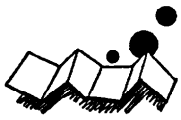
解　説



VLSI 設計の新手法†

富沢　治††　徳田　健††　榎本龍弥††

1. はじめに

VISI製造技術の進歩は著しく10万トランジスタ以上の集積度を持ったチップが出現している[1]．現在の設計技術を用いてこのチップの設計に60人・年，さらにデバッグに60人・年要したと言われている．理論的限界値として VLSI チップが1000万トランジスタまで集積されるとすれば，複雑な設計プロセスを簡略化する方法がなければこのようなチップを設計するのに6000人・年を要する事になる．このことは現在既に存在するチップの製造技術と設計技術のギャップがさらにひろがりつつある事を示している．現在製造技術の許しうる集積度に近いレベルで設計可能なものは非常に規則性の高いメモリチップのみである．この設計技術上の問題に対する1つの解決案として上げられるものにMEAD と CONWAY 氏によって提唱されている設計手法がある[2),3)]．もともと集積回路に素人であるシステム設計者が VLSI を設計できる方法として，特に大学における教育用として考えられたものであるため，この手法で設計されたマイクロプロセッサが汎用品として半導体事業をなすことに対しては異論もあるところであるが少なくともこの基本的概念は，システム設計者によるカスタム VLSI 設計に一つの方向を与えるものであると考えられる．本稿ではMEAD と CONWAY の設計手法を紹介しこの応用例としてカリフォルニア大学バークレイ校において開発されている32ビットマイクロプロセッサ RISC チップの紹介を行う．

2. MEAD と CONWAY による VLSI 設計手法

2.1 設計の基本概念

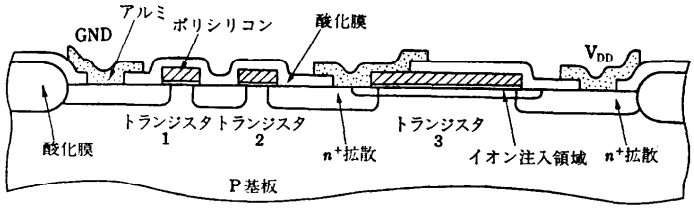
非常に複雑な VLSI 設計過程をできるかぎり簡単化するために取り入れられている基本的な考えかたは構造化設計の概念である．構造化設計は“規則的設計”と“階層的設計”の2つの要素を持ちこのうち階層的設計は複雑なシステムの設計に従来から良く用いられてきたものである．階層的設計手法を用いる事により各階層を設計チームのメンバに分割して割り当てることもできるし，また共通の部分は一度設計すれば繰り返し用いる事ができる．規則性を考慮した設計は複雑さを低減する点において大きな効果を発揮する．すなわち最初から規則性と相互結線関係を考慮してサブユニットを設計しておけばサブユニットの繰り返しでより大きなユニットを設計することができ，このとき効率が悪くかつ時間を要する配線問題を大幅に緩和することができる．VLSI の構造化設計はいわゆる構造化プログラミングと類似の概念でありまず設計過程がトップダウン的にすすめられ，1つのシステムの要求仕様から機能的なブロックに分割され相互的な配置を考慮してフロアプランがたてられる．続いてセルの大きさ，結線等のインタフェースが決定されたのちセルの詳細なレイアウトが実施される．以後ボトムアップ的にセルが組み合わされてチップが構成される．従来の非構造的ボトムアップ設計に用いられてきた NAND, NOR 等の基本論理ゲートに代わり非常に単純で規則的な相互接続によりシステムを構成するのに適した回路方式も与えられている．さらにセルの物理的レイアウトに必要なレイアウトルールに関しては一般に半導体メーカで用いられている複雑な設計基準に代わり，λ と呼ばれる基本単位の整数倍もしくは 0.5 倍からなる単純な設計基準系が与えられている．

2.2 デバイス構造とその表記法

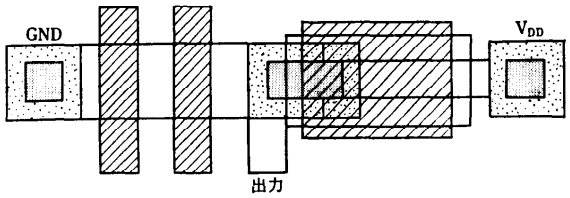
シリコン基板上に形成された VLSI のデバイス構造を NAND ゲートを例にとり図-1(a)に示す．NAND ゲートを構成しているコンポーネントは図-1(c)の等価回路図に示すトランジスタ(1)，(2)と負

† A New VLSI Design Methodology by Osamu TOMISAWA, Takeshi TOKUDA and Tatsuya ENOMOTO (Mitsubishi Electric Corp., LSI R & D LABORATORY).

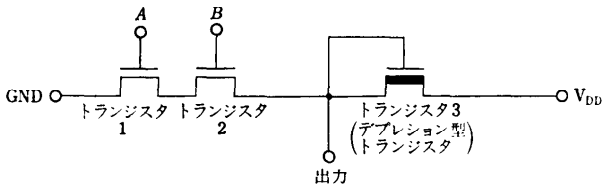
†† 三菱電機(株) LSI 研究所



(a) デバイス構造



(b) 幾何的配置



(c) 等価回路図

図-1 NAND ゲート

荷となるデプレッショントランジスタ（ゲートソース間に 0 V が印加されているときでも導通している）である．これらのトランジスタはソース/ドレインに対応する拡散領域とゲートとなるポリシリコンからなっており，デプレッショントランジスタの部分はゲートポリシリコンの直下にしきい値電圧を負にするためのイオン注入領域が設けられている．このデバイス構造はいくつかのマスクの組みを用いて写真製版，拡散等の技術を用いて形成される．VLSI の設計は最終的にはこの製造のためのマスクのレイアウトを作成することにある．図-1(b)はマスクの幾何的レイアウトを重ね書きしたものである．レイアウト設計を行うためには製造技術と関連する個々の工程に許される寸法を守る必要があり，これがレイアウト設計基準と呼ばれるものである．設計基準は例えばポリシリコンゲートのゲート長，金属配線と拡散領域の接続をとるためのコンタクト孔の大きさ等を与えるものであり，その製造技術のレベルに応じてきまっている．実際の半導体メーカで用いられているものは，VLSI チップのサイズがそのまま製品のコストに繋がるため，製造技術のレベル内でできるかぎり小さくするよう配慮されてお

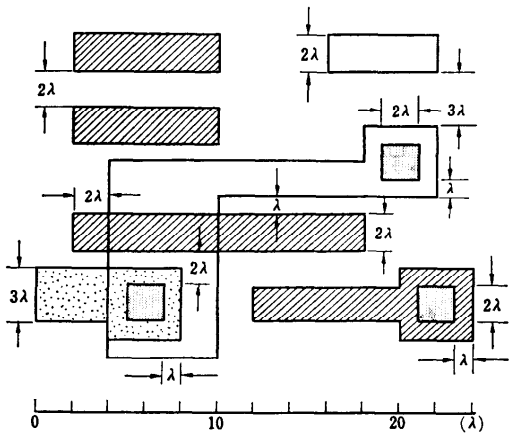


図-2 レイアウト設計基準

り，一般には数 10 以上の細かい取り決めがなされている．複雑な設計基準はレイアウト設計をより複雑にし，かつ設計ミスをもたらすことから MEAD と CONWAY は非常に単純化した設計基準を与えている．この例を図-2 に示す．基本的にはユニットとなる最小寸法を λ として，この λ の整数倍，場合に

よっては0.5倍を用いて設計基準体系を作っている．レイアウト図に対してデバイスの幾何学的詳細を記述するかわりに回路のトポロジを示す図がスティック線図でありこの例を図-3 に示す．ここでは点線，実線等で各層を識別してあるが米国で標準的に用いられているスティック線図の色分けは，拡散層及びトランジスタ領域が緑．デプレッショントランジスタのイオン注入層が黄色，ポリシリコンが赤，金属配線が青，コンタクト孔が黒である．

2.3 基本回路

VLSI の構造化設計を進める上で，単純かつ規則的

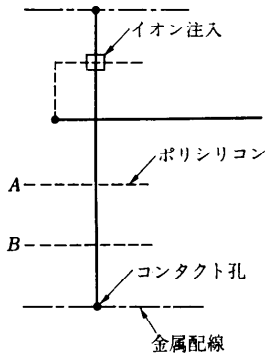
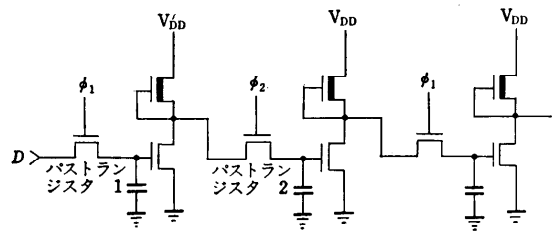
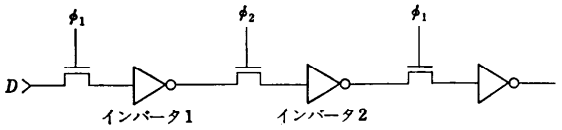


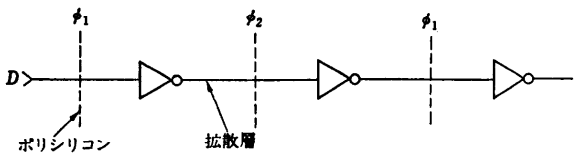
図-3 スティック線図 (NAND ゲートの例)



(a) 回 路 図



(b) トランジスタ/ロジック混合表記



(c) ロジック/スティック線図混合表記

図-4 ダイナミックシフトレジスタ

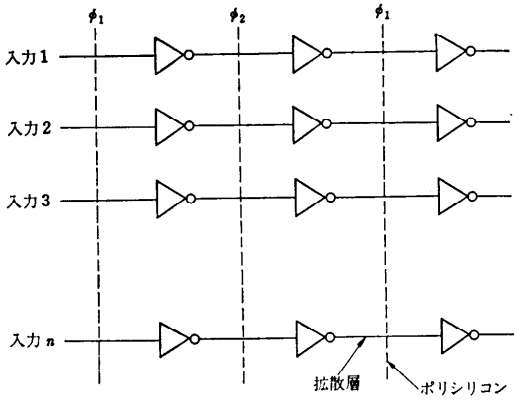


図-5 ワード構成のダイナミックシフトレジスタ

な相互接続によりサブシステムが構成されるような回路形状が必要になる．このため MEAD と CONWAY は，インバータ回路をパストランジスタ（トランスミッションゲートとも呼ばれる）で結合して何段も重ねていく手法を基本としている．まずこの一例としてシフトレジスタを取り上げつづいて ALU の設計を例にとって説明する．図-4 はダイナミックシフトレジスタをトランジスタ回路図及び，トランジスタ/ロジック混合表記，ロジック/スティック線図混合表記で示す．クロック $\phi1$ が "H" レベルになるとパストランジスタ1が導通し入力データ D が第一段目のインバータの入力に伝えられる．データ D の "H","L" に対応して入力容量が充電もしくは放電されインバータ1の出力は "L" もしくは "H" になる．$\phi1$ が "L" レベルになるとパストランジスタ1が遮断し充電/放電された入力容量の電位は保持され，$\phi1$ が再び "H" になって入力 D のレベルが伝わるまで値は変わらない．クロック $\phi2$ が "H" レベルになると，インバータ1の出力がパストランジスタを介してインバータ2の入力に伝えられる．以下同様にして信号が伝えられ図-4 の回路はシフトレジスタとして動作する．従来の MSI/SSI で構成されているシステムをそのまま LSI 化する場合にはシフトレジスタはいわゆるスタティックなフリップフロップを用いた回路に置き換えられ同じ機能を実現するにも数倍の素子を必要とする．このダイナミックシフトレジスタを隣り合わせに並べると図-5 に示す通り1ワードを並列にシフトさせるレジスタが実現できる．パストランジスタは拡散領域に直角にポリ

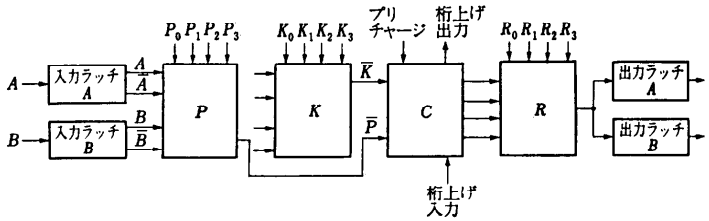


図-6 ビットスライス ALU ブロック図[2)]

シリコンを配置することによって構成されるため単純なトポロジで回路を構成することが可能である．次にもう少し複雑な論理回路の例としてALU(Arithmetic Logic Unit)の設計について述べる．**図-6**は，ビットスライスの ALU のブロック図を示す．ここで機能ブロック P, K は，ALU の2つの入力 A,B に対して桁上げ回路で行うべき内容を決定する．機能ブロック P の出力は桁上げ回路を信号が伝搬すべきかどうかを決定する．一方，機能ブロック K の出力は，桁上げ信号を抹消すべきか否かを決めるものである．ブロック R は，桁上げ入力信号と桁上げ伝搬信号の組み合わせにより，出力信号を生成する．桁上げ信号の伝搬，抹消，結果の処理の各機能ブロックは，ランダムロジックで実現する代わりに**図-7** に示す汎用的な論理機能ブロックが使われている．この回路は2つの入力を変数とする16種の論理関数を実現することができる．**図-8** のスティック線図に示すとおり，縦に拡散領域がはしり，横にポリシリコン層が通る構造でありこれらの交差点にトランジスタが構成される．この構造で図-7 と同一の等価回路を得るために，トランジスタが不要である交差点には，イオン注入が施され，デプレッショントランジスタが形成される．すなわち，デプレッショントランジスタの部分はポリシリコンゲートの電位の "H", "L" に拘らず常に導通しているため論理的にはトランジスタが存在しないのと等価になる．桁上げ回路は，従来高速性の点でキャリールックアヘッド方式がよく用いられてきたが，ここでは**図-9** に示すような，回路形状が簡単でありかつそれほどスピードの点で不利ではないマンチェスタ型桁上げ回路が採用されている．この回路は桁上げ伝搬信号が入ればパストランジスタを開き，桁上げ信号をより上位のビットに伝搬させ桁上げ抹消信号が入れば出力を "L" にするものである．ALU としての各種機能は図-6 のブロック図中 P_0-P_3, K_0-K_3, R_0-R_3, の合計12の制御線により決定する事ができる．これらの $P, K, R,$ の制御信号は全ビットに共通に与えることが

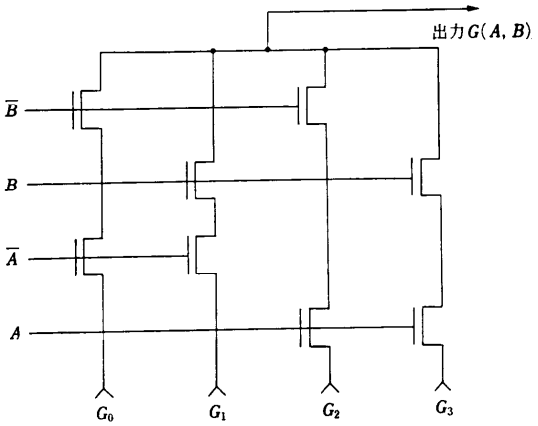


図-7 汎用論理機能ブロック[2)]

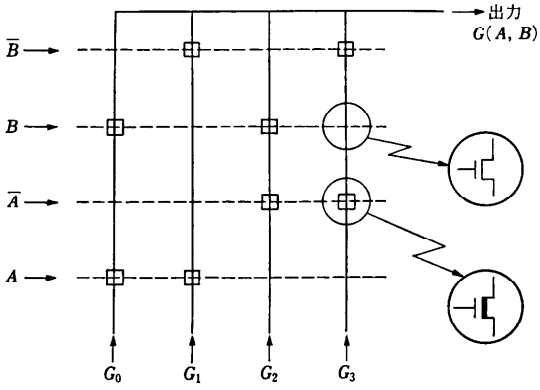


図-8 汎用論理機能ブロックのスティック線図

できるため，図-6 の回部を縦に並べて任意のビット数の ALU を設計する事ができる．

3. RISC (Reduced Instruction Set Computer)

3.1 コンピュータ設計と VLSI 技術

コンピュータのコストは一般に (1)マシーンを作るためのハードウェアコスト，(2)プログラミングコ

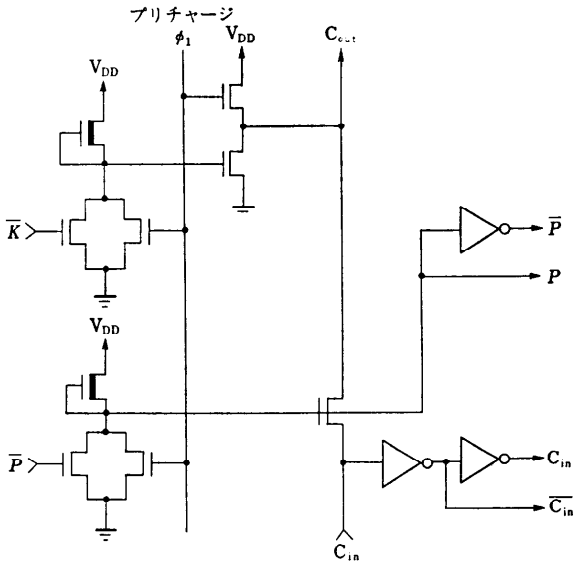


図-9 マンチェスタ型桁上げ回路[2]

スト，(3)アーキテクチャに係わる最初のハードウェア，及びプログラムのデバックのコストからなる．今までのコンピュータがより複雑なマシーンに向って進んできたのは，以下のような理由によりコスト的に有利であったからである．すなわちコアメインメモリのスピードが CPU に比べて一桁遅く性能向上のために高級な命令を付加する必要があった事，また IC メモリの進歩によってマイクロプログラム制御が容易になり命令数を簡単に増やせるようになった事さらに，メモリが高価であるためにコード密度をあげるのに複雑な命令セットが有利であった事などである．しかしながらメモリが高速化され CPU とメモリのスピードの差は小さくなりまた特別の命令は必ずしも単純な命令の組み合わせに比べて速くはないという事実も明らかになってきた．複雑なコンピュータは設計期間が長くなると同時に設計ミスが多くなるという欠点も有する．このような背景のもとに開発されたのが RISC (Reduced Instruction Set Computer) である[4),7)]．VLSI 化を前提に考えると現状のレイアウト設計基準では1チップの VLSI コンピュータはより単純な RISC の方で早く実現できるしまた VLSI 技術が2年で倍の集積密度をもたらすならば2年以内に設計，デバックが完了できるものは VLSI 技術を効率よく取り込むことができる．複雑なコンピュータに比べて RISC ではチップ面積を小さくできるためチップキャッシュ等にチップを有効に利用する事が可能である．制御用 PLA 等が小さくなりマシーンのクリティカルパスのゲート数も少なくなるので高速化に適している．以下にカリフォルニア大学バークレイ校にて開発が進められている32ビット VLSI プロセッサ RISC の設計について述べる．

3.2 RISC のマイクロアーキテクチャ

単純，規則的な RISC ではほとんどすべての命令は次の基本パターンで実行される．すなわち，(1)2つのレジスタの読みだし，(2)読みだされたデータに対する演算実行，(3)レジスタへの結果の書き込み，である．ジャンプ，コールおよびリターン命令ではプログラムカウンタの内容とオフセットが加えられその結果が適当なプログラムカウンタラッチにストアされる．ロードとストア命令だけは例外でありメモリアクセスの時間をとるために2サイクルを要する．これらを基本として同一のアーキテクチャをもつ2種類の VLSI チップ，RISC I と RISC II が設計された．カリフォルニア大学のカラーであるゴールドとブルーにちなんでこれらの2種類のチップに対してそれぞれ，ゴールドとブルーというニックネームが付けられている．

(a) RISC I

RISC I の CPU は以下の機能ブロックからなる．レジスタファイル，ALU，シフタ，プログラムカウンタレジスタ，データ I/O ラッチ，プログラムステータスワード (PSW) レジスタと，コントロールのための命令レジスタ，命令デコーダ，タイミング回路である．同時に2つのオペランドが必要とされるため，レジスタファイルは2ポートセルから構成され少なくとも2つの独立したバスを持つ必要がある．一般にレジスタはスピードを考慮してダイナミックにプリチャージされたバスを選択的にディスチャージすることによって読みだされる．従ってこの方式によれば(1)レジスタ読みだし，(2)ALU 動作，(3)レジスタ書き込み，(4)次のサイクルのためのバスプリチャージ，の4つの位相を必要とする．そこで第三のバスを設けこのバスによってレジスタが書き込まれている間に第一および第二のバスがプリチャージされるように設計された．この基本構成を**図-10** に示す．

(b) RISC II

RISC はチップ面積の大半をレジスタファイルに用いる関係上レジスタセルの大きさは重要な要素となる．RISCI の設計過程で，3バス方式のレジスタセル

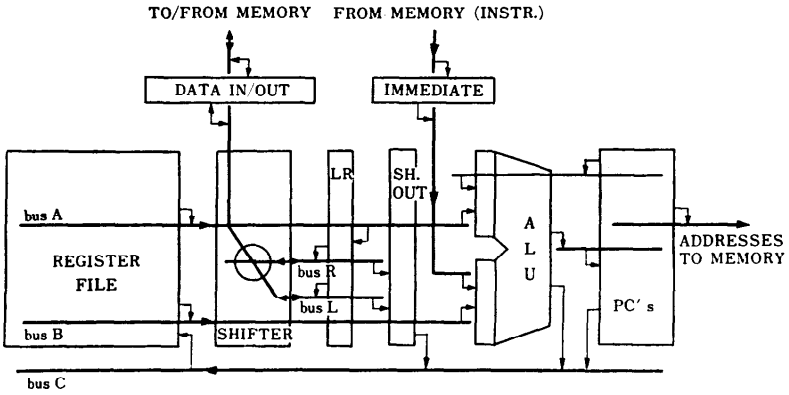


図-10 RISC I の基本構成[6]

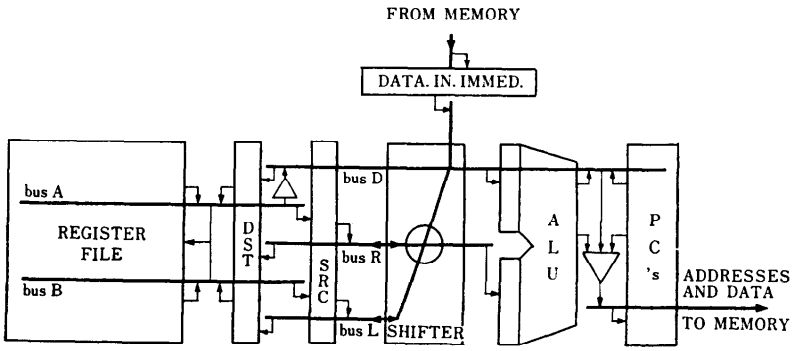


図-11 RISC II の基本構成[6]

は占有面積の点で不利である事が明らかになり，小型のレジスタセルの検討の結果，いわゆる6トランジスタ型スタッティック RAM 用のセルを RISCII のレジスタセルとして用いる事が決定された．読みだしは2つのプリチャージされたビット線の一方を選択的に放電する事で行われる．2つのビット線のうち一方は他方の否定信号であり，通常の RAM のようにセンスアンプを用いない代わり2つのビット線を用いて2ポートの読みだしを実行する．書き込みは，RAM の場合と同様に2つのビット線に相補型の信号を印加して行われる．レジスタセルが小型であるためチップサイズが小さくなり，さらにデータパスを横切るポリシリコンコントロール線が短くなるため，RC 時定数による遅延時間が小さくなり高速化が可能である．さらに，レジスタへの書き込みは次の命令実行と並列に行うことにより性能の改善が計られている．演算結果はテンポラリのラッチに保持され，引き続く算術，論理，シフト演算時にレジスタファイルに書きこまれる．従って演算結果が次の命令で必要な時は，これらの値は直接 ALU に転送される．RISC II では，次の3サイクルで命令が実行される．すなわち，(1)命令フェッチとデコード，(2)レジスタ読みだし，演算，結果のラッチ，(3)レジスタファイルへの書き込み，である．RISC II のブロック図を 図-11 に示す．

3.3 VLSI 機能ブロック

ここでは，RISC の VLSI 設計においてチップ面積に占める割合の高いレジスタファイルとシフタ部分について述べる[7]．RISC アーキテクチャの大きな特徴の1つである多重ウインドウレジスタファイルスキーム[4]を実現するためには，同時に2つのオペランドを読みだせるよう2ポートアクセスが可能なメモリが必要である．カリフォルニア工科大学の OM2 チップ[2]に用いられている2ポートレジスタセルを図-12に示す．読みだしは，ダイナミックにプリチャージしたビット線を放電して行われる．リフレッシュクロックは，セルへの書き込みを容易にするため，フィード

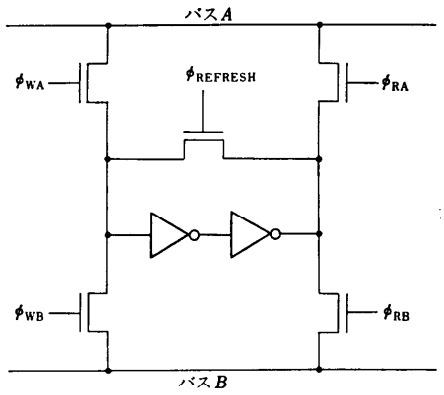
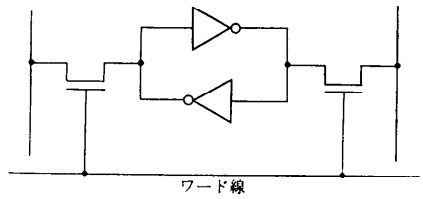
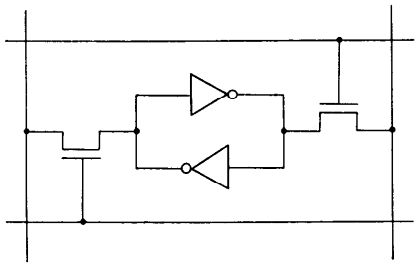


図-12 2ポートレジスタセル



(a)



(b)

図-13 スタッティックRAM用メモリセル(a)及び2ポートレジスタセル(b)[7]

バックループを遮断するためのものであり，書き込み時以外はこのループが形成されている．この種のレジスタセルは HP 社の 32 ビット CPU をはじめとして広く用いられている．3バス方式をとり，書き込みが1つの専用バスからおこなわれる RISC I では，書き込みポートを1つにした同タイプのレジスタセルが用いられている．RISC II チップでは，セルの面積を小さくし，リフレッシュ線を除去するためにスタッティック RAM で用いられている**図-13**(a)に示すシングルポートのセルを変形した**図-13**(b)のセルが用いられた．シングルポートのセルと異なり各ビットに対して1組のワード線が存在し，各ワード線は，セル内の1つのデバイスを駆動する．2つのアクセスを同時

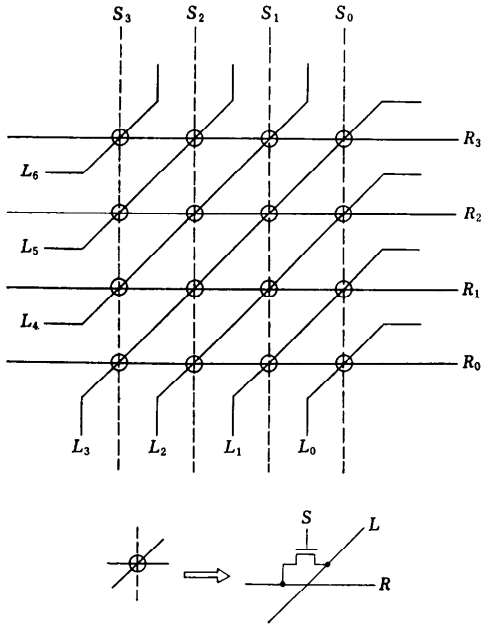


図-14 RISC に採用されているシフタの配線トポロジ[7]

に実行するために，デコーダは2つある．セルの読みだしは，単純化を目的としてセンスアンプ回路を用いず，プリチャージしたビット線を放電する方式を採っている．従ってビット線のセンシングは差動型ではなく，シングルエンド型で行われる．この方式は容量の大きなビット線を数 VOLT にわたって放電する必要があるためセンスアンプを用いるタイプに比べスピードが遅い欠点を持つ．書き込みは，両方のビット線に相補的な信号を印加して行われる．

シフタは，コンピュータ，その他のデジタルシステムに用いられる一般的な機能ブロックの1つであり，もっとも複雑なシフトは任意の量だけシフトもしくは回転が可能である．シフタを実現するアプローチにはカスケード接続したマルチプレクサを用いるものや，クロスバースイッチによるものがある．規則的であり，かつ高速性が期待できることから，RISC ではクロスバースイッチタイプのシフタが採用されている．基本的にはカリフォルニア工科大学の OM 2 チップに用いられているもののトポロジを変えたもので，これを**図-14** に示す．ここで実線はデータバスであり，破線はシフト量を指定する．○印はパストランジスタが置かれる位置を示す．例えば右シフトを行うとすると入力のワードが $L_0 \cdots L_3$ に，符号拡張部が $L_4 \cdots L_6$ に置かれ出力は R バスに現れる．

3.4 設計ツール

RISC I の設計に用いられた CAD ツールは図-15 に示すように，グラフィックエディタからシミュレーション，検証用に至る数多くのツールの集合体であり，これらは VAX 11/780 上の UNIX オペレーティングシステムのもとに有機的に結合されてシステムが構成されている．設計は ISPS (Instruction Set Processor Specification) によるアーキテクチャの記述から始まり論理設計，回路設計，レイアウト設計の後グラフィックエディタである CAESAR に入力されここで CIF (Caltech Intermediate Form) ファイルに変換される．この CIF ファイルはプロット，設計基準チェックプログラム等で検証される．さらに，CIF ファイルから MEXTRA(Manhattan Circuits Extractor) によって回路情報が抽出されこの結果を用いて回路シミュレーションプログラム SPICE の実行や MOSSIM によるスイッチレベルシミュレーションが実施される．一方マルチレベルシミュレータである SLANG による機能，論理レベルでの記述に基づくシミュレーション結果と MOSSIM によるシミュレーション結果が比較され最終的な検証がすむと，CIF ファイルはマスク製作のために ARPA ネットワークを介して送り出される．複数人によるプロジェクトの共同作業を効率良くするため SCCS(Source Code Control System) と呼ばれるプログラムを導入し，これらのプログラムやファイル類の変更に対する履歴を残しかつ，同一のファイルが異なる人間によって同時に編集されるのを禁止するなどの管理が行われている．さらに，電子メールが多用され設計者間の意志の疎通が十分に計られるような設計環境が作られている．

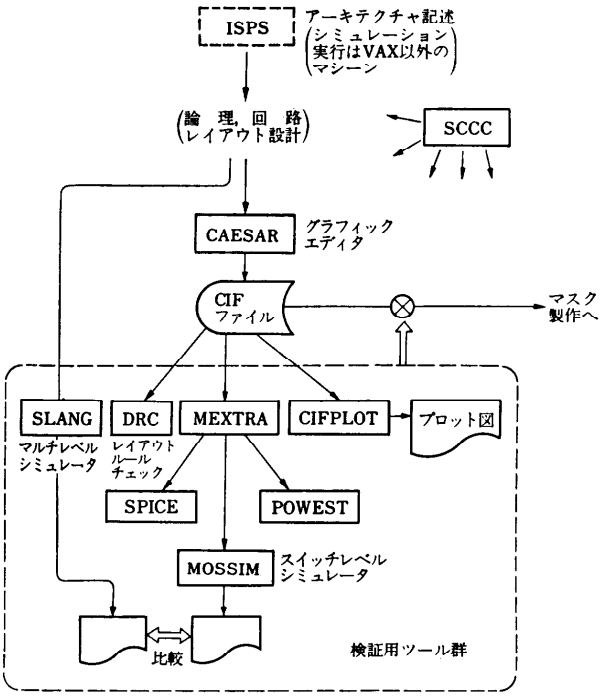


図-15 RISC 設計に用いられた CAD ツール

4. おわりに

本稿では MEAD と CONWAY によって提唱されている VLSI 設計手法の紹介と，カリフォルニア大学バークレイ校にて研究開発がすすめられている 32 ビット CPU RISC について解説を行った．データベースを中心にした標準セル/マクロセル方式による自動配置配線アプローチと異なり，この方式で VLSI を設計するためには，VLSI デバイス，プロセスに関する知識をより必要とし，かつ設計期間も長くなるであろうが集積度，性能ではより優れた結果を期待出来ると思われる．MEAD CONWAY 方式の設計手法も今後半導体メーカに取り入れられ，修正，改良が加えられた後，ゲートアレイ，標準セル方式と並んでカスタム VLSI 開発に用いられていくものとおもわれる．ユーザによる VLSI 設計を考慮すれば，大学もしくは，企業における，システム設計者に対する VLSI 教育がますます必要とされてくるだろう．

参考文献

1) Lattin, W.: A 32-bit VLSI Micromainframe Computer System, ISSCC. Dig. Tech. Papers pp. 110-111, New York (1981).
2) Mead, C. and Conway, L.: Introduction to VLSI systems, Addison Wesley Publishing Co. (1980).
3) Trimberger, S., Rowson, J. A., Lang, C. R. and Gray, J. P.: A Strucured Design Methodology and Associated Software Tools, IEEE Trans. Circuits and Systems, Vol. CAS 28, No. 7, pp. 618-633 (1981).
4) Patterson, D. A. and Ditzel, D. R.: The Case for the Reduced Instruction Set Computer, Computer Architecture News, Vol. 8, No. 6, pp. 25-33 (1980).
5) Patterson, D. A. and Sequin, C. H.: RISC I

A Reduced Instruction Set VLSI Computer, Proc. 8th Intl. Symp. on Computer Arch., Minneapolis, Minn (May 14, 1981).
6) Fitzpatric, D. T., Foderaro, J. K., Katevenis, M. G. H., Landman, H. A., Peek, J. B., Peshkess, Z., Sequin, C. H., Sherburne, R. W., Patterson, D. A. and Van Dyke, K. S.: VLSI Implementation of a Reduced Instruction Set Computer, Proc. CMU Conf. on VLSI System and Computers. (Oct. 19-21, 1981).
7) Sherburne, R. W., Katevenis, M. G. H., Patterson, D. A. and Sequin, D. H.: Datapath Design for RISC, Proc. Conf. on Adv. Research in VLSI, pp. 53-62 (Jan. 1982).

(昭和58年3月14日受付)