LSIのデザイン オートメーションへの応用

熊野　長次郎

LSIの設計は，演算，論理，データ処理機能を物理的に実現するために数多くのステップから成り立っている．このため，スイッチング理論，ソフトウェア工学，グラフ理論，数値計算，電磁気学，等の基本的な知識が要求される．設計の自動化を実現するためには，これらの素養をもった人たち（必ずしも専門家である必要はないらしい）が集まり，巨大なプログラム群を作成し，その間のインターフェイスを上手にとらねばならない．また自動設計システムを運用するために適切なハードウェア構成も十分に検討する必要がある．

ここでは，LSI設計の概要を述べ，スーパーコンピュータがどの設計分野に適しているかについて述べる．

1. LSIの概要

LSIに不慣れな読者のために，その概要を簡単に説明する．LSIはシリコン単結晶のウェーハと呼ばれる円板上に論理回路を構成したもので，配線用のピンが配置されている容器に収納されている．論理回路は機能ブロックに分けて，階層的に設計するため，物理的な構成も図1に示すような構造をもつ．まず，セルと呼ばれる基本回路（1～数十ゲート）があり，これをもとに，ALU(演算論理ユニット)，CU(制御

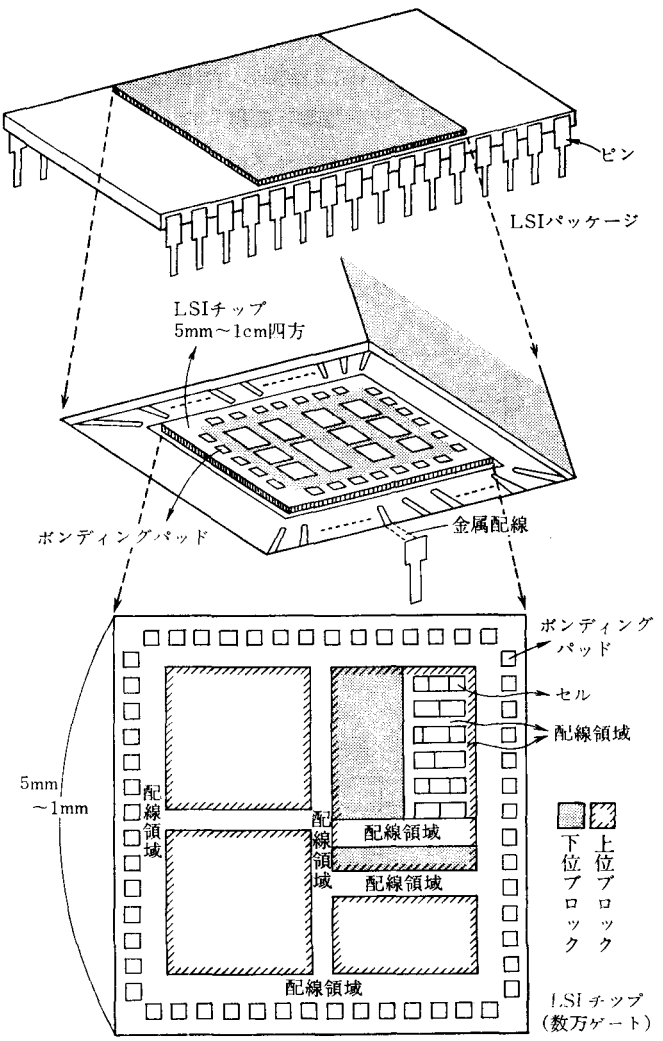

図1　LSIの構成

くまの　ちょうじろう　三菱総合研究所

ユニット)，汎用レジスタ群，デコーダ・エンコーダ，ROM (読出し専用メモリ)，RWM (読み出し書込みメモリ．この中，任意の時刻に任意の場所にアクセス可能なものをRAMと呼ぶ) 等に対応するブロックが階層的に構築されている．LSIには，これらのすべての機能をもつものから，単機能のみ有するものまであり，市場ニーズに合わせて作成される．

任意の基本回路は，AND，OR等の基本ゲートまたはフリップ・フロップで構成される．出力が入力によってのみ決まり，過去の状態に無関係な回路を組合せ回路と呼ぶ．任意の組合せ回路はNAND(またはNOR)を用いて表わすことができる(図2参照)．たとえば，NANDを数万ゲート整然と配置したものを作っておき，ユーザー・ニーズに合わせて配線工程のみ，後に行なえば生産性が大幅に上昇することは容易に想像される．性能は専用に最適設計したものと比べて劣るが，

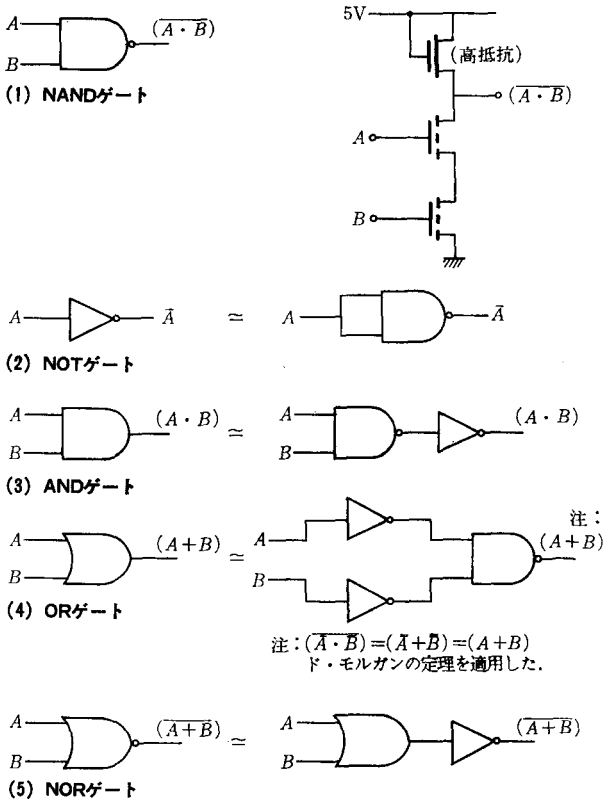

図 2　NAND ゲートと他のゲートとの関係

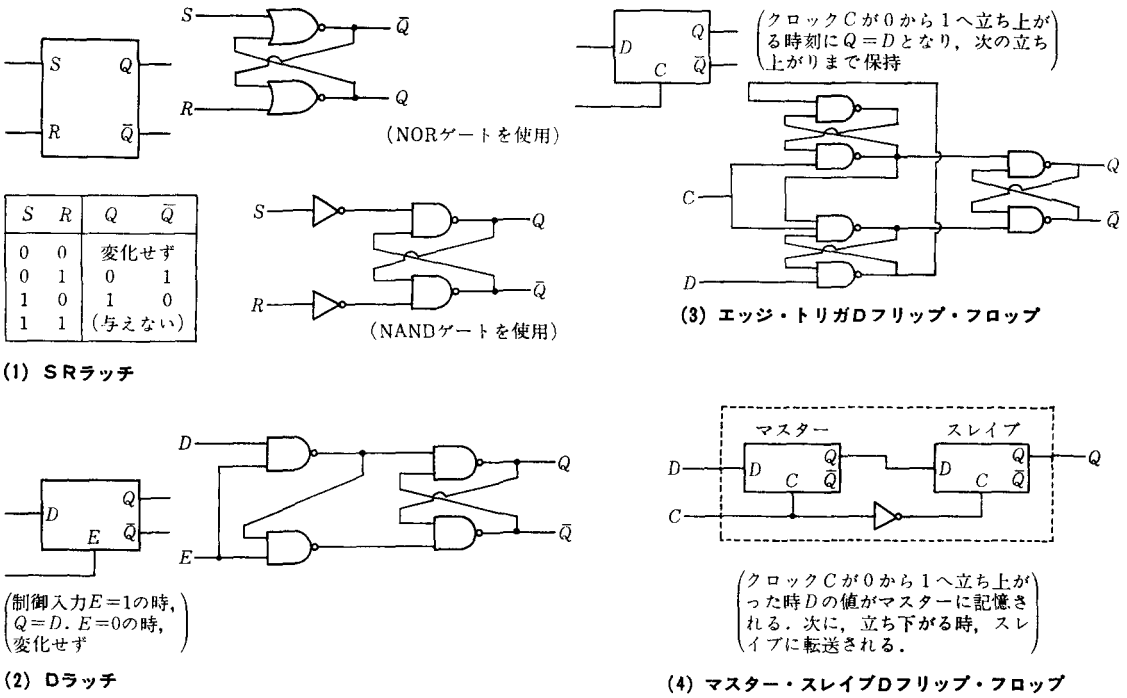

図 3　フリップ・フロップのゲート回路表現

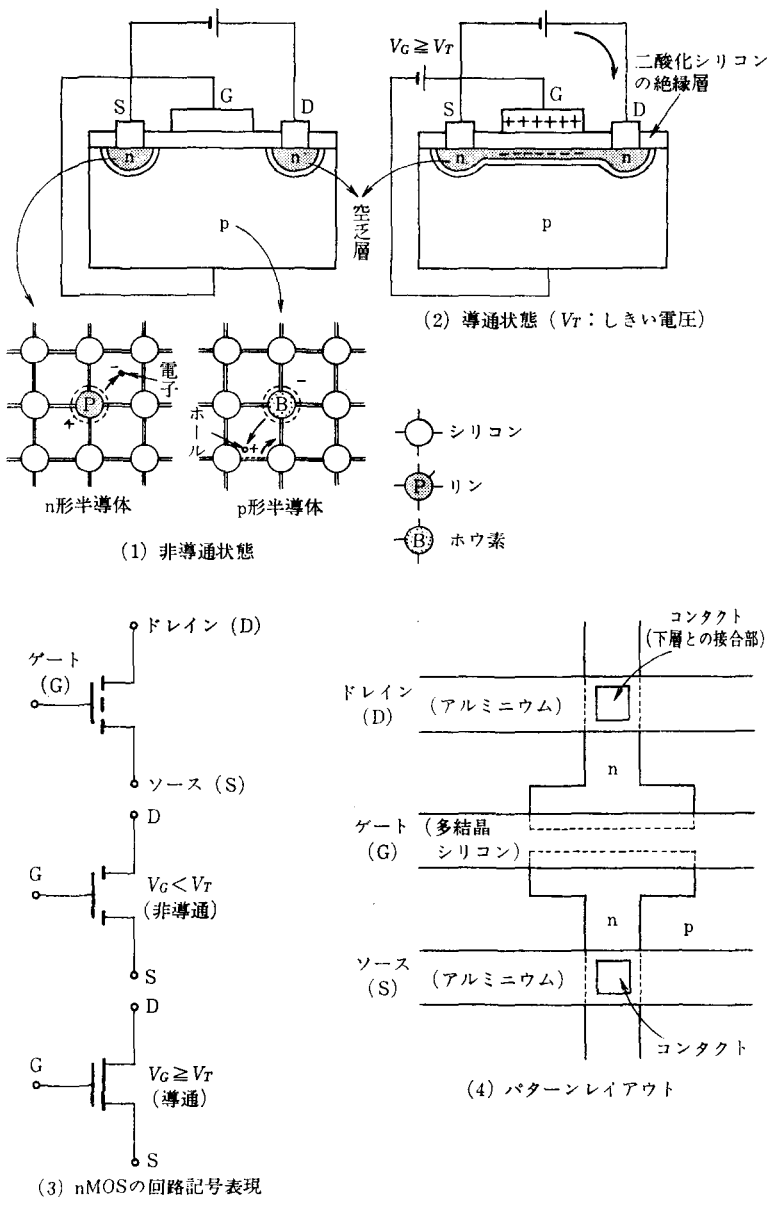

図 4　nMOS の構造とレイアウト

多品種少量の LSI では設計段階に多くのコストをかけられず，この方面にきわめて有効である．

このような方式の LSI はマスタ・スライスと呼ばれ，PLA(Programable Logic Array)とともにセミカスタム LSI の代表的なものとして位置づけられている．

他方，入力および，過去の状態によって出力が決まる回路を順序回路と呼ぶ．順序回路を表現するためには，基本ゲートの他にフリップ・フロップと呼ばれる1ビット分の情報を記憶する回路が必要である．フリップ・フロップは，図3に示すようにゲートをラッチ(latch：かんぬきを意味する）することにより合成される．このため，ラッチと呼ばれることも多い．順序回路の動特性は過去の入力にも依存するため，組合せ回路以上に重要である．各ゲートの動作のタイミングを取るために，クロック(周期的なパルスの列)で同期する場合がある．これを同期式順序回路と呼び，クロックのないものは非同期式順序回路と呼ぶ．SR ラッチはフリップ・フロップの基本型であり，S=R=0 の時，Q は変化せず，S=R=1 は定義されない．S=1(set) の時，Q=1 と設定され，R=1 (reset)の時，Q=0 とクリアされる．(図3で動特性をチェックされたい．)

以上の議論で，システム設計されたものを，論理図に落とすところで，どのような作業が発生するかを想像するのに必要最小限の知識が得られたと思う．次は，これらのゲートが物理的にどのように実現されるかについて概説する．

NAND を例にとる．NAND は実際に，図4に示す MOS でも実現可能であるし，バイポーラ・トランジスタを利用した TTL でも実現可能である．前者は，電荷を運ぶのが自由電子か正孔の一方（図4では，自由電子がキャリヤである．これを nMOS と呼ぶ．このほうが p型の MOS よ

り移動度が大きいので，高速動作の回路には適している.)であるのに対し，後者は，その両方である．ここでは，簡明なMOSで説明する．図4(1)のnpn接合に電界を印加したものでは，接合面に広がる空乏層のため電流は流れない．ところが，同図(2)のように二酸化シリコンの絶縁層をはさみ，電極Gに閾値以上の電圧を加えると，電界誘導によりGの下部にn型の領域が生成される．これがSとDのn型領域を結ぶチャネルとなり，電流が流れる．以上がnMOSデバイスの動作原理である．(詳細な挙動は，**2.2** 基本回路解析で述べる.)(3)は，このデバイスの回路記号であり，(4)は，上部から見たマスク・パターンのレイアウト図である．このnMOSを図2の(1)のように接続すればNANDゲートが得られる．(読者各位が確認されたい.)

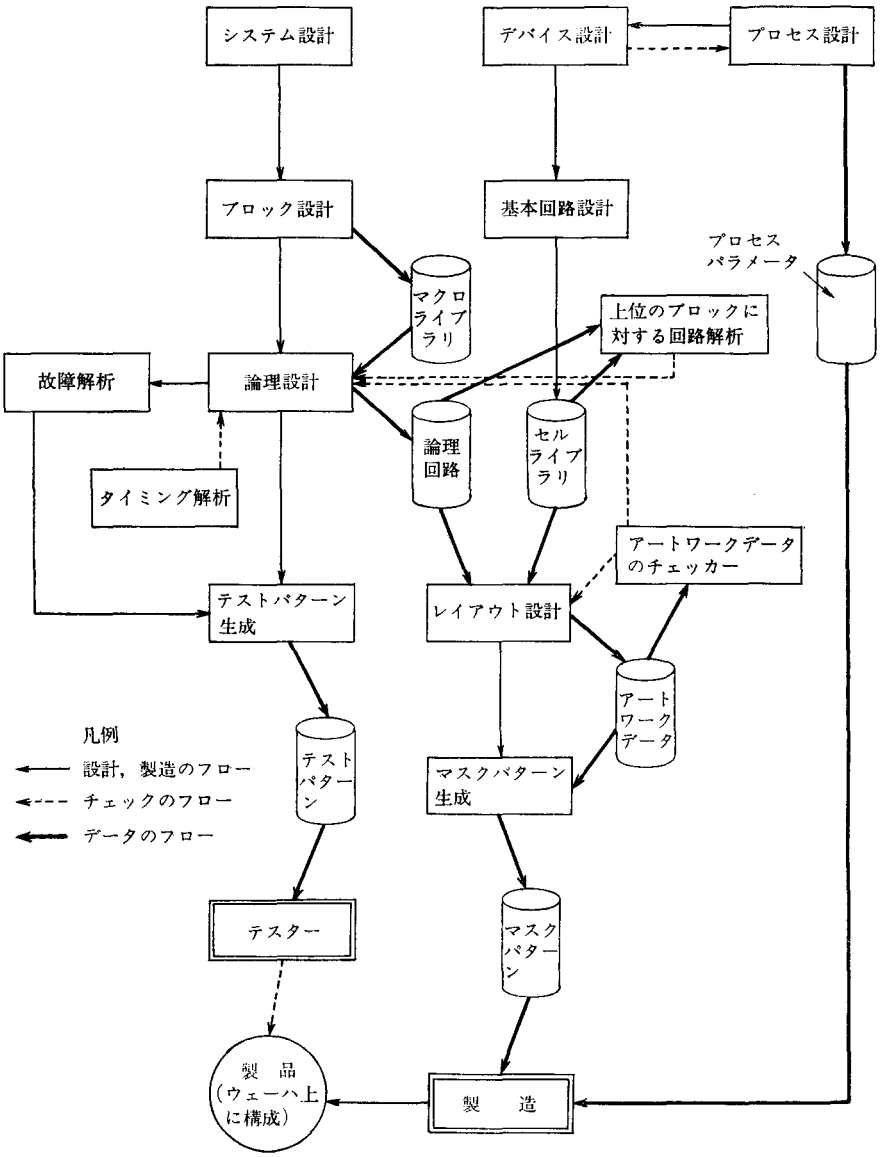

図 5　LSIの設計・製造フロー

nMOSとpMOSを組み合わせたCMOSと呼ばれるデバイスは，常にnチャネルかpチャネルの一方が非導通となり，0Vと5Vの電源間に電流が流れないため消費電力が少なくて済む．MOSのゲート部は絶縁層をはさみコンデンサを形成しているが，この充電・放電時間が動作遅延時間になるため，その容量が小さいことが重要である．サファイアは絶縁特性がよく，容量がきわめて小さいため，CMOSに応用され，バイポーラ素子に近い動作速度が得られている．

2.　LSIの設計

LSIの設計は図5に示すフローで構成される．「1. LSIの概要」を読まれた読者は，図中の設計

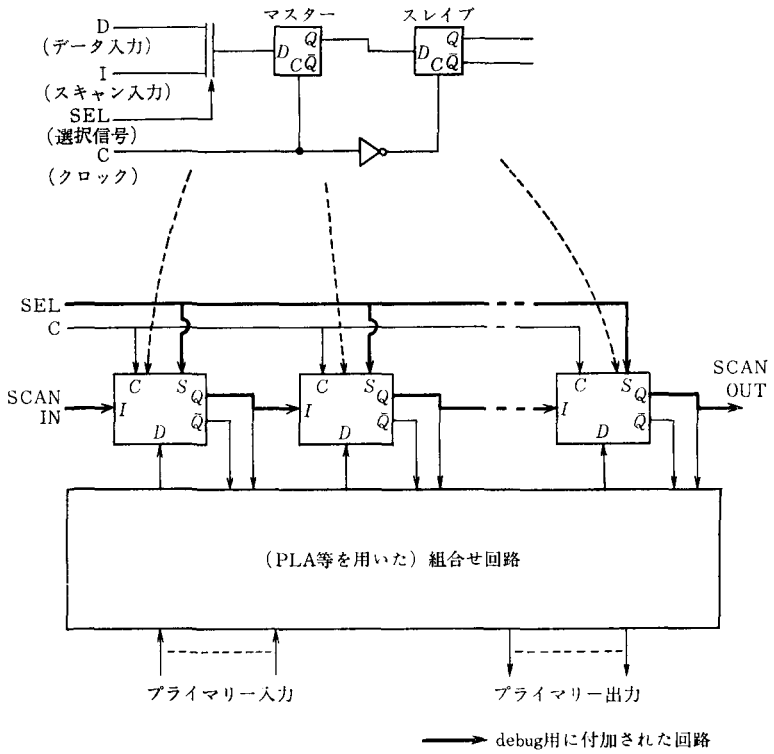

図 6 スキャン・パス方式(試験容易化設計)

・解析フェイズでどのようなことをするのか容易に想像し得るであろう．LSIの場合は，ソフトウェア等と異なり，いちど完成したものに対して訂正がきかない．このため，各段階でチェックが入るのが大きな特徴である．

また，フロー全体の中で，マクロ・ライブラリやセル・ライブラリをもとに，論理回路図やアートワーク図が有機的関連をもって生成されるのがわかるであろう．このため，全体が1個のデータベース・システムの管理下で処理されるのが望ましい．

しかしながら，個々に見ても巨大なプログラムを管理し，複雑なハードウェア構成を管理するシステムを作成するには莫大な投資とリスク負担を覚悟しなければならない．現在のところ，上記の要求をすべて満足するシステムは見当らない．

設計は大きく分けて，論理設計，基本回路設計，レイアウト設計から構成される．以下では，各々について注意点を述べる．

2.1 論理設計フェイズ

システム設計されたものを，階層的に処理し，ゲート・レベルの回路を生成すること．適当なテストパターンを入力して，回路が所要の機能を達成しているか否かを，ソフト的にもハード的にもチェックすることがこのフェイズの目的である．

後者については，故障箇所を診断できること．また，効率のよいテストパターン生成が可能なことが重要である．このような故障解析を順序回路に適用する場合，時間軸をもとに展開して，擬似的な組合せ回路として取り扱うため回路規模が増大する．このため，順序回路に対しては，試験容易化設計が行なわれるのが一般的である．この代表的なものが図6に示すスキャン・パス方式である．これは，ソフトウェアで言えば，ダイナミックなデバッグ機能を標準的に保有するプログラム設計方式と言えるであろう．すなわち，選択信号をシフト・レジスタ動作状態にし，テストパターンを各フリップ・フロップに設定する．次に，プライマリ入力にテストパターンを設定し，選択信号を反転して，プライマリ出力を観測する．さらに，1クロック進めプライマリ出力を観測し，選択信号をシフト・レジスタ動作状態にして，フリップ・フロップの内容を順次読み出し，結果をチェックするといった検査機能をもつ．IBM の LSS*D も同様の主旨をもつ設計方式である．

実回路では，ゲート遅延があるため，論理シミュレータも遅延を考慮したモデルでなければなら

* Level Sensitive Scan Design

ない．各端子ごとにノミナルな遅延を割当てる標準遅延モデルが最もよく使用されている．最悪時のタイミングを検査するために遅延の幅を割当てる最大—最小遅延モデルが使用されることもあるが，計算時間を多く費すため大規模回路にはあまり適用されていない．

2.2 基本回路設計フェイズ

このフェイズは，論理回路を物理的に実現するためのセル・ライブラリを定義するもので，プロセス解析，デバイス解析，回路解析の手法が利用される．デバイス・モデルとして，解析的なモデル（たとえば，バイポーラ・トランジスタのEbers-Moll モデル，Gummel-Poon モデル等）を利用し，回路解析のみで，このフェイズを処理する場合も多い．回路解析は比較的大規模な上位ブロックの遅延特性を解析するためにもよく利用される．回路には，時定数の大きく異なる素子が混在するため，過渡解析にはインプリシット積分が用いられる．このため，各タイム・ステップごとに大型の非線形方程式を（したがって，大型の線形方程式を逐次的に）解く必要があり，大型疎マトリクスのLU分解に工夫がされている．さらに大規模な回路の遅延を解析するために電子回路と論理回路のハイブリッド・シミュレータも発表されている．

プロセス技術の微細化とともに，プロセス解析・デバイス解析の必要性が高まってきた．デバイス解析とは，デバイス中の電圧，電子・正孔濃度，電流密度を詳細に解析するもので，以下の方程式を解くことにより得られる．

① ポアソン方程式

$$\varepsilon \,\mathrm{div}\cdot \mathrm{grad}\,\varphi = q(n-p-N)$$

② 電子・正孔電流の連続式

$$q\partial n/\partial t = \mathrm{div}\, J_n - qR$$

$$q\partial p/\partial t = -\mathrm{div}\, J_p - qR$$

③ 電子・正孔のドリフト・拡散電流特性式

$$J_n = -q(\mu_n n \,\mathrm{grad}\,\varphi - D_n \,\mathrm{grad}\, n)$$

$$J_p = -q(\mu_p p \,\mathrm{grad}\,\varphi + D_p \,\mathrm{grad}\, p)$$

ただし，

ε：半導体の誘電率，q：電子電荷，N：不純物濃度（ドナの場合は正，アクセプタの場合は負），φ：静電位，n：電子密度，p：正孔密度，J_n：電子電流密度，J_p：正孔電流密度，R：再結合速度，μ_n：電子移動度，μ_p：正孔移動度，D_n：電子拡散係数，D_p：正孔拡散係数である．

実際に解く場合は，ヤコビ行列を対称なバンド構造に近づけるため，以下の変換を行なう．

$$n = n_i \exp[(\varphi-\varphi_n)/V_T]$$

$$p = n_i \exp[(\varphi_p-\varphi)/V_T]$$

ただし，

n_i：真性キャリア密度，V_T：熱電位（$=kT/q$，k：ボルツマン定数，T：絶対温度），φ_n：電子の擬フェルミ準位，φ_p：正孔の擬フェルミ準位である．

pn接合面では，Nが急激に変化するためメッシュを細かくとる必要があるが，2次・3次元解析ではヤコビ行列の規模が急増し，計算時間を膨大に消費する．

ところで，不純物濃度Nは，イオン打ち込みによる不純物分布，不純物の堆積（プレデポジション）とドライブイン，エピタキシャル成長，熱酸化による不純物の再分配等のプロセス・パラメータによって決まる．（たとえば，イオン打ち込みの場合は，イオンビーム電流，注入時間，エネルギー等がそれである．これにより，ガウス型の濃度分布がほぼ決定される．）これをシミュレートするのがプロセス解析である．デバイス解析と連動し，最適なプロセス・パラメータを選定することが望まれている．

2.3 レイアウト設計フェイズ

以上のフェイズで，LSIの論理的・物理的な構成を決め，本フェイズで具体的にウェーハ上に配置・配線することを決める．

本来なら，完全自動でマスクパターンが生成されることが望ましいが，自由度が多すぎて実用的なシステムは得られていない．しかしながら，配

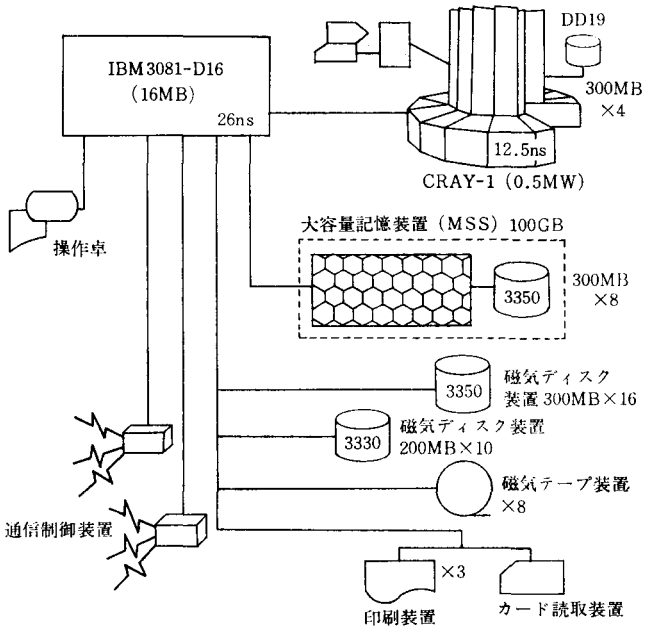

図 7 スーパーコンピュータシステム構成図

置決定後の配線プログラムに関しては，有効なプログラムが発表されている．実際，マスタ・スライスに関しては，自動配線プログラムを提供しているメーカーが多い．

フルカスタム LSI に関しては，会話型のグラフィック・エディタにより，セルやブロックを階層的に設定し，各階層レベルで自動配線プログラムを利用する方法がとられている．この場合，人手設計が入るので，幾何検査（DRC）や接続検査が必要となる．前者は，パターンの最小幅や最小間隔等が守られているか否かをチェックするもので，実用に供している．後者は，パターンから回路素子とその接続関係を抽出し，簡単なチェックを行なうほかに，論理シミュレータ等にかけて，初期の設計どおりのものが得られたか否かをチェックするものであるが，任意のデバイスに対して回路抽出することはむずかしく，DRC ほど実績をあげていない．

最近，コンパクション(空間圧縮)機能がついたシンボリック設計プログラムが発表された．

これは，ラフ・レイアウトを与えれば，パターン・ルールを満足する範囲で空間圧縮をしたアートワーク図を出力するものである．これを用いれば，DRC は不要となり，設計工数も大幅に短縮されると思われる．

3. LSI 設計のハードウェア構成

2. では LSI 設計にどのようなフェイズがあり，それがどのような特徴をもつかについて詳しく述べた．ここでは，これに適したハードウェア構成について考え，スーパーコンピュータがどの分野と深く係わりをもつか検討する．

3.1 スーパーコンピュータ CRAY 1

まず，スーパーコンピュータについて触れてみよう．この定義はあまり定かではないが，50 MIPS（1秒間に5000万命令を処理すること）以上の処理速度をもつ計算機を，一般にこう呼んでいる．現在のところ，この名に最もふさわしいものは CRAY 1 で，150MIPS の処理速度をもつと言われている．高速演算処理能力を達成するため，パイプライン方式，ベクトル・レジスタ，チェイニング方式を採用している．したがって，ベクトル化可能な演算（たとえば，ベクトルのスカラー倍や内積演算）に対して無類の強さを発揮する．

この高速演算能力をバック・アップするため，フロント・エンド処理は図 7 に示すように別のプロセッサ(この図では，IBM 3081）が担当している．

具体的な性能を，回路解析プログラム SPICE 2 のベンチマーク・テストをもとに評価してみよう．表 1 からわかるように，IBM の超大型計算機3081と比較して，約 7 倍程度の処理速度は最低限保障されそうだということがわかる．(ケース 1 は，モデルが小さく，計算処理より入出力処理の比重が高いため，性能比が低く出ていると考えら

れる.) これは，両者の MIPS 比の約10という値とよく一致している．このことを，LSI 設計プログラムで述べれば，従来と同程度の時間で，論理設計では1ケタ多いテストパターンを入力でき，2次元デバイス解析では一方向に対して約3倍のメッシュを生成でき，回路解析では約10倍の時間にわたり過渡解析ができることになる．

3.2 LSI 設計のハードウェア構成

論理設計で階層的に回路を構築する場合や，それと対応するパターンを生成するアートワークでは会話型の図形入出力装置が欠かせない．また局所的な論理解析，自動配線，DRC はスーパーミニコンで十分行なうことができる．このレベルの処理では，早いレスポンスが要求されるので，スーパーミニコンを中心とし光ファイバー等を利用したローカル・エリア・ネットワーク（LAN）を組むことが望ましい．しかしながら，回路規模が大きくなると超大型計算機に頼らざるを得ない．

さらに，大規模な回路解析やデバイス解析のような数値解析は，3.1で述べたようにスーパーコンピュータを利用するのが得策であろう.(CRAY 1 は，整数演算よりも実数演算にその特質をよく表わす.)

結論として，現在考えられる理想的なハードウェア構成は，スーパーコンピュータと超大型計算機がフロント・エンド接続され，LAN の中心をなすスーパーミニコンと超大型計算機が RJE(リモート・ジョブ・エントリー) 接続されるネットワークであろう．

近い将来，上記の CAD システムと製造装置，テスター等が直接接続され CAM システムとみなされることになるであろうし，少し先の話になるかもしれないが，このようなシステムで設計・製造された VLSI が，さらに高速並列処理機能を有するデータ・フロー型マシンのプロセッサーとして組み込まれ，3次元のデバイス解析を行なうためのハードウェアを提供することになるかもしれない．

表 1　SPICE 2 の CPU 時間の比較

ケース	モデル（節点数）	Ⓐ IBM3081	Ⓑ CRAY 1	Ⓐ/Ⓑ 性能比
1	Operational Amplifier -バイポーラ (26)	10.10秒	2.28秒	4.4
2	Flip-Flop 特性 -MOS 100ns まで (146)	74.35秒	10.42秒	7.1
3	同上 600ns まで (同上)	228.35秒	33.25秒	6.9
4	-バイポーラ (371)	1710.62秒	25.12秒	68.1

参考文献

[1] E. S. ヤン（後藤俊成，中田良平，岡本考太郎，共訳）：半導体デバイスの基礎，マグロウヒル好学社，(1981)

[2] 室賀三郎：論理設計とスイッチング理論 LSI，VLSI の設計基礎，bit 別冊，(1981)

[3] 創刊10周年記念特集　LSI 設計日本電子産業展望，日経エレクトロニクス，1981年4月13日号，No. 262

[4] 「拍車がかかる VLSI 向け CAD システムの開発」，日経エレクトロニクス，1980年12月22日号，No. 254