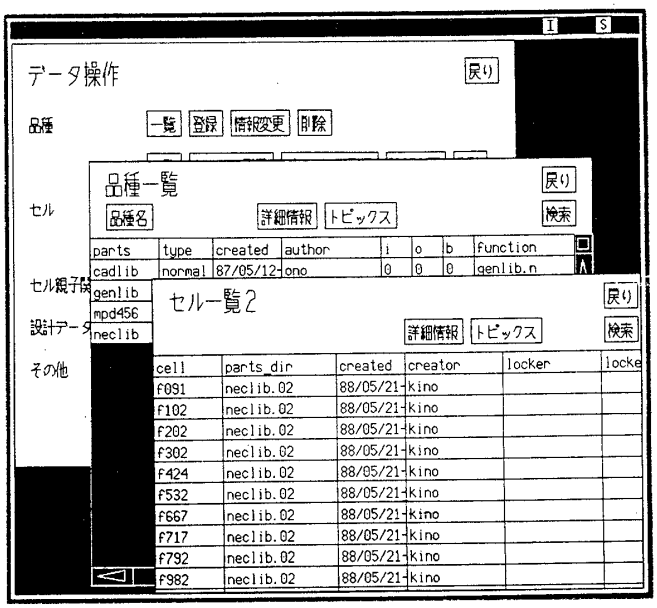

図５　データ検索例

４．まとめ

ここでは、管理ＤＢの中に情報を格納し、それを参照しながら設計データをアクセスして管理する手法の一例として、複数ユーザ・複数品種管理、データ格納場所管理、整合性保証、データ検索を簡単に紹介した。この手法によって設計データ管理の目標を満足させることができた。ファイルの格納場所の管理にとどまらず整合性の保証が成されることによって、より高品質な設計を支援することができるものと考える。

５．おわりに

本稿中で紹介した設計データ管理システムは試作システムが完成し評価中である。今後は評価とともに、管理の拡充及び、このデータ管理機能を取り入れた統合設計支援環境を構築していくつもりである。

参考文献

[1] Katz, R.H., M. Anwarrudin, E. Chang, " A Version Server for Computer-Aided Design Data ", 23rd ACM/IEEE Design Automation Conference (1986)